

Bachelorstudiengang

# Informatik

Modulhandbuch

WS 2019/2020  
SS 2019

Prüfungsordnungsversion: 2009s

Modulhandbuch generiert aus *UnivIS*  
Stand: 13.10.2019 16:43





# Informatik (Bachelor of Science)

WS 2019/2020, SS 2019; Prüfungsordnungsversion: 2009s

## Grundlagen- und Orientierungsprüfung

### Algorithmen und Datenstrukturen

- Algorithmen und Datenstrukturen, 10 ECTS, Felix Freiling, Norbert Oster, WS 2019/2020 23

### Konzeptionelle Modellierung

- Konzeptionelle Modellierung, 5 ECTS, Richard Lenz, WS 2019/2020 26

### Grundlagen der Technischen Informatik

- Grundlagen der Technischen Informatik, 7.5 ECTS, Jürgen Teich, WS 2019/2020 28

### Parallele und Funktionale Programmierung

- Parallele und Funktionale Programmierung, 5 ECTS, Michael Philippsen, Norbert Oster, SS 2019 30

### Grundlagen der Schaltungstechnik

- Grundlagen der Schaltungstechnik, 5 ECTS, Albert Heuberger, Florian Rittner, SS 2019 32

### Rechnerkommunikation

- Rechnerkommunikation, 5 ECTS, Anatoli Djanatliev, SS 2019 34

### Systemprogrammierung

- Systemprogrammierung, 10 ECTS, Wolfgang Schröder-Preikschat, Jürgen Kleinöder, SS 2019, 2 Sem. 36

### Mathematik für INF 1

- Mathematik C1, 7.5 ECTS, Serge Kräutle, u. a. Hochschullehrer, WS 2019/2020 38

### Mathematik für INF 2

- Mathematik C2, 7.5 ECTS, Serge Kräutle, SS 2019 40

## weitere Pflichtmodule

### Grundlagen der Rechnerarchitektur und -organisation

- Grundlagen der Rechnerarchitektur und -organisation, 5 ECTS, Dietmar Fey, Marc Reichenbach, SS 2019 42

### Software-Entwicklung in Großprojekten

- Software-Entwicklung in Großprojekten, 5 ECTS, Francesca Saglietti, WS 2019/2020 44

### Berechenbarkeit und Formale Sprachen

- Berechenbarkeit und Formale Sprachen, 7.5 ECTS, Rolf Wanka, WS 2019/2020 46

### Theorie der Programmierung

- Theorie der Programmierung, 7.5 ECTS, Lutz Schröder, Christoph Rauch, SS 2019 48

### Algorithmik kontinuierlicher Systeme

- Algorithmik kontinuierlicher Systeme, 7.5 ECTS, Ulrich Rude, SS 2019 50

### Implementierung von Datenbanksystemen

- Implementierung von Datenbanksystemen, 5 ECTS, Klaus Meyer-Wegener, WS 2019/2020 52

### Mathematik für INF 3

- Mathematik C3, 7.5 ECTS, Serge Kräutle, u. a. Hochschullehrer, WS 2019/2020 54

### Mathematik für INF 4

- Mathematik C4, 7.5 ECTS, Wigand Rathmann, SS 2019 56

## Seminare, Praktika, Bachelorarbeit

### Seminar

- Software Architecture (PROJ 5-ECTS), 5 ECTS, Martin Jung, Dirk Riehle, SS 2019 58
- "Hallo Welt!" für Fortgeschrittene, 5 ECTS, N.N, SS 2019 60
- Seminar Theoretische Informatik, 5 ECTS, Lutz Schröder, SS 2019 62
- Systems- and Networks-on-a-Chip für INF, 5 ECTS, Stefan Wildermann, Behnaz Pourmohseni, Hananeh Aliee, WS 2019/2020 64
- Design Patterns und Anti-Patterns, 5 ECTS, N.N, SS 2019 66
- Nailing your Thesis (PROJ 5-ECTS), 5 ECTS, Dirk Riehle, SS 2019 67
- Einführung in die Kryptografie, 5 ECTS, N.N, WS 2019/2020 69
- Algorithmen der Simulationstechnik, 5 ECTS, Ulrich Rude, WS 2019/2020 70
- Themen der Kategorientheorie, 5 ECTS, Stefan Milius, SS 2019 71
- Seminar Energieinformatik (B.Sc.), 5 ECTS, Marco Pruckner, WS 2019/2020 73
- Product Management (VUE 5-ECTS), 5 ECTS, Dirk Riehle, WS 2019/2020 74
- Free/Libre, and Open Source Software (5-ECTS), 5 ECTS, Dirk Riehle, SS 2019 76
- Seminar Graphische Datenverarbeitung, 5 ECTS, Günther Greiner, Frank Bauer, SS 2019 78
- Nailing your Thesis (VUE 5-ECTS), 5 ECTS, Dirk Riehle, SS 2019 79
- Seminar Wissensrepräsentation und -verarbeitung, 5 ECTS, Michael Kohlhase, Dennis Müller, SS 2019 81
- Product Management (PROJ 5-ECTS), 5 ECTS, Dirk Riehle, WS 2019/2020 82
- Ausgewählte Kapitel aus "The Art of Computer Programming", 5 ECTS, Oliver Keszöcze, SS 2019 84
- Seminar Kommunikationssysteme (B.Sc.), 5 ECTS, N.N, SS 2019 86
- Hardware-based Cryptography, 5 ECTS, Dominique Schröder, Marc Reichenbach, SS 2019 87
- Algorithmische Schönheiten - Algorithms Unplugged, 5 ECTS, Rolf Wanka, SS 2019 88
- Seminar nominale Mengen und Automate, 5 ECTS, Lutz Schröder, SS 2019 89
- Seminar Machine Learning and Data Analytics for Industry 4.0, 5 ECTS, An Nguyen, SS 2019 91
- Seminar Automatische Analyse von Stimm-, Sprech- und Sprachstörungen bei Sprachpathologien, 5 ECTS, Tino Haderlein, Elmar Nöth, SS 2019 93
- Advanced Design and Programming (5-ECTS), 5 ECTS, Dirk Riehle, WS 2019/2020 94
- Neuartige Rechnerarchitekturen, 5 ECTS, Marc Reichenbach, Philipp Holzinger, Dietmar Fey, SS 2019 96
- Advanced Competitive Programming, 5 ECTS, Paul Wild, WS 2019/2020 98
- IT-Sicherheits-Seminar (Bachelor), 5 ECTS, Felix Freiling, WS 2019/2020 100

### Blender Seminar

#### Free/Libre, and Open Source Software (5-ECTS)

#### Hardware-based Cryptography

#### Konzepte von Betriebssystem-Komponenten

#### Machine Learning [5 ECTS]

- Machine Learning, 5 ECTS, Christopher Mutschler, WS 2019/2020 101

#### Nailing your Thesis (PROJ 5-ECTS)

#### Nailing your Thesis (VUE 5-ECTS)

#### Neuartige Rechnerarchitekturen

#### Seminar 'Hallo Welt!' für Fortgeschrittene

#### Seminar Algorithmische Schönheiten - Algorithms Unplugged

#### Seminar Design Patterns und Anti-Patterns

#### Seminar Kommunikationssysteme (B.Sc.)

- Seminar Kommunikationssysteme (B.Sc.), 5 ECTS, N.N, WS 2019/2020 103

<b>Seminar Machine Learning Algorithms</b>	
<b>Seminar Machine Learning and Data Analytics for Industry 4.0</b>	
• Seminar Machine Learning and Data Analytics for Industry 4.0, 5 ECTS, Björn Eskofier, An Nguyen, Franz Köferl, Philipp Schlieper, WS 2019/2020	104
<b>Seminar Sprachtechnologie für Sprachpathologien</b>	
• Seminar Automatische Analyse von Stimm-, Sprech- und Sprachstörungen bei Sprachpathologien, 5 ECTS, Elmar Nöth, Tino Haderlein, WS 2019/2020	106
<b>Seminar Theoretische Informatik</b>	
• Seminar Theoretische Informatik, 5 ECTS, Lutz Schröder, WS 2019/2020	108
<b>Seminar Wissensrepräsentation und -verarbeitung</b>	
• Seminar Wissensrepräsentation und -verarbeitung, 5 ECTS, Michael Kohlhase, Dennis Müller, WS 2019/2020	110
<b>Seminar nominale Mengen und Automaten</b>	
<b>Seminar: Ausgewählte Kapitel aus "The Art of Computer Programming"</b>	
<b>Software Architecture (PROJ 5-ECTS)</b>	
<b>Themen der Kategorientheorie</b>	
<b>Praktikum Informatik</b>	
• Nailing your Thesis (VUE+PROJ 10-ECTS), 10 ECTS, Dirk Riehle, SS 2019	111
• Software Architecture (VUE+PROJ 10-ECTS), 10 ECTS, Martin Jung, Dirk Riehle, SS 2019	113
• Product Management (VUE+PROJ 10-ECTS), 10 ECTS, Dirk Riehle, WS 2019/2020	115
• Praktikum Software Engineering: Verfahren und Werkzeuge, 10 ECTS, Marc Spisländer, Xiaochen Wu, SS 2019	117
• Software-Anwendungen mit KI (VUE 10-ECTS), 10 ECTS, Dirk Riehle, SS 2019	118
• IoT Security, 10 ECTS, Philipp Klein, WS 2019/2020	120
• Praktikum Enterprise Computing, 10 ECTS, Peter Wilke, SS 2019	122
• Praktikum angewandte Systemsoftwaretechnik, 10 ECTS, N.N, SS 2019	123
• Praktikum: Entwicklung interaktiver eingebetteter Systeme, 10 ECTS, Joachim Falk, Mehmet Akif Özkan, WS 2019/2020	125
• Hackerpraktikum (Bachelor), 10 ECTS, Tilo Müller, WS 2019/2020	127
• Supercomputing Praktikum, 10 ECTS, Johannes Hofmann, SS 2019	129
• Praktikum Mustererkennung, 10 ECTS, Andreas Maier, Vincent Christlein, WS 2019/2020	130
• HPC Software Projekt, 10 ECTS, Harald Köstler, SS 2019	131
• Praktikum: Lego Mindstorms, 10 ECTS, Stefan Wildermann, SS 2019	132
<b>Grafik-Praktikum Game Programming</b>	
• GraPra, 10 ECTS, Alexander Lier, WS 2019/2020	134
<b>HPC Software Projekt</b>	
<b>Nailing your Thesis (VUE+PROJ 10-ECTS)</b>	
<b>Praktikum "Smart Cameras"</b>	
<b>Praktikum Enterprise Computing</b>	
<b>Praktikum Lego Mindstorms</b>	
<b>Praktikum Software Engineering in der Praxis</b>	
<b>Praktikum Software Engineering: Verfahren und Werkzeuge</b>	
• Praktikum Software Engineering: Verfahren und Werkzeuge, 10 ECTS, N.N, WS 2019/2020	136
<b>Praktikum angewandte Systemsoftwaretechnik</b>	
• Praktikum angewandte Systemsoftwaretechnik, 10 ECTS, N.N, WS 2019/2020	137
<b>Software Architecture (VUE+PROJ 10-ECTS)</b>	

## Software-Anwendungen mit KI (VUE 10-ECTS)

### Supercomputing Praktikum

- Supercomputing Praktikum, 10 ECTS, Alexander Ditter, WS 2019/2020 139

Bachelorarbeit 140

## Wahlpflichtbereich (5. und 6. Semester)

### Wahlpflichtmodule

#### Vertiefungsrichtung Datenbanksysteme

- SWAT-Intensivübungen [als eBT-Aufbau], 5 ECTS, Peter Schwab, Demian Vöhringer, SS 2019 141

#### Database Practice with Oracle (DBP)

#### Database Systems II (Option 1)

#### SWAT-Intensivübung

#### eBusiness Technologies und Evolutionäre Informationssysteme

- eBusiness Technologies und Evolutionäre Informationssysteme, 5 ECTS, Christoph P. Neumann, Richard Lenz, Florian Irmert, WS 2019/2020 143

#### Vertiefungsrichtung Diskrete Simulation

- Dienstgüte von Kommunikationssystemen, 5 ECTS, Reinhard German, SS 2019 146

#### Dienstgüte von Kommunikationssystemen (Vorlesung und Übung)

#### Simulation und Modellierung I

- Simulation und Modellierung 1 - VÜ, 5 ECTS, Reinhard German, WS 2019/2020 148

#### Vertiefungsrichtung Elektronik und Informationstechnik

- Satellitenkommunikation, 5 ECTS, Christian Rohde, SS 2019 150

#### Globale Navigationssatellitensysteme

- Globale Navigationssatellitensysteme, 5 ECTS, Jörn Thielecke, WS 2019/2020 152

#### Kommunikationsstrukturen

- Kommunikationsstrukturen, 5 ECTS, Jürgen Frickel, WS 2019/2020 154

#### Satellitengestützte Ortsbestimmung

- Satellitengestützte Ortsbestimmung, 5 ECTS, Jörn Thielecke, WS 2019/2020 156

#### Satellitenkommunikation

- Satellitenkommunikation, 5 ECTS, Christian Rohde, SS 2019 158

#### Vertiefungsrichtung Graphische Datenverarbeitung

- Interaktive Computergraphik und Globale Beleuchtungsberechnung, 10 ECTS, Marc Stamminger, SS 2019 160
- Interaktive Computergraphik, 5 ECTS, Marc Stamminger, SS 2019 161
- Applied Visualization, 5 ECTS, Roberto Grosso, SS 2019 163

#### Applied Visualization

- Applied Visualization, 5 ECTS, Roberto Grosso, SS 2019 165

#### Computer Graphics

- Computergraphik-VU, 5 ECTS, Marc Stamminger, WS 2019/2020 167

#### Computergraphik (Vorlesung mit Übung und Praktikum)

- Computergraphik-VUP, 7.5 ECTS, Marc Stamminger, WS 2019/2020 170

#### Geometric Modeling (Vorlesung mit Übung)

- Geometrische Modellierung - VU, 5 ECTS, Marc Stamminger, Roberto Grosso, WS 2019/2020 172

<b>Informationsvisualisierung</b>	
• Informationsvisualisierung, 5 ECTS, Roberto Grosso, WS 2019/2020	174
<b>Interaktive Computergraphik</b>	
<b>Interaktive Computergraphik und globale Beleuchtungsberechnung</b>	
<b>Visual Computing in Medicine</b>	
• Visual Computing in Medicine, 5 ECTS, Peter Hastreiter, Thomas Wittenberg, WS 2019/2020, 2 Sem.	176
<b>Vertiefungsrichtung Hardware-Software-Co-Design</b>	
• Verifikation digitaler Systeme, 5 ECTS, Oliver Keszöcze, SS 2019	179
• Cyber-Physical Systems, 5 ECTS, Torsten Klie, SS 2019	181
• Hardware-Software-Co-Design, 5 ECTS, Jürgen Teich, SS 2019	183
• Approximate Computing, 5 ECTS, Oliver Keszöcze, Jürgen Teich, SS 2019	185
• Parallele Systeme mit erweiterter Übung, 7.5 ECTS, Jürgen Teich, Frank Hannig, SS 2019	186
• Hardware-Software-Co-Design mit erweiterter Übung, 7.5 ECTS, Jürgen Teich, SS 2019	189
• Parallele Systeme, 5 ECTS, Jürgen Teich, Frank Hannig, SS 2019	191
• Security in Embedded Hardware, 5 ECTS, Jürgen Teich, SS 2019	194
• Entwurf und Analyse eingebetteter Netzwerke des Automobilbaus, 5 ECTS, Hritam Dutta, Fedor Smirnov, SS 2019	196
<b>Cyber-Physical Systems</b>	
<b>Cyber-Physical Systems</b>	
• Cyber-Physical Systems, 5 ECTS, Torsten Klie, WS 2019/2020	198
• Cyber-Physical Systems, 5 ECTS, Torsten Klie, SS 2019	200
<b>Cyber-Physical Systems</b>	
• Cyber-Physical Systems, 5 ECTS, Torsten Klie, WS 2019/2020	202
<b>Eingebettete Systeme (Vorlesung mit erweiterten Übungen)</b>	
• Eingebettete Systeme mit erweiterter Übung, 7.5 ECTS, Jürgen Teich, Frank Hannig, WS 2019/2020	204
<b>Eingebettete Systeme (Vorlesung mit Übungen)</b>	
• Eingebettete Systeme, 5 ECTS, Jürgen Teich, WS 2019/2020	206
<b>Entwurf und Analyse eingebetteter Netzwerke des Automobilbaus</b>	
<b>Ereignisgesteuerte Systeme</b>	
• Ereignisgesteuerte Systeme, 5 ECTS, Stefan Wildermann, WS 2019/2020	208
<b>Hardware-Software-Co-Design</b>	
<b>Hardware-Software-Co-Design (Vorlesung mit erweiterter Übung)</b>	
<b>Organic Computing</b>	
• Organic Computing, 5 ECTS, Rolf Wanka, SS 2019	210
<b>Parallele Systeme</b>	
<b>Parallele Systeme</b>	
• Parallele Systeme, 5 ECTS, Jürgen Teich, Frank Hannig, SS 2019	212
<b>Parallele Systeme (Vorlesung mit erweiterten Übungen)</b>	
<b>Reconfigurable Computing</b>	
<b>Reconfigurable Computing (Lecture with Exercises)</b>	
• Reconfigurable Computing, 5 ECTS, Jürgen Teich, WS 2019/2020	215
<b>Reconfigurable Computing (Lecture with Extended Exercises)</b>	
• Reconfigurable Computing with Extended Exercises, 7.5 ECTS, Jürgen Teich, WS 2019/2020	217
<b>Security in Embedded Hardware</b>	

<b>Verifikation digitaler Systeme</b>	
• Verifikation digitaler Systeme, 5 ECTS, N.N, WS 2019/2020	220
<b>Vertiefungsrichtung IT-Sicherheit</b>	
• Cryptocurrencies, 5 ECTS, Dominique Schröder, SS 2019	222
• Software Reverse Engineering V+Ü, 5 ECTS, Tilo Müller, SS 2019	224
• Forensische Informatik, 5 ECTS, Felix Freiling, Christian Moch, Stefan Vömel, SS 2019	226
• Security in Embedded Hardware, 5 ECTS, Jürgen Teich, SS 2019	227
<b>Angewandte IT-Sicherheit</b>	
• Angewandte IT-Sicherheit, 5 ECTS, Felix Freiling, WS 2019/2020	229
<b>Cryptocurrencies</b>	
<b>Einführung in die IT-Sicherheit (EinfITSec)</b>	
• Einführung in die IT-Sicherheit, 5 ECTS, Felix Freiling, WS 2019/2020	230
<b>Forensische Informatik (Vorlesung mit Übung)</b>	
<b>Security and Privacy in Pervasive Computing (Vorlesung mit Übung)</b>	
• Security and Privacy in Pervasive Computing, 5 ECTS, Zinaida Benenson, WS 2019/2020	232
<b>Security in Embedded Hardware</b>	
<b>Software Reverse Engineering V+Ü</b>	
<b>Vertiefungsrichtung Kommunikationssysteme</b>	
• Fahrzeugkommunikation, 5 ECTS, Kai-Steffen Jens Hielscher, SS 2019	234
• Smart Grids und Elektromobilität, 5 ECTS, Marco Pruckner, SS 2019	236
• Fahrzeugkommunikation, 2.5 ECTS, Kai-Steffen Jens Hielscher, SS 2019	238
• Human Computer Interaction, 5 ECTS, Björn Eskofier, SS 2019	240
• Automotive Systems & Software Engineering, 5 ECTS, Christian Allmann, SS 2019	242
• Dienstgüte von Kommunikationssystemen, 5 ECTS, Reinhard German, SS 2019	244
<b>Automotive Systems and Software Engineering (Vorlesung mit Übung)</b>	
<b>Dienstgüte von Kommunikationssystemen (Vorlesung und Übung)</b>	
<b>Fahrzeugkommunikation</b>	
<b>Fahrzeugkommunikation (Vorlesung mit Übung)</b>	
<b>Human Computer Interaction</b>	
<b>Kommunikationssysteme</b>	
• Kommunikationssysteme-VÜ, 5 ECTS, Reinhard German, WS 2019/2020	246
<b>Modellierung, Optimierung und Simulation von Energiesystemen</b>	
• Modellierung, Optimierung und Simulation von Energiesystemen, 5 ECTS, Marco Pruckner, WS 2019/2020	248
<b>Smart Grid</b>	
<b>Vertiefungsrichtung Künstliche Intelligenz</b>	
• Künstliche Intelligenz II, 7.5 ECTS, Michael Kohlhase, SS 2019	250
• Logik-Basierte Wissensrepräsentation für Mathematisch/Technisches Wissen, 5 ECTS, Michael Kohlhase, SS 2019	252
<b>Künstliche Intelligenz I</b>	
• Künstliche Intelligenz I, 7.5 ECTS, Michael Kohlhase, WS 2019/2020	253
<b>Künstliche Intelligenz II</b>	
<b>Logik-Basierte Sprachverarbeitung</b>	
• Logik-Basierte Sprachverarbeitung, 5 ECTS, Michael Kohlhase, Dennis Müller, WS 2019/2020	255
<b>Logik-basierte Wissensrepräsentation für mathematisch/technisches Wissen</b>	



<b>Ontologien im Semantic Web</b>	
• Ontologien im Semantic Web, 7.5 ECTS, Lutz Schröder, WS 2019/2020	257
<b>Vertiefungsrichtung Medieninformatik</b>	
• Multimedia-Technik, 5 ECTS, Klaus Meyer-Wegener, SS 2019	259
• Human Computer Interaction, 5 ECTS, Björn Eskofier, SS 2019	261
• Gestalterische Grundlagen der Medien, 5 ECTS, Wolfgang Keller, WS 2019/2020	263
<b>Digitale Bildgestaltung</b>	
<b>Human Computer Interaction</b>	
<b>Multimedia-Technik</b>	
<b>Music Processing Analysis - Lecture and Exercise</b>	
• Music Processing Analysis - Lecture & Exercise, 5 ECTS, Meinard Müller, WS 2019/2020	264
<b>Vertiefungsrichtung Mustererkennung</b>	
• Interventional Medical Image Processing (Online-Kurs), 5 ECTS, Andreas Maier, SS 2019	267
• Human Computer Interaction, 5 ECTS, Björn Eskofier, SS 2019	269
• Diagnostic Medical Image Processing (VHB-Kurs), 5 ECTS, Andreas Maier, SS 2019	271
• IT-Modernisierung, 5 ECTS, Peter Wilke, SS 2019	273
<b>Biomedizinische Signalanalyse</b>	
• Biomedizinische Signalanalyse, 5 ECTS, Felix Kluge, Björn Eskofier, WS 2019/2020	275
<b>Computer Vision with Exercises</b>	
<b>Diagnostic Medical Image Processing</b>	
• Diagnostic Medical Image Processing (VHB-Kurs), 5 ECTS, Andreas Maier, WS 2019/2020	278
<b>Diagnostic Medical Image Processing mit Übung</b>	
<b>Human Computer Interaction</b>	
<b>IT-Modernisierung</b>	
<b>Interventional Medical Image Processing</b>	
• Interventional Medical Image Processing (Online-Kurs), 5 ECTS, Andreas Maier, WS 2019/2020	280
<b>Interventional Medical Image Processing mit Übung</b>	
<b>Introduction to Pattern Recognition</b>	
<b>Introduction to Pattern Recognition</b>	
<b>Maschinelles Lernen für Zeitreihen</b>	
• Maschinelles Lernen für Zeitreihen, 5 ECTS, Björn Eskofier, Oliver Amft, WS 2019/2020	282
<b>Maschinelles Lernen für Zeitreihen Deluxe</b>	
• Maschinelles Lernen für Zeitreihen Deluxe, 7.5 ECTS, Björn Eskofier, Oliver Amft, WS 2019/2020	284
<b>Music Processing Analysis - Lecture and Exercise</b>	
• Music Processing Analysis - Lecture & Exercise, 5 ECTS, Meinard Müller, WS 2019/2020	264
<b>Pattern Recognition</b>	
• Pattern Recognition, 5 ECTS, Elmar Nöth, Sebastian Käßler, WS 2019/2020	286
<b>Pattern Recognition (Lecture + Exercises)</b>	
• Pattern Recognition Deluxe, 7.5 ECTS, Elmar Nöth, Sebastian Käßler, WS 2019/2020	288
<b>Vertiefungsrichtung Programmiersysteme</b>	
• Programmierung und Architekturen von Cluster-Rechnern, 7.5 ECTS, Dietmar Fey, Thorsten Blaß, SS 2019	291
• Analyse und Design objektorientierter Softwaresysteme mit UML, 5 ECTS, Detlef Kips, Martin Jung, SS 2019	293

• Testen von Softwaresystemen, 5 ECTS, Norbert Oster, Klaudia Dussa-Zieger, SS 2019	295
<b>Analyse und Design objektorientierter Softwaresysteme mit der Unified Modeling Language (UML)</b>	
<b>Geschichte der Programmiersprachen</b>	
<b>Grundlagen des Übersetzerbaus</b>	
• Grundlagen des Übersetzerbaus, 7.5 ECTS, Michael Philippsen, WS 2019/2020	297
<b>Parallele Algorithmen</b>	
<b>Programmierung und Architekturen von Cluster-Rechnern</b>	
• Programmierung und Architekturen von Cluster-Rechnern, 7.5 ECTS, Dietmar Fey, Thorsten Blaß, SS 2019	300
<b>Testen von Softwaresystemen</b>	
<b>Vertiefungsrichtung Rechnerarchitektur</b>	
• CPU Entwurf mit VHDL, 7.5 ECTS, Marc Reichenbach, SS 2019	302
• Architectures of Supercomputers / Architekturen von Superrechnern, 5 ECTS, Johannes Hofmann, Dietmar Fey, WS 2019/2020	304
• FPGA-Online Basic Course with VHDL, 5 ECTS, Marc Reichenbach, Dietmar Fey, SS 2019	306
• CPU Entwurf mit VHDL (Schwerpunkt Prozessorentwurf), 5 ECTS, Marc Reichenbach, SS 2019	308
• Heterogene Rechnerarchitekturen Online, 5 ECTS, Marc Reichenbach, Thomas Heller, Johannes Hofmann, SS 2019	310
• CPU Entwurf mit VHDL (Schwerpunkt VHDL), 5 ECTS, Marc Reichenbach, SS 2019	312
• Programmierung und Architekturen von Cluster-Rechnern, 7.5 ECTS, Dietmar Fey, Thorsten Blaß, SS 2019	314
<b>Architectures of Supercomputers (with Exercises)</b>	
<b>CPU Entwurf mit VHDL (CPU)</b>	
<b>CPU Entwurf mit VHDL (Schwerpunkt Prozessorentwurf)</b>	
<b>CPU Entwurf mit VHDL (Schwerpunkt VHDL)</b>	
<b>Einführung digitaler ASIC Entwurf</b>	
<b>Einführung digitaler ASIC Entwurf (Vorlesung mit Übung)</b>	
<b>FPGA-Online Basic Course with VHDL</b>	
• FPGA-Online Basic Course with VHDL, 5 ECTS, Dietmar Fey, Michael Schmidt, Marc Reichenbach, WS 2019/2020	316
<b>FPGA-Online Basic Course with VHDL</b>	
<b>Heterogene Rechnerarchitekturen Online</b>	
• Heterogene Rechnerarchitekturen Online, 5 ECTS, Marc Reichenbach, Thomas Heller, Johannes Hofmann, WS 2019/2020	318
<b>Programmierung und Architekturen von Cluster-Rechnern</b>	
• Programmierung und Architekturen von Cluster-Rechnern, 7.5 ECTS, Dietmar Fey, Thorsten Blaß, SS 2019	300
<b>Rechnerarchitektur (Vorlesung mit Übung und Rechnerübung)</b>	
• Rechnerarchitektur (Vorlesung mit Übung und Rechnerübung), 7.5 ECTS, Dietmar Fey, WS 2019/2020	320
<b>Rechnerarchitektur (Vorlesung mit Übung)</b>	
• Rechnerarchitektur, 5 ECTS, Dietmar Fey, WS 2019/2020	322
<b>Virtuelle Maschinen (Vorlesung mit Übung und Laborübung)</b>	
• Virtuelle Maschinen - V+EÜ, 7.5 ECTS, Volkmar Sieh, WS 2019/2020	324

## **Virtuelle Maschinen (Vorlesung mit Übungen)**

- Virtuelle Maschinen - V+Ü, 5 ECTS, Volkmar Sieh, WS 2019/2020 326

## **Vertiefungsrichtung Software Engineering**

- Software Architecture (PROJ 5-ECTS), 5 ECTS, Martin Jung, Dirk Riehle, SS 2019 328
- IT-Modernisierung, 5 ECTS, Peter Wilke, SS 2019 330
- Analyse und Design objektorientierter Softwaresysteme mit UML, 5 ECTS, Detlef Kips, Martin Jung, SS 2019 332
- Nailing your Thesis (PROJ 5-ECTS), 5 ECTS, Dirk Riehle, SS 2019 334
- Free/Libre, and Open Source Software (5-ECTS), 5 ECTS, Dirk Riehle, SS 2019 336
- Product Management (VUE 5-ECTS), 5 ECTS, Dirk Riehle, WS 2019/2020 338
- Nailing your Thesis (VUE 5-ECTS), 5 ECTS, Dirk Riehle, SS 2019 340
- Testen von Softwaresystemen, 5 ECTS, Norbert Oster, Klaudia Dussa-Zieger, SS 2019 342
- Organisation und Qualitätskontrolle im Software Engineering - Option A, 5 ECTS, Francesca Saglietti, SS 2019, 2 Sem. 344
- Product Management (PROJ 5-ECTS), 5 ECTS, Dirk Riehle, WS 2019/2020 346
- Software-Projektmanagement, 5 ECTS, Bernd Hindel, WS 2019/2020 348
- Organisation und Qualitätskontrolle im Software Engineering - Option B, 5 ECTS, N.N, SS 2019, 2 Sem. 350
- Erweiterte Grundlagen des Software Engineering - Option A, 10 ECTS, Francesca Saglietti, SS 2019, 2 Sem. 352
- Konstruktives Software Engineering, 5 ECTS, Francesca Saglietti, SS 2019 354
- Software-Anwendungen mit KI (VUE 10-ECTS), 10 ECTS, Dirk Riehle, SS 2019 356
- Praktische Softwaretechnik, 5 ECTS, Detlef Kips, Martin Jung, Bernd Hindel, Dirk Riehle, Norbert Oster, WS 2019/2020 358
- Advanced Design and Programming (5-ECTS), 5 ECTS, Dirk Riehle, WS 2019/2020 360
- Product Management (VUE+PROJ 10-ECTS), 10 ECTS, Dirk Riehle, WS 2019/2020 362
- Grundlagen des Software Engineering, 7.5 ECTS, Francesca Saglietti, SS 2019 364
- Erweiterte Grundlagen des Software Engineering - Option B, 10 ECTS, Francesca Saglietti, SS 2019, 2 Sem. 366
- Bewertung der Softwarezuverlässigkeit, 5 ECTS, Francesca Saglietti, Maria Bonner, SS 2019 368
- Software Architecture (VUE+PROJ 10-ECTS), 10 ECTS, Martin Jung, Dirk Riehle, SS 2019 369

## **Analyse und Design objektorientierter Softwaresysteme mit der Unified Modeling Language (UML)**

### **Fehlertolerierende Softwarearchitekturen (Vorlesung mit Übung)**

### **Free/Libre, and Open Source Software (5-ECTS)**

### **Grundlagen des Software Engineering**

- Grundlagen des Software Engineering, 7.5 ECTS, Francesca Saglietti, SS 2019 371

### **Grundlagen des Software Engineering mit Seminar Design Patterns und Anti-Patterns**

### **Grundlagen des Software Engineering mit Seminar Einführung in die Kryptografie**

### **Grundlagen des Software Engineering mit prakt. Übungen Software Engineering in der Praxis**

### **IT-Modernisierung**

### **Konstruktives Software Engineering**

### **Nailing your Thesis (PROJ 5-ECTS)**

### **Nailing your Thesis (VUE 5-ECTS)**

### **Organisation und Qualitätskontrolle im modernen Software Engineering (mit Seminar Design Patterns und Anti-Patterns)**

### **Organisation und Qualitätskontrolle im modernen Software Engineering (mit Seminar Einführung in die Kryptografie)**

<b>Organisation und Qualitätskontrolle im modernen Software Engineering (mit prakt. Übungen Software Engineering in der Praxis)</b>	
<b>Software Architecture (PROJ 5-ECTS)</b>	
<b>Software Architecture (VUE+PROJ 10-ECTS)</b>	
<b>Software-Anwendungen mit KI (VUE 10-ECTS)</b>	
<b>Softwarezuverlässigkeit (Vorlesung mit Übung)</b>	
<b>Test- und Analyseverfahren zur Software-Verifikation und Validierung</b>	
<b>Testen von Softwaresystemen</b>	
<b>Vertiefungsrichtung Systemsimulation</b>	
• High End Simulation in Practice, 7.5 ECTS, Harald Köstler, SS 2019	373
• Programming Techniques for Supercomputers (Lecture and Tutorial), 7.5 ECTS, Gerhard Wellein, SS 2019	374
• Programming Techniques for Supercomputers (Lecture), 5 ECTS, Gerhard Wellein, SS 2019	376
<b>Advanced Programming Techniques (Vorlesung mit Übung)</b>	
• Advanced Programming Techniques, 7.5 ECTS, Harald Köstler, WS 2019/2020	378
<b>High End Simulation in Practice (Vorlesung mit Übung)</b>	
<b>Programmiertechniken für Supercomputer</b>	
• Programming Techniques for Supercomputers (Lecture), 5 ECTS, Gerhard Wellein, SS 2019	380
<b>Programming Techniques for Supercomputers (Lecture and Tutorial)</b>	
<b>Simulation und Wissenschaftliches Rechnen</b>	
• Simulation und Wissenschaftliches Rechnen 1, 7.5 ECTS, Ulrich Rüde, Christoph Pflaum, WS 2019/2020	382
<b>Vertiefungsrichtung Theoretische Informatik</b>	
• Organic Computing, 5 ECTS, Rolf Wanka, SS 2019	384
• Introduction to Finite Model Theory, 7.5 ECTS, Tadeusz Litak, WS 2019/2020	386
• Introduction to Dependently Typed Programming, 7.5 ECTS, Sergey Goncharov, WS 2019/2020	388
• Monad-Based Programming, 7.5 ECTS, Sergey Goncharov, SS 2019	390
• Praktische Semantik von Programmiersprachen, 7.5 ECTS, Tadeusz Litak, SS 2019	391
• Randomisierte Algorithmen, 7.5 ECTS, Rolf Wanka, SS 2019	393
• Approximationsalgorithmen, 7.5 ECTS, Rolf Wanka, SS 2019	394
<b>Algebra des Programmierens (Vorlesung mit Übung)</b>	
• Algebra des Programmierens, 7.5 ECTS, Henning Urbat, Stefan Milius, WS 2019/2020	396
<b>Algebraische und Logische Aspekte der Automatentheorie</b>	
• Algebraische und Logische Aspekte der Automatentheorie, 7.5 ECTS, Stefan Milius, WS 2019/2020	398
<b>Approximationsalgorithmen (Vorlesung mit Übung)</b>	
<b>Effiziente kombinatorische Algorithmen (Vorlesung mit Übung)</b>	
• Effiziente kombinatorische Algorithmen, 7.5 ECTS, Rolf Wanka, WS 2019/2020	400
<b>Einführung in die algorithmische Kryptographie</b>	
• Einführung in die moderne Kryptographie, 7.5 ECTS, Dominique Schröder, WS 2019/2020	402
<b>Formale Methoden der Softwareentwicklung</b>	
• Formale Methoden der Softwareentwicklung, 7.5 ECTS, Tadeusz Litak, WS 2019/2020	404
<b>Kommunikation und Parallele Prozesse</b>	
<b>Monad-Based Programming</b>	

## Mündliche Prüfung zu Praktische Semantik von Programmiersprachen

### Ontologien im Semantic Web

- Ontologien im Semantic Web, 7.5 ECTS, Lutz Schröder, WS 2019/2020 257

### Organic Computing

- Organic Computing, 5 ECTS, Rolf Wanka, SS 2019 210

### Randomisierte Algorithmen (Vorlesung mit Übung)

#### Vertiefungsrichtung Verteilte Systeme und Betriebssysteme

- Betriebssystemtechnik, 5 ECTS, Wolfgang Schröder-Preikschat, SS 2019 406
- Betriebssystemtechnik (V + EÜ), 7.5 ECTS, Wolfgang Schröder-Preikschat, SS 2019 408
- Verteilte Systeme - V+Ü, 5 ECTS, Tobias Distler, Jürgen Kleinöder, SS 2019 410
- Automotive Systems & Software Engineering, 5 ECTS, Christian Allmann, SS 2019 412
- Echtzeitsysteme 2 - Verlässliche Echtzeitsysteme (V+Ü), 5 ECTS, Peter Ulbrich, Peter Wagemann, SS 2019 414
- Echtzeitsysteme 2 - Verlässliche Echtzeitsysteme (V+EÜ), 7.5 ECTS, Peter Ulbrich, Peter Wagemann, SS 2019 416
- Verteilte Systeme - V+EÜ, 7.5 ECTS, Tobias Distler, Jürgen Kleinöder, SS 2019 418

### Automotive Systems and Software Engineering (Vorlesung mit Übung)

#### Betriebssysteme (Vorlesung mit erweiterten Übungen)

- Betriebssysteme-V+EÜ, 7.5 ECTS, Volkmar Sieh, Wolfgang Schröder-Preikschat, WS 2019/2020 420

#### Betriebssysteme (Vorlesung mit Übungen)

- Betriebssysteme-V+Ü, 5 ECTS, Volkmar Sieh, Wolfgang Schröder-Preikschat, WS 2019/2020 422

#### Betriebssysteme und Echtzeitsysteme (Vorlesungen mit Übungen)

- Betriebssysteme und Echtzeitsysteme - V+Ü, 10 ECTS, Volkmar Sieh, Peter Ulbrich, WS 2019/2020 424

#### Betriebssystemtechnik (V+EÜ)

#### Betriebssystemtechnik (Vorlesung mit Übungen)

#### Echtzeitsysteme (Vorlesung mit Übungen)

- Echtzeitsysteme-V+Ü, 5 ECTS, Peter Ulbrich, WS 2019/2020 428

#### Echtzeitsysteme 2 - Verlässliche Echtzeitsysteme

#### Echtzeitsysteme mit erweiterten Übungen

- Echtzeitsysteme-V+EÜ, 7.5 ECTS, Peter Ulbrich, WS 2019/2020 432

#### Middleware - Cloud Computing (Vorlesung mit erweiterten Übungen)

- Middleware - Cloud Computing V+EÜ, 7.5 ECTS, Tobias Distler, WS 2019/2020 436

#### Middleware - Cloud Computing (Vorlesung mit Übungen)

- Middleware - Cloud Computing V+Ü, 5 ECTS, Tobias Distler, WS 2019/2020 439

#### Nebenläufige Systeme

- Nebenläufige Systeme, 5 ECTS, Wolfgang Schröder-Preikschat, WS 2019/2020 442

#### Verlässliche Echtzeitsysteme (Vorlesung mit Übungen)

#### Verteilte Systeme (Vorlesung mit erweiterten Übungen)

#### Verteilte Systeme (Vorlesung mit Übungen)

#### Virtuelle Maschinen (Vorlesung mit Übung und Laborübung)

- Virtuelle Maschinen - V+EÜ, 7.5 ECTS, Volkmar Sieh, WS 2019/2020 324

#### Virtuelle Maschinen (Vorlesung mit Übungen)

- Virtuelle Maschinen - V+Ü, 5 ECTS, Volkmar Sieh, WS 2019/2020 326

## Vertiefungsrichtung Informatik in der Bildung

### Vertiefungsrichtung Medizinische Informatik

#### Informationssysteme im Gesundheitswesen

- Informationssysteme im Gesundheitswesen, 5 ECTS, Hans-Ulrich Prokosch, WS 2019/2020 444

#### Technologien zur Prozessunterstützung im Gesundheitswesen

## Nebenfach

Im Bachelor-Studiengang Informatik sind im Rahmen des Nebenfachs Module im Umfang von 15 ECTS zu belegen.

Im Folgenden sind die Nebenfächer dargestellt, die in der FPO Informatik vorgesehen sind bzw. für die ausgearbeitete Modulkataloge vorliegen. Da das UnivIS-Modulverzeichnis noch nicht für alle Fachrichtungen - vor allem außerhalb der Technischen Fakultät - vollständig aufgebaut ist, empfehlen wir, in Zweifelsfällen zusätzlich in den Modulübersichten auf der Nebenfach-Webseite der Informatik-Studiengänge nachzusehen und über die Suche-Funktion (Lehrveranstaltungen) von UnivIS nach den entsprechenden Lehrveranstaltungen zu suchen.

## Nebenfach Astronomie

### Astronomie

## Nebenfach Betriebswirtschaftslehre

Das Nebenfach Betriebswirtschaft besteht aus einem Pflichtbereich im Umfang von 10 ECTS und einem Wahlbereich von 5 ECTS.

### Pflichtbereich

#### E-Commerce

### Wahlbereich

#### Prozess- und Informationsmanagement

#### Mobile service business

#### Marketing-Fallstudien

- Marketing Fallstudien, 5 ECTS, Christian Götz, WS 2019/2020 446

#### Betriebswirtschaftslehre für Ingenieure

- BWL für Ingenieure, 5 ECTS, Kai-Ingo Voigt, WS 2019/2020, 2 Sem. 448

## Nebenfach Biologie

### Allgemeine Biologie I

- Allgemeine Biologie I: Biologie für Nebenfächler, 5 ECTS, Johann Helmut Brandstätter, Ingrid Brehm, Renato Frischknecht, Michael Lebert, Gerald Seidel, WS 2019/2020 450

### Allgemeine Biologie II

- Allgemeine Biologie II: Morphologie und Anatomie der Organismen, 5 ECTS, Michael Lebert, Ralph Rübsam, WS 2019/2020 452

### Zoologie

- Ökologie und Diversität A, 5 ECTS, Jürgen Schmidl, Andreas Feigenspan, WS 2019/2020 454

### Allgemeine Mikrobiologie

#### Mikrobiologische Übungen

- Mikrobiologische Übungen, 5 ECTS, Gerald Seidel, WS 2019/2020 456

## Ökologie und Diversität B

- Ökologie und Diversität B, 5 ECTS, Ruth Stadler, Jürgen Schmidl, Regula Muheim-Lenz, Ulrike Daigl, , SS 2019 457

## Nebenfach Chemie

### Nebenfach Chemie für Informatikstudierende

### Nebenfach Chemie- und Bioingenieurwesen

Im Nebenfach Chemie- und Bioingenieurwesen sind die folgenden drei Module zu belegen.

#### Abfallaufbereitung

- Abfallaufbereitung, 5 ECTS, Stefanos Georgiadis, WS 2019/2020 459

#### Entsorgung medizinischer Abfälle

- Entsorgung medizinischer Abfälle ohne Praktikum, 5 ECTS, Stefanos Georgiadis, SS 2019 461

#### Umweltbioverfahrenstechnik

- Umweltbioverfahrenstechnik ohne Praktikum, 5 ECTS, Roman Breiter, SS 2019 463

## Nebenfach Elektrotechnik, Elektronik und Informationstechnik

### Allgemeine Elektrotechnik

#### Grundlagen der Elektrotechnik I

- Grundlagen der Elektrotechnik I, 7.5 ECTS, Georg Fischer, Christopher Beck, Sarah Linz, N.N., WS 2019/2020 466

#### Grundlagen der Elektrotechnik II

- Grundlagen der Elektrotechnik II, 5 ECTS, Klaus Helmreich, SS 2019 468

#### Praktikum Grundlagen der Elektrotechnik für EEI,ME,BP,INF,Math

- Praktikum Grundlagen der Elektrotechnik für EEI, ME, BP, INF, MATH, 2.5 ECTS, Georg Fischer, Lorenz-Peter Schmidt, Reinhard Lerch, SS 2019, 3 Sem. 470

### Automatisierungstechnik

#### Grundlagen der Elektrotechnik III

- Grundlagen der Elektrotechnik III, 5 ECTS, Stefan J. Rupitsch, WS 2019/2020 472

#### Einführung in die Systemtheorie

- Einführung in die Systemtheorie, 5 ECTS, Knut Graichen, wissenschaftliche Mitarbeiter/innen, SS 2019 474

#### Regelungstechnik A (Grundlagen)

Für Studienbeginner im SS 2011 und SS 2012 des Studiengangs EEI findet Regelungstechnik A im 4. FS statt.

- Regelungstechnik A (Grundlagen), 5 ECTS, Knut Graichen, WS 2019/2020 476

### Informationstechnik

#### Signale und Systeme 1

- Signale und Systeme I, 5 ECTS, André Kaup, Jürgen Seiler, WS 2019/2020 478

#### Signale und Systeme 2

- Signale und Systeme II, 5 ECTS, André Kaup, Christian Herglotz, Fabian Brand, SS 2019 480

#### Information Theory and Coding / Informationstheorie und Codierung

- Information Theory and Coding, 5 ECTS, Ralf Müller, WS 2019/2020 482

## Mikroelektronik

### Halbleiterbauelemente

- Halbleiterbauelemente, 5 ECTS, Tobias Dirnecker, Tobias Stolzke, SS 2019 485
- Halbleiterbauelemente, 5 ECTS, Tobias Dirnecker, Christian Martens, WS 2019/2020 487

### Entwurf integrierter Schaltungen I

- Entwurf Integrierter Schaltungen I, 5 ECTS, Sebastian M. Sattler, WS 2019/2020 489

### Schaltungstechnik

- Schaltungstechnik, 5 ECTS, Robert Weigel, SS 2019 491

## Elektrische Energie- und Antriebstechnik

### Grundlagen der Elektrotechnik I

- Grundlagen der Elektrotechnik I, 7.5 ECTS, Georg Fischer, Christopher Beck, Sarah Linz, N.N., WS 2019/2020 466

### Grundlagen der Elektrotechnik II

- Grundlagen der Elektrotechnik II, 5 ECTS, Klaus Helmreich, SS 2019 468

### Grundlagen der elektrischen Antriebstechnik

### Grundlagen der elektrischen Energieversorgung

## Nebenfach Englische Linguistik

### Seminarmodul L-UF Linguistics

### Basismodul II Linguistics

## Nebenfach Geowissenschaften

### Geowissenschaften für Studierende der Informatik

- Das System Erde, 5 ECTS, Anette Regelous, WS 2019/2020 492
- Der Mensch im geowissenschaftlichen Umfeld II, 5 ECTS, Matthias Göbbels, SS 2019 493

### Der Mensch im geowissenschaftlichen Umfeld I

### Der Mensch im geowissenschaftlichen Umfeld II

### Z-Edu-Geo 1 - Das System Erde

- Das System Erde, 5 ECTS, Anette Regelous, WS 2019/2020 495

## Nebenfach Japanologie

### Nebenfach Japanologie für Informatikstudierende

### Grundlagen Japanologie 1

### Japanisch 1

## Nebenfach Maschinenbau

### Fertigungsautomatisierung und Produktionstechnik

### Produktionstechnik I + II

- Produktionstechnik I + II, 5 ECTS, Marion Merklein, Dietmar Drummer, Jörg Franke, Michael Schmidt, Nico Hanenkamp, WS 2019/2020, 2 Sem. 496

### Ringvorlesung Systemtechnik für Nebenfach Informatik

### Automatisierte Produktionsanlagen

- Automatisierte Produktionsanlagen, 5 ECTS, Jörg Franke, WS 2019/2020 498



## Technische Mechanik

### Dynamik starrer Körper

- Dynamik starrer Körper (3V+2Ü+2T), 7.5 ECTS, Sigrid Leyendecker, WS 2019/2020 500

### Statik und Festigkeitslehre

- Statik und Festigkeitslehre, 7.5 ECTS, Kai Willner, Gunnar Possart, Martin Jerschl, Maximilian Volkan Baloglu, Lucie Spannraft, WS 2019/2020 502

## Qualitätsmanagement und Messtechnik

### Grundlagen der Messtechnik

- Grundlagen der Messtechnik, 5 ECTS, Tino Hausotte, Assistenten, SS 2019 505
- Grundlagen der Messtechnik, 5 ECTS, Tino Hausotte, Andreas Gröschl, Martin Heigl, WS 2019/2020 511

### Wahlpflichtmodule Qualitätsmanagement und Messtechnik für Nebenfach Informatik

- Fertigungsmesstechnik II, 5 ECTS, N.N, SS 2019 517
- Hauptseminar Messtechnik, 2.5 ECTS, Tino Hausotte, Assistenten, SS 2019 520
- Qualitätsmanagement II - Phasenübergreifendes Qualitätsmanagement, 2.5 ECTS, N.N., SS 2019 522
- Rechnergestützte Messtechnik, 5 ECTS, Tino Hausotte, SS 2019 524

### Fertigungsmesstechnik I

- Fertigungsmesstechnik I, 5 ECTS, Tino Hausotte, WS 2019/2020 528

### Fertigungsmesstechnik II

- Fertigungsmesstechnik II, 5 ECTS, N.N, SS 2019 532

### Hauptseminar Messtechnik

- Hauptseminar Messtechnik, 2.5 ECTS, T. Hausotte, Assistenten, WS 2019/2020 535

### Hauptseminar Qualitätsmanagement

#### Phasenübergreifendes Qualitätsmanagement (QM II)

- Qualitätsmanagement II - Phasenübergreifendes Qualitätsmanagement, 2.5 ECTS, N.N., SS 2019 537

### Prozess- und Temperaturmesstechnik

- Prozess- und Temperaturmesstechnik, 5 ECTS, Tino Hausotte, WS 2019/2020 539

### Qualitätsmanagement I - Qualitätstechniken für die Produktentstehung

- Qualitätsmanagement I - Qualitätstechniken für die Produktentstehung, 2.5 ECTS, Tino Hausotte, WS 2019/2020 542

### Rechnergestützte Messtechnik

- Rechnergestützte Messtechnik, 5 ECTS, Tino Hausotte, SS 2019 544

## Grundlagen der Produktentwicklung

### Technische Darstellungslehre 2

- Technische Darstellungslehre II, 2.5 ECTS, Benjamin Gerschütz, SS 2019 548

### Technische Produktgestaltung

- Technische Produktgestaltung, 5 ECTS, Sandro Wartzack, Benjamin Schleich, SS 2019 550

### Grundlagen der Produktentwicklung

- Grundlagen der Produktentwicklung, 7.5 ECTS, Stephan Tremmel, WS 2019/2020 554

## Computerintegrierte Produktion

### Automatisierte Produktionsanlagen

- Automatisierte Produktionsanlagen, 5 ECTS, Jörg Franke, WS 2019/2020 498

<b>Automotive Engineering</b>	
• Automotive Engineering, 2.5 ECTS, Ingo Kriebitzsch, WS 2019/2020	558
<b>Die Werkzeugmaschine als mechatronisches System</b>	
• Die Werkzeugmaschine als mechatronisches System, 2.5 ECTS, Siegfried Russwurm, WS 2019/2020	560
<b>Industrie 4.0 - Anwendungsszenarien und Grundlagen</b>	
• Industrie 4.0 - Anwendungsszenarien in Produktion und Service, 2.5 ECTS, Ulrich Löwen, WS 2019/2020	562
<b>International Supply Chain Management</b>	
• International Supply Chain Management, 5 ECTS, Jörg Franke, WS 2019/2020	564
<b>Programmierung humanoider Roboter</b>	
• Einführung in die Programmierung humanoider Roboter, 5 ECTS, Jörg Franke, Assistenten, WS 2019/2020	566
<b>Technische Grundlagen des ressourcenschonenden und intelligenten Wohnens</b>	
• Technische Grundlagen des ressourcenschonenden und intelligenten Wohnens, 2.5 ECTS, Jörg Franke, Julian Praß, WS 2019/2020	568
<b>Produktionssystematik</b>	
• Produktionssystematik, 5 ECTS, Jörg Franke, SS 2019	570
<b>Nebenfach Mathematik</b>	
Aus dem nachfolgenden Angebot sind Module im Umfang von 15 ECTS zu wählen. Es wird empfohlen, mit den Dozenten Rücksprache zu halten.	
<b>Einführung in die Numerik</b>	
<b>Gewöhnliche Differentialgleichungen</b>	
<b>Elementare Partielle Differentialgleichungen</b>	
<b>Einführung Funktionalanalysis</b>	
<b>Kombinatorische Optimierung</b>	
• Lineare und Kombinatorische Optimierung, 10 ECTS, Dieter Weninger, WS 2019/2020	572
<b>Lineare und konvexe Optimierung</b>	
<b>Diskretisierung und numerische Optimierung</b>	
<b>Optimierung für Ingenieure mit Praktikum</b>	
• Optimierung für Ingenieure (mit Praktikum), 7.5 ECTS, Johannes Hild, Martin Gugat, SS 2019	574
<b>Numerik I für Ingenieure</b>	
• Numerik I für Ingenieure, 5 ECTS, Wilhelm Merz, J. Michael Fried, Nicolas Neuß, u.a., WS 2019/2020	577
<b>Numerik II für Ingenieure</b>	
• Numerik II für Ingenieure, 5 ECTS, Wilhelm Merz, SS 2019	578
<b>Discrete optimization I</b>	
• Discrete Optimization, 5 ECTS, Andreas Bärmann, WS 2019/2020	579
• Diskrete Optimierung I, 5 ECTS, Andreas Bärmann, WS 2019/2020	581
<b>Lineare und Kombinatorische Optimierung</b>	
• Lineare und Kombinatorische Optimierung, 10 ECTS, Dieter Weninger, WS 2019/2020	583
<b>Algebra</b>	
• Algebra, 10 ECTS, Friedrich Knop, WS 2019/2020	585
<b>Angewandte Mathematik</b>	
• Kryptographie I, 10 ECTS, Wolfgang Ruppert, WS 2019/2020	587

<b>Discrete optimization II</b>	
• Discrete Optimization II, 10 ECTS, Alexander Martin, SS 2019	589
<b>Diskretisierung und numerische Optimierung</b>	
• Diskretisierung und numerische Optimierung, 10 ECTS, Martin Burger, SS 2019	591
<b>Elementare Zahlentheorie</b>	
• Elementare Zahlentheorie, 5 ECTS, Christina Birkenhake, WS 2019/2020	593
<b>Funktionalanalysis</b>	
• Funktionalanalysis I, 10 ECTS, Hermann Schulz-Baldes, SS 2019	595
<b>Funktionentheorie I</b>	
• Funktionentheorie I, 5 ECTS, Yasmine Sanderson, SS 2019	597
<b>Geometrie</b>	
• Geometrie, 5 ECTS, Yasmine Sanderson, WS 2019/2020	599
• Topologie, 5 ECTS, Karl-Hermann Neeb, Catherine Meusburger, SS 2019	600
<b>Gewöhnliche Differentialgleichungen</b>	
• Gewöhnliche Differentialgleichungen, 10 ECTS, Andreas Knauf, SS 2019	602
<b>Introduction to Statistics and Statistical Programming</b>	
• Introduction to Statistics and Statistical Programming, 5 ECTS, Christoph Richard, SS 2019	604
<b>Mathematische Modellierung Praktikum</b>	
• Mathematische Modellierung Praxis, 5 ECTS, Nadja Ray, Maria Neuss-Radu, Serge Kräutle, WS 2019/2020	606
<b>Mathematische Modellierung Theorie</b>	
• Mathematische Modellierung Theorie, 5 ECTS, Nadja Ray, WS 2019/2020	608
<b>Nichtlineare Optimierung</b>	
• Nichtlineare Optimierung, 10 ECTS, Michael Stingl, WS 2019/2020	610
<b>Numerische Mathematik</b>	
• Numerische Mathematik, 10 ECTS, Florian Frank, WS 2019/2020	612
<b>Optimierung für Ingenieure</b>	
• Optimierung für Ingenieure, 5 ECTS, Johannes Hild, Martin Gugat, SS 2019	614
<b>Optimization in industry and economy</b>	
• Graph Routing and applications, 5 ECTS, Francisco Javier Zaragoza Martínez, WS 2019/2020	616
• Optimization in Industry and Economy, 5 ECTS, Martin Schmidt, Frauke Liers, Francisco Javier Zaragoza Martínez, WS 2019/2020	617
<b>Partielle Differentialgleichungen I</b>	
• Partielle Differentialgleichungen I, 10 ECTS, Frank Duzaar, WS 2019/2020	619
<b>Robust optimization II</b>	
<b>Robuste Optimierung</b>	
• Einführung in die Robuste Optimierung, 5 ECTS, Frauke Liers, SS 2019	621
<b>Topologie</b>	
• Topologie, 5 ECTS, Karl-Hermann Neeb, Catherine Meusburger, SS 2019	623
<b>Topologie</b>	
<b>Einführung in die Darstellungstheorie</b>	
• Einführung in die Darstellungstheorie, 10 ECTS, Catherine Meusburger, SS 2019	625

## Nebenfach Medizin (Medizinische Informatik)

Im Bachelor-Studiengang Informatik sind im Rahmen des Nebenfachs Medizin (Medizinische Informatik) Module im Umfang von 15 ECTS zu belegen.

Die Module sind nach vier Schwerpunkten gegliedert. In jedem Fall wird empfohlen, das schwerpunktübergreifende Modul "Einführung in die Medizinische Informatik" zu belegen.

Wird eine thematische Vertiefung angestrebt, können die weiteren Module - so weit möglich - aus einem Schwerpunkt gewählt werden. Es ist aber auch zulässig, zur Gewinnung einer größeren thematischen Breite, Module aus verschiedenen Schwerpunkten zu kombinieren.

Das das Modulverzeichnis noch nicht für allen Fachrichtungen vollständig aufgebaut ist, empfehlen wir in den Fällen, in denen hier noch keine UnivIS-Module zugeordnet sind, im Zweifelsfall über die Suche-Funktion (Lehrveranstaltungen) von UnivIS nach den entsprechenden Lehrveranstaltungen zu suchen.

### Schwerpunktübergreifende Module

#### Einführung in die Medizinische Informatik

- Einführung in die Medizinische Informatik für Informatik-Nebenfachstudierende, 5 ECTS, 627  
Wolfgang Rödle, Wolfgang Uter, Werner Adler, Oliver Amft, Meik Kunz, WS 2019/2020

### Schwerpunkt Informationssysteme im Gesundheitswesen

#### Informationssysteme im Gesundheitswesen

- Informationssysteme im Gesundheitswesen, 5 ECTS, Hans-Ulrich Prokosch, WS 444  
2019/2020

#### Technologien zur Prozessunterstützung im Gesundheitswesen

#### Wissensbasierte Systeme in der Medizin 1

- Wissensbasierte Systeme in der Medizin 1, 5 ECTS, Stefan Kraus, Dennis Toddenroth, 629  
Wolfgang Rödle, Hans-Ulrich Prokosch, SS 2019

### Schwerpunkt Physiologie

#### Anatomie und Physiologie für Nichtmediziner

- Grundlagen der Anatomie und Physiologie für Medizintechniker, Naturwissenschaftler und 631  
Ingenieure, 5 ECTS, Clemens Forster, Michael Eichhorn, SS 2019, 2 Sem.

### Schwerpunkt Biometrie

#### Biomedizinische Signalanalyse

- Biomedizinische Signalanalyse, 5 ECTS, Felix Kluge, Björn Eskofier, WS 2019/2020 275

## Nebenfach Nordische Philologie

### Nebenfach Nordische Philologie für Informatikstudierende

- Nebenfach Skandinavistik, 15 ECTS, Karina Brehm, , Patrick Ledderose, WS 2019/2020, 633  
2 Sem.

## Nebenfach Philosophie

### Philosophie systematisch

### Philosophiegeschichte im Überblick

## Nebenfach Physik

### Nebenfach Physik

## Nebenfach Psychologie

### Nebenfach Psychologie

Psychologie als Nebenfach für Informatik

Anwendungsmodul

Einführungsmodul

Grundlagenmodul

Grundlagen der Psychologie für Nichtpsychologen

Nebenfach Rechtswissenschaften

Nebenfach Rechtswissenschaften für Informatikstudierende

- Recht als Nebenfach für Informatikstudierende, 15 ECTS, N.N, WS 2019/2020, 2 Sem. 635

Nebenfach Romanistik

Nebenfach Romanistik für Informatikstudierende

Nebenfach Frankoromanistik für Informatikstudierende

Basismodul Französische Sprachpraxis 1

Basismodul Französische Sprachpraxis 2

Basismodul Einführung in die Französische Literaturwissenschaft

Basismodul Einführung in die Französische Sprachwissenschaft

Basismodul Französische Sprachpraxis 1

Basismodul Französische Sprachpraxis 2

Nebenfach Iberoromanistik für Informatikstudierende

Spanische Sprachpraxis 1

Basismodul Spanische Literaturwissenschaft

Basismodul Spanische Sprachwissenschaft

Nebenfach Itoloromanistik für Informatikstudierende

Basismodul Italienische Sprachpraxis 1

Basismodul Italienische Literaturwissenschaft

Basismodul Italienische Sprachwissenschaft

Nebenfach Sinologie

Modul 1: Modernes Chinesisch 1

Modul 2: Modernes Chinesisch 2

Modernes Chinesisch 1

Modernes Chinesisch 2

Nebenfach Soziologie

Nebenfach Soziologie

Nebenfach Politische Wissenschaften

Nebenfach Politische Wissenschaften für Informatikstudierende

Basismodul Außereuropäische Regionen I

Basismodul Außereuropäische Regionen II  
 Basismodul Internationale Beziehungen I  
 Basismodul Internationale Beziehungen II  
 Basismodul Politische Systeme I  
 Basismodul Politische Systeme II  
 Basismodul Politische Theorie und Ideengeschichte I  
 Basismodul Politische Theorie und Ideengeschichte II  
 Einführung in die Politikwissenschaft  
 Grundlagen der Logik in der Informatik

## Hauptseminar

Blender Seminar  
 Free/Libre, and Open Source Software (5-ECTS)  
 Hardware-based Cryptography  
 Konzepte von Betriebssystem-Komponenten  
 Machine Learning [5 ECTS]  
     • Machine Learning, 5 ECTS, Christopher Mutschler, WS 2019/2020 101  
 Nailing your Thesis (PROJ 5-ECTS)  
 Nailing your Thesis (VUE 5-ECTS)  
 Neuartige Rechnerarchitekturen  
 Seminar 'Hallo Welt!' für Fortgeschrittene  
 Seminar Algorithmische Schönheiten - Algorithms Unplugged  
 Seminar Design Patterns und Anti-Patterns  
 Seminar Kommunikationssysteme (B.Sc.)  
     • Seminar Kommunikationssysteme (B.Sc.), 5 ECTS, N.N, WS 2019/2020 103  
 Seminar Machine Learning Algorithms  
 Seminar Machine Learning and Data Analytics for Industry 4.0  
     • Seminar Machine Learning and Data Analytics for Industry 4.0, 5 ECTS, Björn Eskofier, 104  
       An Nguyen, Franz Köferl, Philipp Schlieper, WS 2019/2020  
 Seminar Sprachtechnologie für Sprachpathologien  
     • Seminar Automatische Analyse von Stimm-, Sprech- und Sprachstörungen bei Sprachpa- 106  
       thologien, 5 ECTS, Elmar Nöth, Tino Haderlein, WS 2019/2020  
 Seminar Theoretische Informatik  
     • Seminar Theoretische Informatik, 5 ECTS, Lutz Schröder, WS 2019/2020 108  
 Seminar Wissensrepräsentation und -verarbeitung  
     • Seminar Wissensrepräsentation und -verarbeitung, 5 ECTS, Michael Kohlhase, Dennis 110  
       Müller, WS 2019/2020  
 Seminar nominale Mengen und Automaten  
 Seminar: Ausgewählte Kapitel aus "The Art of Computer Programming"  
 Software Architecture (PROJ 5-ECTS)  
 Themen der Kategorientheorie

---

**Modulbezeichnung: Algorithmen und Datenstrukturen (AuD)** **10 ECTS**  
 (Algorithms and Data Structures)

Modulverantwortliche/r: Michael Philippsen  
 Lehrende: Felix Freiling, Norbert Oster

---

Startsemester: WS 2019/2020	Dauer: 1 Semester	Turnus: jährlich (WS)
Präsenzzeit: 120 Std.	Eigenstudium: 180 Std.	Sprache: Deutsch

---

**Lehrveranstaltungen:**

Algorithmen und Datenstrukturen (WS 2019/2020, Vorlesung, 4 SWS, Felix Freiling)  
 Tafelübungen zu Algorithmen und Datenstrukturen (WS 2019/2020, Übung, 2 SWS, Norbert Oster et al.)  
 Rechnerübungen zu Algorithmen und Datenstrukturen (WS 2019/2020, Übung, 2 SWS, Norbert Oster et al.)

---

**Inhalt:**

- Grundlagen der Programmierung
- Datenstrukturen
- Objektorientierung
- JAVA-Grundkenntnisse
- Aufwandsabschätzungen
- Grundlegende Algorithmen

**Lernziele und Kompetenzen:**

**A - Fachkompetenz:**

Die Studierenden...

1.) Grundlagen der Programmierung in Java

- interpretieren Syntaxdiagramme für grundlegende Programmstrukturen und übertragen diese in entsprechenden Java-Code
- deklarieren und verwenden Variablen mit adäquatem Java-Datentyp (primitive Typen, Reihungen, Zeichenketten)
- überprüfen die Zulässigkeit der Variablendeklaration und -Wertzuweisung nach Java-Typ-Regeln
- bestimmen den Datentyp und den Wert eines Java-Ausdrucks mit primitivem Datentyp und zugehörigen Operatoren
- überführen einfache mathematische Ausdrücke in Java-Code
- werten zusammengesetzte Bedingungen nach den Regeln der strikten bzw. faulen Auswertung für Java aus
- konzipieren zu einer gegebenen Aufgabenstellung einen Algorithmus
- implementieren einfache Algorithmen in Java unter Verwendung verschiedener Kontrollstrukturen
- bestimmen die Gültigkeitsbereiche der Variablen anhand der Blockstruktur eines Java-Programms
- strukturieren Java-Code in Methoden und entwickeln wiederverwendbare Funktionen

2.) Rekursion

- beurteilen den Typ der Rekursion für gegebene Java-Methoden
- entwerfen rekursive Algorithmen zu einer gegebenen Problemstellung unter Anwendung des Induktionssprinzips, des Teile-und-Herrsche-Prinzips sowie des Rücksetzverfahrens und implementieren diese jeweils in Java
- entwickeln effizientere Lösungen, indem sie rekursive Methoden in endrekursive bzw. iterative Methoden umwandeln, implementieren diese jeweils in Java-Code und bewerten deren Laufzeit- und Speicherverbrauch
- bewerten und verbessern rekursive Lösungen unter Verwendung von Dynamischer Programmierung und implementieren diese in Java-Code

3.) Aufwandsanalyse

- analysieren den Laufzeitaufwand und den Speicherbedarf verschiedener Implementierungen
- klassifizieren den asymptotischen Laufzeitaufwand anhand der Komplexitätsklassen des O-Kalküls

- unterscheiden verschiedene Sortierverfahren (Blasensortierung, Sortieren durch Auswählen/Einfügen, Haldensortierung, Sortieren durch Verschmelzen/Zerlegen/Fachverteilen) hinsichtlich ihres Laufzeit- und Speicherplatzbedarfs
- 4.) Objekt-Orientierte Programmierung in Java
- implementieren Java-Klassen gemäß textueller oder graphischer (UML) Spezifikation
  - wenden Verfahren zur systematischen Ableitung von Klassen und Attributen (Hauptwortextraktion), ihren statischen Beziehungen (Vererbung, Polymorphie, Assoziationen) und ihrem dynamischen Zusammenspiel (CRC, Kollaboration) aus einer textuellen Problemstellung an und entwickeln so kleine objekt-orientierte Java-Programme
  - instantiiieren Klassen und verwenden Objektvariablen sachgerecht
  - unterscheiden statische und dynamische Bindung gemäß Polymorphie-Konzept von Java und wenden die Erkenntnisse sachgerecht bei der Entwicklung eigener Applikationen an
- 5.) Robustes Programmieren
- wenden Checklisten an, um typische Programmierfehler im Vorfeld zu vermeiden oder nach der Programmierung zu identifizieren
  - benutzen verschiedenen Möglichkeiten zur Absicherung gegen Fehlersituationen und zur Fehlerrückmeldung (Rückgabewert, Ausnahmebehandlung)
  - wenden Junit zum Testen von Java-Programmen an
  - setzen Verfahren und Werkzeuge zur systematischen Lokalisierung und Behebung von Programmfehlern an (Debugging) und verbessern ihre Lösungen auf diese Weise iterativ
- 6.) Elementare Datentypen
- übertragen eine Spezifikation in Form eines Abstrakten Datentyps (ADT) in ein gleichwertiges Java-Modul
  - erstellen eine formale Spezifikation eines Datentyps in Form eines Abstrakten Datentyps (ADT) aus einer gegebenen textuellen Beschreibung
  - verstehen die grundlegende Behälterdatentypen (Liste, Stapel, Schlange, Streutabelle) und deren Eigenschaften (insbesondere Laufzeit- und Speicherplatzbedarf ihrer Operationen)
  - verwenden generische Behälterdatentypen sachgerecht in eigenen Programmen
  - kennen die Verfügbarkeit generischer Behälterdatentypen in der Java-API und erschließen sich bei Bedarf selbst neue Datentypen sowie deren Funktionen aus der zugehörigen API-Spezifikation für die Verwendung in eigenen Programmen
- 7.) Bäume und Graphen
- bewerten verschiedene Baum- und Graphdarstellungen hinsichtlich Zeitaufwand und Speicherbedarf
  - unterscheiden und klassifizieren die grundlegenden Baum-Arten (Suchbaum, AVL-Baum, Halde)
  - wenden die Grundoperationen (Einfügen, Suchen, Löschen, ggf. Restrukturieren) anhand von Beispieldaten auf gegebene Bäume artgerecht an
  - implementieren und verwenden verschiedene Baumstrukturen sachgerecht in eigenen Java-Programmen
  - führen verschiedene Durchlaufmöglichkeiten (Tiefensuche (DFS), Breitensuche (BFS)) für Graphen und Bäume auf Beispieldaten aus und setzen diese zielführend in eigenen Java-Programmen ein
  - wenden grundlegende Graphalgorithmen (Dijkstra, Floyd, Prim, Kruskal) auf Beispieldaten an und implementieren diese Verfahren in Java-Code

### **B - Selbst- und Sozialkompetenz:**

Die Studierenden...

- organisieren sich selbständig zu Gruppen und koordinieren in gegenseitiger Absprache den organisatorischen und technischen Ablauf der Gruppenarbeiten
- kommunizieren und erarbeiten gemeinsam Lösungen für theoretische Fragestellungen und praktische Programmieraufgaben in Rahmen von Gruppenaufgaben
- planen und wenden zielgerichtet Maßnahmen zu gegenseitigen Qualitätssicherung der eingereichten Lösungen an (prüfen wechselseitig die Gruppenabgaben)
- verantworten gemeinsam das Ergebnis ihrer Gruppenarbeit, deren Bewertung für beide Gruppenpartner gleichermaßen gilt

### **Literatur:**

Lehrbuch: Saake, Sattler: „Algorithmen und Datenstrukturen - Eine Einführung mit JAVA“



---

## Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

### [1] Informatik (Bachelor of Science): 1. Semester

(Po-Vers. 2009s | Grundlagen- und Orientierungsprüfung)

Dieses Modul ist daneben auch in den Studienfächern "Berufspädagogik Technik (Bachelor of Science)", "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Bachelor of Science)", "Informatik (1. Staatsprüfung für das Lehramt an Gymnasien)", "Informatik (1. Staatsprüfung für das Lehramt an Hauptschulen)", "Informatik (1. Staatsprüfung für das Lehramt an Mittelschulen)", "Informatik (1. Staatsprüfung für das Lehramt an Realschulen)", "Informatik (Bachelor of Arts (2 Fächer))", "Informations- und Kommunikationstechnik (Bachelor of Science)", "Mathematik (Bachelor of Science)", "Physische Geographie (Bachelor of Science)", "Technomathematik (Bachelor of Science)", "Wirtschaftsinformatik (Bachelor of Science)" verwendbar.

---

## Studien-/Prüfungsleistungen:

Algorithmen und Datenstrukturen (Prüfungsnummer: 30501)

Prüfungsleistung, Klausur, Dauer (in Minuten): 120

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Algorithmen und Datenstrukturen

weitere Erläuterungen:

- Zur Klausur sind KEINE Hilfsmittel zugelassen - insbesondere KEINE elektronischen Geräte mit eigenem Betriebssystem (z.B. Handy, SmartWatch o.ä.).
- Bei den schriftlichen Prüfungen kann ein zweisprachiges Wörterbuch verwendet werden. Es darf sich dabei auch um ein Fachwörterbuch handeln. Ergänzungen oder Anmerkungen sind nicht erlaubt. Die Kandidaten werden gebeten, ihre Wörterbücher an den jeweiligen Prüfungstagen bei den Aufsichtern zur Kontrolle vorzulegen. Elektronische Wörterbücher sind ausdrücklich verboten.
- Die Klausur muss mit einem dokumentenechten Stift (Kugelschreiber, Füller) ausgefüllt werden. Bleistifte, Buntstifte o.ä. sind NICHT zugelassen.

Prüfungssprache: Deutsch

Erstablingung: WS 2019/2020, 1. Wdh.: SS 2020

1. Prüfer: Philipp/Oster/Riehle/Stammin/Brinda (ps0566)

Übungen zu Algorithmen und Datenstrukturen (Prüfungsnummer: 30502)

Studienleistung, Übungsleistung

Zugeordnete Lehrveranstaltungen:

- Tafelübungen zu Algorithmen und Datenstrukturen
- Rechnerübungen zu Algorithmen und Datenstrukturen

weitere Erläuterungen:

Bearbeitung wöchentlicher Übungsblätter, die je zur Hälfte aus Einzel- bzw. Gruppenaufgaben bestehen. Für den unbenoteten Übungsschein sind sowohl 60% der möglichen Einzelpunkte als auch 60% der Gruppenpunkte erforderlich.

Erstablingung: WS 2019/2020, 1. Wdh.: SS 2020

1. Prüfer: Philipp/Oster/Riehle/Stammin/Brinda (ps0566)

---

**Modulbezeichnung:** **Konzeptionelle Modellierung (KonzMod)** **5 ECTS**  
(Conceptual Modelling)

Modulverantwortliche/r: Richard Lenz  
Lehrende: Richard Lenz

---

Startsemester: WS 2019/2020	Dauer: 1 Semester	Turnus: jährlich (WS)
Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Deutsch

---

**Lehrveranstaltungen:**

Ausschlussbedingung: Wer dieses Modul ablegt, darf das Modul DBNF nicht mehr ablegen.

Konzeptionelle Modellierung (WS 2019/2020, Vorlesung, 2 SWS, Richard Lenz)

Übungen zu Konzeptionelle Modellierung (WS 2019/2020, Übung, 2 SWS, David Haller)

Intensivierungsübungen zu Konzeptionelle Modellierung (WS 2019/2020, Übung, 2 SWS, Anwesenheitspflicht, David Haller)

---

**Empfohlene Voraussetzungen:**

Gewünscht "Algorithmen und Datenstrukturen" und "Grundlagen der Logik und Logikprogrammierung"

---

**Inhalt:**

- Grundlagen der Modellierung
- Datenmodellierung am Beispiel Entity-Relationship-Modell
- Modellierung objektorientierter Systeme am Beispiel UML
- Relationale Datenmodellierung und Anfragemöglichkeiten
- Grundlagen der Metamodellierung
- XML
- Multidimensionale Datenmodellierung
- Domänenmodellierung und Ontologien

**Lernziele und Kompetenzen:**

Die Studierenden:

- definieren grundlegende Begriffe aus der Datenbankfachliteratur
- erklären die Vorteile von Datenbanksystemen
- erklären die verschiedenen Phasen des Datenbankentwurfs
- benutzen das Entity-Relationship Modell und das erweiterte Entity-Relationship Modell zur semantischen Datenmodellierung
- unterscheiden verschiedene Notationen für ER-Diagramme
- erläutern die grundlegenden Konzepte des relationalen Datenmodells
- bilden ein gegebenes EER-Diagramm auf ein relationales Datenbankschema ab
- erklären die Normalformen 1NF, 2NF, 3NF, BCNF und 4NF
- definieren die Operationen der Relationenalgebra
- erstellen Datenbanktabellen mit Hilfe von SQL
- lösen Aufgaben zur Datenselektion und Datenmanipulation mit Hilfe von SQL
- erklären die grundlegenden Konzepte der XML
- erstellen DTDs für XML-Dokumente
- benutzen XPATH zur Formulierung von Anfragen an XML-Dokumente
- definieren die grundlegenden Strukturelemente und Operatoren des multidimensionalen Datenmodells
- erklären Star- und Snowflake-Schema
- benutzen einfache UML Use-Case Diagramme
- benutzen einfache UML-Aktivitätsdiagramme
- erstellen UML-Sequenzdiagramme
- erstellen einfache UML-Klassendiagramme
- erklären den Begriff Meta-Modellierung
- definieren den Begriff der Ontologie in der Informatik

- definieren die Begriffe RDF und OWL

**Literatur:**

- Alfons Kemper, Andre Eickler: Datenbanksysteme : Eine Einführung. 6., aktualis. u. erw. Aufl. Oldenbourg, März 2006. - ISBN-10: 3486576909
- Bernd Oestereich: Analyse und Design mit UML 2.1. 8. Aufl. Oldenbourg, Januar 2006. - ISBN-10: 3486579266
- Ian Sommerville: Software Engineering. 8., aktualis. Aufl. Pearson Studium, Mai 2007. - ISBN-10: 3827372577
- Horst A. Neumann: Objektorientierte Softwareentwicklung mit der Unified Modeling Language. (UML). Hanser Fachbuch, März 2002. - ISBN-10: 3446188797
- Rainer Eckstein, Silke Eckstein: XML und Datenmodellierung. Dpunkt Verlag, November 2003. - ISBN-10: 3898642224

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

**[1] Informatik (Bachelor of Science): 1. Semester**

(Po-Vers. 2009s | Grundlagen- und Orientierungsprüfung)

Dieses Modul ist daneben auch in den Studienfächern "Berufspädagogik Technik (Bachelor of Science)", "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Master of Science)", "Digitale Geistes- und Sozialwissenschaften (Bachelor of Arts (2 Fächer))", "Informatik (1. Staatsprüfung für das Lehramt an Gymnasien)", "Informatik (1. Staatsprüfung für das Lehramt an Hauptschulen)", "Informatik (1. Staatsprüfung für das Lehramt an Mittelschulen)", "Informatik (1. Staatsprüfung für das Lehramt an Realschulen)", "Informatik (Bachelor of Arts (2 Fächer))", "Information and Communication Technology (Master of Science)", "Informations- und Kommunikationstechnik (Master of Science)", "International Information Systems (IIS) (Master of Science)", "Maschinenbau (Master of Science)", "Mathematik (Bachelor of Science)", "Medizintechnik (Master of Science)", "Physik (Master of Science)", "Physische Geographie (Bachelor of Science)", "Technomathematik (Bachelor of Science)", "Wirtschaftsinformatik (Bachelor of Science)", "Wirtschaftsingenieurwesen (Master of Science)" verwendbar.

---

**Studien-/Prüfungsleistungen:**

Konzeptionelle Modellierung (Klausur) (Prüfungsnummer: 31301)

(englische Bezeichnung: Written examination in conceptual modelling)

Prüfungsleistung, Klausur mit MultipleChoice, Dauer (in Minuten): 90

Anteil an der Berechnung der Modulnote: 100%

Erstablegung: WS 2019/2020, 1. Wdh.: SS 2020

1. Prüfer: Richard Lenz (100185)

---

---

**Modulbezeichnung:** Grundlagen der Technischen Informatik (GTI) 7.5 ECTS  
 (Fundamentals of Computer Engineering)

Modulverantwortliche/r: Jürgen Teich  
 Lehrende: Jürgen Teich

---

Startsemester: WS 2019/2020	Dauer: 1 Semester	Turnus: jährlich (WS)
Präsenzzeit: 90 Std.	Eigenstudium: 135 Std.	Sprache: Deutsch

---

**Lehrveranstaltungen:**

Grundlagen der Technischen Informatik (WS 2019/2020, Vorlesung, 4 SWS, Jürgen Teich et al.)  
 Übung zu Grundlagen der Technischen Informatik (WS 2019/2020, Übung, 2 SWS, Michael Witterauf et al.)  
 Praktische Übungen zu Grundlagen der Technischen Informatik (WS 2019/2020, Praktikum, Marcel Brand et al.)  
 Übungen zu Grundlagen der Technischen Informatik (SS 2020, Übung, 2 SWS, Tobias Schwarzer et al.)  
 Praktikum zu Grundlagen der Technischen Informatik (SS 2020, Praktikum, Marcel Brand et al.)

---

**Inhalt:**

Aufbau und Prinzip von Rechnern, Daten und ihre Codierung, Boolesche Algebra und Schaltalgebra, Schaltnetze (Symbole, Darstellung), Optimierung von Schaltnetzen (Minimierung Boolescher Funktionen), Realisierungsformen von Schaltnetzen (ROM, PLA, FPGA), Automaten und Schaltwerke (Moore/Mealy, Zustandskodierung und -minimierung), Flipflops, Register, Zähler, Speicher (RAM, ROM), Taktung und Synchronisation, Realisierungsformen von Schaltwerken, Realisierung der Grundrechenarten Addition/Subtraktion, Multiplikation und Division, Gleitkommazahlen (Darstellung, Fehler, Rundung, Standards, Einheiten), Steuerwerksentwurf, Spezialeinheiten und Co-Prozessoren, Mikrocontroller; vorlesungsbegleitende Einführung und Beschreibung der Schaltungen mit VHDL.

**Lernziele und Kompetenzen:**

*Fachkompetenz*

*Wissen*

- Die Studierenden veranschaulichen fundierte theoretische und praxisorientierte Grundlagen der Informationstheorie, Rechnerarithmetik, Digitaltechnik und des Schaltungsentwurfs.
- Die Studierenden führen den Entwurf, die Synthese und das Testen von digitalen Schaltungen auf programmierbarer Hardware (FPGAs) durch.

*Verstehen*

- Die Studierenden verstehen, dass Hardware heutzutage mit Software am Rechner entwickelt und simuliert wird.
- Die Studierenden verstehen den Schaltungsentwurf mittels einer Beschreibungssprache (VHDL).

*Anwenden*

- Die Studierenden erarbeiten und diskutieren verschiedene Lösungswege für die Datencodierung sowie den Entwurf und die Optimierung von digitalen Hardwareschaltungen.

*Selbstkompetenz*

- Die Studierenden erlernen die Fähigkeit, digitale Schaltungen und Systeme eigenständig zu konzipieren und zu implementieren.

**Literatur:**

siehe Webseite: <http://www12.informatik.uni-erlangen.de/edu/gti>

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

[1] **Informatik (Bachelor of Science): 2. Semester**

(Po-Vers. 2009s | Grundlagen- und Orientierungsprüfung)

Dieses Modul ist daneben auch in den Studienfächern "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Bachelor of Science)", "Informatik (1. Staatsprüfung für das Lehramt an Gymnasien)", "Informatik (Bachelor of Arts (2 Fächer))", "Medizintechnik (Bachelor of Science)" verwendbar.

---

### **Studien-/Prüfungsleistungen:**

Klausur zu Grundlagen der Technischen Informatik (Prüfungsnummer: 31101)

(englische Bezeichnung: Examination (Klausur) on Foundations of Computer Engineering)

Prüfungsleistung, Klausur, Dauer (in Minuten): 120

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Grundlagen der Technischen Informatik
- Übung zu Grundlagen der Technischen Informatik

weitere Erläuterungen:

Auf Basis der Bewertung zweier während des Semesters angebotener Miniklausuren können Bonuspunkte erworben werden, die zu dem Ergebnis einer bestandenen Klausur hinzugerechnet werden.

Erstablingung: WS 2019/2020, 1. Wdh.: SS 2020

1. Prüfer: Jürgen Teich (100115)

Kurztest, praktische Übungen zu Grundlagen der Technischen Informatik (Übungsleistung) (Prüfungsnummer: 31102)

(englische Bezeichnung: Tutorial Credit: Foundations of Computer Engineering)

Studienleistung, Übungsleistung

Zugeordnete Lehrveranstaltungen:

- Praktische Übungen zu Grundlagen der Technischen Informatik

weitere Erläuterungen:

Der Schein wird vergeben auf erfolgreiches Absolvieren eines Kurztests und von praktischen Übungen.

Erstablingung: WS 2019/2020, 1. Wdh.: SS 2020

1. Prüfer: Jürgen Teich (100115)

---

---

**Modulbezeichnung:** **Parallele und Funktionale Programmierung (PFP)** **5 ECTS**  
 (Parallel and Functional Programming)

Modulverantwortliche/r: Michael Philippsen  
 Lehrende: Michael Philippsen, Norbert Oster

---

Startsemester: SS 2019	Dauer: 1 Semester	Turnus: jährlich (SS)
Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Deutsch

---

**Lehrveranstaltungen:**

Parallele und Funktionale Programmierung (SS 2019, Vorlesung, 2 SWS, Michael Philippsen et al.)  
 Übungen zu Parallele und funktionale Programmierung (SS 2019, Übung, 2 SWS, Marius Kamp)  
**Wiederholungsübungen im Sommersemester**

---

**Inhalt:**

- Grundlagen der funktionale Programmierung
- Grundlagen der parallelen Programmierung
- Datenstrukturen
- Objektorientierung
- Scala-Kenntnisse
- Erweiterte JAVA-Kenntnisse
- Aufwandsabschätzungen
- Grundlegende Algorithmen

**Lernziele und Kompetenzen:**

Die Studierenden

- erlernen die Grundlagen der funktionalen Programmierung anhand der Programmiersprache Scala
- verstehen paralleles Programmieren mit Java
- kennen fundamentale Datenstrukturen und Algorithmen
- können funktionale und parallele Algorithmen entwickeln und analysieren

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

**[1] Informatik (Bachelor of Science): 2. Semester**

(Po-Vers. 2009s | Grundlagen- und Orientierungsprüfung)

Dieses Modul ist daneben auch in den Studienfächern "Berufspädagogik Technik (Bachelor of Science)", "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Bachelor of Science)", "Informatik (1. Staatsprüfung für das Lehramt an Gymnasien)", "Informatik (1. Staatsprüfung für das Lehramt an Hauptschulen)", "Informatik (1. Staatsprüfung für das Lehramt an Mittelschulen)", "Informatik (1. Staatsprüfung für das Lehramt an Realschulen)", "Informatik (Bachelor of Arts (2 Fächer))", "Mathematik (Bachelor of Science)", "Technomathematik (Bachelor of Science)", "Wirtschaftsinformatik (Bachelor of Science)" verwendbar.

---

**Studien-/Prüfungsleistungen:**

Parallele und Funktionale Programmierung (Prüfungsnummer: 30401)

(englische Bezeichnung: Parallel and Functional Programming)

Prüfungsleistung, Klausur, Dauer (in Minuten): 60

Anteil an der Berechnung der Modulnote: 100%

weitere Erläuterungen:

- Zur Klausur sind KEINE Hilfsmittel zugelassen - insbesondere KEINE elektronischen Geräte mit eigenem Betriebssystem (z.B. Handy, SmartWatch o.ä.).
- Bei den schriftlichen Prüfungen kann ein zweisprachiges Wörterbuch verwendet werden. Es darf sich dabei auch um ein Fachwörterbuch handeln. Ergänzungen oder Anmerkungen sind nicht erlaubt. Die Kandidaten werden gebeten, ihre Wörterbücher an den jeweiligen Prüfungstagen bei den Aufsichten zur Kontrolle vorzulegen. Elektronische Wörterbücher sind ausdrücklich verboten.

- Die Klausur muss mit einem dokumentenechten Stift (Kugelschreiber, Füller) ausgefüllt werden.  
Bleistifte, Buntstifte o.ä. sind NICHT zugelassen.  
Prüfungssprache: Deutsch

Erstablingung: SS 2019, 1. Wdh.: WS 2019/2020  
1. Prüfer: Philippsen/Oster (ps567)

---

---

**Modulbezeichnung: Grundlagen der Schaltungstechnik (GdS)** **5 ECTS**  
(Basics of Electrical Circuit Technology)

Modulverantwortliche/r: Albert Heuberger

Lehrende: Albert Heuberger, Florian Rittner

Startsemester: SS 2019

Dauer: 1 Semester

Turnus: jährlich (SS)

Präsenzzeit: 60 Std.

Eigenstudium: 90 Std.

Sprache: Deutsch

**Lehrveranstaltungen:**

Grundlagen der Schaltungstechnik (SS 2019, Vorlesung, 2 SWS, Albert Heuberger et al.)

Übung Grundlagen der Schaltungstechnik (SS 2019, Übung, 2 SWS, Florian Rittner)

**Inhalt:**

In der Lehrveranstaltung werden grundlegende Zusammenhänge elektrotechnischer Systeme, welche die Grundlage fast aller digitalen Datenverarbeitungssystemen bilden, behandelt. Zu Beginn werden elektrotechnischen Grundbegriffe und mathematische Grundlagen vermittelt. Daraufhin werden die Techniken zur Analyse von Gleich- und Wechselstromnetzwerken erläutert. Die grundlegenden elektrotechnischen Bauelemente Widerstand, Kondensator und Spule werden eingeführt und ihre Eigenschaften untersucht. Nachfolgend werden nichtlineare Bauelemente, wie Diode, Transistor und Operationsverstärker betrachtet und analysiert. Die Netzwerkanalyse wird anschließend auf Schaltvorgänge ausgeweitet. Außerdem wird das Funktionsprinzip von CMOS-Schaltungen erläutert und einfache digitale logische Grundsaltungen behandelt. Abschließend wird ein Überblick über Prinzipien der Datenspeicherung auf mikroelektronischer Basis gegeben, sowie Schaltungen zu Digital-Analog- und Analog-Digital-Wandlung vorgestellt und diskutiert.

Elektrotechnische Grundlagen

- Mathematische Grundlagen
- Netzwerkanalyse Gleichstromfall
- Netzwerkanalyse Wechselstromfall

Elektronische Bauelemente

- Widerstand, Kondensator, Spule
- Diode, Transistor, Operationsverstärker

Einfache dynamische Vorgänge in Schaltungen

- Schaltvorgänge

Digitale Schaltungstechnik

- Funktionsprinzip von CMOS Schaltungen
- Einführung logischer Grundsaltungen in CMOS
- Prinzipien mikroelektronischer Datenspeicher

Schaltungen zur Realisierung der Schnittstelle zwischen Analog- und Digitaltechnik

- Digital-Analog-Wandler
- Analog-Digital-Wandler

**Lernziele und Kompetenzen:**

Die Studierenden:

- verstehen die Grundprinzipien elektrotechnischer Netzwerke
- kennen zugrundeliegende Annahmen und Voraussetzungen
- können einfache Schaltungen im Gleichstrom-, Wechselstromfall und bei Schaltvorgängen berechnen
- können die Funktionsweise einfacher digitaler Logikschaltungen erklären

**Literatur:**

Tietze, U. ; Schenk, Ch. : Halbleiter-Schaltungstechnik. Berlin: Springer.

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

[1] **Informatik (Bachelor of Science): 1. Semester**

(Po-Vers. 2009s | Grundlagen- und Orientierungsprüfung)



Dieses Modul ist daneben auch in den Studienfächern "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Bachelor of Science)", "Medizintechnik (Master of Science)" verwendbar.

---

### **Studien-/Prüfungsleistungen:**

Grundlagen der Schaltungstechnik (Prüfungsnummer: 30901)

Prüfungsleistung, Klausur, Dauer (in Minuten): 90

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Grundlagen der Schaltungstechnik
- Übung Grundlagen der Schaltungstechnik

Erstablingung: SS 2019, 1. Wdh.: WS 2019/2020

1. Prüfer: Albert Heuberger (100228)

Grundlagen der Schaltungstechnik (Übungen) (Prüfungsnummer: 30902)

Studienleistung, Übungsleistung

Zugeordnete Lehrveranstaltungen:

- Übung Grundlagen der Schaltungstechnik

weitere Erläuterungen:

unbenotete Studienleistung, zu erwerben durch Bestehen einer von zwei 45-minütigen Zwischenklausuren

Erstablingung: SS 2019, 1. Wdh.: keine Angabe

1. Prüfer: Albert Heuberger (100228)

---

---

**Modulbezeichnung:** Rechnerkommunikation (RK) 5 ECTS  
(Computer Communication)

Modulverantwortliche/r: Reinhard German

Lehrende: Anatoli Djanatliev

---

Startsemester: SS 2019	Dauer: 1 Semester	Turnus: jährlich (SS)
Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Deutsch

---

**Lehrveranstaltungen:**

Rechnerkommunikation (SS 2019, Vorlesung, 2 SWS, Anatoli Djanatliev)

Rechnerkommunikation Übungen (SS 2019, Übung, 2 SWS, Lorenz Ammon et al.)

---

**Inhalt:**

Die Vorlesung vermittelt die Grundlagen der Rechnerkommunikation und durchläuft von oben nach unten die Schichten des Internets:

- Anwendungsschicht
- Transportschicht
- Netzwerkschicht
- Sicherungsschicht
- Physikalische Schicht

Anschließend wird Sicherheit als übergreifender Aspekt behandelt. An verschiedenen Stellen werden analytische Modelle eingesetzt, um Wege für eine quantitative Auslegung von Kommunikationsnetzen aufzuzeigen. Die Übung beinhaltet praktische und theoretische Aufgaben zum Verständnis der einzelnen Schichten.

**Lernziele und Kompetenzen:**

Die Studierenden erwerben

- Kenntnisse über zentrale Mechanismen, Protokolle und Architekturen der Rechnerkommunikation (Topologie, Schicht, Adressierung, Wegsuche, Weiterleitung, Flusskontrolle, Überlastkontrolle, Fehlersicherung, Medienzugriff, Bitübertragung) am Beispiel des Internets und mit Ausblicken auf andere Netztechnologien
- Kenntnisse über Sicherheit, Leistung und Zuverlässigkeit bei der Rechnerkommunikation
- praktische Erfahrung in der Benutzung und Programmierung von Rechnernetzen

**Literatur:**

Lehrbuch: Kurose, Ross. Computer Networking: A Top-Down Approach Featuring the Internet. 6th Ed., Addison Wesley, 2013

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

[1] **Informatik (Bachelor of Science): 2. Semester**

(Po-Vers. 2009s | Grundlagen- und Orientierungsprüfung)

Dieses Modul ist daneben auch in den Studienfächern "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Bachelor of Science)", "Informatik (1. Staatsprüfung für das Lehramt an Gymnasien)", "Informatik (Bachelor of Arts (2 Fächer))", "Informations- und Kommunikationstechnik (Bachelor of Science)", "Mathematik (Bachelor of Science)", "Medizintechnik (Bachelor of Science)", "Wirtschaftsinformatik (Bachelor of Science)" verwendbar.

---

**Studien-/Prüfungsleistungen:**

Rechnerkommunikation (Klausur) (Prüfungsnummer: 31501)

(englische Bezeichnung: Examination (Klausur) on Computer communications)

Prüfungsleistung, Klausur, Dauer (in Minuten): 90

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Rechnerkommunikation
- Rechnerkommunikation Übungen

Erstablingung: SS 2019, 1. Wdh.: WS 2019/2020

1. Prüfer: Anatoli Djanatliev (100361)

Hausaufgaben zu Rechnerkommunikation (Übungsleistung) (Prüfungsnummer: 31502)

(englische Bezeichnung: Tutorial credit: Computer communications)

Studienleistung, Übungsleistung

Zugeordnete Lehrveranstaltungen:

- Rechnerkommunikation Übungen

weitere Erläuterungen:

Die Modulprüfung besteht aus:

- unbenotete Studienleistung, zu erwerben durch erfolgreiche Teilnahme an den Übungen

Erstablingung: SS 2019, 1. Wdh.: WS 2019/2020

1. Prüfer: Anatoli Djanatliev (100361)

---

<b>Modulbezeichnung: Systemprogrammierung (SP)</b>		<b>10 ECTS</b>
(Systems Programming)		
Modulverantwortliche/r:	Wolfgang Schröder-Preikschat	
Lehrende:	Wolfgang Schröder-Preikschat, Jürgen Kleinöder	
Startsemester: SS 2019	Dauer: 2 Semester	Turnus: jährlich (SS)
Präsenzzeit: 120 Std.	Eigenstudium: 180 Std.	Sprache: Deutsch

#### Lehrveranstaltungen:

- Systemprogrammierung 1 (SS 2019, Vorlesung, 2 SWS, Wolfgang Schröder-Preikschat et al.)
- Übungen zu Systemprogrammierung 1 (SS 2019, Übung, 2 SWS, Simon Ruderich et al.)
- Rechnerübungen zu Systemprogrammierung 1 und 2 (SS 2019, Übung, 2 SWS, Simon Ruderich et al.)
- Systemprogrammierung 2 (WS 2019/2020, Vorlesung, 2 SWS, Jürgen Kleinöder et al.)
- Übungen zu Systemprogrammierung 2 (WS 2019/2020, Übung, 2 SWS, Simon Ruderich et al.)
- Rechnerübungen zu Systemprogrammierung 1 und 2 (WS 2019/2020, Übung, 2 SWS, Simon Ruderich et al.)
- Übungen zu Systemprogrammierung 1 (für Wiederholer) (WS 2019/2020, Übung, 2 SWS, Simon Ruderich et al.)

#### Inhalt:

- Grundlagen von Betriebssystemen (Adressräume, Speicher, Dateien, Prozesse, Koordinationsmittel; Betriebsarten, Einplanung, Einlastung, Virtualisierung, Nebenläufigkeit, Koordination/Synchronisation)
- Abstraktionen/Funktionen UNIX-ähnlicher Betriebssysteme
- Programmierung von Systemsoftware
- C, Make, UNIX-Shell (Solaris, Linux, MacOS X)

#### Lernziele und Kompetenzen:

Die Studierenden

- erwerben fundierte Kenntnisse über Grundlagen von Betriebssystemen
- verstehen Zusammenhänge, die die Ausführungen von Programmen in vielschichtig organisierten Rechensystemen ermöglichen
- erkennen Gemeinsamkeiten und Unterschiede zwischen realen und abstrakten (virtuellen) Maschinen
- erlernen die Programmiersprache C
- entwickeln Systemprogramme auf Basis der Systemaufrufchnittstelle UNIX-ähnlicher Betriebssysteme

#### Literatur:

- Lehrbuch: Betriebssysteme - Grundlagen, Entwurf, Implementierung, Wolfgang Schröder-Preikschat, 2008

#### Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

##### [1] Informatik (Bachelor of Science): 2-3. Semester

(Po-Vers. 2009s | Grundlagen- und Orientierungsprüfung)

Dieses Modul ist daneben auch in den Studienfächern "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Bachelor of Science)", "Informatik (1. Staatsprüfung für das Lehramt an Gymnasien)", "Informatik (Bachelor of Arts (2 Fächer))", "Informations- und Kommunikationstechnik (Bachelor of Science)", "Medizintechnik (Bachelor of Science)", "Medizintechnik (Master of Science)", "Wirtschaftsinformatik (Bachelor of Science)" verwendbar.

#### Studien-/Prüfungsleistungen:

Systemprogrammierung (Klausur) (Prüfungsnummer: 31801)

(englische Bezeichnung: Examination (Klausur) on System programming)

Prüfungsleistung, Klausur, Dauer (in Minuten): 120

Anteil an der Berechnung der Modulnote: 100%

weitere Erläuterungen:

Die Klausur enthält einen kleinen Multiple-Choice-Anteil.

Die Rahmen der Übungen gestellten Übungsaufgaben können abgegeben werden und werden in diesem Fall bewertet. Auf Basis des Ergebnisses dieser Bewertungen können bis zu 10 % Bonuspunkte erworben werden, die zu dem Ergebnis einer bestandenen Klausur hinzugerechnet werden.

Zu Beginn des Teils Systemprogrammierung 2 wird eine Miniklausur angeboten, deren Ergebnis wie eine Übungsaufgabe behandelt wird.

Erstablingung: WS 2019/2020, 1. Wdh.: SS 2020, 2. Wdh.: WS 2020/2021

1. Prüfer: Jürgen Kleinöder (100163)

---

---

**Modulbezeichnung: Mathematik C1 (IngMathC1)** **7.5 ECTS**  
 (Mathematics C1)

Modulverantwortliche/r: Martin Gugat

Lehrende: Serge Kräutle, u. a. Hochschullehrer

---

Startsemester: WS 2019/2020

Dauer: 1 Semester

Turnus: jährlich (WS)

Präsenzzeit: 90 Std.

Eigenstudium: 135 Std.

Sprache: Deutsch

---

**Lehrveranstaltungen:**

Mathematik für Ingenieure C1: INF, IP, ILS, PhM (WS 2019/2020, Vorlesung, 4 SWS, Serge Kräutle)  
 Übung zur Mathematik für Ingenieure C1 (WS 2019/2020, Übung, 2 SWS, Serge Kräutle)

---

**Inhalt:**

**Grundlagen**

Aussagenlogik, Mengen, Relationen, Abbildungen

**Zahlensysteme**

natürliche, ganze, rationale und reelle Zahlen, komplexe Zahlen

**Vektorräume**

Grundlagen, Lineare Abhängigkeit, Spann, Basis, Dimension, euklidische Vektor- und Untervektorräume, affine Räume

**Matrizen, Lineare Abbildungen, Lineare Gleichungssysteme**

Matrixalgebra, Lösungsstruktur linearer Gleichungssysteme, Gauß-Algorithmus, inverse Matrizen, Matrixtypen, lineare Abbildungen, Determinanten, Kern und Bild, Eigenwerte und Eigenvektoren, Basis, Ausgleichsrechnung

**Grundlagen Analysis einer Veränderlichen**

Grenzwert, Stetigkeit, elementare Funktionen, Umkehrfunktionen

**Lernziele und Kompetenzen:**

Die Studierenden

- erklären grundlegende Begriffe und Strukturen der Mathematik
- erklären den Aufbau von Zahlensystemen im Allgemeinen und der Obengenannten im Speziellen
- rechnen mit komplexen Zahlen in Normal- und Polardarstellung und Wechseln zwischen diesen Darstellungen
- berechnen lineare Abhängigkeiten, Unterräume, Basen, Skalarprodukte, Determinanten
- vergleichen Lösungsmethoden zu linearen Gleichungssystemen
- bestimmen Lösungen zu Eigenwertproblemen
- überprüfen Eigenschaften linearer Abbildungen und Matrizen
- überprüfen die Konvergenz von Zahlenfolgen
- ermitteln Grenzwerte und überprüfen Stetigkeit
- entwickeln Beweise anhand grundlegender Beweismethoden aus den genannten Themenbereichen
- kennen eine regelmäßige selbstständige Nachbereitung und Anwendung des Vorlesungsstoffes

**Literatur:**

Skripte des Dozenten

W. Merz, P. Knabner, Mathematik für Ingenieure und Naturwissenschaftler, Springer, 2013

Fried, Mathematik für Ingenieure I für Dummies I, Wiley A. Hoffmann, B. Marx, W. Vogt, Mathematik für Ingenieure 1, Pearson

v. Finckenstein et.al: Arbeitsbuch Mathematik fuer Ingenieure: Band I Analysis und Lineare Algebra. Teubner-Verlag 2006, ISBN 9783835100343

Meyberg, K., Vachenaer, P.: Höhere Mathematik 1. 6. Auflage, Sprinbger-Verlag, Berlin, 2001

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

[1] **Informatik (Bachelor of Science): 1. Semester**

(Po-Vers. 2009s | Grundlagen- und Orientierungsprüfung)

Dieses Modul ist daneben auch in den Studienfächern "International Production Engineering and Management (Bachelor of Science)", "Materialphysik (Bachelor of Science)" verwendbar.

---

**Studien-/Prüfungsleistungen:**

Mathematik C1 (Prüfungsnummer: 45801)

Prüfungsleistung, Klausur, Dauer (in Minuten): 90

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Mathematik für Ingenieure C1: INF, IP, ILS, PhM

Erstablingung: WS 2019/2020, 1. Wdh.: SS 2020

1. Prüfer: Serge Kräutle (060254)

Mathematik C1 Übungen (Prüfungsnummer: 45802)

Studienleistung, Übungsleistung

Zugeordnete Lehrveranstaltungen:

- Übung zur Mathematik für Ingenieure C1

weitere Erläuterungen:

Erwerb der Übungsleistung durch Lösung der wöchentlichen Hausaufgaben. Die Lösungen sind in handschriftlicher Form abzugeben.

Erstablingung: WS 2019/2020, 1. Wdh.: keine Angabe

1. Prüfer: Serge Kräutle (060254)

---

---

**Modulbezeichnung: Mathematik C2 (IngMathC2)** **7.5 ECTS**  
 (Mathematics C2)

Modulverantwortliche/r: Martin Gugat  
 Lehrende: Serge Kräutle

---

Startsemester: SS 2019	Dauer: 1 Semester	Turnus: jährlich (SS)
Präsenzzeit: 84 Std.	Eigenstudium: 141 Std.	Sprache: Deutsch

---

**Lehrveranstaltungen:**

Mathematik für Ingenieure C2: INF, IP, ILS, PhM (SS 2019, Vorlesung, 4 SWS, Serge Kräutle)  
 Übung zur Mathematik für Ingenieure C2: INF, IP, ILS, PhM (SS 2019, Übung, 2 SWS, Serge Kräutle et al.)

---

**Inhalt:**

**Differentialrechnung einer Veränderlichen:**

Ableitung mit Rechenregeln, Mittelwertsätze, L'Hospital, Taylor-Formel, Kurvendiskussion

**Integralrechnung einer Veränderlichen:**

Riemann-Integral, Hauptsatz der Infinitesimalrechnung, Mittelwertsätze, Partialbruchzerlegung, uneigentliche Integration

**Folgen und Reihen:**

reelle und komplexe Zahlenfolgen, Konvergenzbegriff und -sätze, Folgen und Reihen von Funktionen, gleichmäßige Konvergenz, Potenzreihen, iterative Lösung nichtlinearer Gleichungen

**Grundlagen Analysis mehrerer Veränderlicher:**

Grenzwert, Stetigkeit, Differentiation, partielle Ableitungen, totale Ableitung, allgemeine Taylor-Formel

**Lernziele und Kompetenzen:**

Die Studierenden

- analysieren Funktionen einer reellen Veränderlichen mit Hilfe der Differentialrechnung
- berechnen Integrale von Funktionen mit einer reellen Veränderlichen
- stellen technisch-naturwissenschaftliche Problemstellungen mit mathematischen Modellen dar und lösen diese
- erklären den Konvergenzbegriff bei Folgen und Reihen
- berechnen Grenzwerte und rechnen mit diesen
- analysieren und klassifizieren Funktionen mehrerer reeller Veränderlicher an Hand grundlegender Eigenschaften
- wenden grundlegende Beweistechniken in o.g. Bereichen an
- erkennen die Vorzüge einer regelmäßigen Nachbereitung und Vertiefung des Vorlesungsstoffes

**Literatur:**

Skripte des Dozenten

G. Baron und P. Kirschenhofer: Einführung in die Mathematik für Informatiker 1 und 2, Springer, 1989 und 1990

K.-H. Kiyek und F. Schwarz: Mathematik für Informatiker 1 und 2, Teubner, 1989 und 1990

K. Finck von Finckenstein, J. Lehn et. al., Arbeitsbuch für Ingenieure, Band I, Teubner

M. Fried: Mathematik für Ingenieure I für Dummies. Wiley

M. Fried: Mathematik für Ingenieure II für Dummies. Wiley

W. Merz, P. Knabner: Mathematik für Ingenieure und Naturwissenschaftler, Springer, 2013

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

[1] **Informatik (Bachelor of Science): 2. Semester**

(Po-Vers. 2009s | Grundlagen- und Orientierungsprüfung)

Dieses Modul ist daneben auch in den Studienfächern "International Production Engineering and Management (Bachelor of Science)", "Materialphysik (Bachelor of Science)", "Mathematik Export" verwendbar.

---

**Studien-/Prüfungsleistungen:**



Klausur Mathematik C 2 (Prüfungsnummer: 45901)

(englische Bezeichnung: Examination (Klausur) on Mathematics C 2)

Prüfungsleistung, Klausur, Dauer (in Minuten): 90

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Mathematik für Ingenieure C2: INF, IP, ILS, PhM

Erstablingung: SS 2019, 1. Wdh.: WS 2019/2020

1. Prüfer: Alexander Prechtel (060123)

Übung Mathematik C 2 (Prüfungsnummer: 45902)

(englische Bezeichnung: Tutorial mathematics C 2)

Studienleistung, Übungsleistung

Zugeordnete Lehrveranstaltungen:

- Übung zur Mathematik für Ingenieure C2: INF, IP, ILS, PhM

weitere Erläuterungen:

Erwerb der Übungsleistung durch Lösung der wöchentlichen Hausaufgaben. Die Lösungen sind in handschriftlicher Form abzugeben.

Erstablingung: SS 2019, 1. Wdh.: keine Angabe

1. Prüfer: Alexander Prechtel (060123)

---

---

**Modulbezeichnung:** Grundlagen der Rechnerarchitektur und -organisation (GRa) 5 ECTS  
 (Foundations of Computer Architecture and Organization)

Modulverantwortliche/r: Dietmar Fey

Lehrende: Dietmar Fey, Marc Reichenbach

Startsemester: SS 2019

Dauer: 1 Semester

Turnus: jährlich (SS)

Präsenzzeit: 60 Std.

Eigenstudium: 90 Std.

Sprache: Deutsch

---

### Lehrveranstaltungen:

Grundlagen der Rechnerarchitektur und -organisation (SS 2019, Vorlesung, 2 SWS, Dietmar Fey)

Übungen zu Grundlagen der Rechnerarchitektur und -organisation (SS 2019, Übung, 2 SWS, Sebastian Rachuj)

---

### Inhalt:

Ziel der Vorlesung ist, die Grundlagen beim Aufbau eines Rechners zu vermitteln. Dies beinhaltet die Grundkomponenten, wie das Leitwerk, das Rechenwerk, das Speicherwerk und das Ein-/Ausgabewerk. Ausgehend vom klassischen von Neumann-Rechner wird der Bogen bis zu den Architekturen moderner Rechner und Prozessoren geschlagen.

Grundprinzipien der Ablaufsteuerung bei der Bearbeitung von Befehlen werden ebenso behandelt wie Aufbau und Funktionsweise eines Caches und die Architektur von Speichern im Allgemeinen. Das Konzept der Mikroprogrammierung wird erläutert. Ferner wird der Einstieg in die hardwarenahe Programmierung moderner CPUs mittels Assembler vorgestellt und erprobt. Aufbau und Funktionsweise peripherer Einheiten und Bussysteme werden ebenfalls behandelt.

Die Studierenden sollen am Ende der Vorlesung den Aufbau und die Funktionsweise der Architektur eines Rechners, z.B. eines PCs, und des darin enthaltenen Prozessors nicht nur kennen, sondern auch die Gründe für deren Zustandekommen verstanden haben.

### Lernziele und Kompetenzen:

Nach dem Besuch der Lehrveranstaltung kennen die Studierenden die Grundkomponenten eines Rechners, z. B. eines PCs, und können diese auch im Zusammenspiel als Gesamtsystem erklären, sowie die Eigenheiten verschiedener Architekturen diskutieren. Sie können die Funktionsweise von Grundkomponenten wie Leitwerk, Rechenwerk, Speicherwerk, Ein-/Ausgabewerk, Bussystemen, sowie peripherer Komponenten erläutern und in die Struktur eines Computersystems einordnen. Sie kennen den Aufbau von Caches, bzw. von Speichern im Allgemeinen und verstehen die Funktionsweise der Ablaufsteuerung, insbesondere in Bezug auf die Abarbeitung von Befehlen. Weiterhin können die Studierenden Konzepte der Mikroprogrammierung unterscheiden, sowie hardwarenahe Programme in Assembler verstehen, modifizieren und erstellen.

### Literatur:

Hennessy/Patterson: Computer Architecture - A quantitative approach, 4.Auflage, 2006, MorganKaufmann.

Patterson/Hennessy: Computer Organization & Design, 4.Auflage, 2008, MorganKaufmann.

Stallings, Computer Organization & Architecture, 8.Auflage, 2009, Prentice Hall.

Märtinger, Rechnerarchitekturen, 2001, Fachbuchverlag Leipzig.

---

### Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

[1] **Informatik (Bachelor of Science): 3. Semester**

(Po-Vers. 2009s | weitere Pflichtmodule)

Dieses Modul ist daneben auch in den Studienfächern "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Bachelor of Science)", "Informations- und Kommunikationstechnik (Bachelor of Science)" verwendbar.

---

### Studien-/Prüfungsleistungen:

Grundlagen der Rechnerarchitektur und -organisation (Vorlesung mit Übungen) (Prüfungsnummer: 30801)

Prüfungsleistung, Klausur, Dauer (in Minuten): 90

Anteil an der Berechnung der Modulnote: 100%

weitere Erläuterungen:

Bonuspunkte für die erfolgreiche Bearbeitung der Übungsaufgaben werden bei Bestehen auf die Klausur angerechnet

Erstablesung: SS 2019, 1. Wdh.: WS 2019/2020

1. Prüfer: Dietmar Fey (100101)

---

**Organisatorisches:**

Scheinerwerb durch Klausur

---

**Modulbezeichnung:** **Software-Entwicklung in Großprojekten (SoSy3)** **5 ECTS**  
 (Software Development in Large Projects)

Modulverantwortliche/r: Francesca Saglietti  
 Lehrende: Francesca Saglietti

---

Startsemester: WS 2019/2020	Dauer: 1 Semester	Turnus: jährlich (WS)
Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Deutsch

---

**Lehrveranstaltungen:**

Softwareentwicklung in Großprojekten (Softwaresysteme 3) (WS 2019/2020, Vorlesung, 2 SWS, Francesca Saglietti)  
 Übungen zu Softwareentwicklung in Großprojekten (Softwaresysteme 3) (WS 2019/2020, Übung, 2 SWS, Marc Spisländer)

---

**Inhalt:**

- Einführung in die einzelnen Phasen der Softwareentwicklung: Anforderungsanalyse, Spezifikation, Entwurf, Implementierung, Test, Wartung
- Beispielhafter Einsatz ausgewählter repräsentativer Verfahren zur Unterstützung dieser Entwicklungsphasen
- Ergonomische Prinzipien Benutzungsoberfläche
- Objektorientierte Analyse und Design mittels UML
- Entwurfsmuster als konstruktive, wiederverwendbare Lösungsansätze für ganze Problemklassen
- Automatisch unterstützte Implementierung aus UML-Diagrammen
- Teststrategien
- Refactoring zur Unterstützung der Wartungsphase

**Lernziele und Kompetenzen:**

Die Studierenden

- wenden auf Basis der bereits erworbenen Programmierkenntnisse systematische und strukturierte Vorgehensweisen (wie das Wasserfall- und V-Modell) zur Bewältigung der Komplexität im Zusammenhang mit dem „Programmieren-im-Großen“ an;
- benutzen ausgewählte Spezifikationssprachen (wie Endliche Automaten, Petri-Netze und OCL), um komplexe Problemstellungen eindeutig zu formulieren und durch ausgewählte Entwurfsverfahren umzusetzen;
- wenden UML-Diagramme (wie Use Case-, Klassen-, Sequenz- und Kommunikationsdiagramme) zum Zweck objektorientierter Analyse- und Design-Aktivitäten an;
- reproduzieren allgemeine Entwurfslösungen wiederkehrender Probleme des Software Engineering durch Verwendung von Entwurfsmustern;
- erfassen funktionale und strukturelle Testansätze;
- setzen Refactoring-Strategien zur gezielten Erhöhung der Software-Änderungsfreundlichkeit um.

**Literatur:**

Lehrbuch der Softwaretechnik (Band 1), Helmut Balzert, 2000

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

**[1] Informatik (Bachelor of Science): 4. Semester**

(Po-Vers. 2009s | weitere Pflichtmodule)

Dieses Modul ist daneben auch in den Studienfächern "Berufspädagogik Technik (Bachelor of Science)", "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Bachelor of Science)", "Informatik (1. Staatsprüfung für das Lehramt an Gymnasien)", "Informatik (1. Staatsprüfung für das Lehramt an Hauptschulen)", "Informatik (1. Staatsprüfung für das Lehramt an Mittelschulen)", "Informatik (1. Staatsprüfung für das Lehramt an Realschulen)", "International Information Systems (IIS) (Master of Science)", "Mathematik (Bachelor of Science)", "Medizintechnik (Bachelor of Science)", "Wirtschaftsinformatik (Bachelor of Science)" verwendbar.

**Studien-/Prüfungsleistungen:**

Software-Entwicklung in Großprojekten (Klausur) (Prüfungsnummer: 31601)

(englische Bezeichnung: Examination (Klausur) on Software Development in Large Projects)

Prüfungsleistung, Klausur, Dauer (in Minuten): 90

Anteil an der Berechnung der Modulnote: 100%

Erstablingung: WS 2019/2020, 1. Wdh.: SS 2020

1. Prüfer: Francesca Saglietti (100112)

---

**Modulbezeichnung:** Berechenbarkeit und Formale Sprachen (BFS) 7.5 ECTS  
(Theory of Computation and Formal Languages)

Modulverantwortliche/r: Rolf Wanka  
Lehrende: Rolf Wanka

Startsemester: WS 2019/2020      Dauer: 1 Semester      Turnus: jährlich (WS)  
Präsenzzeit: 90 Std.      Eigenstudium: 135 Std.      Sprache: Deutsch

**Lehrveranstaltungen:**

Berechenbarkeit und Formale Sprachen (WS 2019/2020, Vorlesung, 4 SWS, Rolf Wanka)  
Übungen zu Berechenbarkeit und Formale Sprachen (WS 2019/2020, Übung, Rolf Wanka et al.)

**Inhalt:**

- Registermaschinen und Turingmaschinen als Modelle des Berechenbaren, die Churchsche These und unentscheidbare Probleme
- NP-Vollständigkeit und das P-NP-Problem
- Endliche Automaten
- Grammatiken und die Chomsky-Hierarchie
- Kontextfreie Grammatiken und Kontextfreie Sprachen
- Kellerautomaten

**Lernziele und Kompetenzen:**

Die Studierenden

- erwerben fundierte Kenntnisse über die Grenzen der Berechenbaren, insbesondere lernen sie, wie man beweist, dass bestimmte Aufgaben unlösbar sind bzw. dass sie vermutlich nicht schnell gelöst werden können;
- lernen die wesentlichen Techniken kennen, mit denen man Programmiersprachen beschreiben und syntaktisch korrekte Programme erkennen kann;
- erwerben fundierte Kenntnisse in den Beweis- und Analyse-Methoden der algorithmisch orientierten Theoretischen Informatik

**Literatur:**

- I. Wegener. Theoretische Informatik. Teubner
- J. Hopcroft, J. Ullman. Introduction to Automata Theory, Languages and Computation

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

[1] **Informatik (Bachelor of Science): 4. Semester**

(Po-Vers. 2009s | weitere Pflichtmodule)

Dieses Modul ist daneben auch in den Studienfächern "Mathematik (Bachelor of Science)" verwendbar.

**Studien-/Prüfungsleistungen:**

Berechenbarkeit und Formale Sprachen (Vorlesung mit Übungen) (Prüfungsnummer: 30101)

(englische Bezeichnung: Theory of Computation and Formal Languages)

Prüfungsleistung, Klausur, Dauer (in Minuten): 90

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Berechenbarkeit und Formale Sprachen
- Übungen zu Berechenbarkeit und Formale Sprachen

Erstabelleung: WS 2019/2020, 1. Wdh.: SS 2020

1. Prüfer: Rolf Wanka (100367)

## Übungen zu Berechenbarkeit und Formale Sprachen (Prüfungsnummer: 30102)

(englische Bezeichnung: Theory of Computation and Formal Languages (Exercises))

Studienleistung, Übungsleistung

Zugeordnete Lehrveranstaltungen:

- Übungen zu Berechenbarkeit und Formale Sprachen

weitere Erläuterungen:

Zum erfolgreichen Absolvieren der Übungsleistung werden gefordert:

- Mindestens 50% der Übungspunkte
- Mind. einmal vorrechnen in der Übungsgruppe

Erstabledung: WS 2019/2020, 1. Wdh.: WS 2020/2021

1. Prüfer: Rolf Wanka (100367)

---

---

**Modulbezeichnung:** Theorie der Programmierung (ThProg) 7.5 ECTS  
 (Theory of Programming)

Modulverantwortliche/r: Lutz Schröder

Lehrende: Lutz Schröder, Christoph Rauch

Startsemester: SS 2019

Dauer: 1 Semester

Turnus: jährlich (SS)

Präsenzzeit: 90 Std.

Eigenstudium: 135 Std.

Sprache: Deutsch

---

**Lehrveranstaltungen:**

Theorie der Programmierung (SS 2019, Vorlesung, 4 SWS, Lutz Schröder)

Übungen zu Theorie der Programmierung (SS 2019, Übung, 2 SWS, Christoph Rauch et al.)

Intensivübung zu Theorie der Programmierung (SS 2019, Übung, 2 SWS, Christoph Rauch)

---

**Inhalt:**

- Termersetzungssysteme, Normalisierung, Konfluenz
- Getypter und ungetypter Lambda-Kalkül
- Semantik von Programmiersprachen, Anfänge der Bereichstheorie
- Datentypen, Kodatentypen, Induktion und Koinduktion, Rekursion und Korekursion
- Programmverifikation, Floyd-Hoare-Kalkül
- Reguläre Sprachen und endliche Automaten
- Beschriftete Transitionssysteme, Bisimulation und Temporallogik

**Lernziele und Kompetenzen:**

*Fachkompetenz*

*Wissen*

Die Studierenden geben elementare Definitionen und Fakten zu den behandelten Formalismen wieder.

*Verstehen*

Die Studierenden

- erläutern Grundbegriffe der Syntax und Semantik von Formalismen und setzen diese zueinander in Bezug
- beschreiben und erklären grundlegende Algorithmen zu logischem Schließen und Normalisierung
- beschreiben wichtige Konstruktionen von Modellen, Automaten und Sprachen

*Anwenden*

Die Studierenden

- verfassen formale Spezifikationen sequentieller und nebenläufiger Programme
- verifizieren einfache Programme gegenüber ihrer Spezifikation durch Anwendung der relevanten Kalküle
- setzen formale Sprachen mit entsprechenden Automaten in Beziehung
- führen einfache Beweise über Programme mittels Induktion und Koinduktion

*Analysieren*

Die Studierenden

- wählen für gegebene Verifikationsprobleme geeignete Formalismen aus
- erstellen einfache Meta-Analysen formaler Systeme, etwa Konfluenzprüfung von Termersetzungssystemen
- führen einfache Meta-Beweise über Formalismen mittels Induktion und Koinduktion

*Lern- bzw. Methodenkompetenz*

Die Studierenden beherrschen das grundsätzliche Konzept des Beweises als hauptsächliche Methode des Erkenntnisgewinns in der theoretischen Informatik. Sie überblicken abstrakte Begriffsarchitekturen.

*Sozialkompetenz*

Die Studierenden lösen abstrakte Probleme in kollaborativer Gruppenarbeit.

**Literatur:**

- Glynn Winskel, Formal Semantics of Programming Languages, MIT Press, 1993



- Michael Huth, Mark Ryan, Logic in Computer Science, Cambridge University Press, 2. Auflage 2004
- Henk Barendregt, The lambda-Calculus: Its Syntax and Semantics, North Holland, 1984
- John E. Hopcroft, Jeffrey D. Ullman and Rajeev Motwani, Introduction to Automata Theory, Languages, and Computation, 3rd ed., Prentice Hall, 2006
- Franz Baader, Tobias Nipkow, Term Rewriting and All That, Cambridge University Press, 1999

---

### Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

#### [1] Informatik (Bachelor of Science)

(Po-Vers. 2009s | weitere Pflichtmodule)

Dieses Modul ist daneben auch in den Studienfächern "Mathematik (Bachelor of Science)" verwendbar.

---

### Studien-/Prüfungsleistungen:

Theorie des Programmierens (Klausur) (Prüfungsnummer: 31211)

Prüfungsleistung, Klausur, Dauer (in Minuten): 90

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Theorie der Programmierung
- Übungen zu Theorie der Programmierung

weitere Erläuterungen:

Die im Rahmen der Übungen gestellten Übungsaufgaben können abgegeben werden und werden in diesem Fall bewertet. Auf Basis des Ergebnisses dieser Bewertungen können bis zu 15% Bonuspunkte erworben werden, die zu dem Ergebnis einer bestandenen Klausur hinzugerechnet werden.

Erstablingung: SS 2019, 1. Wdh.: WS 2019/2020 (nur für Wiederholer)

1. Prüfer: Lutz Schröder (100190)

---

---

**Modulbezeichnung:** **Algorithmik kontinuierlicher Systeme (AlgoKS)** **7.5 ECTS**  
 (Algorithms and data structures for processing continuous data)

Modulverantwortliche/r: Ulrich Rüde  
 Lehrende: Ulrich Rüde

---

Startsemester: SS 2019	Dauer: 1 Semester	Turnus: jährlich (SS)
Präsenzzeit: 90 Std.	Eigenstudium: 135 Std.	Sprache: Deutsch

---

**Lehrveranstaltungen:**

Algorithmik kontinuierlicher Systeme (SS 2019, Vorlesung, 4 SWS, Harald Köstler)  
 Übung zu Algorithmik kontinuierlicher Systeme (SS 2019, Übung, 2 SWS, Marco Heisig et al.)  
 Rechnerübung zu Algorithmik kontinuierlicher Systeme (SS 2019, Übung, Marco Heisig et al.)

---

**Inhalt:**

- Grundlagen kont. Datenstrukturen (Gleitpunktzahlen, Rundungsfehleranalyse und Kondition, Diskretisierung und Quantisierung, Abtasttheorem, FFT)
- Algorithmische Lineare Algebra (direkte und iterative Verfahren für lin. Gleichungssysteme, Ausgleichsprobleme)
- Datenstrukturen für geometrische Objekte, Interpolation, Approximation, Grundlagen geometrischer Modellierung, Volumen- und Flächenberechnung.
- Kontinuierliche und diskrete Optimierung, nichtlineare Probleme.
- Grundlagen der Simulation: Algorithmen zur Lösung von Differentialgleichungen

**Lernziele und Kompetenzen:**

Die Studierenden erwerben fundierte Kenntnisse über Datenstrukturen und Algorithmen zur Behandlung kontinuierlicher Probleme. Die erworbenen Kompetenzen sind sowohl theoretische-analytischer Art (Analyse von Komplexität, Konvergenz, Fehlerentwicklung) als auch von praktischer Natur (Implementierung der Algorithmen in einer objekt-orientierten Programmiersprache).

Die Studierenden planen und bearbeiten kleine Programmierprojekte so, dass sie zeitgerecht fertig gestellt werden. Sie erwerben damit insbesondere die Grundlagen, die für ein vertieftes Studium in den Bereichen Systemsimulation, Mustererkennung, Graphischer Datenverarbeitung unabdingbar sind.

*Fachkompetenz*

*Wissen*

Die Studierenden

- geben die Definition von Gleitpunktzahlen wieder
- reproduzieren Formel zur Berechnung von Flächen und Volumina

*Verstehen*

Die Studierenden

- erklären die Kondition Problemen
- veranschaulichen Methoden der Freiformflächenmodellierung
- erläutern das Abtasttheorem und die Fouriertransformation

*Anwenden*

Die Studierenden

- implementieren Algorithmen zur Lösung von linearen Gleichungssystemen
- lösen Interpolation- und Approximationsaufgaben
- berechnen iterativ Lösungen von nichtlinearen Gleichungen

*Analysieren*

Die Studierenden

- klassifizieren Optimierungsprobleme
- erforschen lineare Ausgleichsprobleme

*Lern- bzw. Methodenkompetenz*

Die Studierenden

- beherrschen Analyse und Lösung kontinuierlicher Probleme durch Diskretisierung, Implementierung und Rekonstruktion.

### Sozialkompetenz

Die Studierenden

- lösen Aufgaben der Algorithmen kontinuierlicher Problem in Gruppenarbeit

---

### Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

#### [1] Informatik (Bachelor of Science): 3. Semester

(Po-Vers. 2009s | weitere Pflichtmodule)

Dieses Modul ist daneben auch in den Studienfächern "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Bachelor of Science)", "Informations- und Kommunikationstechnik (Bachelor of Science)", "Medizintechnik (Bachelor of Science)" verwendbar.

---

### Studien-/Prüfungsleistungen:

Algorithmik kontinuierlicher Systeme (Vorlesung mit Übungen) (Prüfungsnummer: 30001)

Prüfungsleistung, Klausur, Dauer (in Minuten): 90

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Algorithmik kontinuierlicher Systeme
- Übung zu Algorithmik kontinuierlicher Systeme

Erstablingung: SS 2019, 1. Wdh.: WS 2019/2020

1. Prüfer: Ulrich Rüde (100111), 2. Prüfer: Harald Köstler (100124)

Übungen zu Algorithmik kontinuierlicher Systeme (Prüfungsnummer: 30002)

Studienleistung, Übungsleistung

Zugeordnete Lehrveranstaltungen:

- Übung zu Algorithmik kontinuierlicher Systeme
- Rechnerübung zu Algorithmik kontinuierlicher Systeme

weitere Erläuterungen:

Für das Bestehen der Übungen müssen mindestens

- 50% der Theorieaufgaben und
- 50% der Programmieraufgaben

Prüfungssprache: Deutsch

Erstablingung: SS 2019, 1. Wdh.: WS 2019/2020

1. Prüfer: Ulrich Rüde (100111), 2. Prüfer: Harald Köstler (100124)

---

**Modulbezeichnung:** Implementierung von Datenbanksystemen (IDB) **5 ECTS**  
(Implementation of Database Systems)

Modulverantwortliche/r: Klaus Meyer-Wegener  
Lehrende: Klaus Meyer-Wegener

Startsemester: WS 2019/2020      Dauer: 1 Semester      Turnus: jährlich (WS)  
Präsenzzeit: 60 Std.      Eigenstudium: 90 Std.      Sprache: Deutsch

**Lehrveranstaltungen:**

Implementierung von Datenbanksystemen (WS 2019/2020, Vorlesung, 2 SWS, Klaus Meyer-Wegener)  
Übungen zu Implementierung von Datenbanksystemen (WS 2019/2020, Übung, 2 SWS, Demian Vöhringer)

**Vorhergehende Module:**

Konzeptionelle Modellierung

**Inhalt:**

Die Vorlesung soll einführen in den Aufbau und die Architektur von Datenbanksystemen, die Modularisierung und Schichtenbildung mit Abstraktionen verwenden. Schwerpunkt sind deshalb systemtechnische Aspekte von Datenbanksystemen.

Ausgangspunkt einer Reihe von aufeinander aufbauenden Abstraktionen ist die Speicherung von Daten auf Hintergrundspeichern. Die erste Abstraktion ist die Datei. Dann werden Sätze eingeführt und auf verschiedene Weisen in Blöcken organisiert (sequenziell, mit Direktzugriff, indexsequentiell). Das schließt die Organisation eines Blockpuffers und Zugriffspfade (Indexstrukturen) unterschiedlichen Typs ein. Als zweite große Abstraktion werden Datenmodelle eingeführt und hier insbesondere das relationale. Dazu gehören sowohl Strukturen als auch Anfragesprachen wie SQL.

Der zweite Teil befasst sich mit der Realisierung der Leistungen eines Datenbanksystems unter Verwendung der vorher eingeführten Sätze und Zugriffspfade ("top-down"). Das umfasst die Anfrageverarbeitung und -optimierung, aber auch die Mechanismen zur Protokollierung von Aktionen und zur Wiederherstellung von Datenzuständen nach einem Fehler oder Ausfall. Ein Schichtenmodell fasst abschließend die Aufgaben in einer Architektur für Datenbank-Verwaltungssysteme zusammen. Ziel der Vorlesung ist es also, ein grundlegendes Verständnis für den Aufbau und die Funktionsweise eines Datenbanksystems zu vermitteln.

**Lernziele und Kompetenzen:**

Die Studierenden

- kennen das Schichtenmodell eines Datenbankverwaltungssystems;
- verstehen das Prinzip der Datenunabhängigkeit (Datenabstraktion);
- beherrschen das Aufbauprinzip einer Software-Schicht;
- unterscheiden die Begriffe "Datenbank", "Datenbanksystem" und "Datenbankverwaltungssystem";
- unterscheiden die Begriffe "Datenmodell" und "Schema";
- zeigen das Konzept der blockorientierten Datei mit ihren Zugriffsoperationen auf;
- unterscheiden einen Satz von einem Block;
- erklären das Konzept der sequentiellen Satzdatei;
- schildern das Prinzip der Wechselpuffertechnik;
- charakterisieren den Schlüsselzugriff auf Sätze;
- stellen Gestreute Speicherung (Hashing) auf der Basis von Blöcken (Buckets) dar;
- formulieren die Funktionsweise des Virtuellen Hashings;
- fassen die Funktionsweise eines B-Baums zusammen;
- unterscheiden die Dienste eines B-Baums von denen des Hashings;
- können für eine Folge von Schlüsselwerten einen B-Baum aufbauen;
- unterscheiden einen B-Baum von einem B-Stern-Baum (B+-Baum);
- veranschaulichen einen Bitmap-Index;
- unterscheiden die Primär- und Sekundärorganisation von Sätzen;
- zählen Ersetzungsstrategien der Pufferverwaltung auf und vergleichen sie;

- benennen die Dienste einer Pufferverwaltung;
- erklären die Konzepte "Seite" und "Segment" im Gegensatz zu "Block" und "Datei";
- unterscheiden direkte und indirekte Seitenzuordnung;
- interpretieren in Programmiersprachen eingebettete Anfragesprachen und Datenbank-Unterprogrammaufrufe;
- charakterisieren Datenbank-Transaktionen;
- kennen die Aufrufe zur Definition von Transaktionen;
- erläutern die spaltenweise Abspeicherung von Relationen;
- diskutieren die algebraische Optimierung von Anfragen;
- stellen Planoperatoren eines Datenbanksystems dar;
- unterscheiden Planoperatoren für den Verbund;
- beschreiben Kostenformeln für die Abschätzung von Anfrageausführungen;
- schildern die verschiedenen Anomalien im Mehrbenutzerbetrieb;
- beschreiben die Serialisierbarkeit von Transaktionen;
- erläutern das Konzept der Sperren in Datenbanksystemen;
- unterscheiden physische und logische Konsistenz;
- kennen die vier Recovery-Klassen;
- erläutern die verschiedenen Arten von Sicherungspunkten.

**Literatur:**

KEMPER, Alfons ; EICKLER, André: Datenbanksysteme : Eine Einführung. 9., aktual. u. erweit. Aufl. München : Oldenbourg, 2013. - ISBN 978-3-486-72139-3. - Kapitel 7 bis 11  
 KEMPER, Alfons ; WIMMER, Martin: Übungsbuch Datenbanksysteme. 2., aktual. u. erweit. Aufl. München : Oldenbourg, 2009. - ISBN 978-3-486-59001-2. - Kapitel 7 bis 11  
 HEUER, Andreas ; SAAKE, Gunter: Datenbanken : Konzepte und Sprachen. 3., aktual. u. erw. Aufl. Bonn : mitp, 2007. - ISBN 3-8266-1664-2  
 HÄRDER, Theo ; RAHM, Erhard: Datenbanksysteme : Konzepte und Techniken der Implementierung. Berlin : Springer, 1999 - ISBN 3-540-65040-7  
 SAAKE, Gunter ; HEUER, Andreas: Datenbanken : Implementierungstechniken. 2., aktual. u. erw. Aufl. Bonn : mitp, 2005. ISBN 3-8266-1438-0

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

**[1] Informatik (Bachelor of Science): 4. Semester**

(Po-Vers. 2009s | weitere Pflichtmodule)

Dieses Modul ist daneben auch in den Studienfächern "Berufspädagogik Technik (Master of Education)", "Informatik (1. Staatsprüfung für das Lehramt an Gymnasien)", "Informatik (1. Staatsprüfung für das Lehramt an Hauptschulen)", "Informatik (1. Staatsprüfung für das Lehramt an Mittelschulen)", "Informatik (1. Staatsprüfung für das Lehramt an Realschulen)", "Informatik (Bachelor of Arts (2 Fächer))", "Information and Communication Technology (Master of Science)", "Informations- und Kommunikationstechnik (Master of Science)", "Mathematik (Bachelor of Science)", "Wirtschaftsinformatik (Bachelor of Science)" verwendbar.

**Studien-/Prüfungsleistungen:**

Implementierung von Datenbanksystemen (Prüfungsnummer: 30201)

Prüfungsleistung, Klausur mit MultipleChoice, Dauer (in Minuten): 90

Anteil an der Berechnung der Modulnote: 100%

Erstablingung: WS 2019/2020, 1. Wdh.: SS 2020, 2. Wdh.: WS 2020/2021

1. Prüfer: Klaus Meyer-Wegener (100153)

---

**Modulbezeichnung: Mathematik C3 (IngMathC3)** **7.5 ECTS**  
 (Mathematics C3)

Modulverantwortliche/r: Martin Gugat

Lehrende: Serge Kräutle, u. a. Hochschullehrer

---

Startsemester: WS 2019/2020

Dauer: 1 Semester

Turnus: jährlich (WS)

Präsenzzeit: 90 Std.

Eigenstudium: 135 Std.

Sprache: Deutsch

---

**Lehrveranstaltungen:**

Mathematik für Ingenieure C3:INF (WS 2019/2020, Vorlesung, 4 SWS, Frauke Liers)

Übungen zur Mathematik für Ingenieure C3:INF (WS 2019/2020, Übung, 2 SWS, Frauke Liers)

---

**Inhalt:**

**Anwendung der Differentialrechnung im  $\mathbb{R}^n$**

Extremwertaufgaben, Extremwertaufgaben mit Nebenbedingungen, Lagrange-Multiplikatoren, Theorem über implizite Funktionen, Anwendungsbeispiele

**Optimierung und Algebraische Strukturen**

Mathematische Grundlagen der linearen Optimierung und geometrische Interpretation, Simplex, Konvexität, Dualität Binäre Operationen, Monoide, Halbgruppen, Gruppen, Homomorphismen, Ringe, Körper, Vektorräume über endlichen Körpern, Einführung in Kryptographie und Kanalcodierung

**Gewöhnliche Differentialgleichungen**

Explizite Lösungsmethoden, Existenz- und Eindeutungssätze, Lineare Differentialgleichungen, Systeme von Differentialgleichungen, Eigen- und Hauptwertaufgaben, Fundamentalsysteme, Stabilität

**Lernziele und Kompetenzen:**

Die Studierenden

- klassifizieren verschiedene Extremwertaufgaben anhand der Nebenbedingungen und kennen die grundlegende Existenzaussagen
- erschließen den Unterschied zur eindimensionalen Kurvendiskussion,
- wenden die verschiedenen Extremwertaufgaben bei Funktionen mehrerer Veränderlicher mit und ohne Nebenbedingungen,
- beherrschen die grundlegenden Begriffe der Algebra,
- klassifizieren gewöhnliche Differentialgleichungen nach Typen,
- wenden elementare Lösungsmethoden auf Anfangswertprobleme bei gewöhnlichen Differentialgleichungen an,
- wenden allgemeine Existenz- und Eindeutigkeitsresultate an,
- erschließen den Zusammenhang zwischen Analysis und linearer Algebra,
- wenden die erlernten mathematischen Methoden auf die Ingenieurwissenschaften an,
- beachten die Vorzüge einer regelmaessigen selbstaendigen Nachbereitung und Vertiefung des Vorlesungsstoffes.

**Literatur:**

Skripte des Dozenten

M. Fried, Mathematik für Ingenieure II für Dummies, Wiley

A. Hoffmann, B. Marx, W. Vogt: *Mathematik für Ingenieure 1,2* Pearson

K. Finck von Finckenstein, J. Lehn et. al.: *Arbeitsbuch für Ingenieure, Band I und II*, Teubner

H. Heuser: *Gewöhnliche Differentialgleichungen* Teubner

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

[1] **Informatik (Bachelor of Science): 4. Semester**

(Po-Vers. 2009s | weitere Pflichtmodule)

---

**Studien-/Prüfungsleistungen:**

Mathematik C3 (Prüfungsnummer: 46001)

Prüfungsleistung, Klausur, Dauer (in Minuten): 90

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Mathematik für Ingenieure C3:INF

Erstablingung: WS 2019/2020, 1. Wdh.: SS 2020

1. Prüfer: Frauke Liers (060129)

Mathematik C3 Übungen (Prüfungsnummer: 46002)

Studienleistung, Übungsleistung

Zugeordnete Lehrveranstaltungen:

- Übungen zur Mathematik für Ingenieure C3:INF

weitere Erläuterungen:

Erwerb der Übungsleistung durch Lösung der wöchentlichen Hausaufgaben. Die Lösungen sind in handschriftlicher Form abzugeben.

Erstablingung: WS 2019/2020, 1. Wdh.: keine Angabe

1. Prüfer: Frauke Liers (060129)

---

---

**Modulbezeichnung:** **Mathematik C4 (IngMathC4)** **7.5 ECTS**  
 (Mathematics C4)

Modulverantwortliche/r: Martin Gugat  
 Lehrende: Wigand Rathmann

---

Startsemester: SS 2019	Dauer: 1 Semester	Turnus: jährlich (SS)
Präsenzzeit: 84 Std.	Eigenstudium: 141 Std.	Sprache: Deutsch

---

**Lehrveranstaltungen:**

Mathematik für Ingenieure C4: INF (SS 2019, Vorlesung, 4 SWS, Wilhelm Merz)  
 Übungen zur Mathematik für Ingenieure C4: INF (SS 2019, Übung, 2 SWS, Wilhelm Merz)

---

**Inhalt:**

**Allgemeine und Diskrete Wahrscheinlichkeitsrechnung:**

- Rechenregeln für Wahrscheinlichkeiten
- Bedingte Wahrscheinlichkeiten, stochastische Unabhängigkeit, Produktexperimente
- Charakterisierung diskreter Wahrscheinlichkeitsverteilungen durch Wahrscheinlichkeitsfunktion, erzeugende Funktion, Momente
- Eigenschaften und Anwendungsgebiete der wichtigsten diskreten Verteilungen
- Markoffketten

**Kontinuierliche Wahrscheinlichkeitsrechnung:**

- Charakterisierung von Wahrscheinlichkeitsverteilungen auf dem  $\mathbb{R}^n$  durch Verteilungsfunktionen und Dichten
- Verteilung, Erwartungswert und Varianz von Zufallsvariablen
- Funktionen von Zufallsvariablen
- Mehrdimensionale Normalverteilung
- Grenzwertsätze

**Statistische Datenanalyse:**

- Statistische Analysemethoden: Parameterschätzung, Konfidenzbereiche, Signifikanztests
- Lineare statistische Modelle: Regression und Varianzanalyse

**Lernziele und Kompetenzen:**

Die Studierenden

- untersuchen die grundlegende Begriffe und Methoden der diskreten und kontinuierlichen Wahrscheinlichkeitsrechnung,
- berechnen charakteristische Größen (Erwartungswerte, Momente, erz. Funktion)
- erstellen stochastische Modelle in typischen Anwendungssituationen,
- kennen Eigenschaften von diskreten und kontinuierlichen Verteilungen und nutzen diese bei der Lösung von Problemstellungen,
- beherrschen und verwenden Rechenverfahren für stochastische Fragestellungen,
- untersuchen die Eigenschaften von Schätzern,
- wenden Verfahren der beschreibenden und mathematischen Statistik, insbesondere Schätzer und Signifikanztests
- analysieren Modelle, die auf homogenen Markoffketten basieren
- schätzen die Vorzüge einer regelmäßigen Nachbereitung und Vertiefung des Vorlesungsstoffes

**Literatur:**

- Skripte des Dozenten
  - Gerhard Hübner, Stochastik, Vieweg + Teubner, 2009
  - Ulrich Krengel, Einführung in die Wahrscheinlichkeitstheorie und Statistik, Vieweg Verlag
  - John A. Rice, Mathematical Statistics and Data Analysis, Duxbury Press
- 

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

[1] **Informatik (Bachelor of Science): 5. Semester**

(Po-Vers. 2009s | weitere Pflichtmodule)



Dieses Modul ist daneben auch in den Studienfächern "Mathematik Export" verwendbar.

---

**Studien-/Prüfungsleistungen:**

Klausur Mathematik C 4 (Prüfungsnummer: 46101)

(englische Bezeichnung: Examination (Klausur) on Mathematics C 4)

Prüfungsleistung, Klausur, Dauer (in Minuten): 90

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Mathematik für Ingenieure C4: INF

Erstablingung: SS 2019, 1. Wdh.: WS 2019/2020 (nur für Wiederholer)

1. Prüfer: Wilhelm Merz (060249)

Übung Mathematik C 4 (Prüfungsnummer: 46102)

(englische Bezeichnung: Tutorial Certificate: Mathematics C 4)

Studienleistung, Übungsleistung

Zugeordnete Lehrveranstaltungen:

- Übungen zur Mathematik für Ingenieure C4: INF

weitere Erläuterungen:

Erwerb der Übungsleistung durch Lösung der wöchentlichen Hausaufgaben. Die Lösungen sind in handschriftlicher Form abzugeben.

Erstablingung: SS 2019, 1. Wdh.: keine Angabe

1. Prüfer: Wilhelm Merz (060249)

---

---

**Modulbezeichnung:** **Software Architecture (PROJ 5-ECTS) (OSS-ARCH-PROJ) 5 ECTS**  
 (Software Architecture (PROJ 5-ECTS))

Modulverantwortliche/r: Martin Jung, Dirk Riehle

Lehrende: Martin Jung, Dirk Riehle

Startsemester: SS 2019

Dauer: 1 Semester

Turnus: jährlich (SS)

Präsenzzeit: k.A. Std.

Eigenstudium: 150 Std.

Sprache: Deutsch

---

**Lehrveranstaltungen:**

Software Architecture (PROJ) (SS 2019, Sonstige Lehrveranstaltung, 2 SWS, Martin Jung et al.)

---

**Inhalt:**

This course teaches students concepts, methods, and tools of software architecture.

Die Architektur eines Softwaresystems beschreibt die wesentlichen Komponenten des Systems, ihre Beziehungen und Struktur, sowie das Verhalten und die Dynamik der Beziehungen und Struktur dieser Komponenten. Dieser Kurs vermittelt in einer Vorlesung zunächst die folgenden Aspekte von Softwarearchitektur:

- Grundlegende Bausteine und ihre Beziehungen
- Softwarearchitekturbeschreibungssprachen
- Softwarearchitekturstile und -muster
- Bibliotheken, Rahmenwerke und Plattformen
- Formale sowie de-facto Industriestandards
- Die Softwarearchitekturen von Beispielsystemen
- Nicht technische Kriterien in der Architektur
- Werkzeuge für Softwarearchitekten
- Vorgehensmodelle der Softwarearchitektur
- Architekturgetriebene Entwicklung
- Die Rolle und Funktion der Softwarearchitektin

Studierende können eine oder beide von zwei Komponenten wählen:

- VUE (Vorlesung + Übungen), 4 SWS, 5 ECTS
- PROJ (kleines Projekt), 2 SWS, 5 ECTS. Die Projekte werden von unseren Industriepartnern bereitgestellt. Hier dokumentieren, analysieren und bewerten Studierende die Softwarearchitektur eines realen Softwaresystems. Diese verschiedenen Aspekte werden im Laufe des Semesters inkrementell abgearbeitet und am Ende dem Industriepartner in einer Präsentation vorgestellt.

Der Unterricht findet als 3h-Block während der Vorlesungszeit statt. Der Zeitplan befindet sich hier: <http://goo.gl/ZXJjg> . Der Zeitplan enthält auch einen Link auf den zur Veranstaltung gehörigen StudOn Kurs. Bitte registrieren Sie sich auf StudOn sobald wie möglich.

**Lernziele und Kompetenzen:**

- Ganzheitliches Verständnis des Konzepts "Softwarearchitektur"
- Befähigung zur Bewertung, Auswahl und Konstruktion problemangemessener Architekturen
- Kenntnis architekturgetriebener Entwicklungsmethodik und entsprechender Werkzeuge
- Kenntnis der typischen Verantwortlichkeiten und der Methodik eines Softwarearchitekten

**Literatur:**

- <http://goo.gl/ou7mja>

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

[1] **Informatik (Bachelor of Science): 3-5. Semester**

(Po-Vers. 2009s | Seminare, Praktika, Bachelorarbeit)

Dieses Modul ist daneben auch in den Studienfächern "International Information Systems (IIS) (Master of Science)" verwendbar.

---

**Studien-/Prüfungsleistungen:**

Software Architecture (PROJ 5-ECTS) (Prüfungsnummer: 568512)

(englische Bezeichnung: Software Architecture (PROJ 5-ECTS))

Prüfungsleistung, Seminarleistung

Anteil an der Berechnung der Modulnote: 100%

weitere Erläuterungen:

- Unterricht
- Hausaufgaben
- Mündliche Prüfung
- Projektarbeit

Erstablingung: SS 2019, 1. Wdh.: keine Angabe

1. Prüfer: Martin Jung (100803), 2. Prüfer: Dirk Riehle (100100)

---

---

**Modulbezeichnung:** "Hallo Welt!" für Fortgeschrittene (inf2-algo) 5 ECTS  
 ("Hello World!" for Advanced Learners)

Modulverantwortliche/r: Michael Philippsen  
 Lehrende: Daniela Novac, Michael Baer

---

Startsemester: SS 2019	Dauer: 1 Semester	Turnus: jährlich (SS)
Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Deutsch

---

**Lehrveranstaltungen:**

"Hallo Welt!" für Fortgeschrittene (SS 2019, Seminar, 3 SWS, Anwesenheitspflicht, Daniela Novac et al.)

---

**Inhalt:**

Programmierungswettbewerbe wie der **International Collegiate Programming Contest (ICPC)** der ACM bieten die Möglichkeit, die eigenen Programmier- und Teamfähigkeiten an einer Vielzahl algorithmischer Probleme aus ganz verschiedenen Gebieten wie Geometrie, Kombinatorik, String-Verarbeitung und Zahlentheorie zu testen. Dabei treten die Studenten in 3er-Teams an, haben aber nur einen Computer zur Verfügung. Oft ist die Teamstrategie entscheidend für den Erfolg der Gruppe. In diesem Seminar werden wichtige Algorithmen zur Lösung von Problemen aus den verschiedenen Gebieten in wöchentlichen, studentischen Vorträgen vorgestellt und Standardverfahren eingeübt. Neben den Vorträgen werden die aktuell zu lösenden Aufgaben in einer simulierten Wettbewerbssituation in 3er-Teams besprochen und Lösungsansätze in der Gesamtgruppe diskutiert. Das Seminar bereitet auf die Teilnahme am Programmierungswettbewerb der Universität Erlangen-Nürnberg Ende des Sommersemesters vor. Es besteht Teilnahmepflicht für diesen Wettbewerb.

**Lernziele und Kompetenzen:**

Nach der Teilnahme an diesem Seminar sind die Studierenden in der Lage:

- erforderliche Literatur aufzufinden, zu analysieren und zu bewerten,
- sich eigenständig in ein Themengebiet einzuarbeiten,
- Grundzüge der Präsentationstechniken anzuwenden,
- eine Präsentation mit Begleitmaterial für ein Fachpublikum zu entwickeln,
- einen Vortrag im vorgegebenen Zeitrahmen durchzuführen,
- Sachverhalte unter Fachleuten zu diskutieren,
- grundlegende und fortgeschrittene Algorithmen aus diversen Themengebieten darzulegen,
- den Zusammenhang zwischen der Laufzeit und dem Speicherverbrauch von Programmen und der theoretischen Komplexität (O-Kalkül) anzugeben,
- algorithmische Probleme zu analysieren und die gelernten Algorithmen zur Lösung anzuwenden,
- Lösungsideen in Kleingruppen zu entwickeln und diese in der Gesamtgruppe zu präsentieren.

**Literatur:**

Skiena/Revilla, Programming Challenges. The Programming Contest Training Manual. Springer 2003.  
 Cormen/Leiserson/Rivest/Stein, Introduction to Algorithms. MIT Press 2001.

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

[1] **Informatik (Bachelor of Science): 3-5. Semester**

(Po-Vers. 2009s | Seminare, Praktika, Bachelorarbeit)

---

**Studien-/Prüfungsleistungen:**

Seminar "Hallo Welt!" für Fortgeschrittene (Prüfungsnummer: 319374)

(englische Bezeichnung: "Hello World!" for Advanced Learners)

Prüfungsleistung, Seminarleistung

Anteil an der Berechnung der Modulnote: 100%

weitere Erläuterungen:

Scheinkriterien für Bachelor-Seminarschein:

- 45 - 60 Minuten Vortrag, aus jedem Gebiet mindestens eine gelöste Programmieraufgabe, 30 Aufgaben insgesamt, Anwesenheit bei den Vorträgen, Prüfung über den eigenen Vortrag (ECTS: 5)

Scheinkriterien für Master-Seminarschein:

- 45 - 60 Minuten Vortrag, aus jedem Gebiet mindestens eine gelöste Programmieraufgabe, 30 Aufgaben insgesamt (10 schwere Aufgaben), Anwesenheit bei den Vorträgen, Prüfung über den eigenen Vortrag und zwei weitere Themengebiete (ECTS: 5)

Scheinkriterien für Schlüsselqualifikation:

- 45 - 60 Minuten Vortrag, aus jedem Gebiet mindestens eine gelöste Programmieraufgabe, 30 Aufgaben insgesamt, Anwesenheit bei den Vorträgen, nicht mit Informatik als Haupt-/Nebenfach möglich (ECTS: 5)

Erstablingung: SS 2019, 1. Wdh.: WS 2019/2020

1. Prüfer: Michael Philippsen (100113)

---

**Organisatorisches:**

AuD, Spaß am Programmieren, Teilnahme am ICPC

**Bemerkungen:**

Anmeldung über eMail an [hallowelt@i2.cs.fau.de](mailto:hallowelt@i2.cs.fau.de). Eine Anmeldung sollte eine ganz kurze (1-2 Sätze) Erklärung enthalten, die uns vermuten lässt, dass du motiviert und fähig bist, die Aufgaben zu lösen (z.B. Teilnahme an früheren ICPCs (Ergebnis irrelevant) oder gute Note in AuD).

---

**Modulbezeichnung:** Seminar Theoretische Informatik (ThInfSem) 5 ECTS  
(Theoretical Computer Science (Seminar))

Modulverantwortliche/r: Lutz Schröder  
Lehrende: Lutz Schröder

---

Startsemester: SS 2019	Dauer: 1 Semester	Turnus: halbjährlich (WS+SS)
Präsenzzeit: 30 Std.	Eigenstudium: 120 Std.	Sprache: Englisch

---

**Lehrveranstaltungen:**

Seminar Theoretische Informatik (SS 2019, Oberseminar, 2 SWS, Lutz Schröder)

---

**Inhalt:**

- . Theory of concurrency
  - Programming semantics
- . Categories in computer science
  - Logic in computer science
  - Theory of artificial intelligence

**Lernziele und Kompetenzen:**

*Fachkompetenz*

*Wissen*

Die Studierenden geben grundlegende Definitionen und Resultate aus einem vertieften Bereich der Theoretischen Informatik wieder.

*Verstehen*

Die Studierenden erläutern grundlegende algorithmische und deduktive Verfahren.

*Anwenden*

Die Studierenden erläutern Anwendungen theoretischer Methoden in konkreten informatischen Kontexten.

*Erschaffen*

Die Studierenden bereiten Resultate des Gebiets und ihre Herleitung eigenständig zur Präsentation auf.

*Lern- bzw. Methodenkompetenz*

Die Studierenden erarbeiten sich selbständig den Inhalt wissenschaftlicher Veröffentlichungen und geben diesen in einem strukturierten Vortrag verständlich wieder.

*Sozialkompetenz*

Die Studierenden arbeiten in einer größeren Gruppe an komplexen Problemen und erläutern eigene Einsichten den Mitstudierenden in verständlicher Form.

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

[1] **Informatik (Bachelor of Science): 3-5. Semester**

(Po-Vers. 2009s | Seminare, Praktika, Bachelorarbeit)

---

**Studien-/Prüfungsleistungen:**

Seminar Theoretische Informatik (Prüfungsnummer: 863761)

(englische Bezeichnung: Theoretical Computer Science (Seminar))

Prüfungsleistung, Seminarleistung

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Seminar Theoretische Informatik

weitere Erläuterungen:

Die Prüfungsleistung besteht in einer Ausarbeitung und einem 90-minütigen erfolgreichen Vortrag.

Prüfungssprache: Englisch

Erstablesung: SS 2019, 1. Wdh.: WS 2019/2020 (nur für Wiederholer)

1. Prüfer: Lutz Schröder (100190)



---

**Modulbezeichnung:** **Systems- and Networks-on-a-Chip für INF (SNoC-INF)** **5 ECTS**  
 (Systems- and Networks-on-a-Chip for INF)

Modulverantwortliche/r: Stefan Wildermann

Lehrende: Stefan Wildermann, Behnaz Pourmohseni, Hananeh Aliee

Startsemester: WS 2019/2020

Dauer: 1 Semester

Turnus: jährlich (WS)

Präsenzzeit: 30 Std.

Eigenstudium: 120 Std.

Sprache: Englisch

---

**Lehrveranstaltungen:**

Systems- and Networks-on-a-Chip (WS 2019/2020, Seminar, 2 SWS, Stefan Wildermann et al.)

---

**Inhalt:**

Eingebettete Systeme spielen im alltäglichen Leben eine immer größere Rolle. Gleichzeitig nimmt die Komplexität dieser Systeme immer weiter zu. Durch die heutige Technologie ist es möglich, Millionen, in naher Zukunft Milliarden von Transistoren auf einem Chip zu platzieren. Dies führt dazu, dass häufig das komplette eingebettete System, ein sogenanntes System-on-a-Chip (SoC), auf einem einzigen Chip realisiert werden kann. Die Vorteile einer verbesserten Performanz, niedrigerem Energieverbrauch sowie sinkenden Kosten sind dabei unter anderem durch die Wiederverwendung bestehender Komponenten bedingt. Eine der Herausforderungen bestehender SoCs besteht darin, eine korrekte und zuverlässige Kommunikation zwischen den Komponenten herzustellen. Aus diesem Grund wird den Komponenten eine netzwerkartige Kommunikation zur Verfügung gestellt, wodurch sogenannte Networks-on-a-Chip (NoCs) entstehen.

Dieses Seminar beschäftigt sich mit der Problematik von Design, Synthese und Analyse bestehender und zukünftiger Systems- und Networks-on-a-Chip. Hierbei soll vor allem die Vereinbarkeit verschiedener Anforderungen an das System wie Kosten, Platz- und Energieverbrauch oder Zuverlässigkeit in den verschiedenen Phasen der Entwicklung betrachtet werden.

**Lernziele und Kompetenzen:**

*Fachkompetenz*

*Verstehen*

- Die Studierenden tragen die wesentlichen Inhalte einer ausgewählten wissenschaftlichen Veröffentlichung auf dem Gebiet der MPSoCs vor.
- Die Studierenden veranschaulichen den grundlegenden Kontext der Veröffentlichung sowie deren wesentliche Neuerungen.
- Die Studierenden bereiten den Inhalt der ausgewählten wissenschaftlichen Veröffentlichung sowie benötigte Grundlagen in einer Ausarbeitung auf.

*Lern- bzw. Methodenkompetenz*

- Die Studierenden arbeiten sich selbstständig in eine wissenschaftliche Veröffentlichung ein und suchen hierbei selbstständig nach verwandten Arbeiten, um den Kontext der Veröffentlichung zu verstehen und aufzubereiten.

*Sozialkompetenz*

- Die Studierenden beteiligen sich aktiv an den Vorträgen der jeweils anderen Studierenden durch fachbezogene Fragen zum Thema wie auch Rückmeldung zu Vortragsstil.

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

[1] **Informatik (Bachelor of Science): 3-5. Semester**

(Po-Vers. 2009s | Seminare, Praktika, Bachelorarbeit)

---

**Studien-/Prüfungsleistungen:**



Systems- and Networks-on-a-Chip für INF (Prüfungsnummer: 834345)

Prüfungsleistung, Seminarleistung

Anteil an der Berechnung der Modulnote: 100%

weitere Erläuterungen:

Die Bewertung der Prüfungsleistung setzt sich zusammen aus 50% Seminarvortrag und 50% schriftlicher Ausarbeitung.

Prüfungssprache: Deutsch oder Englisch

Erstablingung: WS 2019/2020, 1. Wdh.: keine Angabe

1. Prüfer: Jürgen Teich (100115)

---

**Organisatorisches:**

Die Auswahl dieses Moduls schließt die Auswahl des Modules „Systems- and Networks-on-a-Chip (SNoC)“ aus.

---

**Modulbezeichnung:** Design Patterns und Anti-Patterns (DPAP) **5 ECTS**  
(Design Patterns and Anti Patterns)

Modulverantwortliche/r: Francesca Saglietti  
Lehrende: Xiaochen Wu, Andreas Neubaum

---

Startsemester: SS 2019	Dauer: 1 Semester	Turnus: jährlich (SS)
Präsenzzeit: 30 Std.	Eigenstudium: 120 Std.	Sprache: Deutsch

---

**Lehrveranstaltungen:**

Design Patterns und Anti-Patterns (SS 2019, Seminar, 2 SWS, Anwesenheitspflicht, Xiaochen Wu et al.)

---

**Inhalt:**

In diesem Seminar werden zahlreiche in der Praxis bewährte Entwurfsmuster präsentiert, sowie typische Fehler, die während des gesamten Lebenszyklus' eines Software-Systems auftreten können, zusammen mit ihrer Erkennung und Behebung vorgestellt.

**Lernziele und Kompetenzen:**

Die Studierenden

- erläutern etablierte Design Patterns sowie bekannte Anti-Patterns;
- klassifizieren sie und charakterisieren deren Stärken und Schwächen bzw. Negativfolgen;
- tragen vor Publikum über wissenschaftliche Ergebnisse vor;
- recherchieren selbständig Fachliteratur;
- fassen wissenschaftliche Erkenntnisse in Schriftform zusammen;
- nutzen Verbesserungshinweise des Betreuers zur Analyse eigener Stärken und Schwächen und leiten daraus Konsequenzen für ihr künftiges Lern-Handeln ab;
- können komplexe fachbezogene Inhalte klar und zielgruppengerecht präsentieren und argumentativ vertreten.

**Literatur:**

Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides: Design Patterns - Elements of Reusable Object-Oriented Software, Addison-Wesley Longman, 1994.

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

[1] **Informatik (Bachelor of Science): 3-5. Semester**

(Po-Vers. 2009s | Seminare, Praktika, Bachelorarbeit)

---

**Studien-/Prüfungsleistungen:**

Seminar Design Patterns und Anti-Patterns (Prüfungsnummer: 905912)

Prüfungsleistung, Seminarleistung

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Design Patterns und Anti-Patterns

weitere Erläuterungen:

Die Bewertung der Prüfungsleistung setzt sich zusammen aus Seminarvortrag (Gewichtung 1/2), Ausarbeitung (Gewichtung 1/4) und 15-minütiger mündlicher Prüfung (Gewichtung 1/4). Jede dieser Einzelleistungen muss mit mindestens 4,0 bestanden werden.

Prüfungssprache: Deutsch oder Englisch

Erstablingung: SS 2019, 1. Wdh.: WS 2019/2020

1. Prüfer: Francesca Saglietti (100112)

---

**Modulbezeichnung:** **Nailing your Thesis (PROJ 5-ECTS) (OSS-NYT-PROJ)** **5 ECTS**  
(Nailing your Thesis (PROJ 5-ECTS))

Modulverantwortliche/r: Dirk Riehle  
Lehrende: Dirk Riehle

Startsemester: SS 2019                      Dauer: 1 Semester                      Turnus: halbjährlich (WS+SS)  
Präsenzzeit: k.A. Std.                      Eigenstudium: 150 Std.                      Sprache: Englisch

**Lehrveranstaltungen:**

Nailing your Thesis (PROJ) (SS 2019, Sonstige Lehrveranstaltung, 2 SWS, Dirk Riehle)

**Inhalt:**

This course teaches students how to perform scientific research for their final thesis or a research paper. The goal is to prepare students for a Bachelor or Master research thesis.

The course covers the following topics:

- Science and society
- The research process
- Exploratory research
- Confirmatory research
- Writing a thesis/paper
- The scientific community

Students can choose one or both of two components:

- VUE (lecture + exercise), 4 SWS, 5 ECTS. VUE combines lectures with homework and exercises. VUE is run as a 3h block.
- PROJ (small research project), 2 SWS, 5 ECTS. In PROJ, students perform a small research project, either individually or in teams. The available projects will be presented at the beginning of the course. Students perform the research, write a paper, and hold a presentation about their work.

If you only want to take OSS-NYT-PROJ please still come to OSS-NYT-VUE class on the first day to learn about the projects.

The overall schedule can be found at <https://goo.gl/VqoFO> . Please sign up for the course on StudOn (link accessible through schedule spreadsheet) as soon as possible.

**Lernziele und Kompetenzen:**

- Understand how to perform research
- Understand how to write a research thesis

**Literatur:**

The syllabus, schedule, literature, and more can be found at <http://nythesis.com>

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

[1] **Informatik (Bachelor of Science): 3-5. Semester**

(Po-Vers. 2009s | Seminare, Praktika, Bachelorarbeit)

Dieses Modul ist daneben auch in den Studienfächern "International Information Systems (IIS) (Master of Science)" verwendbar.

**Studien-/Prüfungsleistungen:**

Nailing your Thesis (PROJ 5-ECTS) (Prüfungsnummer: 404439)

(englische Bezeichnung: Nailing your Thesis (PROJ 5-ECTS))

Prüfungsleistung, Seminarleistung

Anteil an der Berechnung der Modulnote: 100%

weitere Erläuterungen:

- Exercise participation
- Homework assignments

- Project work

Erstablingung: SS 2019, 1. Wdh.: keine Angabe

1. Prüfer: Dirk Riehle (100100)

---

---

**Modulbezeichnung:** Einführung in die Kryptografie (Krypto) 5 ECTS  
(Introduction To Cryptography)

Modulverantwortliche/r: Marc Spisländer

Lehrende: Andreas Neubaum, Xiaochen Wu

---

Startsemester: WS 2019/2020

Dauer: 1 Semester

Turnus: jährlich (WS)

Präsenzzeit: 30 Std.

Eigenstudium: 120 Std.

Sprache: Deutsch

---

**Lehrveranstaltungen:**

Einführung in die Kryptografie (WS 2019/2020, Seminar, 2 SWS, Anwesenheitspflicht, Andreas Neubaum et al.)

---

**Inhalt:**

In diesem Seminar werden symmetrische und asymmetrische Verfahren zur Identifikation, Verschlüsselung und Signierung behandelt. Dazu werden sowohl die mathematischen Grundlagen vermittelt als auch die entsprechenden Algorithmen vorgestellt.

**Lernziele und Kompetenzen:**

Die Studierenden

- erläutern wesentliche Konzepte der modernen Kryptografie;
  - klassifizieren Kryptoverfahren und charakterisieren deren Stärken und Schwächen;
  - tragen vor Publikum über wissenschaftliche Ergebnisse vor;
  - recherchieren selbständig Fachliteratur;
  - fassen wissenschaftliche Erkenntnisse in Schriftform zusammen;
  - nutzen Verbesserungshinweise des Betreuers zur Analyse eigener Stärken und Schwächen und leiten daraus Konsequenzen für ihr künftiges Lern-Handeln ab;
  - können komplexe fachbezogene Inhalte klar und zielgruppengerecht präsentieren und argumentativ vertreten.
- 

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

**[1] Informatik (Bachelor of Science): 3-5. Semester**

(Po-Vers. 2009s | Seminare, Praktika, Bachelorarbeit)

Dieses Modul ist daneben auch in den Studienfächern "International Information Systems (IIS) (Master of Science)" verwendbar.

---

**Studien-/Prüfungsleistungen:**

Seminar Einführung in die Kryptografie (Prüfungsnummer: 153330)

Prüfungsleistung, Seminarleistung

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Einführung in die Kryptografie

weitere Erläuterungen:

Die Bewertung der Prüfungsleistung setzt sich zusammen aus Seminarvortrag (Gewichtung 1/2), Ausarbeitung (Gewichtung 1/4) und 15-minütiger mündlicher Prüfung (Gewichtung 1/4). Jede dieser Einzelleistungen muss mit mindestens 4,0 bestanden werden.

Erstabelleung: WS 2019/2020, 1. Wdh.: SS 2020

1. Prüfer: Francesca Saglietti (100112)

---

---

**Modulbezeichnung:** **Algorithmen der Simulationstechnik (AlgoSimT)** **5 ECTS**  
 (Algorithms of Simulation Technology)

Modulverantwortliche/r: Ulrich Rüde, Marco Heisig

Lehrende: Ulrich Rüde

---

Startsemester: WS 2019/2020

Dauer: 1 Semester

Turnus: jährlich (WS)

Präsenzzeit: 30 Std.

Eigenstudium: 120 Std.

Sprache: Deutsch

---

**Lehrveranstaltungen:**

Algorithmen der Simulationstechnik (WS 2019/2020, Seminar, 2 SWS, Ulrich Rüde)

---

**Vorhergehende Module:**

Algorithmik kontinuierlicher Systeme

---

**Inhalt:**

Dieses Seminar soll im Anschluss an die Vorlesung Algorithmik kontinuierlicher Systeme den Bereich Simulationstechnik vertiefen. Das Seminar wendet sich damit in erster Linie an Bachelorstudenten der Informatik und Medizintechnik, oder auch Computational Engineering, die sich speziell für Simulationsthemen interessieren.

Die Vorträge und die jeweils anschließenden Diskussionen werden im Januar und Februar in Blockform an drei Nachmittagen stattfinden. Für die erfolgreiche Teilnahme ist die Vorbereitung des Vortrags und das Halten des Vortrags erforderlich, sowie die Ausarbeitung eines Handouts zum Vortrag und die Teilnahme an den weiteren Seminarterminen. In die Benotung gehen ein: Vortrag (50%), Vortragsfolien (30%), und Handout (20%).

**Lernziele und Kompetenzen:**

Studenten lernen in dieser Lehrveranstaltung, sich intensiv in ein Thema einzuarbeiten und in einem freien Vortrag das Thema vorzustellen. Darüber hinaus erhalten die Studenten einen Einblick in die Disziplin des wissenschaftlichen Rechnens und der Computersimulation.

**Literatur:**

Bungartz, H.-J., Zimmer, S., Buchholz, M., Pflüger, D.: Modellbildung und Simulation: Eine anwendungsorientierte Einführung, Springer, 2013, DOI: 10.1007/978-3-642-37656-6

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

[1] **Informatik (Bachelor of Science): 3-5. Semester**

(Po-Vers. 2009s | Seminare, Praktika, Bachelorarbeit)

---

**Studien-/Prüfungsleistungen:**

Algorithmen der Simulationstechnik (Prüfungsnummer: 949119)

(englische Bezeichnung: Algorithms of Simulation Technology)

Prüfungsleistung, Seminarleistung, Dauer (in Minuten): 45

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Algorithmen der Simulationstechnik

weitere Erläuterungen:

Vortrag, sowie bis zu 10 Seiten Ausarbeitung.

Prüfungssprache: Deutsch

Erstabelleung: WS 2019/2020, 1. Wdh.: keine Angabe

1. Prüfer: Ulrich Rüde (100111)

---

**Modulbezeichnung:** Themen der Kategorientheorie (TopCat) 5 ECTS  
 (Topics in Category Theory)

Modulverantwortliche/r: Stefan Milius  
 Lehrende: Stefan Milius

---

Startsemester: SS 2019	Dauer: 1 Semester	Turnus: unregelmäßig
Präsenzzeit: 30 Std.	Eigenstudium: 120 Std.	Sprache: Deutsch oder Englisch

---

**Lehrveranstaltungen:**  
 Themen der Kategorientheorie (SS 2019, Seminar, 2 SWS, Stefan Milius)

---

**Empfohlene Voraussetzungen:**  
 Grundkenntnisse der Kategorientheorie.

**Inhalt:**

Verschiedene Themen der Kategorientheorie werden in Form eines "Reading Course" von den Teilnehmern erarbeitet und vorgetragen.

Die Themenauswahl kann dabei flexibel auf die Interessen der Teilnehmer zugeschnitten werden. Mögliche, teilweise an "Algebra des Programmierens" unmittelbar anschließende, Themen sind z.B.:

- freie Konstruktionen, universelle Pfeile und adjungierte Funktoren
- Äquivalenzfunktoren
- Monaden: Eilenberg-Moore und Kleisli-Kategorien; Freie Monaden; Becks Satz
- Kartesisch abgeschlossene Kategorien
- Vollständige Halbordnungen (cpos), Einbettungen/Projektionen, Limes-Kolimes-Koinzidenz, Lösung rekursive Domaingleichungen
- Kan Erweiterungen
- (symmetrische) monoidale Kategorien
- Faktorisierungsstrukturen

**Lernziele und Kompetenzen:**

*Fachkompetenz*

*Wissen*

Die Studierenden geben Definitionen und Ergebnisse aus der Kategorientheorie und ihren Anwendungen in der Informatik wieder.

*Verstehen*

Die Studierenden erläutern grundlegende Methoden und Beweise der Kategorientheorie.

*Anwenden*

Die Studierenden wenden die Konzepte und Beweismethoden an, um einfache Problemstellungen kategoriell zu beschreiben und entsprechende Aussagen zu beweisen.

*Erschaffen*

Die Studierenden bereiten Resultate des Gebiets und ihre Herleitung eigenständig zur Präsentation auf.

*Lern- bzw. Methodenkompetenz*

Die Studierenden erarbeiten sich selbständig den Inhalt von Fachbüchern und wissenschaftlichen Veröffentlichungen und geben diesen in einem strukturierten Vortrag verständlich wieder.

*Sozialkompetenz*

Die Studierenden arbeiten in einer größeren Gruppe an komplexen Problemen und erläutern eigene Einsichten den Mitstudierenden in verständlicher Form.

**Literatur:**

- S. Mac Lane: Categories for the Working Mathematician, 2nd edition, Springer-Verlag, 1998.
- J. Adamek, H. Herrlich and G.E. Strecker: Abstract and Concrete Categories: The Joy of Cats, 2nd edition, Dover Publications, 2009.
- S. Awodey: Category Theory, 2nd edition, Oxford University Press, 2011.
- E. Riehl: Category Theory in Context, Dover Publications, 2016.

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

**[1] Informatik (Bachelor of Science): 3-5. Semester**

(Po-Vers. 2009s | Seminare, Praktika, Bachelorarbeit)

---

**Studien-/Prüfungsleistungen:**

Themen der Kategorientheorie (Prüfungsnummer: 819238)

(englische Bezeichnung: Topics in Category Theory)

Prüfungsleistung, Seminarleistung

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Themen der Kategorientheorie

weitere Erläuterungen:

Die Prüfung besteht in erfolgreichen Präsentationen im Laufe der Lehrveranstaltung und evtl. einer mündlichen Abschlussprüfung.

Prüfungssprache: Deutsch oder Englisch

Erstablesung: SS 2019, 1. Wdh.: keine Angabe

1. Prüfer: Stefan Milius (100294)

---



---

**Modulbezeichnung:** Seminar Energieinformatik (B.Sc.) (EI-BaSem) 5 ECTS  
(Seminar Energy Informatics (B.Sc.))

Modulverantwortliche/r: Marco Pruckner  
Lehrende: Marco Pruckner

---

Startsemester: WS 2019/2020	Dauer: 1 Semester	Turnus: unregelmäßig
Präsenzzeit: 30 Std.	Eigenstudium: 120 Std.	Sprache: Deutsch

---

**Lehrveranstaltungen:**

Seminar Energieinformatik (WS 2019/2020, Seminar, 2 SWS, Marco Pruckner)

---

**Inhalt:**

Teilnehmende arbeiten sich selbständig anhand der ausgewählten wissenschaftlichen Literatur in ein vorgeschlagenes Thema aus dem Bereich der Energieinformatik ein, erstellen dazu eine schriftliche Ausarbeitung, bereiten einen Seminarvortrag vor und präsentieren ihn vor Lehrenden, interessierten wissenschaftlichen Mitarbeitern des Instituts und Kommilitonen der Lehrveranstaltung.

**Lernziele und Kompetenzen:**

Zu den zu erwerbenden Kompetenzen zählen

- die Literaturrecherche,
- korrektes Zitieren,
- die Fähigkeit auszuwählen, welche Aspekte in der Ausarbeitung und im Vortrag behandelt werden,
- zielgruppengerechtes Schreiben,
- die Verwendung von Textverarbeitungswerkzeugen für die Präsentation - und die Ausarbeitung,
- sicheres Auftreten beim Vortragen eines wissenschaftlichen Themas sowie
- fachspezifische Fragen zum ausgewählten Themengebiet zu beantworten.

Am Ende jeder Lehreinheit

- stellen die Teilnehmenden gezielt Fragen zum vorgetragenen Thema,
  - diskutieren über fachliche Aspekte der Präsentation,
  - üben konstruktive Kritik an der Darstellung des Themas und
  - bewerten die Präsentation aus vortragstechnischer Sicht.
- 

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

[1] **Informatik (Bachelor of Science): 3-5. Semester**

(Po-Vers. 2009s | Seminare, Praktika, Bachelorarbeit)

---

**Studien-/Prüfungsleistungen:**

Seminar Energieinformatik (B.Sc.) (Prüfungsnummer: 666379)

(englische Bezeichnung: Seminar Energy Informatics (B.Sc.))

Prüfungsleistung, Seminarleistung

Anteil an der Berechnung der Modulnote: 100%

weitere Erläuterungen:

Der Scheinerwerb erfolgt durch Seminarbesuch, Vortrag und Abgabe einer schriftlichen Seminararbeit.

Prüfungssprache: Deutsch oder Englisch

Erstablingung: WS 2019/2020, 1. Wdh.: keine Angabe

1. Prüfer: Marco Pruckner (100381)

---

---

**Modulbezeichnung:** **Product Management (VUE 5-ECTS) (OSS-PROD-VUE)** **5 ECTS**  
 (Product Management (VUE 5-ECTS))

Modulverantwortliche/r: Dirk Riehle  
 Lehrende: Dirk Riehle

---

Startsemester: WS 2019/2020	Dauer: 1 Semester	Turnus: jährlich (WS)
Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Englisch

---

**Lehrveranstaltungen:**

Product Management (VUE) (WS 2019/2020, Vorlesung mit Übung, 2 SWS, Dirk Riehle)

---

**Inhalt:**

This course teaches students the concepts, methods, and tools of software product management. Product management is an important function in software development organizations. A product manager conceives and defines new products. His or her task is to understand the market incl. customers, to develop a product vision from that understanding, to translate it into product requirements, define those requirements on a by-feature basis and work with engineering to ensure these features are properly realized in the product under development.

- Role, tasks, and responsibilities of a product manager
- Process, methods, techniques and tools of product management
- Managing incremental/sustaining as well as disruptive innovation
- Open source product management; new trends in product management

Students can choose one or both of two components:

- VUE (lecture + homework), 2 SWS, 5 ECTS. VUE uses teaching cases as commonly used in MBA programs. The teaching cases are available for free at <http://pmbycase.com>.
- PROJ (small project), 2 SWS, 5 ECTS. In PROJ, students perform a small product management project, either individually or in teams. In such a project, students write a business plan for a market opportunity or a startup. There are two sources of projects: Those supplied by the professorship and those supplied by students. If we supply any projects, we will tell you about them in the first lecture. Student-supplied projects can be suggested by students to Prof. Riehle at least one week before class starts and must meet the requirements described here: <https://goo.gl/sNPi1i>

PROD projects are run as shared projects, in which all participants contribute and get to participate in the project results. Read more at <https://wp.me/pDU66-2p4>.

Class is run as two 90min blocks. The first block discusses the teaching cases. The second block is a coaching session for the projects (10 ECTS only).

The overall schedule can be found at <http://goo.gl/tTA10> . Please sign up for the course on StudOn (link accessible through schedule spreadsheet) as soon as possible.

**Lernziele und Kompetenzen:**

- Understand the role, function, and responsibilities of a product manager
- Understand key concepts, methods, and tools of software product management
- Understand different business situations, incl. Incremental vs. disruptive innovation

**Literatur:**

- <http://goo.gl/41Dgsr>
- 

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

**[1] Informatik (Bachelor of Science): 3-5. Semester**

(Po-Vers. 2009s | Seminare, Praktika, Bachelorarbeit)

Dieses Modul ist daneben auch in den Studienfächern "International Information Systems (IIS) (Master of Science)", "Maschinenbau (Master of Science)", "Mechatronik (Bachelor of Science)", "Mechatronik (Master of Science)" verwendbar.

---

**Studien-/Prüfungsleistungen:**

Product Management (VUE 5-ECTS) (Prüfungsnummer: 949245)

(englische Bezeichnung: Product Management (VUE 5-ECTS))

Prüfungsleistung, Seminarleistung

Anteil an der Berechnung der Modulnote: 100%

weitere Erläuterungen:

- In-class participation
- Homework assignments
- Project work

Erstablingung: WS 2019/2020, 1. Wdh.: keine Angabe

1. Prüfer: Dirk Riehle (100100)

---

---

**Modulbezeichnung:** Free/Libre, and Open Source Software (5-ECTS) (OSS-FLOSS) 5 ECTS  
(Free/Libre, and Open Source Software (5-ECTS))

Modulverantwortliche/r: Dirk Riehle  
Lehrende: Dirk Riehle

---

Startsemester: SS 2019	Dauer: 1 Semester	Turnus: jährlich (SS)
Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Englisch

---

**Lehrveranstaltungen:**

Free/Libre, and Open Source Software (VUE) (SS 2019, Vorlesung mit Übung, 4 SWS, Dirk Riehle)

---

**Inhalt:**

This course teaches students engineering, product management, and business practices of open source. It takes the classic form of a lecture with associated weekly exercises.

The topics of the lectures are:

- Introduction to open source
- Open source and intellectual property
- Open source project communities
- Open source software engineering
- Corporate open source governance
- Strategic corporate open sourcing
- Open source developer foundations
- Open source user foundations
- Open source distributors
- Single-vendor open source firms
- Open source labor economics

More details can be found in the course syllabus at <http://goo.gl/gEjk2>.

Students of computer science (Informatik) should note that this course is less about technology and more about economics and the software industry.

The course combines weekly lectures with homework. Homework typically involves reading materials and summarizing these readings in short one-page documents. Some homework consists of installing a tool (FOSSology) and using it to create a bill-of-material for open source projects, including checking for license compliance. Students should be able to install and use the tool.

The overall schedule can be found at <http://goo.gl/V6gYj>. Please sign up for the course on StudOn (link accessible through schedule spreadsheet) as soon as possible.

**Lernziele und Kompetenzen:**

- Understand how open source projects work
- Understand key aspects of the software industry
- Understand how open source is changing the software industry

**Literatur:**

- <http://goo.gl/D8qnu>
- 

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

[1] **Informatik (Bachelor of Science): 3-5. Semester**

(Po-Vers. 2009s | Seminare, Praktika, Bachelorarbeit)

Dieses Modul ist daneben auch in den Studienfächern "International Information Systems (IIS) (Master of Science)" verwendbar.

---

**Studien-/Prüfungsleistungen:**

Free/Libre, and Open Source Software (5-ECTS) (Prüfungsnummer: 171904)

(englische Bezeichnung: Free/Libre, and Open Source Software (5-ECTS))

Prüfungsleistung, Seminarleistung

Anteil an der Berechnung der Modulnote: 100%

weitere Erläuterungen:

- Exercise participation
- Homework assignments

Erstablingung: SS 2019, 1. Wdh.: keine Angabe

1. Prüfer: Dirk Riehle (100100)

---

---

**Modulbezeichnung:** Seminar Graphische Datenverarbeitung (GraGS) 5 ECTS  
(Computer Graphics Seminar)

Modulverantwortliche/r: Günther Greiner  
Lehrende: Günther Greiner, Frank Bauer

---

Startsemester: SS 2019	Dauer: 1 Semester	Turnus: jährlich (SS)
Präsenzzeit: 80 Std.	Eigenstudium: 70 Std.	Sprache: Deutsch

---

**Lehrveranstaltungen:**

Blender Seminar (SS 2019, Hauptseminar, Anwesenheitspflicht, Dominik Penk et al.)

---

**Inhalt:**

Vermitteln grundlegender Kenntnisse über den Umgang mit 3D-Modellierungstools und zur Planung von Projekten.

**Lernziele und Kompetenzen:**

*Fachkompetenz*

*Wissen*

- Darlegen von mathematischen Grundlagen verschiedener Beleuchtungsmodelle
- Reproduzieren von Konzepten der Szenengestaltung und Beleuchtung
- Mathematische Grundlagen zu Interpolationsverfahren reproduzieren können
- Auflisten von Lizenzmodellen für eigene Werke
- Berichten über verschiedene Strategien zur Projekt- und Teamplanung

*Verstehen*

- Beschreiben von verschiedenen Verfahren zur Keyframe- oder Vertexinterpolation
- Aufzeigen von Problemen bei der Erstellung von Geometrie
- Illustrieren von Problemen des Photon-Tracing

*Anwenden*

- Wissen über Szenengestaltung und Beleuchtung auf konkrete Beispiele anwenden können
- Fähigkeit zur Erstellung eigener Materialshader auf der Grundlage vorgegebener Formeln/Vorschriften demonstrieren

*Analysieren*

- Selbstständige Projektplanung, Teamorganisation und Umsetzung eines 3D-Projektes in vorgegebenem Zeitfenstern
- 

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

[1] **Informatik (Bachelor of Science): 3-5. Semester**

(Po-Vers. 2009s | Seminare, Praktika, Bachelorarbeit)

---

**Studien-/Prüfungsleistungen:**

Blender Seminar (Prüfungsnummer: 921878)

Prüfungsleistung, Seminarleistung

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Blender Seminar

weitere Erläuterungen:

Die Seminarnote besteht aus dem Seminarvortrag und der Abschlussarbeit

Prüfungssprache: Deutsch

Erstablingung: SS 2019, 1. Wdh.: keine Angabe

1. Prüfer: Günther Greiner (100159), 2. Prüfer: Frank Bauer (100413)

---

**Modulbezeichnung:** **Nailing your Thesis (VUE 5-ECTS) (OSS-NYT-VUE)** **5 ECTS**  
(Nailing your Thesis (VUE 5-ECTS))

Modulverantwortliche/r: Dirk Riehle

Lehrende: Dirk Riehle

Startsemester: SS 2019

Dauer: 1 Semester

Turnus: halbjährlich (WS+SS)

Präsenzzeit: 60 Std.

Eigenstudium: 90 Std.

Sprache: Englisch

**Lehrveranstaltungen:**

Nailing your Thesis (VUE) (SS 2019, Vorlesung mit Übung, 4 SWS, Dirk Riehle)

**Inhalt:**

This course teaches students how to perform scientific research for their final thesis or a research paper. The goal is to prepare students for a Bachelor or Master research thesis.

The course covers the following topics:

- Science and society
- The research process
- Exploratory research
- Confirmatory research
- Writing a thesis/paper
- The scientific community

Students can choose one or both of two components:

- VUE (lecture + exercise), 4 SWS, 5 ECTS. VUE combines lectures with homework and exercises. VUE is run as a 3h block.
- PROJ (small research project), 2 SWS, 5 ECTS. In PROJ, students perform a small research project, either individually or in teams. The available projects will be presented at the beginning of the course. Students perform the research, write a paper, and hold a presentation about their work.

If you only want to take OSS-NYT-PROJ please still come to OSS-NYT-VUE class on the first day to learn about the projects.

The overall schedule can be found at <https://goo.gl/VqoFO> . Please sign up for the course on StudOn (link accessible through schedule spreadsheet) as soon as possible.

**Lernziele und Kompetenzen:**

- Understand how to perform research
- Understand how to write a research thesis

**Literatur:**

The syllabus, schedule, literature, and more can be found at <http://nythesis.com>

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

**[1] Informatik (Bachelor of Science): 3-5. Semester**

(Po-Vers. 2009s | Seminare, Praktika, Bachelorarbeit)

Dieses Modul ist daneben auch in den Studienfächern "International Information Systems (IIS) (Master of Science)", "Maschinenbau (Bachelor of Science)", "Maschinenbau (Master of Science)", "Mechatronik (Bachelor of Science)", "Mechatronik (Master of Science)", "Wirtschaftsingenieurwesen (Bachelor of Science)" verwendbar.

**Studien-/Prüfungsleistungen:**

Nailing your Thesis (VUE 5-ECTS) (Prüfungsnummer: 304439)

(englische Bezeichnung: Nailing your Thesis (VUE 5-ECTS))

Prüfungsleistung, Seminarleistung

Anteil an der Berechnung der Modulnote: 100%

weitere Erläuterungen:

- Exercise participation
- Homework assignments
- Project work

Erstablingung: SS 2019, 1. Wdh.: keine Angabe  
1. Prüfer: Dirk Riehle (100100)

---



---

**Modulbezeichnung:** Seminar Wissensrepräsentation und -verarbeitung (WuV) 5 ECTS  
(Knowledge Representation and -processing)

Modulverantwortliche/r: Michael Kohlhase

Lehrende: Michael Kohlhase, Dennis Müller

---

Startsemester: SS 2019

Dauer: 1 Semester

Turnus: halbjährlich (WS+SS)

Präsenzzeit: 30 Std.

Eigenstudium: 120 Std.

Sprache: Deutsch oder Englisch

---

**Lehrveranstaltungen:**

Seminar Wissensrepräsentation und -verarbeitung (SS 2019, Oberseminar, 2 SWS, Michael Kohlhase)

---

**Inhalt:**

Dieses Seminar behandelt ausgewählte Themen aus dem Gebiet der Wissensrepräsentation und -Verarbeitung, mit einem Fokus auf mathematisches Wissen.

Die Vortragsthemen sind sehr unterschiedlich im Schwierigkeitsgrad, sie reichen von Einführungsthemen für ambitionierte Bachelor-Studenten bis zur Forschungsfront für Doktoranden. Sie spiegeln recht gut die Forschungsinteressen der KWARC Gruppe wieder. Daher ist dieses Seminar sehr gut geeignet um in die Arbeitsgruppe einzusteigen (z.B. für eine Promotion).

**Lernziele und Kompetenzen:**

Aktuelle Fachliteratur verstehen, Forschungsthemen für ein Fachpublikum verständlich aufbereiten und vortragen, Kommunikation mit Experten.

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

[1] **Informatik (Bachelor of Science): 3-5. Semester**

(Po-Vers. 2009s | Seminare, Praktika, Bachelorarbeit)

---

**Studien-/Prüfungsleistungen:**

Seminar Wissensrepräsentation und -verarbeitung (Prüfungsnummer: 635405)

(englische Bezeichnung: Knowledge Representation and -processing)

Prüfungsleistung, Seminarleistung

Anteil an der Berechnung der Modulnote: 100%

weitere Erläuterungen:

Zum Bestehen dieser Lehrveranstaltung muss ein Vortrag im Seminar gehalten und der Inhalt schriftlich ausgearbeitet werden.

Prüfungssprache: Deutsch oder Englisch

Erstablesung: SS 2019, 1. Wdh.: keine Wdh.

1. Prüfer: Michael Kohlhase (100451)

---

---

**Modulbezeichnung:** Product Management (PROJ 5-ECTS) (OSS-PROD-PROJ) 5 ECTS  
 (Product Management (PROJ 5-ECTS))

Modulverantwortliche/r: Dirk Riehle  
 Lehrende: Dirk Riehle

---

Startsemester: WS 2019/2020	Dauer: 1 Semester	Turnus: jährlich (WS)
Präsenzzeit: k.A. Std.	Eigenstudium: 150 Std.	Sprache: Englisch

---

**Lehrveranstaltungen:**

Product Management (PROJ) (WS 2019/2020, Sonstige Lehrveranstaltung, 2 SWS, Dirk Riehle)

---

**Inhalt:**

This course teaches students the concepts, methods, and tools of software product management. Product management is an important function in software development organizations. A product manager conceives and defines new products. His or her task is to understand the market incl. customers, to develop a product vision from that understanding, to translate it into product requirements, define those requirements on a by-feature basis and work with engineering to ensure these features are properly realized in the product under development.

- Role, tasks, and responsibilities of a product manager
- Process, methods, techniques and tools of product management
- Managing incremental/sustaining as well as disruptive innovation
- Open source product management; new trends in product management

Students can choose one or both of two components:

- VUE (lecture + homework), 2 SWS, 5 ECTS. VUE uses teaching cases as commonly used in MBA programs. The teaching cases are available for free at <http://pmbycase.com>.
- PROJ (small project), 2 SWS, 5 ECTS. In PROJ, students perform a small product management project, either individually or in teams. In such a project, students write a business plan for a market opportunity or a startup. There are two sources of projects: Those supplied by the professorship and those supplied by students. If we supply any projects, we will tell you about them in the first lecture. Student-supplied projects can be suggested by students to Prof. Riehle at least one week before class starts and must meet the requirements described here: <https://goo.gl/sNPi1i>

PROD projects are run as shared projects, in which all participants contribute and get to participate in the project results. Read more at <https://wp.me/pDU66-2p4>.

Class is run as two 90min blocks. The first block discusses the teaching cases. The second block is a coaching session for the projects (10 ECTS only).

The overall schedule can be found at <http://goo.gl/tTAI0> . Please sign up for the course on StudOn (link accessible through schedule spreadsheet) as soon as possible.

**Lernziele und Kompetenzen:**

- Understand the role, function, and responsibilities of a product manager
- Understand key concepts, methods, and tools of software product management
- Understand different business situations, incl. Incremental vs. disruptive innovation

**Literatur:**

- <http://goo.gl/41Dgsr>

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

[1] **Informatik (Bachelor of Science): 3-5. Semester**

(Po-Vers. 2009s | Seminare, Praktika, Bachelorarbeit)

Dieses Modul ist daneben auch in den Studienfächern "International Information Systems (IIS) (Master of Science)", "Maschinenbau (Master of Science)", "Mechatronik (Bachelor of Science)" verwendbar.

---

**Studien-/Prüfungsleistungen:**

Product Management (PROJ 5-ECTS) (Prüfungsnummer: 738728)

(englische Bezeichnung: Product Management (PROJ 5-ECTS))

Prüfungsleistung, Seminarleistung

Anteil an der Berechnung der Modulnote: 100%

weitere Erläuterungen:

- In-class participation
- Homework assignments
- Project work

Erstablingung: WS 2019/2020, 1. Wdh.: keine Angabe

1. Prüfer: Dirk Riehle (100100)

---

---

**Modulbezeichnung:** Ausgewählte Kapitel aus "The Art of Computer Programming" (SEM-TACP) **5 ECTS**  
 (Selected Chapters from "The Art of Computer Programming")

Modulverantwortliche/r: Oliver Keszöcze  
 Lehrende: Oliver Keszöcze

---

Startsemester: SS 2019	Dauer: 1 Semester	Turnus: jährlich (SS)
Präsenzzeit: 30 Std.	Eigenstudium: 120 Std.	Sprache: Deutsch oder Englisch

---

**Lehrveranstaltungen:**

Seminar: Ausgewählte Kapitel aus "The Art of Computer Programming" (SS 2019, Seminar, 2 SWS, Oliver Keszöcze)

---

**Inhalt:**

Donald E. Knuth ist vielen Studierenden der naturwissenschaftlichen und technischen Fächer hauptsächlich durch sein Textsatzsystem LaTeX bekannt. Sein wissenschaftliches Hauptwerk ist jedoch das Buch "The Art of Computer Programming" (TAOCP). Anders als der Titel suggeriert, handelt es sich bei dem Buch um kein Programmierhandbuch, sondern vielmehr um ein allumfassendes Kompendium von Algorithmen und Datenstrukturen. Es wird zu Recht als "die Bibel aller fundamentalen Algorithmen" bezeichnet. Insbesondere interessant ist, dass Knuth durch seine Herangehensweise bzw. Darstellung einer völlig neuen Sichtweisen auf schon bekannte Probleme erlaubt. Dieses Seminar widmet sich ausgesuchten Kapiteln aus TAOCP. Teilnehmer setzen sich mit den Inhalten auseinander und stellen diese in einem Vortrag und einer schriftlichen Ausarbeitung vor. Die Ausarbeitung kann einen Programmieranteil enthalten. Nach Rücksprache können auch eigene Kapitel ausgesucht werden. Auf Wunsch kann der Vortrag und/oder die Ausarbeitung auf Englisch verfasst werden.

**Lernziele und Kompetenzen:**

*Fachkompetenz*

*Verstehen*

Die Studierenden tragen grundlegende Inhalte aus dem Buch "The Art of Computer Programming" vor.

*Anwenden*

Die Studierenden können eigenständig neue Beispiele und Anwendungsszenarien aufstellen.

*Evaluieren (Beurteilen)*

Die Studierenden können die vorgestellten Inhalte in einen wissenschaftlichen Kontext setzen und ihre Güte beurteilen.

*Lern- bzw. Methodenkompetenz*

Die Studierenden arbeiten sich selbstständig in einen Abschnitt aus "The Art of Computer Programming" ein und suchen hierbei selbstständig nach verwandten Arbeiten, um den Kontext zu verstehen und aufzuarbeiten. Die Studierenden lernen, wissenschaftliche Inhalt mündlich sowie schriftlich leicht verständlich darzustellen.

*Selbstkompetenz*

Die Studierenden können eigene Stärken und Schwächen im Bereich ihrer Präsentationstechniken reflektieren und die eigene Entwicklung planen.

*Sozialkompetenz*

Die Studierenden beteiligen sich aktiv an den Vorträgen der jeweils anderen Studierenden durch fachbezogene Fragen zum Thema wie auch Rückmeldung zu Vortragsstil.

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

[1] **Informatik (Bachelor of Science): 3-5. Semester**

(Po-Vers. 2009s | Seminare, Praktika, Bachelorarbeit)

---

**Studien-/Prüfungsleistungen:**

Seminar: Ausgewählte Kapitel aus "The Art of Computer Programming" (Prüfungsnummer: 886484)

(englische Bezeichnung: Seminar: Selected Chapters from "The Art of Computer Programming")

Prüfungsleistung, Seminarleistung, Dauer (in Minuten): 30

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Seminar: Ausgewählte Kapitel aus "The Art of Computer Programming"

weitere Erläuterungen:

Die Bewertung der Prüfungsleistung setzt sich zusammen aus 50% Seminarvortrag (30 min) und 50% schriftlicher Ausarbeitung.

Prüfungssprache: Deutsch oder Englisch

Erstabelgung: SS 2019, 1. Wdh.: keine Angabe

1. Prüfer: Jürgen Teich (100115)

---

**Modulbezeichnung:** Seminar Kommunikationssysteme (B.Sc.) (KS-BaSem) 5 ECTS  
(Seminar Communication Systems (B.Sc.))

Modulverantwortliche/r: Reinhard German  
Lehrende: Marco Pruckner

Startsemester: SS 2019	Dauer: 1 Semester	Turnus: unregelmäßig
Präsenzzeit: 30 Std.	Eigenstudium: 120 Std.	Sprache: Deutsch oder Englisch

**Lehrveranstaltungen:**

Seminar Energieinformatik (SS 2019, Seminar, 2 SWS, Marco Pruckner)

**Inhalt:**

Teilnehmende arbeiten sich selbständig anhand der ausgewählten wissenschaftlichen Literatur in ein vorgeschlagenes Thema aus dem Bereich der Kommunikationssysteme ein, erstellen dazu eine schriftliche Ausarbeitung, bereiten einen Seminarvortrag vor und präsentieren ihn vor Lehrenden, interessierten wissenschaftlichen Mitarbeitern des Instituts und Kommilitonen der Lehrveranstaltung.

**Lernziele und Kompetenzen:**

Zu den zu erwerbenden Kompetenzen zählen

- die Literaturrecherche,
- korrektes Zitieren,
- die Fähigkeit auszuwählen, welche Aspekte in der Ausarbeitung und im Vortrag behandelt werden,
- zielgruppengerechtes Schreiben,
- die Verwendung von Textverarbeitungswerkzeugen für die Präsentation - und die Ausarbeitung,
- sicheres Auftreten beim Vortragen eines wissenschaftlichen Themas sowie
- fachspezifische Fragen zum ausgewählten Themengebiet zu beantworten.

Am Ende jeder Lehreinheit

- stellen die Teilnehmenden gezielt Fragen zum vorgetragenen Thema,
- diskutieren über fachliche Aspekte der Präsentation,
- üben konstruktive Kritik an der Darstellung des Themas und
- bewerten die Präsentation aus vortragstechnischer Sicht.

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

[1] **Informatik (Bachelor of Science): 3-5. Semester**

(Po-Vers. 2009s | Seminare, Praktika, Bachelorarbeit)

**Studien-/Prüfungsleistungen:**

Seminar Kommunikationssysteme (B.Sc.) (Prüfungsnummer: 609624)

(englische Bezeichnung: Seminar Communication Systems (B.Sc.))

Untertitel: Seminar Energieinformatik

(englischer Untertitel Seminar Computer Science for Energy Systems)

Prüfungsleistung, Seminarleistung

Anteil an der Berechnung der Modulnote: 100%

weitere Erläuterungen:

Der Scheinerwerb erfolgt durch Seminarbesuch und Vortrag.

Prüfungssprache: Deutsch oder Englisch

Erstabllegung: SS 2019, 1. Wdh.: keine Angabe

1. Prüfer: Marco Pruckner (100381)

---

**Modulbezeichnung:** **Hardware-based Cryptography (HWCrypt)** **5 ECTS**  
(Hardware-based Cryptography)

Modulverantwortliche/r: Dominique Schröder, Marc Reichenbach  
Lehrende: Dominique Schröder, Marc Reichenbach

---

Startsemester: SS 2019	Dauer: 1 Semester	Turnus: unregelmäßig
Präsenzzeit: 30 Std.	Eigenstudium: 120 Std.	Sprache: Englisch

---

**Lehrveranstaltungen:**

Hardware-based Cryptography (SS 2019, Seminar, Anwesenheitspflicht, Dominique Schröder et al.)

---

**Inhalt:**

The course covers cryptographic schemes as well as their practical / hardware applications.

The topics covered are the following:

- Random Number Generator
- Encryption
- Oblivious Transfer
- Garbled Circuits
- Controlled Functional Encryption
- Hardware Foundations
- Image Processing in Hardware

More advanced topics may be covered if time permits.

**Lernziele und Kompetenzen:**

On successfully passing the course, the students understand how to work with scientific literature on the a.-m. topics. They also understand the practical implementation of cryptographic schemes in hardware.

**Literatur:**

Scientific papers that will be referred to during each lecture

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

[1] **Informatik (Bachelor of Science): 3-5. Semester**

(Po-Vers. 2009s | Seminare, Praktika, Bachelorarbeit)

---

**Studien-/Prüfungsleistungen:**

Hardware-based Cryptography (Prüfungsnummer: 908923)

(englische Bezeichnung: Hardware-based Cryptography)

Untertitel: HWCrypt

(englischer Untertitel HWCrypt)

Prüfungsleistung, Seminarleistung

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Hardware-based Cryptography

weitere Erläuterungen:

Written seminar work to be handed in:

- approx. 15 pages seminar paper
- presentation slides / script

Oral presentation (30mins + 10mins for questions)

Grading: 60% written seminar work + 40% oral presentation

Prüfungssprache: Englisch

Erstablingung: SS 2019, 1. Wdh.: WS 2019/2020 (nur für Wiederholer)

1. Prüfer: Dominique Schröder (100450), 2. Prüfer: Marc Reichenbach (100455)

---

---

**Modulbezeichnung:** Algorithmische Schönheiten - Algorithms Unplugged (AlgosUnplgdd)  
(Algorithms Unplugged) **5 ECTS**

Modulverantwortliche/r: Rolf Wanka  
Lehrende: Rolf Wanka

---

Startsemester: SS 2019	Dauer: 1 Semester	Turnus: jährlich (SS)
Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Deutsch

---

**Lehrveranstaltungen:**

Algorithmische Schönheiten - Algorithms Unplugged (SS 2019, Seminar, 2 SWS, Rolf Wanka)

---

**Inhalt:**

Viele Algorithmen lösen nicht "einfach nur" das Problem, für das sie ausgedacht worden sind, sie sind oft auch ästhetisch sehr ansprechend. Sie benutzen anschauliche Ideen auf überraschend schlaue Art und Weise, oder sie verwenden Ideen aus einem Bereich bewundernswert clever in einem anderen Einsatzbereich wieder.

Ziel dieses Seminars ist es, einige dieser Algorithmen kennenzulernen. Die Teilnehmerinnen und Teilnehmer bekommen Texte, lesen diese und suchen zusätzlich einige der Hintergrundaufsätze und stellen "ihre" Algorithmen in einem Vortrag und einer schriftlichen Ausarbeitung vor.

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

[1] **Informatik (Bachelor of Science): 3-5. Semester**

(Po-Vers. 2009s | Seminare, Praktika, Bachelorarbeit)

---

**Studien-/Prüfungsleistungen:**

Seminar Algorithmische Schönheiten - Algorithms Unplugged (Prüfungsnummer: 871263)

Prüfungsleistung, Seminarleistung, Dauer (in Minuten): 45

Anteil an der Berechnung der Modulnote: 100%

weitere Erläuterungen:

Vortrag: 45 Minuten + Ausarbeitung: 5 Seiten. Abgabe der Ausarbeitung nach den Vorträgen. Genauer Termin wird bekannt gegeben.

Erstabelleung: SS 2019, 1. Wdh.: keine Wdh.

1. Prüfer: Rolf Wanka (100367)

---



**Modulbezeichnung:** Seminar nominale Mengen und Automate (NomAut) 5 ECTS  
(Nominal Sets and Automata (Seminar))

Modulverantwortliche/r: Lutz Schröder  
Lehrende: Lutz Schröder

Startsemester: SS 2019 Dauer: 1 Semester Turnus: unregelmäßig  
Präsenzzeit: 30 Std. Eigenstudium: 120 Std. Sprache: Deutsch oder Englisch

**Lehrveranstaltungen:**

Seminar nominale Mengen und Automaten (SS 2019, Seminar, 2 SWS, Lutz Schröder)

**Inhalt:**

- Mengen, Atome und Symmetrien
- Nominale Mengen
- Automaten über unendlichen Alphabeten und Datenalphabeten
- Registerautomaten
- Nominale Automaten
- Nominale Ausdruckskalküle
- Temporallogik mit Freeze-Quantoren

**Lernziele und Kompetenzen:**

*Fachkompetenz*

*Wissen*

Die Studierenden geben grundlegende Definitionen und Resultate zu nominalen Mengen, Daten-sprachen und nominalen Automaten wieder.

*Verstehen*

Die Studierenden erläutern grundlegende algorithmische Verfahren für Nominale Automaten und analysieren Beispiele für nominale Mengen und Konstruktionen auf nominalen Mengen.

*Anwenden*

Die Studierenden teilen Automatenbegriffe und Ausdruckskalküle hinsichtlich ihrer Komplexität und ihrer Ausdrucksstärke ein.

*Erschaffen*

Die Studierenden bereiten Resultate des Gebiets und ihre Herleitung eigenständig zur Präsentation auf.

*Lern- bzw. Methodenkompetenz*

Die Studierenden erarbeiten sich selbständig den Inhalt wissenschaftlicher Veröffentlichungen und geben diesen in einem strukturierten Vortrag verständlich wieder.

*Sozialkompetenz*

Die Studierenden arbeiten in einer größeren Gruppe an komplexen Problemen und erläutern eigene Einsichten den Mitstudierenden in verständlicher Form.

**Literatur:**

- A. Pitts, Nominal Sets, CUP, 2013
- M. Bojanczyk, B. Klin, S. Lasota:  
Automata theory in nominal sets. Logical Methods in Computer Science 10(3), 2014
- M. Kaminski, N. Francez: Finite-Memory Automata. Theor. Comput. Sci. 134(2), 1994: 329-363
- S. Demri, R. Lazic, and A. Sangnier. Model checking Freeze LTL over one-counter automata. In R. Amadio, ed., Foundations of Software Science and Computational Structures, FOSSACS 2008, vol. 4962 of LNCS, pp. 490 - 504. Springer, 2008.
- L. Schröder, D. Kozen, S. Milius, T. Wißmann:  
Nominal Automata with Name Binding. FoSSaCS 2017: 124-142

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

[1] Informatik (Bachelor of Science): 3-5. Semester

(Po-Vers. 2009s | Seminare, Praktika, Bachelorarbeit)

---

**Studien-/Prüfungsleistungen:**

Seminar nominale Mengen und Automaten (Prüfungsnummer: 685386)

(englische Bezeichnung: Nominal Sets and Automata (Seminar))

Prüfungsleistung, Seminarleistung

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Seminar nominale Mengen und Automaten

weitere Erläuterungen:

Die Prüfungsleistung besteht in einer Ausarbeitung und einem 90-minütigen erfolgreichen Vortrag.

Prüfungssprache: Deutsch oder Englisch

Erstablingung: SS 2019, 1. Wdh.: WS 2019/2020 (nur für Wiederholer)

1. Prüfer: Lutz Schröder (100190)

---

---

**Modulbezeichnung: Seminar Machine Learning and Data Analytics for Industry 4.0 (SemMLDA) 5 ECTS**

(Seminar Machine Learning and Data Analytics for Industry 4.0)

Modulverantwortliche/r: An Nguyen

Lehrende: An Nguyen

---

Startsemester: SS 2019

Dauer: 1 Semester

Turnus: jährlich (SS)

Präsenzzeit: 30 Std.

Eigenstudium: 120 Std.

Sprache: Englisch

---

**Lehrveranstaltungen:**

Maschinelles Lernen und Datenanalytik für Industrie 4.0 (SS 2019, Seminar, 2 SWS, Anwesenheitspflicht, Björn Eskofier et al.)

---

**Inhalt:**

Companies in all kinds of industries are producing and collecting rapidly more and more data from various sources. This is enabled by technologies such as the Internet of Things (IoT), Cyber-physical system (CPS) and cloud computing. Hence there is an increasing demand in industry and research for students and graduates with machine learning and data analytics skills in the Industry 4.0 context. In this Seminar the Industry 4.0 term will include the medical device sector. Aim of this seminar is to give students insights about state-of-the-art machine learning and data analytics methods in the Industry 4.0 and Healthcare context. Topics covered will include but are not limited to:

- Best practices for presentation and scientific work
- Overview of current hot topics in the field of machine learning and data analytics for Industry 4.0 (e.g. deep learning for predictive maintenance and process mining for usage analysis)
- Data acquisition (what kind of data can be acquired?) and storage (how can data be stored efficiently?)
- Machine learning and data analytics methodologies (Support vector machines, Hidden Markov models, Deep learning, Process mining) for industrial data (sensor data, event logs)

The seminar will include talks by corresponding lecturer and invited experts in the domain. Furthermore, students will present results from literature research and/or data analytics projects (provided or open source datasets).

**Lernziele und Kompetenzen:**

- Students will develop an understanding of the current hot field of machine learning and data analytics for Industry 4.0
- Students will learn to research and present a topic within the context of machine learning and data analytics for Industry 4.0 independently in a team
- Students will learn to identify opportunities, challenges and limitations of corresponding ML approaches for Industry 4.0
- Students will develop the skill to identify and understand relevant literature and to present their finding in a structured manner

**Literatur:**

- Lei, Yaguo, Naipeng Li, Liang Guo, Ningbo Li, Tao Yan, and Jing Lin. "Machinery Health Prognostics: A Systematic Review from Data Acquisition to RUL Prediction." *Mechanical Systems and Signal Processing* 104 (May 2018): 799 - 834. <https://doi.org/10.1016/j.ymsp.2017.11.016>.
- Rojas, Eric, Jorge Munoz-Gama, Marcos Sepúlveda, and Daniel Capurro. "Process Mining in Healthcare: A Literature Review." *Journal of Biomedical Informatics* 61 (June 1, 2016): 224 - 36. <https://doi.org/10.1016/j.jbi.2016.04.007>.
- Wil M. P. van der Aalst. „Process Mining: Data Science in Action" 2nd edition, Springer 2016. ISBN 978-3-662-49851-4
- Wang, Lihui, and Xi Vincent Wang. *Cloud-Based Cyber-Physical Systems in Manufacturing*. Cham: Springer International Publishing, 2018. <https://doi.org/10.1007/978-3-319-67693-7>.

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

[1] **Informatik (Bachelor of Science): 3-5. Semester**

(Po-Vers. 2009s | Seminare, Praktika, Bachelorarbeit)

---

**Studien-/Prüfungsleistungen:**

Seminar Machine Learning and Data Analytics for Industry 4.0 (Prüfungsnummer: 903776)

(englische Bezeichnung: Seminar Machine Learning and Data Analytics for Industry 4.0)

Prüfungsleistung, Seminarleistung

Anteil an der Berechnung der Modulnote: 100%

weitere Erläuterungen:

Final presentation with demo and paper according to IEEE standards (4 pages excluding references)

Erstablingung: SS 2019, 1. Wdh.: WS 2019/2020

1. Prüfer: Björn Eskofier (100122)

---

**Organisatorisches:**

Application via studOn

**Modulbezeichnung:** Seminar Automatische Analyse von Stimm-, Sprech- und Sprachstörungen bei Sprachpathologien (SemSprachPath) 5 ECTS  
(Seminar Automatic Analysis of Voice, Speech, and Language for Speech Pathologies)

Modulverantwortliche/r: Elmar Nöth  
Lehrende: Tino Haderlein, Elmar Nöth

Startsemester: SS 2019                      Dauer: 1 Semester                      Turnus: unregelmäßig  
Präsenzzeit: 60 Std.                      Eigenstudium: 90 Std.                      Sprache: Deutsch und Englisch

**Lehrveranstaltungen:**

Seminar Automatische Analyse von Stimm-, Sprech- und Sprachstörungen bei Sprachpathologien (SS 2019, Seminar, 4 SWS, Tino Haderlein et al.)

**Inhalt:**

Dieses Seminar befasst sich damit, wie Diagnose und Therapie von unterschiedlichen Sprachpathologien durch Sprachtechnologie unterstützt werden können. Die Teilnehmer sollen in einem Vortrag ausgewählte Sprach-, Sprech- und Stimmstörungen vorstellen und entsprechende Technologien aus dem Bereich der Mustererkennung und Sprachverarbeitung aufzeigen.

**Lernziele und Kompetenzen:**

- Die Studierenden
- erlernen die Literaturrecherche.
  - arbeiten sich selbstständig anhand der gefundenen Literatur in die Thematik der automatischen Analyse von Sprach-, Sprech- und Stimmstörungen ein.
  - wählen einen Schwerpunkt und bereiten diesen im Rahmen einer Präsentation so auf, dass er für andere Teilnehmer des Seminars verständlich ist.
  - lernen die Anforderungen an einen wissenschaftlichen Vortrag auf einer internationalen Konferenz kennen.
  - halten einen Vortrag in der international üblichen Fachsprache Englisch (davon ausgenommen sind Studierende aus dem Ausland, die in Deutschland studieren, um Deutsch zu lernen)

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

- [1] **Informatik (Bachelor of Science): 3-5. Semester**  
(Po-Vers. 2009s | Seminare, Praktika, Bachelorarbeit)

**Studien-/Prüfungsleistungen:**

Seminar Sprachtechnologie für Sprachpathologien (Prüfungsnummer: 349413)

(englische Bezeichnung: Speech Technologies for Speech Pathologies)

Prüfungsleistung, Seminarleistung

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Seminar Automatische Analyse von Stimm-, Sprech- und Sprachstörungen bei Sprachpathologien  
weitere Erläuterungen:

Die Gesamtnote setzt sich zu 50% aus der Bewertung des Vortrags und zu 50% aus der Bewertung der Ausarbeitung zusammen. Ziel des Seminars ist die verständliche Aufbereitung eines Themas für andere Studierende im Rahmen einer Lehrstunde. Die Vortragsdauer beträgt 30 Minuten. Ziel ist es, diese möglichst genau einzuhalten. Die Ausarbeitung umfasst 6-8 Seiten im Stil von IEEE-Konferenzbeiträgen. Vortrag und Ausarbeitung sollten auf Englisch erfolgen.

Erstabelleung: SS 2019, 1. Wdh.: keine Wdh.

1. Prüfer: Elmar Nöth (100183)

---

**Modulbezeichnung:** **Advanced Design and Programming (5-ECTS) (OSS-ADAP) 5 ECTS**  
 (Advanced Design and Programming (5-ECTS))

Modulverantwortliche/r: Dirk Riehle

Lehrende: Dirk Riehle

---

Startsemester: WS 2019/2020

Dauer: 1 Semester

Turnus: jährlich (WS)

Präsenzzeit: 60 Std.

Eigenstudium: 90 Std.

Sprache: Deutsch

---

**Lehrveranstaltungen:**

Advanced Design and Programming (VUE) (WS 2019/2020, Vorlesung mit Übung, 4 SWS, Dirk Riehle)

---

**Vorhergehende Module:**

Algorithmen und Datenstrukturen

---

**Inhalt:**

This course teaches principles and practices of advanced object-oriented design and programming.

Dieser Kurs wird auf Deutsch gehalten.

It consists of a weekly lecture with exercises, homework and self-study, totaling 4 SWS, 5 ECTS. This is a hands-on course and students should be familiar with their Java IDE. Students learn the following concepts:

*Class-Level*

- Method design
- Class design
- Classes and interfaces
- Subtyping and inheritance
- Implementing inheritance
- Design by contract

*Collaboration-Level*

- Values vs. objects
- Role objects
- Type objects
- Object creation
- Collaboration-based design
- Design patterns

*Component-Level*

- Error handling
- Meta-object protocols
- Frameworks and components
- Domain-driven design
- API evolution

The running example is the photo sharing and rating software Wahlzeit, see <https://github.com/dirkriehle/wahlzeit>.

Class is held as a three hour session with a short break in between. The class iterates over short lectures, discussion, and exercise chunks of 10-30min each. Students should bring a laptop with a well-working Java programming setup.

The overall schedule can be found at <http://goo.gl/bePPn>. Please sign up for the course on StudOn (link accessible through schedule spreadsheet) as soon as possible.

**Lernziele und Kompetenzen:**

- Learn to recognize, analyse, and apply advanced concepts of object-oriented design and programming
- Learn to work effectively with a realistic tool set-up, involving an IDE, configuration management, and a service hoster

**Literatur:**

- See <http://goo.gl/BZpU8>

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

**[1] Informatik (Bachelor of Science): 3-5. Semester**

(Po-Vers. 2009s | Seminare, Praktika, Bachelorarbeit)

Dieses Modul ist daneben auch in den Studienfächern "International Information Systems (IIS) (Master of Science)", "Maschinenbau (Master of Science)" verwendbar.

---

**Studien-/Prüfungsleistungen:**

Advanced Design and Programming (5-ECTS) (Prüfungsnummer: 280491)

(englische Bezeichnung: Advanced Design and Programming (5-ECTS))

Prüfungsleistung, Seminarleistung

Anteil an der Berechnung der Modulnote: 100%

weitere Erläuterungen:

- Exercise participation
- Homework assignments

Erstablingung: WS 2019/2020, 1. Wdh.: keine Angabe

1. Prüfer: Dirk Riehle (100100)

---

---

**Modulbezeichnung:** **Neuartige Rechnerarchitekturen (NeuRa)** **5 ECTS**  
(Novel Computer Architectures)

Modulverantwortliche/r: Marc Reichenbach, Dietmar Fey

Lehrende: Marc Reichenbach, Philipp Holzinger, Dietmar Fey

---

Startsemester: SS 2019

Dauer: 1 Semester

Turnus: jährlich (SS)

Präsenzzeit: 60 Std.

Eigenstudium: 90 Std.

Sprache: Deutsch

---

**Lehrveranstaltungen:**

Neuartige Rechnerarchitekturen (SS 2019, Hauptseminar, 4 SWS, Anwesenheitspflicht, Marc Reichenbach et al.)

---

**Inhalt:**

Das Aufkommen der Multi-Kern-Prozessoren in den letzten Jahren hat das parallele Rechnen aus der rein akademischen Nische geholt und es zum Allgemeingut für den Informatiker gemacht. Man spricht in diesem Zusammenhang auch gelegentlich von der Multi-Kern-Revolution.

Am Horizont tauchen bereits die ersten so genannten Viel-Kern-Prozessoren ("many-core processors") mit mehr als 100 Kernen auf. Man unterscheidet je nach Aufbau der Kerne bzgl. homogener und inhomogener Architekturen. Im Seminar sollen anhand ausgewählter Artikel der Stand der Technik und die kommenden Technologien erarbeitet werden.

Einzelne der oben genannten Architekturen können bei Interesse auch erprobt werden. Es sind somit sowohl praktische als auch theoretische Themenstellungen möglich.

**Lernziele und Kompetenzen:**

*Fachkompetenz*

*Wissen*

Lernende können Wissen über die Grundprinzipien modernen Multi-Core Rechnerarchitekturen (Intel, ARM CPUs; AMD, Nvidia GPUs; sonstige Beschleuniger) wiedergeben.

*Verstehen*

Lernende verstehen die unterschiedlichen Ansätze zur Parallelismus der vorgestellten Architekturen.

*Anwenden*

Lernende erklären wie die Parallelisierungstechniken in bestehenden Architekturen eingesetzt werden.

*Evaluieren (Beurteilen)*

Lernende evaluieren die Eignung von Architekturen, bestimmte Probleme effizient auf diese Abbilden zu können.

*Sozialkompetenz*

Lernende können komplexe fachbezogene Inhalte klar und zielgruppenrecht präsentieren und eigene Standpunkte in einer Fachdiskussion argumentativ vertreten.

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

[1] **Informatik (Bachelor of Science): 3-5. Semester**

(Po-Vers. 2009s | Seminare, Praktika, Bachelorarbeit)

---

**Studien-/Prüfungsleistungen:**

Neuartige Rechnerarchitekturen (Prüfungsnummer: 941318)

(englische Bezeichnung: Novel Computer Architectures)

Prüfungsleistung, Seminarleistung

Anteil an der Berechnung der Modulnote: 100%

weitere Erläuterungen:

Vortrag und Abgabe einer Ausarbeitung:

- Theoretisch orientierte Seminarleistung: Vortrag (30-45 Min.) und Ausarbeitung (10-15 Seiten); Notenbildung 50 % Vortrag, 50 % Ausarbeitung



- Praktisch orientierte Seminarleistung: Gruppenvortrag (2-3 Studenten, je 15-20 Min.) und programmtechnische Umsetzung mit schriftlicher Dokumentation; Notenbildung 30 % Vortrag, 70 % Ausarbeitung

Prüfungssprache: Deutsch oder Englisch

Erstablingung: SS 2019, 1. Wdh.: WS 2019/2020

1. Prüfer: Marc Reichenbach (100455), 2. Prüfer: Dietmar Fey (100101)

---

---

**Modulbezeichnung:** **Advanced Competitive Programming (ACP)** **5 ECTS**  
 (Advanced Competitive Programming)

Modulverantwortliche/r: Paul Wild  
 Lehrende: Paul Wild

---

Startsemester: WS 2019/2020	Dauer: 1 Semester	Turnus: unregelmäßig
Präsenzzeit: 30 Std.	Eigenstudium: 120 Std.	Sprache: Deutsch

---

**Lehrveranstaltungen:**

Advanced Competitive Programming (WS 2019/2020, Seminar, 2 SWS, Paul Wild)

---

**Inhalt:**

Es werden verschiedene Algorithmen und algorithmische Methoden vorgestellt, wie sie im Kontext von Programmierwettbewerben zur Anwendung kommen.

- Algorithmen zur schnellen Polynommultiplikation
- Siebmethoden zur Berechnung zahlentheoretischer Funktionen
- Datenstrukturen für intervallbasierte Anfragen auf Arrays und Bäumen
- Methoden zur Optimierung Dynamischer Programmierung
- Sweep-Verfahren aus der Algorithmischen Geometrie
- Suffixarrays und Anwendungen

Bei etwa der Hälfte der Termine finden jeweils Vorträge statt, in denen die verschiedenen Konzepte vorgestellt werden. Bei den verbleibenden Terminen werden diese von den Studierenden umgesetzt und in Übungsaufgaben zur Anwendung gebracht. Anschließend demonstrieren und erklären die Studierenden die Arbeitsweise ihrer Implementierung.

**Lernziele und Kompetenzen:**

Die Studierenden:

- sind vertraut mit diversen Algorithmen aus der Wettbewerbsprogrammierung sowie deren Funktionsweise und Laufzeitkomplexität,
- können algorithmische Probleme analysieren und die gelernten Methoden zur Lösung dieser Probleme einsetzen,
- sind in der Lage die gelernten Algorithmen eigenständig und sauber zu implementieren,
- können die Algorithmen sowie deren Implementierung in klarer und verständlicher Weise präsentieren.

**Literatur:**

- A. Laaksonen: Guide to Competitive Programming, Springer, 2017.
- F. Halim und S. Halim: Competitive Programming 3, the new lower bound of programming contests, Lulu.com, 2013.
- T. Cormen et al.: Introduction to Algorithms, MIT Press, 2001.
- J. Erickson: Algorithms, self published, 2019.

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

[1] **Informatik (Bachelor of Science): 3-5. Semester**

(Po-Vers. 2009s | Seminare, Praktika, Bachelorarbeit)

---

**Studien-/Prüfungsleistungen:**

Advanced Competitive Programming (Prüfungsnummer: 791656)

(englische Bezeichnung: Advanced Competitive Programming)

Prüfungsleistung, Seminarleistung, Dauer (in Minuten): 15

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Advanced Competitive Programming

weitere Erläuterungen:

Voraussetzung ist die erfolgreiche Bearbeitung der Programmieraufgaben.

Prüfungssprache: Deutsch

Erstablingung: WS 2019/2020, 1. Wdh.: SS 2020 (nur für Wiederholer)  
1. Prüfer: Lutz Schröder (100190), 2. Prüfer: Stefan Milius (100294)

---

---

**Modulbezeichnung:** IT-Sicherheits-Seminar (Bachelor) (ITSecSem) 5 ECTS  
 (IT Security Seminar (Bachelor))

Modulverantwortliche/r: Felix Freiling  
 Lehrende: Felix Freiling

---

Startsemester: WS 2019/2020	Dauer: 1 Semester	Turnus: halbjährlich (WS+SS)
Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Deutsch

---

**Lehrveranstaltungen:**

IT-Sicherheits-Konferenzseminar (Bachelor) (WS 2019/2020, Seminar, Felix Freiling et al.)

---

**Inhalt:**

Wechselnde Themen aus dem Bereich IT-Sicherheit.

**Lernziele und Kompetenzen:**

Teilnehmende arbeiten sich selbständig anhand wissenschaftlicher Literatur in ein vorgeschlagenes oder nach Absprache frei gewähltes Thema aus dem Bereich der IT-Sicherheit ein, erstellen dazu eine schriftliche Ausarbeitung, bereiten einen Seminarvortrag vor und halten ihn.

Zu den hier zu erwerbenden Kompetenzen zählen die Literaturrecherche, korrektes Zitieren, die Fähigkeit auszuwählen, welche Aspekte in der Ausarbeitung und im Vortrag behandelt werden, zielgruppen-gerechtes Schreiben sowie der Umgang mit dem Textsatzsystem LaTeX.

In jedem Wintersemester findet das Seminar als Konferenzseminar statt. Hier üben die Studierenden dann auch das Prozedere ein, das beim Einreichen einer wissenschaftlichen Arbeit bei einer Konferenz üblich ist: Unter anderem lernen sie, die Arbeiten anderer Personen im Review-Prozess zu beurteilen und Kritik und Verbesserungsempfehlungen auszusprechen sowie für die eigene Arbeit anzunehmen und umzusetzen.

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

[1] **Informatik (Bachelor of Science): 3-5. Semester**

(Po-Vers. 2009s | Seminare, Praktika, Bachelorarbeit)

---

**Studien-/Prüfungsleistungen:**

IT-Sicherheits-Seminar (Prüfungsnummer: 396551)

Prüfungsleistung, Seminarleistung

Anteil an der Berechnung der Modulnote: 100%

weitere Erläuterungen:

Die Bewertung der Prüfungsleistung setzt sich zusammen aus 40% Seminarausarbeitung, 40% Seminarvortrag und 20% Mitarbeit beim Review-Prozess (im Falle eines Konferenzseminars); anderenfalls 50% Seminarausarbeitung und 50% Seminarvortrag (im Falle eines normalen Seminars).

Erstablingung: WS 2019/2020, 1. Wdh.: keine Angabe

1. Prüfer: Felix Freiling (100128)

---

---

**Modulbezeichnung:** Machine Learning (Inf2-SEM-ML) 5 ECTS  
(Machine Learning)

Modulverantwortliche/r: Michael Philippsen  
Lehrende: Christopher Mutschler

---

Startsemester: WS 2019/2020	Dauer: 1 Semester	Turnus: jährlich (WS)
Präsenzzeit: 30 Std.	Eigenstudium: 120 Std.	Sprache: Deutsch oder Englisch

---

**Lehrveranstaltungen:**

Machine Learning (WS 2019/2020, Seminar, 2 SWS, Tobias Feigl et al.)

---

**Inhalt:**

Dieses Seminar führt in das Themengebiet des maschinellen Lernens ein. Maschinelles Lernen befasst sich mit der Frage, wie Computerprogramme zu konstruieren sind, die automatisch ihr Wissen vergrößern. Ziel des Seminars ist, die wichtigsten Schlüsselalgorithmen aus diesem Gebiet zusammen mit erläuternden Beispielen ihrer Arbeitsweise und der Theorie vorzustellen, die den Kern des Machine Learning ausmachen. Klassische Themengebiete sind hierbei beispielsweise Lernen mit Belohnung, evolutionäre Algorithmen oder statistische Verfahren. Mit der Zeit haben sich nach und nach etablierte Verfahren wie Support Vector Machines, Hidden Markov Modelle oder künstliche Neuronale Netze entwickelt. Das Seminar soll einen übergreifenden Einblick in die Welt des maschinellen Lernen und deren Algorithmen vermitteln.

**Lernziele und Kompetenzen:**

Nach der Teilnahme an diesem Seminar sind die Studierenden in der Lage:

- prinzipielle Vorgehensweisen beim Machine Learning zu erläutern,
- Vor- und Nachteile einzelner Methoden zu untersuchen,
- Chancen und Grenzen des maschinellen Lernens zu erläutern,
- erforderliche Literatur aufzufinden, zu analysieren und zu bewerten,
- sich eigenständig in ein Themengebiet einzuarbeiten,
- Grundzüge der Präsentationstechniken anzuwenden und zu motivieren,
- eine Präsentation mit Begleitmaterial für ein Fachpublikum zu entwickeln,
- einen Vortrag passend für einen vorgegebenen Zeitrahmen durchzuführen,
- Sachverhalte unter Fachleuten zu diskutieren,
- fachspezifische Fragen für das Gebiet zu beantworten,

**Literatur:**

- T. M. Mitchell: Machine Learning, McGraw-Hill, 1997.
  - J. R. Quinlan, C4.5: Programs for Machine Learning, Morgan Kaufmann, 1993.
  - F. V. Jensen, An introduction to Bayesian Networks, UCL Press, 1996.
  - N. Lavrac und S. Dzeroski, Inductive Logic Programming, Techniques and Applications, Ellis Horwood, 1994.
  - J. A. Freeman, Simulating Neural Networks with Mathematica, Addison-Wesley, 1994.
  - J. Hertz, A. Krogh und R. G. Palmer, Introduction to the Theory of Neural Computation, Addison-Wesley, 1991.
  - R. Rojas, Theorie der neuronalen Netze, Springer, 1996.
  - W. Banzhaf, P. Nordin, R. E. Keller und D. Francone, Genetic Programming: An Introduction, Morgan Kaufmann und dpunkt, 1998.1994.
  - M. Mitchel, An Introduction to Genetic Algorithms, MIT-Press, 1996.
  - Michalewicz, Genetic Algorithms + Data Structures = Evolution Programs, Springer, 1992.
  - C. Bishop, Pattern Recognition and Machine Learning, Springer, 2007
- 

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

[1] **Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Seminare, Praktika, Bachelorarbeit | Seminar)

## [2] Informatik (Bachelor of Science)

(Po-Vers. 2009s | Hauptseminar)

Dieses Modul ist daneben auch in den Studienfächern "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Bachelor of Science)", "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Master of Science)", "Informatik (Master of Science)", "Medizintechnik (Master of Science)" verwendbar.

---

### Studien-/Prüfungsleistungen:

Machine Learning [5 ECTS] (Prüfungsnummer: 358246)

(englische Bezeichnung: Machine Learning [5 ECTS])

Prüfungsleistung, Seminarleistung

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Machine Learning

weitere Erläuterungen:

Die Bewertung der Prüfungsleistung setzt sich zusammen aus:

- 1/2 Seminarvortrag
- 1/2 Ausarbeitung

Erstablingung: WS 2019/2020, 1. Wdh.: SS 2020

1. Prüfer: Michael Philippsen (100113)

---

### Organisatorisches:

- Anmeldung per E-Mail an christopher.mutschler@fau.de
- 45-60 Minuten Vortrag
- Erstellung einer Ausarbeitung mit den wesentlichen Punkten des Vortrags (keine Folienkopien, ca. 6-8 Seiten)
- Anwesenheit bei den Vorträgen der anderen Teilnehmer
- Fertigstellung der Folien bis spätestens einer Woche vor dem Vortragstermin, Fertigstellung der Ausarbeitung bis zum Ende des Semesters
- Die Veranstaltung wird im Block an ca. 2 Samstagen gegen Ende der Vorlesungszeit stattfinden.

**Modulbezeichnung:** Seminar Kommunikationssysteme (B.Sc.) (KS-BaSem) 5 ECTS  
(Seminar Communication Systems (B.Sc.))

Modulverantwortliche/r: Reinhard German  
Lehrende: Reinhard German

Startsemester: WS 2019/2020      Dauer: 1 Semester      Turnus: unregelmäßig  
Präsenzzeit: 30 Std.      Eigenstudium: 120 Std.      Sprache: Deutsch oder Englisch

**Lehrveranstaltungen:**

Seminar: Kommunikationssysteme (WS 2019/2020, Seminar, 2 SWS, Reinhard German)

**Inhalt:**

Teilnehmende arbeiten sich selbständig anhand der ausgewählten wissenschaftlichen Literatur in ein vorgeschlagenes Thema aus dem Bereich der Kommunikationssysteme ein, erstellen dazu eine schriftliche Ausarbeitung, bereiten einen Seminarvortrag vor und präsentieren ihn vor Lehrenden, interessierten wissenschaftlichen Mitarbeitern des Instituts und Kommilitonen der Lehrveranstaltung.

**Lernziele und Kompetenzen:**

Zu den zu erwerbenden Kompetenzen zählen

- die Literaturrecherche,
- korrektes Zitieren,
- die Fähigkeit auszuwählen, welche Aspekte in der Ausarbeitung und im Vortrag behandelt werden,
- zielgruppengerechtes Schreiben,
- die Verwendung von Textverarbeitungswerkzeugen für die Präsentation - und die Ausarbeitung,
- sicheres Auftreten beim Vortragen eines wissenschaftlichen Themas sowie
- fachspezifische Fragen zum ausgewählten Themengebiet zu beantworten.

Am Ende jeder Lehreinheit

- stellen die Teilnehmenden gezielt Fragen zum vorgetragenen Thema,
- diskutieren über fachliche Aspekte der Präsentation,
- üben konstruktive Kritik an der Darstellung des Themas und
- bewerten die Präsentation aus vortragstechnischer Sicht.

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

[1] **Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Seminare, Praktika, Bachelorarbeit | Seminar)

[2] **Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Hauptseminar)

**Studien-/Prüfungsleistungen:**

Seminar Kommunikationssysteme (B.Sc.) (Prüfungsnummer: 609624)

(englische Bezeichnung: Seminar Communication Systems (B.Sc.))

Untertitel: Seminar Machine Learning for Computer Networking

(englischer Untertitel Seminar Machine Learning for Computer Networking)

Prüfungsleistung, Seminarleistung

Anteil an der Berechnung der Modulnote: 100%

weitere Erläuterungen:

Der Scheinerwerb erfolgt durch Seminarbesuch und Vortrag.

Prüfungssprache: Deutsch oder Englisch

Erstablingung: WS 2019/2020, 1. Wdh.: keine Angabe

1. Prüfer: Reinhard German (100165)

---

**Modulbezeichnung:** Seminar Machine Learning and Data Analytics for Industry 4.0 (MADI40) **5 ECTS**

(Seminar Machine Learning and Data Analytics for Industry 4.0)

Modulverantwortliche/r: An Nguyen

Lehrende: Björn Eskofier, An Nguyen, Franz Köferl, Philipp Schlieper

---

Startsemester: WS 2019/2020

Dauer: 1 Semester

Turnus: halbjährlich (WS+SS)

Präsenzzeit: 30 Std.

Eigenstudium: 120 Std.

Sprache: Englisch

---

**Lehrveranstaltungen:**

Maschinelles Lernen und Datenanalytik für Industrie 4.0 (WS 2019/2020, Seminar, 2 SWS, Björn Eskofier et al.)

---

**Empfohlene Voraussetzungen:**

Requirements:

- Prior knowledge of machine learning via courses like PA, IntroPR, PR, DL, MLTS, CVP or equivalent (ideally first project experiences) is expected!
- Motivation to explore scientific findings (e.g. via literature research)
- Motivation to code and analyze data

**Vorhergehende Module:**

Pattern Analysis

Deep Learning

Introduction to Pattern Recognition

Introduction to Pattern Recognition Deluxe

Pattern Recognition

Pattern Recognition Deluxe

Maschinelles Lernen für Zeitreihen

Maschinelles Lernen für Zeitreihen Deluxe

---

**Inhalt:**

Companies in all kinds of industries are producing and collecting rapidly more and more data from various sources. This is enabled by technologies such as the Internet of Things (IoT), Cyber-physical system (CPS) and cloud computing. Hence there is an increasing demand in industry and research for students and graduates with machine learning and data analytics skills in the Industry 4.0 context. In this Seminar the Industry 4.0 term will include the medical device sector. Aim of this seminar is to give students insights about state-of-the-art machine learning and data analytics methods and applications in the Industry 4.0 and Healthcare context. Students will mainly work independently on specific topics including implementation and analytical components. Several potential topics will be provided but students are also encouraged to propose their own topics (please discuss with teaching staff beforehand). Topics covered in the seminar will include but are not limited to:

- Best practices for presentation and scientific work
- Data science workflow and tools
- Overview of current hot topics in the field of machine learning and data analytics for Industry 4.0 (e.g. deep learning for predictive maintenance and process mining for usage analysis)
- Machine learning and data analytics methodologies (Support vector machines, Hidden Markov models, Deep learning, Process mining, ect.) for industrial data (sensor data, event logs, ...)
- Object detection in industry application

The seminar will include talks by corresponding lecturer and invited experts in the domain. Furthermore, students will present results from literature research and data analytics projects (provided or open source datasets).

**Lernziele und Kompetenzen:**

- Students will develop an understanding of the current hot field of machine learning and data analytics for Industry 4.0/healthcare



- Students will learn to research and present a topic within the context of machine learning and data analytics for Industry 4.0/healthcare independently
- Students will learn to identify opportunities, challenges and limitations of corresponding ML approaches for Industry 4.0/healthcare
- Students will develop the skill to identify and understand relevant literature and to present their finding in a structured manner
- Students will learn to present implementation and validation results in form of a demonstration and/or report

Literature (selection)

**Literatur:**

- Lei, Yaguo, Naipeng Li, Liang Guo, Ningbo Li, Tao Yan, and Jing Lin. "Machinery Health Prognostics: A Systematic Review from Data Acquisition to RUL Prediction." *Mechanical Systems and Signal Processing* 104 (May 2018): 799 - 834. <https://doi.org/10.1016/j.ymsp.2017.11.016>.
- Rojas, Eric, Jorge Munoz-Gama, Marcos Sepúlveda, and Daniel Capurro. "Process Mining in Healthcare: A Literature Review." *Journal of Biomedical Informatics* 61 (June 1, 2016): 224 - 36. <https://doi.org/10.1016/j.jbi.2016.04.007>.
- Wil M. P. van der Aalst. „Process Mining: Data Science in Action" 2nd edition, Springer 2016. ISBN 978-3-662-49851-4
- Wang, Lihui, and Xi Vincent Wang. *Cloud-Based Cyber-Physical Systems in Manufacturing*. Cham: Springer International Publishing, 2018. <https://doi.org/10.1007/978-3-319-67693-7>.

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

[1] **Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Seminare, Praktika, Bachelorarbeit | Seminar)

[2] **Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Hauptseminar)

Dieses Modul ist daneben auch in den Studienfächern "Informatik (Master of Science)" verwendbar.

**Studien-/Prüfungsleistungen:**

Seminar Machine Learning and Data Analytics for Industry 4.0 (Prüfungsnummer: 903776)

(englische Bezeichnung: Seminar Machine Learning and Data Analytics for Industry 4.0)

Prüfungsleistung, Seminarleistung

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Maschinelles Lernen und Datenanalytik für Industrie 4.0

weitere Erläuterungen:

Final presentation with demo and paper according to IEEE standards (4 pages excluding references)

Erstablingung: WS 2019/2020, 1. Wdh.: SS 2020

1. Prüfer: Björn Eskofier (100122)

**Organisatorisches:**

Application via studOn

**Modulbezeichnung:** Seminar Automatische Analyse von Stimm-, Sprech- und Sprachstörungen bei Sprachpathologien (SemSprachPath) 5 ECTS  
(Seminar Automatic Analysis of Voice, Speech, and Language for Speech Pathologies)

Modulverantwortliche/r: Elmar Nöth  
Lehrende: Elmar Nöth, Tino Haderlein

Startsemester: WS 2019/2020      Dauer: 1 Semester      Turnus: unregelmäßig  
Präsenzzeit: 30 Std.      Eigenstudium: 120 Std.      Sprache: Deutsch und Englisch

**Lehrveranstaltungen:**

Seminar Automatische Analyse von Stimm-, Sprech- und Sprachstörungen bei Sprachpathologien (WS 2019/2020, Seminar, 4 SWS, Elmar Nöth et al.)

**Inhalt:**

Dieses Seminar befasst sich damit, wie Diagnose und Therapie von unterschiedlichen Sprachpathologien durch Sprachtechnologie unterstützt werden können. Die Teilnehmer sollen in einem Vortrag ausgewählte Sprach-, Sprech- und Stimmstörungen vorstellen und entsprechende Technologien aus dem Bereich der Mustererkennung und Sprachverarbeitung aufzeigen.

**Lernziele und Kompetenzen:**

Die Studierenden

- erlernen die Literaturrecherche.
- arbeiten sich selbstständig anhand der gefundenen Literatur in die Thematik der automatischen Analyse von Sprach-, Sprech- und Stimmstörungen ein.
- wählen einen Schwerpunkt und bereiten diesen im Rahmen einer Präsentation so auf, dass er für andere Teilnehmer des Seminars verständlich ist.
- lernen die Anforderungen an einen wissenschaftlichen Vortrag auf einer internationalen Konferenz kennen.
- halten einen Vortrag in der international üblichen Fachsprache Englisch (davon ausgenommen sind Studierende aus dem Ausland, die in Deutschland studieren, um Deutsch zu lernen)

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

[1] **Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Seminare, Praktika, Bachelorarbeit | Seminar)

[2] **Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Hauptseminar)

Dieses Modul ist daneben auch in den Studienfächern "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Bachelor of Science)" verwendbar.

**Studien-/Prüfungsleistungen:**

Seminar Sprachtechnologie für Sprachpathologien (Prüfungsnummer: 349413)

(englische Bezeichnung: Speech Technologies for Speech Pathologies)

Prüfungsleistung, Seminarleistung

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Seminar Automatische Analyse von Stimm-, Sprech- und Sprachstörungen bei Sprachpathologien  
weitere Erläuterungen:

Die Gesamtnote setzt sich zu 50% aus der Bewertung des Vortrags und zu 50% aus der Bewertung der Ausarbeitung zusammen. Ziel des Seminars ist die verständliche Aufbereitung eines Themas für andere Studierende im Rahmen einer Lehrstunde. Die Vortragsdauer beträgt 30 Minuten. Ziel ist es, diese möglichst genau einzuhalten. Die Ausarbeitung umfasst 6-8 Seiten im Stil von IEEE-Konferenzbeiträgen. Vortrag und Ausarbeitung sollten auf Englisch erfolgen.

Erstablingung: WS 2019/2020, 1. Wdh.: keine Wdh.  
1. Prüfer: Elmar Nöth (100183)

---

---

**Modulbezeichnung:** Seminar Theoretische Informatik (ThInfSem) 5 ECTS  
(Theoretical Computer Science (Seminar))

Modulverantwortliche/r: Lutz Schröder  
Lehrende: Lutz Schröder

---

Startsemester: WS 2019/2020	Dauer: 1 Semester	Turnus: halbjährlich (WS+SS)
Präsenzzeit: 30 Std.	Eigenstudium: 120 Std.	Sprache: Englisch

---

**Lehrveranstaltungen:**

Seminar Theoretische Informatik (WS 2019/2020, Oberseminar, 2 SWS, Lutz Schröder)

---

**Inhalt:**

- Theory of concurrency
- Programming semantics
- Categories in computer science
- Logic in computer science
- Theory of artificial intelligence

**Lernziele und Kompetenzen:**

*Fachkompetenz*

*Wissen*

Die Studierenden geben grundlegende Definitionen und Resultate aus einem vertieften Bereich der Theoretischen Informatik wieder.

*Verstehen*

Die Studierenden erläutern grundlegende algorithmische und deduktive Verfahren.

*Anwenden*

Die Studierenden erläutern Anwendungen theoretischer Methoden in konkreten informatischen Kontexten.

*Erschaffen*

Die Studierenden bereiten Resultate des Gebiets und ihre Herleitung eigenständig zur Präsentation auf.

*Lern- bzw. Methodenkompetenz*

Die Studierenden erarbeiten sich selbständig den Inhalt wissenschaftlicher Veröffentlichungen und geben diesen in einem strukturierten Vortrag verständlich wieder.

*Sozialkompetenz*

Die Studierenden arbeiten in einer größeren Gruppe an komplexen Problemen und erläutern eigene Einsichten den Mitstudierenden in verständlicher Form.

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

[1] **Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Seminare, Praktika, Bachelorarbeit | Seminar)

[2] **Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Hauptseminar)

Dieses Modul ist daneben auch in den Studienfächern "Informatik (Master of Science)" verwendbar.

---

**Studien-/Prüfungsleistungen:**

Seminar Theoretische Informatik (Prüfungsnummer: 863761)

(englische Bezeichnung: Theoretical Computer Science (Seminar))

Prüfungsleistung, Seminarleistung

Anteil an der Berechnung der Modulnote: 100%

weitere Erläuterungen:

Die Prüfungsleistung besteht in einer Ausarbeitung und einem 90-minütigen erfolgreichen Vortrag.

Prüfungssprache: Englisch

Erstablingung: WS 2019/2020, 1. Wdh.: SS 2020 (nur für Wiederholer)  
1. Prüfer: Lutz Schröder (100190)

---

---

**Modulbezeichnung:** Seminar Wissensrepräsentation und -verarbeitung (WuV) 5 ECTS  
(Knowledge Representation and -processing)

Modulverantwortliche/r: Michael Kohlhase

Lehrende: Michael Kohlhase, Dennis Müller

---

Startsemester: WS 2019/2020	Dauer: 1 Semester	Turnus: halbjährlich (WS+SS)
Präsenzzeit: 30 Std.	Eigenstudium: 120 Std.	Sprache: Deutsch oder Englisch

---

**Lehrveranstaltungen:**

Seminar Wissensrepräsentation und -verarbeitung (WS 2019/2020, Oberseminar, 2 SWS, Michael Kohlhase et al.)

---

**Inhalt:**

Dieses Seminar behandelt ausgewählte Themen aus dem Gebiet der Wissensrepräsentation und -Verarbeitung, mit einem Fokus auf mathematisches Wissen.

Die Vortragsthemen sind sehr unterschiedlich im Schwierigkeitsgrad, sie reichen von Einführungsthemen für ambitionierte Bachelor-Studenten bis zur Forschungsfront für Doktoranden. Sie spiegeln recht gut die Forschungsinteressen der KWARC Gruppe wieder. Daher ist dieses Seminar sehr gut geeignet um in die Arbeitsgruppe einzusteigen (z.B. für eine Promotion).

**Lernziele und Kompetenzen:**

Aktuelle Fachliteratur verstehen, Forschungsthemen für ein Fachpublikum verständlich aufbereiten und vortragen, Kommunikation mit Experten.

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

[1] **Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Seminare, Praktika, Bachelorarbeit | Seminar)

[2] **Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Hauptseminar)

Dieses Modul ist daneben auch in den Studienfächern "Informatik (Master of Science)" verwendbar.

---

**Studien-/Prüfungsleistungen:**

Seminar Wissensrepräsentation und -verarbeitung (Prüfungsnummer: 635405)

(englische Bezeichnung: Knowledge Representation and -processing)

Prüfungsleistung, Seminarleistung

Anteil an der Berechnung der Modulnote: 100%

weitere Erläuterungen:

Zum Bestehen dieser Lehrveranstaltung muss ein Vortrag im Seminar gehalten und der Inhalt schriftlich ausgearbeitet werden.

Prüfungssprache: Deutsch oder Englisch

Erstablingung: WS 2019/2020, 1. Wdh.: keine Wdh.

1. Prüfer: Michael Kohlhase (100451)

---

---

**Modulbezeichnung:** **Nailing your Thesis (VUE+PROJ 10-ECTS) (OSS-NYT-VUE+PROJ)** **10 ECTS**  
 (Nailing your Thesis (VUE+PROJ 10-ECTS))

Modulverantwortliche/r: Dirk Riehle  
 Lehrende: Dirk Riehle

---

Startsemester: SS 2019	Dauer: 1 Semester	Turnus: halbjährlich (WS+SS)
Präsenzzeit: 60 Std.	Eigenstudium: 240 Std.	Sprache: Englisch

---

**Lehrveranstaltungen:**

Nailing your Thesis (PROJ) (SS 2019, Sonstige Lehrveranstaltung, 2 SWS, Dirk Riehle)  
 Nailing your Thesis (VUE) (SS 2019, Vorlesung mit Übung, 4 SWS, Dirk Riehle)

---

**Inhalt:**

This course teaches students how to perform scientific research for their final thesis or a research paper. The goal is to prepare students for a Bachelor or Master research thesis.

The course covers the following topics:

- Science and society
- The research process
- Exploratory research
- Confirmatory research
- Writing a thesis/paper
- The scientific community

Students can choose one or both of two components:

- VUE (lecture + exercise), 4 SWS, 5 ECTS. VUE combines lectures with homework and exercises. VUE is run as a 3h block.
- PROJ (small research project), 2 SWS, 5 ECTS. In PROJ, students perform a small research project, either individually or in teams. The available projects will be presented at the beginning of the course. Students perform the research, write a paper, and hold a presentation about their work.

If you only want to take OSS-NYT-PROJ please still come to OSS-NYT-VUE class on the first day to learn about the projects.

The overall schedule can be found at <https://goo.gl/VqoFO> . Please sign up for the course on StudOn (link accessible through schedule spreadsheet) as soon as possible.

**Lernziele und Kompetenzen:**

- Understand how to perform research
- Understand how to write a research thesis

**Literatur:**

The syllabus, schedule, literature, and more can be found at <http://nythesis.com>

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

[1] **Informatik (Bachelor of Science): 5-6. Semester**

(Po-Vers. 2009s | Seminare, Praktika, Bachelorarbeit)

Dieses Modul ist daneben auch in den Studienfächern "International Information Systems (IIS) (Master of Science)" verwendbar.

---

**Studien-/Prüfungsleistungen:**

Nailing your Thesis (VUE+PROJ 10-ECTS) (Prüfungsnummer: 917928)

(englische Bezeichnung: Nailing your Thesis (VUE+PROJ 10-ECTS))

Prüfungsleistung, Praktikumsleistung

Anteil an der Berechnung der Modulnote: 100%

weitere Erläuterungen:

- Exercise participation
- Homework assignments
- Project work

Erstablingung: SS 2019, 1. Wdh.: keine Angabe  
1. Prüfer: Dirk Riehle (100100)

---



---

**Modulbezeichnung:** **Software Architecture (VUE+PROJ 10-ECTS) (OSS-ARCH-VUE+PROJ)** **10 ECTS**  
 (Software Architecture (VUE+PROJ 10-ECTS))

Modulverantwortliche/r: Martin Jung, Dirk Riehle  
 Lehrende: Martin Jung, Dirk Riehle

---

Startsemester: SS 2019	Dauer: 1 Semester	Turnus: jährlich (SS)
Präsenzzeit: 60 Std.	Eigenstudium: 240 Std.	Sprache: Deutsch

---

**Lehrveranstaltungen:**

Software Architecture (PROJ) (SS 2019, Sonstige Lehrveranstaltung, 2 SWS, Martin Jung et al.)  
 Softwarearchitektur (SS 2019, Vorlesung mit Übung, 4 SWS, Martin Jung et al.)

---

**Inhalt:**

This course teaches students concepts, methods, and tools of software architecture.

Die Architektur eines Softwaresystems beschreibt die wesentlichen Komponenten des Systems, ihre Beziehungen und Struktur, sowie das Verhalten und die Dynamik der Beziehungen und Struktur dieser Komponenten. Dieser Kurs vermittelt in einer Vorlesung zunächst die folgenden Aspekte von Softwarearchitektur:

- Grundlegende Bausteine und ihre Beziehungen
- Softwarearchitekturbeschreibungssprachen
- Softwarearchitekturstile und -muster
- Bibliotheken, Rahmenwerke und Plattformen
- Formale sowie de-facto Industriestandards
- Die Softwarearchitekturen von Beispielsystemen
- Nicht technische Kriterien in der Architektur
- Werkzeuge für Softwarearchitekten
- Vorgehensmodelle der Softwarearchitektur
- Architekturgetriebene Entwicklung
- Die Rolle und Funktion der Softwarearchitektin

Studierende können eine oder beide von zwei Komponenten wählen:

- VUE (Vorlesung + Übungen), 4 SWS, 5 ECTS
- PROJ (kleines Projekt), 2 SWS, 5 ECTS. Die Projekte werden von unseren Industriepartnern bereitgestellt. Hier dokumentieren, analysieren und bewerten Studierende die Softwarearchitektur eines realen Softwaresystems. Diese verschiedenen Aspekte werden im Laufe des Semesters inkrementell abgearbeitet und am Ende dem Industriepartner in einer Präsentation vorgestellt.

Der Unterricht findet als 3h-Block während der Vorlesungszeit statt. Der Zeitplan befindet sich hier: <http://goo.gl/ZXJjg> . Der Zeitplan enthält auch einen Link auf den zur Veranstaltung gehörigen StudOn Kurs. Bitte registrieren Sie sich auf StudOn sobald wie möglich.

**Lernziele und Kompetenzen:**

- Ganzheitliches Verständnis des Konzepts "Softwarearchitektur"
- Befähigung zur Bewertung, Auswahl und Konstruktion problemangemessener Architekturen
- Kenntnis architekturgetriebener Entwicklungsmethodik und entsprechender Werkzeuge
- Kenntnis der typischen Verantwortlichkeiten und der Methodik eines Softwarearchitekten

**Literatur:**

- <http://goo.gl/ou7mja>

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

[1] **Informatik (Bachelor of Science): 5-6. Semester**

(Po-Vers. 2009s | Seminare, Praktika, Bachelorarbeit)

Dieses Modul ist daneben auch in den Studienfächern "International Information Systems (IIS) (Master of Science)" verwendbar.

**Studien-/Prüfungsleistungen:**

Software Architecture (VUE+PROJ 10-ECTS) (Prüfungsnummer: 359410)

(englische Bezeichnung: Software Architecture (VUE+PROJ 10-ECTS))

Prüfungsleistung, Praktikumsleistung

Anteil an der Berechnung der Modulnote: 100%

weitere Erläuterungen:

- Unterricht
- Hausaufgaben
- Mündliche Prüfung
- Projektarbeit

Erstablingung: SS 2019, 1. Wdh.: keine Angabe

1. Prüfer: Martin Jung (100803), 2. Prüfer: Dirk Riehle (100100)

---

---

**Modulbezeichnung:** Product Management (VUE+PROJ 10-ECTS) (OSS-PROD-VUE+PROJ) 10 ECTS  
 (Product Management (VUE+PROJ 10-ECTS))

Modulverantwortliche/r: Dirk Riehle  
 Lehrende: Dirk Riehle

---

Startsemester: WS 2019/2020	Dauer: 1 Semester	Turnus: jährlich (WS)
Präsenzzeit: 60 Std.	Eigenstudium: 240 Std.	Sprache: Englisch

---

**Lehrveranstaltungen:**

Product Management (VUE) (WS 2019/2020, Vorlesung mit Übung, 2 SWS, Dirk Riehle)  
 Product Management (PROJ) (WS 2019/2020, Sonstige Lehrveranstaltung, 2 SWS, Dirk Riehle)

---

**Inhalt:**

This course teaches students the concepts, methods, and tools of software product management. Product management is an important function in software development organizations. A product manager conceives and defines new products. His or her task is to understand the market incl. customers, to develop a product vision from that understanding, to translate it into product requirements, define those requirements on a by-feature basis and work with engineering to ensure these features are properly realized in the product under development.

- Role, tasks, and responsibilities of a product manager
- Process, methods, techniques and tools of product management
- Managing incremental/sustaining as well as disruptive innovation
- Open source product management; new trends in product management

Students can choose one or both of two components:

- VUE (lecture + homework), 2 SWS, 5 ECTS. VUE uses teaching cases as commonly used in MBA programs. The teaching cases are available for free at <http://pmbycase.com>.
- PROJ (small project), 2 SWS, 5 ECTS. In PROJ, students perform a small product management project, either individually or in teams. In such a project, students write a business plan for a market opportunity or a startup. There are two sources of projects: Those supplied by the professorship and those supplied by students. If we supply any projects, we will tell you about them in the first lecture. Student-supplied projects can be suggested by students to Prof. Riehle at least one week before class starts and must meet the requirements described here: <https://goo.gl/sNPi1i>

PROD projects are run as shared projects, in which all participants contribute and get to participate in the project results. Read more at <https://wp.me/pDU66-2p4>.

Class is run as two 90min blocks. The first block discusses the teaching cases. The second block is a coaching session for the projects (10 ECTS only).

The overall schedule can be found at <http://goo.gl/tTAI0> . Please sign up for the course on StudOn (link accessible through schedule spreadsheet) as soon as possible.

**Lernziele und Kompetenzen:**

- Understand the role, function, and responsibilities of a product manager
- Understand key concepts, methods, and tools of software product management
- Understand different business situations, incl. Incremental vs. disruptive innovation

**Literatur:**

- <http://goo.gl/41DgSr>
- 

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

[1] **Informatik (Bachelor of Science): 5-6. Semester**  
 (Po-Vers. 2009s | Seminare, Praktika, Bachelorarbeit)

Dieses Modul ist daneben auch in den Studienfächern "International Information Systems (IIS) (Master of Science)" verwendbar.

**Studien-/Prüfungsleistungen:**

Product Management (VUE+PROJ 10-ECTS) (Prüfungsnummer: 867375)

(englische Bezeichnung: Product Management (VUE+PROJ 10-ECTS))

Prüfungsleistung, Praktikumsleistung

Anteil an der Berechnung der Modulnote: 100%

weitere Erläuterungen:

- In-class participation
- Homework assignments
- Project work

Erstablingung: WS 2019/2020, 1. Wdh.: keine Angabe

1. Prüfer: Dirk Riehle (100100)

---

**Modulbezeichnung:** **Praktikum Software Engineering: Verfahren und Werkzeuge (SWEPrak)** **10 ECTS**  
(Lab course: software engineering: methods and tools)

Modulverantwortliche/r: Francesca Saglietti  
Lehrende: Marc Spisländer, Xiaochen Wu

Startsemester: SS 2019                      Dauer: 1 Semester                      Turnus: halbjährlich (WS+SS)  
Präsenzzeit: 45 Std.                      Eigenstudium: 255 Std.                      Sprache: Deutsch

**Lehrveranstaltungen:**

Software Engineering in der Praxis (SS 2019, Übung, 3 SWS, Xiaochen Wu)

**Inhalt:**

In den praktischen Übungen werden Werkzeuge zur Entwicklung und zur Analyse komplexer Software vorgestellt, deren industrielle Einsetzbarkeit anschließend von den Teilnehmern anhand für die Praxis repräsentativer Aufgabenstellungen erprobt wird.

**Lernziele und Kompetenzen:**

Die Studierenden

- erläutern Potenzial und Grenzen unterschiedlicher Werkzeuge zur Unterstützung softwaretechnischer Tätigkeiten;
- wenden unterschiedliche Werkzeuge an, um sowohl selbständig als auch in Teams Beispielaufgaben aus dem Bereich der objektorientierten Analyse, des objektorientierten Entwurfs, des Testens, des Beweisens und des Projektmanagements zu lösen;
- erklären erstellte Lösungen und skizzieren mögliche Alternativen.

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

[1] **Informatik (Bachelor of Science): 5-6. Semester**

(Po-Vers. 2009s | Seminare, Praktika, Bachelorarbeit)

**Studien-/Prüfungsleistungen:**

Praktikum Software Engineering in der Praxis (Prüfungsnummer: 102524)

Prüfungsleistung, Praktikumsleistung, Dauer (in Minuten): 120

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Software Engineering in der Praxis

weitere Erläuterungen:

Die Praktikumsleistung besteht aus einer Prüfung am Rechner. Für die Zulassung zu dieser Prüfung ist die Abnahme von 10 Aufgabenblättern erforderlich, die im Rahmen des Praktikums zu bearbeiten sind.

Prüfungssprache: Deutsch oder Englisch

Erstablingung: SS 2019, 1. Wdh.: keine Angabe

1. Prüfer: Francesca Saglietti (100112)

---

**Modulbezeichnung:** **Software-Anwendungen mit KI (VUE 10-ECTS) (OSS-SAKI) 10 ECTS**  
 (Software-Applications with AI (VUE 10-ECTS))

Modulverantwortliche/r: Dirk Riehle

Lehrende: Dirk Riehle

---

Startsemester: SS 2019

Dauer: 1 Semester

Turnus: jährlich (SS)

Präsenzzeit: 60 Std.

Eigenstudium: 240 Std.

Sprache: Deutsch

---

**Lehrveranstaltungen:**

Software-Anwendungen mit KI (VUE) (SS 2019, Vorlesung mit Übung, 2 SWS, Dirk Riehle)

Introduction to Complex Data Analysis in Python (KU) (SS 2019, Kurs, Michael Dorner et al.)

---

**Empfohlene Voraussetzungen:**

Grundlagenveranstaltungen in maschinellem Lernen und künstlicher Intelligenz

---

**Inhalt:**

Dieser Kurs lehrt fortgeschrittene Methoden des maschinellen Lernens resp. der künstlichen Intelligenz anhand von vier nicht-trivialen Anwendungsbeispielen mit realen Daten aus der Industrie.

Jedes der vier Beispiele stellt eine umfangreiche Hausaufgabe für Studierende dar, in der unterschiedliche Problemarten (Korrelation, Klassifikation, etc.) mit unterschiedlichen Methoden (Clustering, Bayesian Networks, etc.) in unterschiedlichen Fachgebieten (Automobilindustrie, Finanzindustrie, etc.) kombiniert werden. Jede Aufgabe wird von einem dazugehörigen Industriepartner mitbetreut.

Die vier Beispiele werden nacheinander abgearbeitet und strukturieren die Kurszeit in vier gleich große Abschnitte von jeweils drei Wochen, von denen jeder Abschnitt dieselbe Struktur hat:

- 1) Vorbereitung auf den anstehenden Abschnitt durch Wiederholung relevanter Literatur
- 2) Einführung in das Problem; Diskussion von Bibliotheken und Vorgehen zur Problemlösung
- 3) Wiederholte Diskussion (zwei weitere Sitzungen) des Problems und der Herangehensweise
- 4) Abgabe der Problemlösung, bestehend aus Erläuterung sowie Quelltext und Ergebnissen

Die Programmierung findet in Python statt. Eine Woche vor Kursbeginn können Studierende, welche Python noch nicht kennen oder ihre Kenntnisse auffrischen wollen, an unserem Einführungskurs zu Python und ML teilnehmen, siehe <https://goo.gl/X6j4nZ>.

Es wird erwartet, dass Studierende aktiv mitarbeiten, sich etwaige fehlende Grundlagen selbst aneignen, und die technischen Aufgaben eigenständig lösen werden. Der Zeitplan ist unter <https://goo.gl/5ynxcQ> einsehbar. Bitte registrieren Sie sich für den Kurs auf StudOn (Verweis über Zeitplan verfügbar).

**Lernziele und Kompetenzen:**

Erwerb praktischer Kompetenz in der Anwendung von ML und KI-Methoden auf reale industrielle Daten

**Literatur:**

(Literaturliste wird zeitnah bereitgestellt)

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

[1] **Informatik (Bachelor of Science): 5-6. Semester**

(Po-Vers. 2009s | Seminare, Praktika, Bachelorarbeit)

Dieses Modul ist daneben auch in den Studienfächern "International Information Systems (IIS) (Master of Science)" verwendbar.

---

**Studien-/Prüfungsleistungen:**

Software-Anwendungen mit KI (VUE 10-ECTS) (Prüfungsnummer: 419238)

(englische Bezeichnung: Software-Applications with AI (VUE 10-ECTS))

Prüfungsleistung, Praktikumsleistung

Anteil an der Berechnung der Modulnote: 100%

weitere Erläuterungen:

Gleichgewichtete Summe der Noten der Hausaufgaben

Erstablingung: SS 2019, 1. Wdh.: keine Angabe  
1. Prüfer: Dirk Riehle (100100)

---

---

**Modulbezeichnung:** IoT Security (IoTSec) 10 ECTS  
(IoT Security)

Modulverantwortliche/r: Felix Freiling  
Lehrende: Philipp Klein

---

Startsemester: WS 2019/2020	Dauer: 1 Semester	Turnus: unregelmäßig
Präsenzzeit: 20 Std.	Eigenstudium: 280 Std.	Sprache: Deutsch

---

**Lehrveranstaltungen:**

IoT Security (WS 2019/2020, Praktikum, Felix Freiling et al.)

---

**Empfohlene Voraussetzungen:**

Grundkenntnisse in Python

---

**Inhalt:**

Dieses Projekt wird in kleinen Teams bearbeitet. Jedes Team muss am Ende einen schriftlichen Bericht einreichen. Die Arbeit der einzelnen Teammitglieder muss hier klar ersichtlich sein.

Jedes Team erhält eine Reihe von IoT- und Smart-Home-Geräten. Diese sollen zunächst zu einem funktionierenden System verbunden werden. Im Anschluss wird detailliert für jedes Gerät betrachtet, welche Daten gespeichert, versendet und erhoben werden.

Zusammengefasster Inhalt:

- Einige Grundlagenvorlesungen zu IoT und Smart Home
- Verknüpfung von diversen IoT-Geräten
- Detaillierte Analyse von IoT-Geräten bezüglich Datenverkehr, Datenschutz und Usability
- Ausarbeitung und Anwendung von Angriffsszenarien auf IoT-Geräte

**Lernziele und Kompetenzen:**

*Fachkompetenz*

*Evaluieren (Beurteilen)*

Lernende können aufgrund Ihrer Erfahrung im Projekt bewerten, ob ein IoT-Gerät als "sicher" einzustufen ist. Diese Bewertung erfolgt entweder auf Grundlage von frei verfügbaren Daten oder über eine selbstständige Evaluation des Geräts.

*Selbstkompetenz*

Lernende können eigenständig, ohne Aufsicht und Anleitung, ein IoT-Gerät detailliert analysieren und zu festgelegten Zeitpunkten Ergebnisse vorweisen.

*Sozialkompetenz*

Lernende können in kleinen Teams effektiv und effizient zusammenarbeiten, die Arbeit gerecht verteilen und gemeinsam einen Bericht anfertigen.

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

[1] **Informatik (Bachelor of Science): 5-6. Semester**

(Po-Vers. 2009s | Seminare, Praktika, Bachelorarbeit)

---

**Studien-/Prüfungsleistungen:**

IoT Security (Prüfungsnummer: 916620)

(englische Bezeichnung: IoT Security)

Prüfungsleistung, Praktikumsleistung

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- IoT Security

weitere Erläuterungen:

Die Note setzt sich zusammen aus Ausarbeitung und Vorträgen

Prüfungssprache: Deutsch oder Englisch



Erstablingung: WS 2019/2020, 1. Wdh.: SS 2020

1. Prüfer: Felix Freiling (100128)

---

**Organisatorisches:**

Die Anmeldung erfolgt persönlich beim Dozenten. Bitte dazu eine Mail an [cs1-websec@cs.fau.de](mailto:cs1-websec@cs.fau.de) schicken. Der Dozent (Philipp Klein) darf direkt geduzt werden.

---

**Modulbezeichnung:** **Praktikum Enterprise Computing (PEC)** **10 ECTS**  
(Lab Class Enterprise Computing)

Modulverantwortliche/r: Peter Wilke

Lehrende: Peter Wilke

---

Startsemester: SS 2019	Dauer: 1 Semester	Turnus: halbjährlich (WS+SS)
Präsenzzeit: 60 Std.	Eigenstudium: 240 Std.	Sprache: Deutsch

---

**Lehrveranstaltungen:**

Praktikum Enterprise Computing (SS 2019, Praktikum, 2 SWS, Peter Wilke)

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

[1] **Informatik (Bachelor of Science): 5-6. Semester**

(Po-Vers. 2009s | Seminare, Praktika, Bachelorarbeit)

---

**Studien-/Prüfungsleistungen:**

Praktikum Enterprise Computing (Prüfungsnummer: 594684)

(englische Bezeichnung: Lab Class Enterprise Computing)

Prüfungsleistung, Praktikumsleistung

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- **Praktikum Enterprise Computing**

weitere Erläuterungen:

In weekly meetings general feedback and suggestions for improvements will be presented. Participation is not obligatory.

Prüfungssprache: Deutsch

Erstablingung: SS 2019, 1. Wdh.: WS 2019/2020

1. Prüfer: Peter Wilke (100155)

---

---

**Modulbezeichnung:** **Praktikum angewandte Systemsoftwaretechnik (PASST)** **10 ECTS**  
 (Laboratory on Applied Systems Software Technology)

Modulverantwortliche/r: Jürgen Kleinöder

Lehrende: Tobias Langer, Maximilian Ott

Startsemester: SS 2019

Dauer: 1 Semester

Turnus: halbjährlich (WS+SS)

Präsenzzeit: 150 Std.

Eigenstudium: 150 Std.

Sprache: Deutsch

---

### Lehrveranstaltungen:

Praktikum angewandte Systemsoftwaretechnik (SS 2019, Praktikum, Anwesenheitspflicht, Tobias Langer et al.)

---

### Inhalt:

Im Praktikum erlernen und verwenden die Studierenden verschiedene Techniken und Prozesse der systemnahen Softwareentwicklung im OpenSource-Umfeld. Nähere Informationen zum konkreten Inhalt in diesem Semester finden sich auf der unter "Weitere Informationen" verlinkten Veranstaltungsseite.

### Lernziele und Kompetenzen:

#### *Fachkompetenz*

##### *Anwenden*

Die Studenten

- verstehen Kodierrichtlinien und sind in der Lage diese situationsgemäß anzuwenden

##### *Analysieren*

Die Studenten

- erkunden gegebenen Programmcode grossen Umfangs und hoher Komplexität
- bestimmen dessen Funktionalität und beschreiben und diskutieren dies

##### *Evaluiieren (Beurteilen)*

Die Studenten

- beurteilen Qualität, Korrektheit und Richtlinienkonformität fremder Programme
- analysieren in Softwaresystemen, insbesondere dem Linux-Kernel auftretende Fehler und Situationsbilder und können solche bewerten und auf ihr Vorkommen testen
- evaluieren und verwenden geeignete Mittel zur Erkennung der Fehlerursachen
- verifizieren die korrekte Behebung eines erkannten Fehlers
- beschreiben, bewerten und kritisieren das eigene und das Vorgehen Dritter bei der Programmentwicklung, Fehlersuche und Integration

##### *Erschaffen*

Die Studenten

- planen und entwickeln Gerätetreiber für PCI- und USB-Geräte für das Betriebssystem Linux
- planen und entwickeln systemnahe Programme zur Interaktion mit Geräten im Zusammenspiel mit den entwickelten Gerätetreibern
- konzipieren, planen und entwickeln systemnahe Software, Systemsoftware oder Bestandteile eines Betriebssystemkerns; erstellen Dokumentation und präsentieren ihr Vorgehen
- erstellen geeignete Maßnahmen (Patches) zur Behebung erkannter Fehler und Probleme
- erzeugen textuelle Beschreibungen der Fehlerbedingungen, -symptome und -ursachen, der Fehlersuche sowie des Patches nach Kriterien relevanter Open-Source-Projekte in englischer Sprache, die geeignet sind, solche Patches und Beschreibungen an diese Open-Source-Projekte weiterzugeben
- entwickeln vorbeschriebene Patches und Beschreibungen und reichen diese bei relevanten Open-Source-Projekten ein
- interagieren mit den externen Entwicklern dieser Projekte mit dem Ziel eingereichtes Material akzeptiert zu bekommen
- erstellen Analysen und Präsentationen eigener und fremder Arbeit und tragen diese in geeigneter Weise vor einem Fachpublikum vor

#### *Lern- bzw. Methodenkompetenz*

Die Studenten

- verwenden gängige Werkzeuge der Softwareentwicklung im Open-Source-Bereich wie git, gdb, kgdb, qemu/kvm und cscope
- verstehen deren Funktionsweise
- verwenden diese erfolgreich in internen Aufgaben und Projekten sowie in der Interaktion mit externen Entwicklern
- interpretieren Code im Hinblick auf dessen vorgesehene Funktion, mögliche Fehler, sinnvolle Erweiterungspunkte und Qualitätsaspekte
- stellen technische, methodische und soziale Sachverhalte geeignet dar

#### *Selbstkompetenz*

Die Studenten

- sind in der Lage mit Kritik und Änderungswünschen umzugehen
- überwinden Berührungsängste im Kontakt mit externen Dritten
- bringen sich konstruktiv und produktiv in Open-Source-Projekte ein

#### *Sozialkompetenz*

Die Studenten

- organisieren selbständig die gemeinsame Bearbeitung der Übungsaufgaben und lösen diese kooperativ in kleinen Gruppen
- kommunizieren erfolgreich in englischer Sprache mit Betreuern und mit externen Entwicklern unter Einhaltung relevanter Protokolle im Open-Source-Umfeld
- gehen professionell mit Kritik an eigener Arbeit um und beziehen berechtigte Kritik in ihre zukünftige Arbeitsweise ein
- verhalten sich angemessen beim kritisieren fremder Arbeit gegenüber dem Ersteller dieser Arbeit oder Dritten
- erkennen und befolgen geschriebene und ungeschriebene Regeln im Umfeld relevanter Open-Source-Projekte; verhalten sich angemessen bei möglichen Konfliktsituationen

#### **Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

##### [1] **Informatik (Bachelor of Science): 5-6. Semester**

(Po-Vers. 2009s | Seminare, Praktika, Bachelorarbeit)

#### **Studien-/Prüfungsleistungen:**

Praktikum angewandte Systemsoftwaretechnik (Prüfungsnummer: 113845)

(englische Bezeichnung: Laboratory on Applied Systems Software Technology)

Prüfungsleistung, Praktikumsleistung

Anteil an der Berechnung der Modulnote: 100%

weitere Erläuterungen:

Es sind semesterbegleitend sechs Übungsaufgaben zu bearbeiten. In der vorlesungsfreien Zeit findet ein Blockpraktikum statt (2 Wochen, Programmierung und zwei Vorträge a ca. 15 Minuten). Arbeit in 2er-Teams.

Übungsaufgaben, Programmierung im Blockpraktikum und Vorträge werden bepunktet. Die Note ergibt sich auf Basis der erreichten Punkte. Gewichtung Übungsaufgaben/Blockpraktikum 40/60.

Erstablægung: SS 2019, 1. Wdh.: keine Angabe

1. Prüfer: Jürgen Kleinöder (100163)

---

**Modulbezeichnung:** **Praktikum: Entwicklung interaktiver eingebetteter Systeme (PR-EES)** **10 ECTS**

(Lab: Developing Cyber-Physical Embedded Systems)

Modulverantwortliche/r: Joachim Falk

Lehrende: Joachim Falk, Mehmet Akif Özkan

---

Startsemester: WS 2019/2020

Dauer: 1 Semester

Turnus: jährlich (WS)

Präsenzzeit: 120 Std.

Eigenstudium: 180 Std.

Sprache: Deutsch oder Englisch

---

**Lehrveranstaltungen:**

Praktikum: Entwicklung interaktiver eingebetteter Systeme (WS 2019/2020, Praktikum, 8 SWS, Joachim Falk)

---

**Inhalt:**

Technische Systeme machen den Alltag immer angenehmer. Unsere Ansprüche steigen dabei stetig und so erwarten wir, dass unsere Geräte immer mehr Funktionen bieten, und gleichzeitig immer einfacher zu bedienen sind. Eingebettete Systeme, also spezialisierte Computer, die direkt in technische Systeme integriert sind, sind dabei seit Jahren das Mittel der Wahl, um unseren Geräten die benötigte Intelligenz zu verpassen. Es hat sich gezeigt, dass es immer wichtiger wird, dass diese Systeme noch mehr mit ihrer Umwelt - und vor allem dem Menschen - interagieren. Ein Paradebeispiel hierfür sind Fahrerassistenzsysteme in modernen Fahrzeugen. Derartige Systeme erhöhen nicht nur den Komfort, sondern vor allem auch die Sicherheit aktueller Automobile. Diese stark mit ihrer Umwelt interagierenden, eingebetteten Systeme nennt man auch Cyberphysical Systems. Das Praktikum „Entwicklung interaktiver eingebetteter Systeme“ behandelt die Entwicklung und Programmierung derartiger Systeme. In drei Abschnitten bietet das Praktikum Einblick in alle Phasen der Entwicklung eingebetteter Systeme. Dabei entwickeln wir ein Objekterkennungssystem, bei dem ein Objekt erkannt werden soll, um dessen Position zur Steuerung der von Ihnen entwickelten Software einzusetzen.

- In Phase I entwickeln wir Filter- und Objekterkennungsalgorithmen, mit deren Hilfe Objekte automatisch in einem Videostrom detektiert werden sollen. Das Erkennen soll möglichst zuverlässig und bei wechselnden Umgebungsbedingungen funktionieren.
- In Phase II portieren wir die entwickelten Algorithmen auf das eingebettete System. Wir verändern allerdings nicht die Programmierung eines Autos, sondern führen die Umsetzung anhand eines virtuellen Prototypen durch. Mittels des virtuellen Prototypen entscheiden wir dann, was in Software und was in Hardware implementiert werden soll.
- In Phase III testen wir unsere Entwicklungen auf einem realen System. Für diesen Zweck steht uns ein am Lehrstuhl entwickelter Demonstrator zur Verfügung. Dadurch lassen sich die entwickelten Filter auf einem realen System testen. Der Demonstrator verfügt über eine Kamera, die den Bereich vor dem Demonstrator überwacht. Die Kameradaten werden anschließend an die selbst entwickelten Module weitergeleitet. Diese erzeugen im Anschluss Steuersignale für einen Motor, der einen verfahrbaren Schlitten antreibt.

**Lernziele und Kompetenzen:**

*Fachkompetenz*

*Wissen*

- Die Studierenden legen Konzepte des modellbasierten Entwurfs eingebetteter Systeme dar.

*Verstehen*

- Die Studierenden veranschaulichen die Hauptaufgaben beim Systementwurf auf verschiedenen Abstraktionsebenen von der Anwendung selbst, dem Gesamtsystem, bis hinunter zu einem Hardwaremodul.
- Die Studierenden schildern den Entwurf eines System von der Idee, über die Spezifikation bis zur Implementierung, der Analyse und letztendlich der Validierung an einem realen mechatronischen Versuchsaufbau.

*Anwenden*

- Die Studierenden setzen die Integration von digitalen Hardware- und Software-Komponenten um.
- Die Studierenden wenden die Programmiersprachen C/C++/SystemC für die Entwicklung von Hardware- und Software-Komponenten an.

#### *Selbstkompetenz*

- Die Studierenden schätzen ihre individuellen Stärken ab, um eine geeignete Aufteilung innerhalb der Gruppe festzulegen.

#### *Sozialkompetenz*

- Die Studierenden erarbeiten gemeinsam Schnittstellendefinitionen über Gruppengrenzen hinweg.
- Die Studierenden erarbeiten kooperativ in Gruppen Lösungskonzepte und implementieren diese gemeinsam.

---

### **Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

#### **[1] Informatik (Bachelor of Science): 5-6. Semester**

(Po-Vers. 2009s | Seminare, Praktika, Bachelorarbeit)

---

### **Studien-/Prüfungsleistungen:**

Praktikum: Entwicklung interaktiver eingebetteter Systeme\_ (Prüfungsnummer: 716025)

Prüfungsleistung, Praktikumsleistung

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Praktikum: Entwicklung interaktiver eingebetteter Systeme

weitere Erläuterungen:

Die Bewertung der Prüfungsleistung setzt sich zusammen aus 1/2 Arbeitsweise im Praktikum und 1/2 Abschlussvortrag.

Erstbelegung: WS 2019/2020, 1. Wdh.: keine Wdh.

1. Prüfer: Jürgen Teich (100115)

---

---

**Modulbezeichnung:** **Hackerpraktikum (Bachelor) (HackBSc)** **10 ECTS**  
 (Hacking Lab (Bachelor))

Modulverantwortliche/r: Tilo Müller  
 Lehrende: Tilo Müller

---

Startsemester: WS 2019/2020	Dauer: 1 Semester	Turnus: jährlich (WS)
Präsenzzeit: 50 Std.	Eigenstudium: 250 Std.	Sprache: Deutsch oder Englisch

---

**Lehrveranstaltungen:**

Hackerpraktikum (Bachelor) (WS 2019/2020, Praktikum, Tilo Müller et al.)

---

**Empfohlene Voraussetzungen:**

- Interesse an IT-Sicherheit.
  - Bereitschaft, viel Zeit in das Praktikum zu investieren.
  - Programmierkenntnisse in C/Assembler und mindestens einer Skriptsprache (bspw. Python).
  - Linux-Kenntnisse.
  - Von großem Vorteil sind darüber hinaus Netzwerk-Kenntnisse und Vorkenntnisse im Bereich IT-Sicherheit.
- 

**Inhalt:**

In diesem Praktikum lernen die Teilnehmer den kritischen Umgang mit offensiver IT-Sicherheit. Es werden prinzipielle Angriffskonzepte erörtert und in einer abgeschotteten Umgebung gezielt zur Anwendung gebracht. Durch diese praktischen Erfahrungen aus der Sichtweise eines "Hackers" werden die Teilnehmer bzgl. sicherheitsrelevanten Fragen sensibilisiert und können die gewonnenen Erkenntnisse dann letztendlich auch zur Absicherung von Systemen einsetzen. Das Hackerpraktikum wird in 5 Übungsblätter zu je 3 Wochen aufgeteilt, wobei die folgenden Themen bearbeitet werden:

- Blatt 1: Netzwerksicherheit (Sniffing, Spoofing, WPA, ...)
- Blatt 2: Webhacking (SQL Injections, XSS, CSRF, ...)
- Blatt 3: Systemsicherheit (Rootkits, Privileges, Suid, ...)
- Blatt 4: Reverse Engineering (Cracking, Malware Analysis, ...)
- Blatt 5: Exploitation (Buffer Overflows, Shellcode, ASLR, ...)

Von den Übungen müssen insgesamt 2/3 und pro Blatt 1/2 der möglichen Punkte erreicht werden. Die Übungen sind in Einzelarbeit abzugeben. Neben den Übungsblättern halten jede Woche 2 Studenten gemeinsam einen Vortrag über das Thema des aktuellen Übungsblattes.

**Lernziele und Kompetenzen:**

Die Studierenden können Schwachstellen in den vorgestellten Themenbereichen identifizieren und beschreiben. Sie können aktuelle Angriffs- und Verteidigungstechniken in konkreten Fällen auswählen und anwenden.

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

[1] **Informatik (Bachelor of Science): 5-6. Semester**

(Po-Vers. 2009s | Seminare, Praktika, Bachelorarbeit)

Dieses Modul ist daneben auch in den Studienfächern "Wirtschaftsinformatik (Bachelor of Science)" verwendbar.

---

**Studien-/Prüfungsleistungen:**

Hackerpraktikum (Prüfungsnummer: 253103)

Prüfungsleistung, Praktikumsleistung

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Hackerpraktikum (Bachelor)

weitere Erläuterungen:

Abgabe von Übungsblättern, eine Präsentation zum Thema eines Übungsblattes.

Erstablingung: WS 2019/2020, 1. Wdh.: keine Angabe  
1. Prüfer: Tilo Müller (100000)

---



---

**Modulbezeichnung:** Supercomputing Praktikum (SuCoPra) 10 ECTS  
(Supercomputing Praktikum)

Modulverantwortliche/r: Johannes Hofmann  
Lehrende: Johannes Hofmann

---

Startsemester: SS 2019	Dauer: 1 Semester	Turnus: jährlich (SS)
Präsenzzeit: 100 Std.	Eigenstudium: 200 Std.	Sprache: Deutsch oder Englisch

---

**Lehrveranstaltungen:**

Supercomputing-Praktikum (SS 2019, Praktikum, 8 SWS, Johannes Hofmann)  
Supercomputing-Praktikum Übung (SS 2019, Übung, Johannes Hofmann)

---

**Inhalt:**

Im Rahmen des Supercomputing Praktikums werden die Studierenden auf die Teilnahme an der Student Cluster Competition (SCC) vorbereitet.

**Lernziele und Kompetenzen:**

*Fachkompetenz*

*Anwenden*

Die Funktionsweise verschiedener Cluster-Rechner kann verglichen, beschrieben und bewertet werden.

*Sozialkompetenz*

Die Studierenden lernen sich selbst in einem Team zu organisieren. Aufgaben, Verantwortlichkeiten und Arbeitsleistung müssen von den Studierenden zu einem großen Teil selbst verwaltet werden.

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

**[1] Informatik (Bachelor of Science): 5-6. Semester**

(Po-Vers. 2009s | Seminare, Praktika, Bachelorarbeit)

---

**Studien-/Prüfungsleistungen:**

Supercomputing Praktikum (Prüfungsnummer: 182798)

(englische Bezeichnung: Student Cluster Competition)

Prüfungsleistung, Praktikumsleistung

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Supercomputing-Praktikum

weitere Erläuterungen:

Die Prüfungsleistung ergibt sich aus der Teilnahme am Praktikum.

Erstablingung: SS 2019, 1. Wdh.: WS 2019/2020

1. Prüfer: Dietmar Fey (100101)

---

---

**Modulbezeichnung: Praktikum Mustererkennung (PME)** **10 ECTS**  
(Lab Class Pattern Recognition)

Modulverantwortliche/r: Andreas Maier

Lehrende: Andreas Maier, Vincent Christlein

---

Startsemester: WS 2019/2020	Dauer: 1 Semester	Turnus: halbjährlich (WS+SS)
Präsenzzeit: 60 Std.	Eigenstudium: 240 Std.	Sprache: Deutsch oder Englisch

---

**Lehrveranstaltungen:**

Praktikum Mustererkennung (WS 2019/2020, Praktikum, Andreas Maier)

---

**Inhalt:**

At the Pattern Recognition Lab we offer project topics that are connected to our current research in the fields of pattern recognition & machine learning, medical image processing, and big data applications. Other than a course with fixed topic, project topics are defined individually. The 10 ECTS project is directed towards students of computer science. Also smaller 5 ECTS projects are available.

**Lernziele und Kompetenzen:**

Die Studierenden

- arbeiten sich in komplexe Softwaresysteme ein und erweitern diese
  - lernen, eigenständig Lösungsvorschläge auszuarbeiten und umzusetzen
  - dokumentieren die von ihnen geschriebene Software
- 

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

[1] **Informatik (Bachelor of Science): 5-6. Semester**

(Po-Vers. 2009s | Seminare, Praktika, Bachelorarbeit)

---

**Studien-/Prüfungsleistungen:**

Praktikum Mustererkennung (Prüfungsnummer: 670545)

(englische Bezeichnung: Lab Class Pattern Recognition)

Prüfungsleistung, Praktikumsleistung

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Praktikum Mustererkennung

weitere Erläuterungen:

The goal of this practical course is to familiarize the students with a Pattern Recognition System.

Erstablingung: WS 2019/2020, 1. Wdh.: SS 2020

1. Prüfer: Andreas Maier (100196)

---

---

**Modulbezeichnung:** HPC Software Projekt (HPCPRO) 10 ECTS  
(HPC Software Projekt)

Modulverantwortliche/r: Harald Köstler  
Lehrende: Harald Köstler

---

Startsemester: SS 2019	Dauer: 1 Semester	Turnus: unregelmäßig
Präsenzzeit: 5 Std.	Eigenstudium: 295 Std.	Sprache: Deutsch oder Englisch

---

**Lehrveranstaltungen:**

HPC Software Projekt (SS 2019, Projektseminar, 8 SWS, Harald Köstler)

---

**Inhalt:**

Anhand eines aktuellen Forschungsthema im Bereich High Performance Computing sollen die Studierenden an die wissenschaftliche Arbeitsweise im Bereich Informatik herangeführt werden. Dazu wird typischerweise in Gruppenarbeit ein größeres Softwarepaket entwickelt und auf eine konkrete Problemstellung aus dem HPC Bereich angewendet. Die Ergebnisse sollen in einem kurzen Bericht zusammengefasst werden.

**Lernziele und Kompetenzen:**

*Fachkompetenz*

*Analysieren*

Auswahl von geeigneten Algorithmen und Datenstrukturen, um ein vorgegebenes Problem effizient zu lösen.

*Erschaffen*

Entwicklung eines Softwarepaketes für eine konkrete Problemstellung aus dem HPC Bereich.

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

[1] **Informatik (Bachelor of Science): 5-6. Semester**

(Po-Vers. 2009s | Seminare, Praktika, Bachelorarbeit)

---

**Studien-/Prüfungsleistungen:**

HPC Software Projekt (Prüfungsnummer: 695344)

(englische Bezeichnung: HPC Software Project)

Prüfungsleistung, Praktikumsleistung

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- HPC Software Projekt

weitere Erläuterungen:

Bewertet wird der abgegebene Programmcode.

Erstablingung: SS 2019, 1. Wdh.: WS 2019/2020

1. Prüfer: Harald Köstler (100124)

---

---

**Modulbezeichnung:** **Praktikum: Lego Mindstorms (PR-LM)** **10 ECTS**  
 (Lab: Lego Mindstorms)

Modulverantwortliche/r: Stefan Wildermann  
 Lehrende: Stefan Wildermann

---

Startsemester: SS 2019	Dauer: 1 Semester	Turnus: jährlich (SS)
Präsenzzeit: 60 Std.	Eigenstudium: 240 Std.	Sprache: Deutsch oder Englisch

---

**Lehrveranstaltungen:**

Praktikum: Lego Mindstorms (SS 2019, Praktikum, Stefan Wildermann)

---

**Inhalt:**

Informatik befasst sich nicht nur mit der Programmierung von Desktop-Rechnern. Vielmehr können Computer in immer mehr Gegenständen unseres Alltags oder in bestimmten technischen Kontexten gefunden werden. Man spricht hier von eingebetteten Systemen.

Auch Roboter stellen solche eingebetteten Systeme dar. Ein Roboter erwacht durch sein Programm zum Leben. Die Programmierung von Robotern stellt einerseits eine Herausforderung dar. Andererseits ist sie aber auch mit viel Spaß verbunden.

In diesem Praktikum werden LEGO Mindstorms Roboter verwendet, die mittels der Sprache Java programmiert werden. Dazu wird das Betriebssystem leJOS verwendet. Ziel der Lehrveranstaltung ist es, ein praktisches Thema als Gruppe zu bearbeiten, Kenntnisse und Fertigkeiten in der Projektorganisation zu erwerben und die Fähigkeit der Problemlösung zu schulen.

**Lernziele und Kompetenzen:**

*Fachkompetenz*

*Erschaffen*

- Die Studierenden erstellen Lösungsideen für die Projekte und implementieren diese in Java für Lego Mindstorms Roboter.

*Selbstkompetenz*

- Die Studierenden schätzen ihre Stärken ab, um eine geeignete Aufteilung innerhalb der Gruppe zu finden.

*Sozialkompetenz*

- Die Studierenden erarbeiten gemeinsam Projektpläne und -dokumentation im Themengebiet Robotik.
  - Die Studierenden organisieren selbstständig die gemeinsame Durchführung des Projekts und führen diese kooperativ in Gruppen durch.
- 

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

**[1] Informatik (Bachelor of Science): 5-6. Semester**

(Po-Vers. 2009s | Seminare, Praktika, Bachelorarbeit)

---

**Studien-/Prüfungsleistungen:**

Praktikum Lego Mindstorms (Prüfungsnummer: 278855)

Prüfungsleistung, Praktikumsleistung

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Praktikum: Lego Mindstorms

weitere Erläuterungen:

Die Bewertung der Prüfungsleistung setzt sich zusammen aus 1/2 Arbeitsweise im Praktikum und 1/2 Dokumentation.

Erstabelleung: SS 2019, 1. Wdh.: keine Wdh.

1. Prüfer: Jürgen Teich (100115)



---

**Modulbezeichnung: GraPra (GraPra)** **10 ECTS**  
 (GraPra)

Modulverantwortliche/r: Marc Stamminger

Lehrende: Alexander Lier

---

Startsemester: WS 2019/2020

Dauer: 1 Semester

Turnus: jährlich (WS)

Präsenzzeit: 60 Std.

Eigenstudium: 240 Std.

Sprache: Deutsch

---

**Lehrveranstaltungen:**

GraPra Game Programming (WS 2019/2020, Praktikum, 10 SWS, Alexander Lier)

---

**Vorhergehende Module:**

Algorithmik kontinuierlicher Systeme

---

**Inhalt:**

Das Grafik-Programmierpraktikum besteht aus vier Teilen:

- Entwickeln eines Bomberman-Spiels (5 Wochen),
- Terrain Rendering (3 Wochen)
- Rendering von Kartendaten (2 Wochen),
- "Freestyle" (3 Wochen).

Die Bearbeitung der Aufgaben erfolgt in Teams von 2-3 Mitgliedern. Im ersten Teil liegt der Fokus auf C++ Programmieren und einem Überblick über die Grafikprogrammierung mit OpenGL. Im zweiten Teil wird die Grafikprogrammierung mit OpenGL vertieft. Im dritten Teil wird ein Level aus OpenStreetMap Daten generiert, in der letzten Aufgabe, Teil vier, stellen sich die Teams eigene (innerhalb des Themengebiets frei wählbare) Aufgaben.

**Lernziele und Kompetenzen:**

Die Studierenden

- erläutern die Stufen und Funktionsweise der Graphikpipeline
- beschreiben und klassifizieren unterschiedliche Rendering-Verfahren zur Berechnung von Beleuchtung und Schatten
- schildern einfache Algorithmen für Kollisionserkennung und -behandlung
- wenden fundierte Kenntnisse in C++, OpenGL und GLSL in der Softwareentwicklung für Animations- und Rendering-Aufgaben an
- implementieren im Rahmen von Projekten die erlernten Rendering-Algorithmen
- benutzen die Kollisionserkennung und -behandlungsalgorithmen in einfachen Animationen

**Literatur:**

- Bjarne Stroustrup, The C++ Programming Language
- OpenGL Red Book
- Tomas Akenine-Möller, Eric Haines und Naty Hoffman, Real-time Rendering

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

**[1] Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Seminare, Praktika, Bachelorarbeit | Praktikum Informatik)

Dieses Modul ist daneben auch in den Studienfächern "Berufspädagogik Technik (Master of Education)" verwendbar.

---

**Studien-/Prüfungsleistungen:**

Grafik-Praktikum Game Programming (Prüfungsnummer: 240715)

Prüfungsleistung, Praktikumsleistung

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- GraPra Game Programming

weitere Erläuterungen:

2/3 Punkte auf Übungsaufgaben, 1/3 Zwischen- und Abschlussvortrag

Erstablingung: WS 2019/2020, 1. Wdh.: SS 2020

1. Prüfer: Marc Stamminger (100173)

---

**Modulbezeichnung:** **Praktikum Software Engineering: Verfahren und Werkzeuge (SWE-PR-10)** **10 ECTS**  
(Practical Exercises Software Engineering: Methods and Tools)

Modulverantwortliche/r: Francesca Saglietti  
Lehrende: Xiaochen Wu, Loui Al Sardy

Startsemester: WS 2019/2020      Dauer: 1 Semester      Turnus: halbjährlich (WS+SS)  
Präsenzzeit: 30 Std.      Eigenstudium: 270 Std.      Sprache: Deutsch

**Lehrveranstaltungen:**

Software Engineering in der Praxis (WS 2019/2020, Übung, 3 SWS, Xiaochen Wu et al.)

**Inhalt:**

In den praktischen Übungen werden Werkzeuge zur Entwicklung und zur Analyse komplexer Software vorgestellt, deren industrielle Einsetzbarkeit anschließend von den Teilnehmern anhand für die Praxis repräsentativer Aufgabenstellungen erprobt wird.

**Lernziele und Kompetenzen:**

Die Studierenden

- erläutern Potenzial und Grenzen unterschiedlicher Werkzeuge zur Unterstützung softwaretechnischer Tätigkeiten;
- wenden unterschiedliche Werkzeuge an, um sowohl selbständig als auch in Teams Beispielaufgaben aus dem Bereich der objektorientierten Analyse, des objektorientierten Entwurfs, des Testens, des Beweisens und des Projektmanagements zu lösen;
- erklären erstellte Lösungen und skizzieren mögliche Alternativen.

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

**[1] Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Seminare, Praktika, Bachelorarbeit | Praktikum Informatik)

Dieses Modul ist daneben auch in den Studienfächern "Berufspädagogik Technik (Master of Education)" verwendbar.

**Studien-/Prüfungsleistungen:**

Praktikum Software Engineering: Verfahren und Werkzeuge\_ (Prüfungsnummer: 301346)

Prüfungsleistung, Praktikumsleistung, Dauer (in Minuten): 120

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Software Engineering in der Praxis

weitere Erläuterungen:

Die Praktikumsleistung besteht aus einer Prüfung am Rechner. Für die Zulassung zu dieser Prüfung ist die Abnahme von 10 Aufgabenblättern erforderlich, die im Rahmen des Praktikums zu bearbeiten sind.

Prüfungssprache: Deutsch

Erstablingung: WS 2019/2020, 1. Wdh.: SS 2020

1. Prüfer: Francesca Saglietti (100112)



---

**Modulbezeichnung:** **Praktikum angewandte Systemsoftwaretechnik (PASST)** **10 ECTS**  
 (Laboratory on Applied Systems Software Technology)

Modulverantwortliche/r: Jürgen Kleinöder

Lehrende: Tobias Langer, Michael Eischer, Florian Schmaus

---

Startsemester: WS 2019/2020	Dauer: 1 Semester	Turnus: halbjährlich (WS+SS)
Präsenzzeit: 150 Std.	Eigenstudium: 150 Std.	Sprache: Deutsch

---

**Lehrveranstaltungen:**

Praktikum angewandte Systemsoftwaretechnik (WS 2019/2020, Praktikum, Anwesenheitspflicht, Tobias Langer et al.)

---

**Vorhergehende Module:**

Systemprogrammierung

---

**Inhalt:**

Im Praktikum erlernen und verwenden die Studierenden verschiedene Techniken und Prozesse der systemnahen Softwareentwicklung im OpenSource-Umfeld. Nähere Informationen zum konkreten Inhalt in diesem Semester finden sich auf der unter "Weitere Informationen" verlinkten Veranstaltungsseite.

**Lernziele und Kompetenzen:**

*Fachkompetenz*

*Anwenden*

Die Studenten

- verstehen Kodierrichtlinien und sind in der Lage diese situationsgemäß anzuwenden

*Analysieren*

Die Studenten

- erkunden gegebenen Programmcode grossen Umfangs und hoher Komplexität
- bestimmen dessen Funktionalität und beschreiben und diskutieren dies

*Evaluiieren (Beurteilen)*

Die Studenten

- beurteilen Qualität, Korrektheit und Richtlinienkonformität fremder Programme
- analysieren in Softwaresystemen, insbesondere dem Linux-Kernel auftretende Fehler und Situationsbilder und können solche bewerten und auf ihr Vorkommen testen
- evaluieren und verwenden geeignete Mittel zur Erkennung der Fehlerursachen
- verifizieren die korrekte Behebung eines erkannten Fehlers
- beschreiben, bewerten und kritisieren das eigene und das Vorgehen Dritter bei der Programm-entwicklung, Fehlersuche und Integration

*Erschaffen*

Die Studenten

- planen und entwickeln Gerätetreiber für PCI- und USB-Geräte für das Betriebssystem Linux
- planen und entwickeln systemnahe Programme zur Interaktion mit Geräten im Zusammenspiel mit den entwickelten Gerätetreibern
- konzipieren, planen und entwickeln systemnahe Software, Systemsoftware oder Bestandteile eines Betriebssystemkerns; erstellen Dokumentation und präsentieren ihr Vorgehen
- erstellen geeignete Maßnahmen (Patches) zur Behebung erkannter Fehler und Probleme
- erzeugen textuelle Beschreibungen der Fehlerbedingungen, -symptome und -ursachen, der Fehlersuche sowie des Patches nach Kriterien relevanter Open-Source-Projekte in englischer Sprache, die geeignet sind, solche Patches und Beschreibungen an diese Open-Source-Projekte weiterzugeben
- entwickeln vorbeschriebene Patches und Beschreibungen und reichen diese bei relevanten Open-Source-Projekten ein
- interagieren mit den externen Entwicklern dieser Projekte mit dem Ziel eingereichtes Material akzeptiert zu bekommen

- erstellen Analysen und Präsentationen eigener und fremder Arbeit und tragen diese in geeigneter Weise vor einem Fachpublikum vor

#### *Lern- bzw. Methodenkompetenz*

Die Studenten

- verwenden gängige Werkzeuge der Softwareentwicklung im Open-Source-Bereich wie git, gdb, kgdb, qemu/kvm und cscope
- verstehen deren Funktionsweise
- verwenden diese erfolgreich in internen Aufgaben und Projekten sowie in der Interaktion mit externen Entwicklern
- interpretieren Code im Hinblick auf dessen vorgesehene Funktion, mögliche Fehler, sinnvolle Erweiterungspunkte und Qualitätsaspekte
- stellen technische, methodische und soziale Sachverhalte geeignet dar

#### *Selbstkompetenz*

Die Studenten

- sind in der Lage mit Kritik und Änderungswünschen umzugehen
- überwinden Berührungsängste im Kontakt mit externen Dritten
- bringen sich konstruktiv und produktiv in Open-Source-Projekte ein

#### *Sozialkompetenz*

Die Studenten

- organisieren selbständig die gemeinsame Bearbeitung der Übungsaufgaben und lösen diese kooperativ in kleinen Gruppen
- kommunizieren erfolgreich in englischer Sprache mit Betreuern und mit externen Entwicklern unter Einhaltung relevanter Protokolle im Open-Source-Umfeld
- gehen professionell mit Kritik an eigener Arbeit um und beziehen berechtigte Kritik in ihre zukünftige Arbeitsweise ein
- verhalten sich angemessen beim kritisieren fremder Arbeit gegenüber dem Ersteller dieser Arbeit oder Dritten
- erkennen und befolgen geschriebene und ungeschriebene Regeln im Umfeld relevanter Open-Source-Projekte; verhalten sich angemessen bei möglichen Konfliktsituationen

### **Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

#### **[1] Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Seminare, Praktika, Bachelorarbeit | Praktikum Informatik)

Dieses Modul ist daneben auch in den Studienfächern "Berufspädagogik Technik (Master of Education)" verwendbar.

### **Studien-/Prüfungsleistungen:**

Praktikum angewandte Systemsoftwaretechnik (Prüfungsnummer: 113845)

(englische Bezeichnung: Laboratory on Applied Systems Software Technology)

Prüfungsleistung, Praktikumsleistung

Anteil an der Berechnung der Modulnote: 100%

weitere Erläuterungen:

Es sind semesterbegeleitend sechs Übungsaufgaben zu bearbeiten. In der vorlesungsfreien Zeit findet ein Blockpraktikum statt (2 Wochen, Programmierung und zwei Vorträge a ca. 15 Minuten). Arbeit in 2er-Teams.

Übungsaufgaben, Programmierung im Blockpraktikum und Vorträge werden bepunktet. Die Note ergibt sich auf Basis der erreichten Punkte. Gewichtung Übungsaufgaben/Blockpraktikum 40/60.

Erstabelleung: WS 2019/2020, 1. Wdh.: keine Angabe

1. Prüfer: Jürgen Kleinöder (100163)

---

**Modulbezeichnung:** Supercomputing Praktikum (SuCoPra) 10 ECTS  
(Supercomputing Praktikum)

Modulverantwortliche/r: Alexander Ditter  
Lehrende: Alexander Ditter

---

Startsemester: WS 2019/2020	Dauer: 1 Semester	Turnus: jährlich (SS)
Präsenzzeit: 100 Std.	Eigenstudium: 200 Std.	Sprache: Deutsch oder Englisch

---

**Lehrveranstaltungen:**

Supercomputing Praktikum (WS 2019/2020, Praktikum, 8 SWS, Alexander Ditter)  
Supercomputing-Praktikum Übung (WS 2019/2020, Übung, Alexander Ditter)

---

**Inhalt:**

Im Rahmen des Supercomputing Praktikums werden die Studierenden auf die Teilnahme an der Student Cluster Competition (SCC) vorbereitet.

**Lernziele und Kompetenzen:**

*Fachkompetenz*

*Anwenden*

Die Funktionsweise verschiedener Cluster-Rechner kann verglichen, beschrieben und bewertet werden.

*Sozialkompetenz*

Die Studierenden lernen sich selbst in einem Team zu organisieren. Aufgaben, Verantwortlichkeiten und Arbeitsleistung müssen von den Studierenden zu einem großen Teil selbst verwaltet werden.

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

**[1] Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Seminare, Praktika, Bachelorarbeit | Praktikum Informatik)

Dieses Modul ist daneben auch in den Studienfächern "Berufspädagogik Technik (Master of Education)", "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Bachelor of Science)" verwendbar.

---

**Studien-/Prüfungsleistungen:**

Supercomputing Praktikum (Prüfungsnummer: 182798)

(englische Bezeichnung: Student Cluster Competition)

Prüfungsleistung, Praktikumsleistung

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Supercomputing Praktikum

weitere Erläuterungen:

Die Prüfungsleistung ergibt sich aus der Teilnahme am Praktikum.

Erstabwegung: WS 2019/2020, 1. Wdh.: SS 2020

1. Prüfer: Dietmar Fey (100101)

---

---

**Modulbezeichnung:** Bachelorarbeit (Bachelor's thesis)

**15 ECTS**

Modulverantwortliche/r: Die Dozenten der Informatik

---

Sprache: Deutsch

Dauer: 1 Semester

---

**Bemerkung:**

§ 42 FPO Informatik: Bachelorarbeit, mündliche Bachelorprüfung

(1) <sup>1</sup>Das Modul Bachelorarbeit umfasst 15 ECTS-Punkte. <sup>2</sup>Es besteht aus der schriftlichen Bachelorarbeit im Umfang von 12 ECTS-Punkten und dem „Begleitseminar mit Referat“ im Umfang von 3 ECTS-Punkten.

(2) <sup>1</sup>Die schriftliche Bachelorarbeit soll ein wissenschaftliches Thema aus dem Bereich der Informatik behandeln. <sup>2</sup>Das Thema für die schriftliche Bachelorarbeit ist in ihrer Anforderung so zu stellen, dass sie bei einer Bearbeitungszeit von ca. 360 Stunden abgeschlossen werden kann. <sup>3</sup>Die Bachelorarbeit wird von einer Hochschullehrerin oder einem Hochschullehrer der Informatik ausgegeben. <sup>4</sup>Ausnahmen hiervon kann der Prüfungsausschuss in begründeten Fällen genehmigen.

(3) <sup>1</sup>Das „Begleitseminar mit Referat“ umfasst ein Referat im Umfang von ca. 30 Minuten über das Thema der schriftlichen Bachelorarbeit und die erfolgreiche Teilnahme am von der Betreuerin oder dem Betreuer durchgeführten Begleitseminar. <sup>2</sup>Der Termin für das Referat wird von der Betreuerin oder dem Betreuer der Bachelorarbeit entweder nach Abgabe oder gegen Ende der Bachelorarbeit festgelegt.

---

---

**Modulbezeichnung:** **SWAT-Intensivübungen [als eBT-Aufbau] (SWAT)** **5 ECTS**  
 (SWAT is a Web Application Tutorial [eBT practical advanced training course])

Modulverantwortliche/r: Klaus Meyer-Wegener  
 Lehrende: Peter Schwab, Demian Vöhringer

---

Startsemester: SS 2019	Dauer: 1 Semester	Turnus: jährlich (SS)
Präsenzzeit: 115 Std.	Eigenstudium: 35 Std.	Sprache: Deutsch

---

**Lehrveranstaltungen:**

- SWAT: Tutorium (SS 2019, Übung, Peter Schwab et al.)
- SWAT: Blockpraktikum (SS 2019, Praktikum, Peter Schwab et al.)
- SWAT: Praktikum (SS 2019, Praktikum, Peter Schwab et al.)

---

**Inhalt:**

- Entwurf und Implementierung einer typischen Web-Applikation
- Kreatives Arbeiten im Team
- Agile Softwareentwicklung
- Verwendung von aktuellen Technologien
- Moderne Programmiertechniken

**Lernziele und Kompetenzen:**

Die Studierenden ...

- konzipieren und implementieren eine mehrschichtige Web-Anwendung.
- bewerten den Arbeitsaufwand von Aufgaben.
- wenden agile Entwicklungsmethoden im Rahmen von Softwareentwicklung an.
- arbeiten kooperativ und verantwortlich in Gruppen und können das eigene Kooperationsverhalten sowie die Zusammenarbeit in der Gruppe kritisch reflektieren und optimieren.
- arbeiten sich eigenständig in Technologien ein, stellen diese Technologien in Präsentationen vor und wenden sie im Projekt an.

**Literatur:**

- Elemental Design Patterns, Smith, 2012
- Patterns of Enterprise Application Architecture, Fowler, 2003
- Scrum mit User Stories, Wirdemann, 2011
- Agile Testing, Crispin and Gregory, 2009
- More Agile Testing, Crispin and Gregory, 2015

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

**[1] Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule)

Dieses Modul ist daneben auch in den Studienfächern "International Information Systems (IIS) (Master of Science)" verwendbar.

---

**Studien-/Prüfungsleistungen:**

SWAT-Intensivübung (Prüfungsnummer: 669768)

Prüfungsleistung, Portfolio

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- SWAT: Tutorium
- SWAT: Blockpraktikum
- SWAT: Praktikum

weitere Erläuterungen:

Die Bewertung der Prüfungsleistung setzt sich zusammen aus 30% Fachvortrag, 50% Praktikum und 20% mündliche Prüfung von etwa 20 Minuten.

Erstablingung: SS 2019, 1. Wdh.: keine Wdh.

1. Prüfer: Klaus Meyer-Wegener (100153)

---

### Organisatorisches:

**Es handelt sich um eine semesterbegleitende Veranstaltung mit einem abschließenden Block-tutorium!**

Alle Lehrveranstaltungen des Moduls müssen besucht werden.

Präsenzzeit während der Vorlesungszeit: 75h (10 x 1,5h Tutorien, 12 x 5h Bearbeitertage)

Eigenstudium und Vortragsvorbereitung: 35h

Blockpraktikum: 40h (5 x 8h Bearbeitertage)

Die Projektsprache ist Deutsch!

Die Anmeldung erfolgt über StudOn (siehe SWAT: Tutorium)

#### **Anwesenheit**

Aufgrund des praktikumsartigen Charakters der Lehrveranstaltung herrscht Anwesenheitspflicht. Versäumt die oder der Studierende nicht mehr als 15 v. H. der Unterrichtszeit aus Gründen, die sie oder er nachweislich nicht zu vertreten hat, so müssen von der oder dem Studierenden angemessene Ersatzleistungen erbracht werden. Werden mehr als 15 v. H. der Unterrichtszeit versäumt, so ist die Veranstaltung erneut zu belegen.

### Bemerkungen:

#### **Empfohlene Vorkenntnisse:**

Bachelor:

- Modul: "Algorithmen und Datenstrukturen" (wg. Objektorientierung)
- Modul: "Konzeptionelle Modellierung" (wg. Datenmodellierung und UML)
- Modul: "Softwareentwicklung in Großprojekten" (wg. Entwurfsmustern und IT-Vorgehensmodellen)
- Modul: "Systemprogrammierung" (wg. Betriebssystem-Architektur)
- Modul: "Rechnerkommunikation" (wg. Transferprotokollen)
- Modul: "Implementierung von Datenbanksystemen" (wg. Schichtenarchitektur, Transaktionen)

Master:

- Veranstaltung: eBusiness Technologies (wg. Scrum und RUP, Advanced XML, OOA&D crash course (Adv. UML), O/R-Mapping, Component Models, Web Basics, Web Services, Presentation Tier (MVC, AJAX, HTML5) )

---

**Modulbezeichnung:** eBusiness Technologies und Evolutionäre Informationssysteme (EBTEIS) **5 ECTS**  
 (eBusiness Technologies and Evolutionary Information Systems)

Modulverantwortliche/r: Richard Lenz

Lehrende: Christoph P. Neumann, Richard Lenz, Florian Irmert

---

Startsemester: WS 2019/2020

Dauer: 1 Semester

Turnus: jährlich (WS)

Präsenzzeit: 60 Std.

Eigenstudium: 90 Std.

Sprache: Deutsch

---

**Lehrveranstaltungen:**

Ausschlussbedingung: Dieses Modul darf nur abgelegt werden, wenn keine der im Modul enthaltenen Lehrveranstaltungen auch noch in einem anderen Modul enthalten ist, das bereits abgelegt wurde.

eBusiness Technologies (WS 2019/2020, Vorlesung, Christoph P. Neumann et al.)

Evolutionäre Informationssysteme (WS 2019/2020, Vorlesung, 2 SWS, Richard Lenz)

---

**Empfohlene Voraussetzungen:**

Programmieren in Java, Datenbanken (SQL)

**Vorhergehende Module:**

Konzeptionelle Modellierung

---

**Inhalt:**

siehe Lehrveranstaltungsbeschreibungen

**Lernziele und Kompetenzen:**

EBT:

Die Studierenden

- identifizieren die wichtigsten Themen des Bereichs eBusiness, von den Anwendungen bis zu den Implementierungen
- verstehen Zusammenhänge der B2B-Integration und der Realisierung von eBusiness-Anwendungen
- wiederholen Grundlagen des Webs
- vergleichen technische Eigenschaften von HTTP-, Web- und Application Servern
- vergleichen Markup Languages (HTML, XML)
- unterscheiden Ansätze zur Schema-Modellierung wie DTD und XML Schema und erkennen die unterschiedliche Leistungsfähigkeit
- verstehen Methoden zur evolutionsfähigen Gestaltung von Datenstrukturen in XML
- unterscheiden Vorgehen bei der Datenhaltung und verschiedene Ansätze für den Datenbankzugriff
- verstehen Objekt-relationale Mapping Frameworks am Beispiel von Hibernate und JPA
- verstehen Komponentenmodelle wie Enterprise JavaBeans (EJB) aus dem JEE Framework
- unterscheiden das EJB Komponentenmodell von den OSGi Bundles und den Spring Beans
- verstehen und unterscheiden grundlegende Web Service Techniken wie SOAP und WSDL
- unterscheiden Herangehensweisen zur dynamischen Generierung von Webseiten
- verstehen grundlegende Eigenschaften eines Java-basierten Front-End-Frameworks am Beispiel von JSF
- verstehen grundlegende Eigenschaften von Service-orientierten Architekturen (SOA)
- verstehen agile Vorgehensmodelle zur Software-Entwicklung am Beispiel von Scrum
- unterscheiden agile Verfahren wie Scrum von iterativ-inkrementellen Verfahren wie RUP
- verstehen die Wichtigkeit von Code-Beispielen um die praktische Anwendbarkeit des theoretischen Wissens zu veranschaulichen.
- können die Code-Beispiele eigenständig zur Ausführung bringen und die praktischen Erfahrungen interpretieren und bewerten
- gestalten eigene Lernprozesse selbständig.
- schätzen ihre eigenen Stärken und Schwächen im Hinblick auf die unterschiedlichen Architektur-Schichten ein (Benutzerinteraktion, Applikationslogik, Schnittstellenintegration, Datenbanksysteme)
- identifizieren eine eigene Vorstellung als zukünftige Software-Architekten und können die eigene Entwicklung planen

- reflektieren durch regelmäßige fachbezogene Fragen des Dozenten Ihre eigene Lösungskompetenz.

EIS:

Die Studierenden:

- definieren die Begriffe "Informationssysteme", "evolutionäre Informationssysteme" und "organisatorisches Lernen"
- grenzen die Begriffe "Wissen" und "Information" gegeneinander ab
- charakterisieren die in der Vorlesung erläuterten Formen der organisatorischen Veränderung
- erklären das SEKI Modell nach Nonaka und Takeuchi
- nennen Beispiele für die in der Vorlesung behandelten Formen der Wissensrepräsentation in IT-Systemen
- nennen typische Erfolgs- und Risikofaktoren für große IT-Projekte
- erklären die Kraftfeldtheorie nach Kurt Lewin
- unterscheiden Typen von Software gemäß der Klassifikation nach Lehman und Belady
- unterscheiden die in der Vorlesung vorgestellten Arten der Software Wartung
- benennen die Gesetzmäßigkeiten der Software-Evolution nach Lehman und Belady
- bewerten die in der Vorlesung vorgestellten Vorgehensmodelle zur Softwareerstellung im Kontext der E-Typ-Software
- nennen die in der Vorlesung vorgestellten Aspekte der Evolutionsfähigkeit von Software
- erklären, wie die in der Vorlesung vorgestellten Methoden zur Trennung von Belangen beitragen
- erklären das Konzept des "Verzögerten Entwurfs"
- erklären die Vor- und Nachteile generischer Datenbankschemata am Beispiel von EAV und EAV/CR
- charakterisieren die in der Vorlesung vorgestellten Architekturkonzepte
- grenzen die in der Vorlesung vorgestellten Integrationsanforderungen gegeneinander ab
- erklären wie Standards zur Systemintegration beitragen und wo die Grenzen der Standardisierung liegen
- erklären das Prinzip eines Kommunikationsservers und der nachrichtenbasierten Integration
- erklären den Begriff "Prozessintegration"
- definieren den Begriff "Enterprise Application Integration" (EAI)
- unterscheiden die in der Vorlesung vorgestellten Integrationsansätze
- erklären die in der Vorlesung vorgestellten Dimensionen der Datenqualität
- unterscheiden die grundlegenden Messmethoden für Datenqualität
- erklären das Maßnahmenportfolio zur Verbesserung der Datenqualität nach Redman
- benennen die in der Vorlesung vorgestellten Methoden zur Verbesserung der Datenqualität

#### Literatur:

siehe Lehrveranstaltungsbeschreibungen

#### Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

##### [1] Informatik (Bachelor of Science)

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule | Vertiefungsrichtung Datenbanksysteme)

Dieses Modul ist daneben auch in den Studienfächern "Informatik (Master of Science)", "International Information Systems (IIS) (Master of Science)", "Mathematik (Bachelor of Science)", "Medizintechnik (Master of Science)" verwendbar.

#### Studien-/Prüfungsleistungen:

eBusiness Technologies und Evolutionäre Informationssysteme (Prüfungsnummer: 710850)

Prüfungsleistung, mündliche Prüfung, Dauer (in Minuten): 30

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- eBusiness Technologies
- Evolutionäre Informationssysteme

Erstablingung: WS 2019/2020, 1. Wdh.: SS 2020 (nur für Wiederholer)



1. Prüfer: Richard Lenz (100185)

---

---

**Modulbezeichnung:** Dienstgüte von Kommunikationssystemen (DKS) 5 ECTS  
 (Quality of Service of Communication Systems)

Modulverantwortliche/r: Reinhard German  
 Lehrende: Reinhard German

---

Startsemester: SS 2019	Dauer: 1 Semester	Turnus: jährlich (SS)
Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Englisch

---

**Lehrveranstaltungen:**

Quality of Service of Communication Systems / Dienstgüte von Kommunikationssystemen (SS 2019, Vorlesung, 2 SWS, Kai-Steffen Jens Hielscher)  
 Dienstgüte von Kommunikationssystemen Übungen (SS 2019, Übung, 2 SWS, Lisa Maile)

---

**Empfohlene Voraussetzungen:**

Rechnerkommunikation, Kommunikationssysteme, Programmierkenntnisse in Java

---

**Inhalt:**

Zunächst wird der Begriff der Dienstgüte (Quality-of-Service, QoS) eingegrenzt und es werden die wichtigsten Ansätze zur Erzielung von Dienstgüte besprochen und in ausgewählten Netztechnologien untersucht. Dann werden unterschiedliche Methoden vorgestellt, mit denen Systeme bezüglich ihrer Dienstgüte bewertet und ausgelegt werden können:

- Netzplanung und -optimierung,
- stochastische Analyse (Markow-Ketten, Warteschlangen),
- Netzwerksimulation,
- deterministische Analyse mit Network Calculus zur Ermittlung von Dienstgütegarantien
- Messung (HW-, SW-, Hybrid-Monitoring, Benchmarks).

Alle Methoden werden an Beispielen demonstriert.

Contents:

We introduce the term quality-of-service (QoS), discuss important approaches to achieve certain degrees of QoS, and show how the implementation in computer networks. Then a number of methodologies to assess and design systems with respect to their QoS:

- network planning and optimization,
- network simulation,
- stochastic analysis (Markov chains, non-Markovian models, queuing systems),
- deterministic analysis with network calculus to determine QoS guarantees
- measurements (hardware, software, and hybrid monitoring, benchmarks).

All methods are illustrated by examples.

**Lernziele und Kompetenzen:**

Die Studierenden erwerben

- Kenntnisse in Methoden zur Modellierung und Bewertung von quantitativen, nicht-funktionalen Eigenschaften von vernetzten Systemen
- Kenntnisse in Mechanismen von vernetzten Systemen zur Erzielung von Dienstgüte

Learning targets and competences:

The students get

- experience in methods to model and evaluate quantitative, non-functional properties of computer networks and related systems
- knowledge of mechanisms of computer networks to achieve quality-of-service

**Literatur:**

- Kurose, Ross. Computer Networking: A Top-Down Approach Featuring the Internet. 6th Ed., Addison Wesley, 2013
- W. Stallings. Data and Computer Communications, 10th ed., Pearson Education, 2014
- W. Stallings. Foundations of Modern Networking: SDN, NFV, QoE, IoT, and Cloud, Pearson Education, 2016

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

**[1] Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule)

Dieses Modul ist daneben auch in den Studienfächern "Information and Communication Technology (Master of Science)", "Informations- und Kommunikationstechnik (Master of Science)", "International Information Systems (IIS) (Master of Science)" verwendbar.

---

**Studien-/Prüfungsleistungen:**

Dienstgüte von Kommunikationssystemen (Vorlesung und Übung) (Prüfungsnummer: 472330)

(englische Bezeichnung: Quality of Service of Communication Systems (Lecture and Exercise))

Prüfungsleistung, Portfolio

weitere Erläuterungen:

Die Modulprüfung besteht aus:

- unbenotete Studienleistung, zu erwerben durch erfolgreiche Teilnahme an den Übungen
- Klausur von 90 Minuten oder mündliche Prüfung bei weniger als 20 Teilnehmern

Erstablingung: SS 2019, 1. Wdh.: WS 2019/2020

1. Prüfer: Kai-Steffen Jens Hielscher (100125)

---

**Organisatorisches:**

Alles Material ist in Englisch, die Vorlesung und die Übungen werden in Englisch gehalten.

---

**Modulbezeichnung:** Simulation und Modellierung 1 - VÜ (SaM 1-VÜ) **5 ECTS**  
 (Simulation and Modeling 1 - L+E)

Modulverantwortliche/r: Reinhard German  
 Lehrende: Reinhard German

---

Startsemester: WS 2019/2020	Dauer: 1 Semester	Turnus: jährlich (WS)
Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Deutsch oder Englisch

---

**Lehrveranstaltungen:**

Simulation and Modeling 1 (WS 2019/2020, Vorlesung, 2 SWS, Reinhard German)  
 Exercises to Simulation and Modeling 1 (WS 2019/2020, Übung, 2 SWS, Lisa Maile et al.)

---

**Empfohlene Voraussetzungen:**

elementare Programmierkenntnisse, vorzugsweise in Java, Mathematikkennnisse in Analysis, wie z.B. im 1. Semester der angewandten Mathematik vermittelt

Recommended background knowledge:

basic programming skills, preferably in Java, mathematics skills in analysis, such as taught in the first semester in applied mathematics.

---

**Inhalt:**

Das Modul vermittelt die Grundlagen der diskreten Ereignissimulation und beinhaltet

- diskrete Simulation
- analytische Modellierung (z.B. Warteschlangen)
- Eingabemodellierung (z.B. Fitting-Verfahren)
- Zufallszahlenerzeugung
- statistische Ausgabeanalyse
- Modellierungsparadigmen (u.a. Ereignis-/Prozessorientierung, Warteschlangen, Automaten, Petri-Netze, UML, graphische Bausteine)
- kontinuierliche und hybride Simulation
- Simulationssoftware
- Fallstudien

Content:

Overview of the various kinds of simulation

- discrete simulation (computational concepts, simulation of queuing systems, simulation in Java, professional simulation tools)
- required probability concepts and statistics, modeling paradigms (e.g., event/process oriented, queuing systems, Petri nets, UML statecharts)
- input modeling (selecting input probability distributions)
- random number generation (linear congruential generators and variants, generating random variates)
- output analysis (warm-up period detection, independent replications, result presentation)
- continuous and hybrid simulation (differential equations, numerical solution, hybrid statecharts)
- simulation software, case studies, parallel and distributed simulation.

**Lernziele und Kompetenzen:**

Die Studierenden

- erwerben Kenntnisse über Verfahren und Realisierungsmöglichkeiten der diskreten Simulation mit Ausblick auf andere Simulationsarten
- erwerben Kenntnisse über statistische Aspekte der Simulation, die für die Anwendung wichtig sind
- wenden statistische Methoden zur Analyse und Bewertung von Eingabe- sowie Ausgabedaten an
- erwerben praktische Erfahrung mit kommerziellen Simulationswerkzeugen
- erwerben Erfahrungen bei der Simulation in verschiedenen Anwendungsbereichen (u.a. Rechnernetze, Fertigungssysteme, Materialflusssysteme)
- entwickeln eigenständig anhand von Beispielaufgaben Simulationsmodelle unter Verwendung verschiedener Modellierungsparadigmen
- können in Gruppen kooperativ und verantwortlich arbeiten

Learning targets and competences:

Students

- gain knowledge about methods and realization possibilities of discrete simulation with an outlook on other types of simulation
- gain knowledge of statistical aspects of simulation that are important for practice
- apply statistical methods for analysis and evaluation of input and output data
- gain hands-on experience with commercial simulation tools
- gain experience in simulation in various fields of application (including computer networks, manufacturing systems, material flow systems)
- independently develop simulation models on the basis of sample tasks using different modeling paradigms
- can work in groups cooperatively and responsibly

#### Literatur:

Law, "Simulation Modeling and Analysis", 5th ed., McGraw Hill, 2014

#### Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

##### [1] Informatik (Bachelor of Science)

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule | Vertiefungsrichtung Diskrete Simulation)

Dieses Modul ist daneben auch in den Studienfächern "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Bachelor of Science)", "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Master of Science)", "Informatik (Master of Science)", "Information and Communication Technology (Master of Science)", "Informations- und Kommunikationstechnik (Master of Science)", "International Information Systems (IIS) (Master of Science)", "Internationales Projektmanagement Großanlagenbau/International Project Management in Systems Engineering (Master of Science)", "Maschinenbau (Bachelor of Science)", "Maschinenbau (Master of Science)", "Mathematik (Bachelor of Science)", "Mechatronik (Master of Science)", "Medizintechnik (Bachelor of Science)" verwendbar.

#### Studien-/Prüfungsleistungen:

Simulation und Modellierung 1 (Vorlesung mit Übungen) (Prüfungsnummer: 70901)

(englische Bezeichnung: Simulation and Modeling 1 (with Exercises))

Prüfungsleistung, Klausur, Dauer (in Minuten): 90

Anteil an der Berechnung der Modulnote: 100%

weitere Erläuterungen:

Die Modulprüfung besteht aus:

- unbenotete Studienleistung, zu erwerben durch erfolgreiche Teilnahme an den Übungen
- Klausur von 90 Minuten oder mündliche Prüfung bei weniger als 20 Teilnehmern

Erstabelleung: WS 2019/2020, 1. Wdh.: WS 2020/2021

1. Prüfer: Reinhard German (100165)

---

**Modulbezeichnung:** **Satellitenkommunikation (SATKOM)** **5 ECTS**  
(Satellite Communications)

Modulverantwortliche/r: Albert Heuberger  
Lehrende: Christian Rohde

---

Startsemester: SS 2019	Dauer: 1 Semester	Turnus: jährlich (SS)
Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Deutsch

---

**Lehrveranstaltungen:**

Satellitenkommunikation (SS 2019, Vorlesung, 2 SWS, Christian Rohde)  
Übung Satellitenkommunikation (SS 2019, Übung, 2 SWS, Jochen Martin-Creuzburg)

---

**Empfohlene Voraussetzungen:**

Keine formalen Voraussetzungen

---

**Inhalt:**

Nach einem historischen Rückblick zur Entwicklung der Satellitenkommunikation werden die einzelnen Komponenten eines typischen Gesamtsystems (Boden- und Raumsegment) näher betrachtet. Hierzu zählt der prinzipielle Aufbau von Trägerraketen (am Beispiel Ariane 5), von Satelliten (Satellitenplattformen, Subsysteme, Nutzlasten), die meist genutzten Umlaufbahnen und die Kommunikationslinks zwischen Bodenstation und Satellit (Uplink) und Satellit und Empfänger (Downlink). Die Besonderheiten der Signalausbreitung und -übertragung über diese großen Entfernungen (verwendete Frequenzen, Signaldispersion, -dämpfung, weitere Störeinflüsse der Weltraumumgebung) werden erklärt und in Beispielen berechnet. Die Architektur transparenter und regenerativer Kommunikationseinheiten wird ausführlich an Beispielen kommerziell verfügbarer Transponder und Onboard-Prozessoren erklärt. Die Prinzipien moderner, standardisierter Verfahren zur Signalaufbereitung und Übertragung von Video-/Bild und Audiosignalen über Satellit (MPEG, DVB-S/-S2 Quellencodierung, Kanalcodierung und Modulation, Kanalzugriff) werden erläutert und diskutiert. Ein Exkurs auf die im Orbit und im kommerziellen Einsatz befindlichen Kommunikationssatelliten für TV- und Breitbandversorgung, von Mobilkommunikationssystemen und eine Vorstellung von Forschungsansätzen für zukünftige Satellitensysteme (Megakonstellationen, Next generation High Throughput Satellites (HTS) sowie die große Dienstvielfalt mithilfe von Satelliten runden die Vorlesung ab. Die in der Vorlesung behandelten physikalischen, elektro- und nachrichtentechnischen Zusammenhänge werden in den ergänzenden Übungen mit Rechenbeispielen vertieft.

**Gliederung der Vorlesung:**

1. Einführung
  - Hauptkomponenten (Space Segment, Ground Segment), Überblick: Satelliten, Launcher, Orbits, Aufgaben, genutzte Frequenzen, Standarddienste, Satellitennetzwerke der größten Satellitenbetreiber
2. Entwicklung der Satellitenkommunikation
  - Meilensteine, Organisationen, Nationale Programme
3. Orbits und Konstellationen
  - Keplersche Gesetze, Beschreibung von Umlaufbahnen, genutzte Orbits, Bodenspuren, erreichbare Abdeckung
4. Trägersysteme
  - Trägerraketen, Entwicklung, Anbietermarkt, Nutzlastfähigkeit, Startplätze, Startverlauf
5. Satellitenaufbau
  - Auswahl aktueller Satellitenplattformen, Satellitenaufbau, Plattformkomponenten, Montageschritte und Tests
6. Satellitennutzlast (Payload)
  - Komponenten, Industrielle Beispiele, Aufbau und Aufgaben der Payload, Transponderarchitekturen, Antennen
7. Signalausbreitung und Leistungsbilanz

- Signalausbreitung, Freiraumverluste, Signaldämpfung, Rauschen, Signal-Rausch-Verhältnis, Linkbudget
- 8. Weltraumumgebung
  - Weltraumumgebungsbedingungen, Einflüsse auf den Satelliten und die Elektronik der Nutzlast (SEE, TID, DD)
- 9., 10., 11. Quellencodierung, Kanalcodierung, Modulationsverfahren, Signalübertragung am Beispiel DVB-S/-S2
  - Audio-, Bild- und Videokompression, FEC, Interleaver, Modulation, Zugriffs-, Duplex-, Multiplexverfahren
- 12. Vielfalt der Kommunikationsdienste, Beispiele moderner, zellulärer Satellitenkommunikationssysteme
  - Iridium, Globalstar, Inmarsat, Thuraya, Orbcom, O3b, ...), kommende Megakonstellationen (OneWeb, SpaceX, ...)
- 13. Neue Märkte, neue Dienste

#### **Lernziele und Kompetenzen:**

Nach der Teilnahme an der Vorlesung "Satellitenkommunikation" verstehen die Studierenden die Funktionsweise moderner satellitengestützter Kommunikationssysteme und wissen um deren Bedeutung für die großflächige und breitbandige Verbreitung von Bild-, Video- und Audiosignalen einschließlich einer Vielfalt an Datendiensten über Satellit. Die theoretischen Grundlagen und Rechenübungen anhand konkreter Beispiele erlauben die prinzipielle Berechnung und Überprüfung grundlegender Leistungsdaten und Systemkenngrößen.

#### **Literatur:**

Skriptum zur Lehrveranstaltung

#### **Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

##### **[1] Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule)

Dieses Modul ist daneben auch in den Studienfächern "Berufspädagogik Technik (Bachelor of Science)", "Berufspädagogik Technik (Master of Education)", "Computational Engineering (Master of Science with Honours)", "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Master of Science)", "Elektrotechnik, Elektronik und Informationstechnik (Bachelor of Science)", "Elektrotechnik, Elektronik und Informationstechnik (Master of Science)", "Informatik (Master of Science)", "Information and Communication Technology (Master of Science)", "Informations- und Kommunikationstechnik (Bachelor of Science)", "Informations- und Kommunikationstechnik (Master of Science)", "Mathematik (Bachelor of Science)", "Wirtschaftsingenieurwesen (Master of Science)" verwendbar.

#### **Studien-/Prüfungsleistungen:**

Satellitenkommunikation (Prüfungsnummer: 34601)

(englische Bezeichnung: Satellite Communication)

Prüfungsleistung, Klausur, Dauer (in Minuten): 90

Anteil an der Berechnung der Modulnote: 100%

Erstablesung: SS 2019, 1. Wdh.: WS 2019/2020

1. Prüfer: Albert Heuberger (100228)

---

**Modulbezeichnung:** Globale Navigationssatellitensysteme (GNSS-V) 5 ECTS  
(Global Navigation Satellite Systems)

Modulverantwortliche/r: Jörn Thielecke  
Lehrende: Jörn Thielecke

---

Startsemester: WS 2019/2020	Dauer: 1 Semester	Turnus: jährlich (WS)
Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Deutsch

---

**Lehrveranstaltungen:**

Globale Navigationssatellitensysteme (WS 2019/2020, Vorlesung, 3 SWS, Jörn Thielecke)  
Übung zu Globale Navigationssatellitensysteme (WS 2019/2020, Übung, 1 SWS, Jörn Thielecke et al.)

---

**Empfohlene Voraussetzungen:**

Keine formalen Voraussetzungen, grundlegende Kenntnisse erforderlich in: linearer Algebra, Signal- & Systemtheorie, Wahrscheinlichkeitstheorie

---

**Inhalt:**

**Hinweis:**

1. Mehrere Übungsstunden werden rechnergestützt (MATLAB) sein, um den Vorlesungsstoff durch eigene praktische Erfahrung zu vertiefen.
2. Eine Laborbesichtigung beim Fraunhofer Institut für Integrierte Schaltungen soll Ihnen Einblick in die einschlägigen Arbeiten zu GPS und Galileo geben.

**Inhalte:**

**1. Überblick: Signale und Systeme**

- Einführung
- GPS - Global Positioning System
- Galileo
- Satellitenergänzungssysteme: EGNOS, WAAS, LAAS
- Mathematische Grundlagen: Navigationssignale, Gold Codes, Cramer-Rao-Schranke für Laufzeitmessungen

**2. Grundlagen und Funktionsweise der Satellitenortung**

- Koordinatensysteme, Zeitsysteme und Orbits
- Ausbreitungsbedingungen und Fehlerquellen
- Positions-, Geschwindigkeits- und Zeitschätzung
- Hochgenaue Positionsschätzung mittels Trägerphase

**3. GNSS Empfänger**

- Signalkonditionierung
- Leistungsfähigkeit der GPS- und Galileo-Signale
- Releschleifen zur Signalverfolgung

**Lernziele und Kompetenzen:**

1. Eine Übersicht über die Möglichkeiten von GPS und Galileo soll Ihre Beurteilungsfähigkeit für neue Anwendungen schärfen.
2. Durch vertiefte Kenntnisse der Grundlagen, Funktionsweise und Fehlerquellen sollen Sie die gelösten Herausforderungen und die Grenzen von GPS und Galileo einschätzen lernen.
3. Sie sollen ein nachrichtentechnisches Verständnis für die Funktionsweise eines GPS-Empfängers erlangen.

**Literatur:**

1. Pratap Misra, Per Enge, „Global Positioning System“, Ganga-Jamuna Press, 2001
  2. E.D. Kaplan, C.J. Hegarty, „Understanding GPS - Principles and Applications“ Artech House, 2. Auflage, 2006
  3. Werner Mansfeld, "Satellitenortung und Navigation", Vieweg, 2004
-



## Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

### [1] Informatik (Bachelor of Science)

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule | Vertiefungsrichtung Elektronik und Informationstechnik)

Dieses Modul ist daneben auch in den Studienfächern "Berufspädagogik Technik (Bachelor of Science)", "Berufspädagogik Technik (Master of Education)", "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Master of Science)", "Elektrotechnik, Elektronik und Informationstechnik (Bachelor of Science)", "Elektrotechnik, Elektronik und Informationstechnik (Master of Science)", "Informatik (Master of Science)", "Informations- und Kommunikationstechnik (Bachelor of Science)", "Informations- und Kommunikationstechnik (Master of Science)", "Mathematik (Bachelor of Science)" verwendbar.

---

## Studien-/Prüfungsleistungen:

Globale Navigationssatellitensysteme (Prüfungsnummer: 64011)

(englische Bezeichnung: Global Navigation Satellite Systems)

Prüfungsleistung, Klausur, Dauer (in Minuten): 90

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Globale Navigationssatellitensysteme
- Übung zu Globale Navigationssatellitensysteme

weitere Erläuterungen:

Klausurergebnis: 100% der Modulnote Bei bestandener Prüfung wird die Note um eine Teilnotenstufe (z.B. von 2,0 auf 1,7) verbessert, wenn Sie:

1. Mindestens 75% der Hausaufgaben bestanden haben und 2. mindestens 75% der Rechnerübungen erfolgreich absolviert haben.

Eine Note besser als 1,0 wird nicht vergeben.

Die Bearbeitungszeit beträgt 90 Minuten, den genauen Abgabetermin entnehmen Sie dem Tafelanschrieb. Zugelassene Hilfsmittel sind ausschließlich Schreib- und Zeichengeräte sowie ein nicht programmierbarer Taschenrechner ohne Kommunikationsfunktion. Ein Bonus wird nur berücksichtigt, wenn die Prüfung auch ohne ihn bestanden ist. Für die Vergabe sind die Regeln der einschlägigen Modulbeschreibung maßgeblich. Überprüfen Sie bitte die Vollständigkeit der ausgegebenen Aufgaben und Hilfsblätter. Die Anzahl ist auf dem jeweiligen Deckblatt angegeben. Bitte verwenden Sie ausschließlich das ausgeteilte Papier. Lösungen, die auf anderem Papier geschrieben wurden, können nicht gewertet werden. Weiteres Papier kann bei der Prüfungsaufsicht angefordert werden. Bearbeiten Sie jede Aufgabe auf einem eigenen Bogen. Schreiben Sie nicht mit Bleistift. Verwenden Sie nicht die Farben Rot oder Grün. Tragen Sie bitte auf allen Lösungsblättern Name und Aufgabennummer ein und unterschreiben Sie den Prüfungsbogen an der dafür vorgesehenen Stelle. Legen Sie bitte während der Prüfung Ihren Lichtbildausweis griffbereit auf den Tisch. Legen Sie am Ende der Klausur Ihre Lösungsblätter und die Hilfsblätter in diesen Mantelbogen, damit alles zusammengeheftet werden kann.

Prüfungssprache: Deutsch

Erstablingung: WS 2019/2020, 1. Wdh.: SS 2020

1. Prüfer: Jörn Thielecke (100224)

---

## Organisatorisches:

Masterstudium, kann ins Bachelorstudium vorgezogen werden. (Wahl- oder Wahlpflichtfach)

---

**Modulbezeichnung: Kommunikationsstrukturen (KOST)** **5 ECTS**  
(Communication Structures)

Modulverantwortliche/r: Jürgen Frickel  
Lehrende: Jürgen Frickel

---

Startsemester: WS 2019/2020	Dauer: 1 Semester	Turnus: jährlich (WS)
Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Deutsch

---

**Lehrveranstaltungen:**

Kommunikationsstrukturen (WS 2019/2020, Vorlesung, 2 SWS, Jürgen Frickel)  
Übungen zu Kommunikationsstrukturen (WS 2019/2020, Übung, 2 SWS, Jürgen Frickel)

---

**Inhalt:**

**Einführung**

- Information und Kommunikation
- Anwendungsgebiete - Kommunikation

**Strukturen und Eigenschaften von Kommunikationssystemen**

- Grundlegende Definitionen und Klassifikationen
- Grundlegende Strukturen

**Protokolle und Schnittstellen**

- Grundlagen
- Basis-Verfahren und Beispiele
- TCP/IP-Protokol
- Referenzmodell nach ISO/OSI
- Sicherungsschicht/Data Link Layer (LLC und MAC)
- Bitübertragungsschicht/Physical Layer
- Übertragungsmedien

**Hardware in Kommunikationsstrukturen**

- HW-Architekturen und Funktionsblöcke
- Digitale und Analoge Komponenten
- Schaltungsdetails von Komponenten

**Grundlagen von Bussystemen**

- Klassifikation
- Funktionale Eigenschaften
- Arbitrierungs-Verfahren

**Leitungsgebundene Anwendungen für Rechnersysteme**

- Bus-Applikationen
- *Baustein-/IC-interne Busse (AMBA, FPI, ConTraBus, ...)*
- *Baugruppeninterne Busse (I2C, Chipsätze+ Bridges, ...)*
- *Busse für Rechensysteme (VME, ISA, PCI, PCIe, AGP, ...)*
- *Peripherie-Busse (ATA, IEC, USB, Firewire, Fibre Channel, Thunderbolt ...)*

**Leitungsgebundene Anwendungen in Systemen**

- Feldkommunikation
- *Automobil, Luftfahrt, Space (CAN, MOST, LIN, MILBus, Spacewire ...)*
- *Industrie, Haustechnik (Profibus, EIB, ...)*
- Weitverkehrsnetze
- *SDH, PDH, ATM, ...*

**Lernziele und Kompetenzen:**

1. Die Studierenden werden in die Lage versetzt die Konzepte und Verfahren vor allem drahtgebundener Kommunikationssysteme anzuwenden.
2. Die Studierenden lernen die Funktionsweise und den Einsatzzweck diverser Kommunikationsprotokolle zu verstehen, und miteinander zu vergleichen.
3. Desweiteren analysieren und klassifizieren Sie grundlegende Strukturen von leitungsgebundenen Kommunikationssystemen anhand ihrer funktionalen Eigenschaften.

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

**[1] Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule | Vertiefungsrichtung Elektronik und Informationstechnik)

Dieses Modul ist daneben auch in den Studienfächern "Berufspädagogik Technik (Bachelor of Science)", "Berufspädagogik Technik (Master of Education)", "Elektrotechnik, Elektronik und Informationstechnik (Bachelor of Science)", "Elektrotechnik, Elektronik und Informationstechnik (Master of Science)", "Informatik (Master of Science)", "Information and Communication Technology (Master of Science)", "Informations- und Kommunikationstechnik (Master of Science)", "Mathematik (Bachelor of Science)", "Medizintechnik (Bachelor of Science)", "Medizintechnik (Master of Science)", "Wirtschaftsingenieurwesen (Bachelor of Science)", "Wirtschaftsingenieurwesen (Master of Science)" verwendbar.

---

**Studien-/Prüfungsleistungen:**

Kommunikationsstrukturen (Prüfungsnummer: 68011)

Prüfungsleistung, Klausur, Dauer (in Minuten): 90

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Kommunikationsstrukturen
- Übungen zu Kommunikationsstrukturen

Erstablingung: WS 2019/2020, 1. Wdh.: SS 2020

1. Prüfer: Jürgen Frickel (100261)

---

**Bemerkungen:**

Vorlesung für Lehramtstudenten: 2 SWS

**Modulbezeichnung:** Satellitengestützte Ortsbestimmung (SatNav) 5 ECTS  
(Global Navigation Satellite Systems)

Modulverantwortliche/r: Jörn Thielecke  
Lehrende: Jörn Thielecke

Startsemester: WS 2019/2020      Dauer: 1 Semester      Turnus: jährlich (WS)  
Präsenzzeit: 60 Std.      Eigenstudium: 90 Std.      Sprache: Deutsch

**Lehrveranstaltungen:**

Globale Navigationssatellitensysteme (WS 2019/2020, Vorlesung, 3 SWS, Jörn Thielecke)  
Übung zu Globale Navigationssatellitensysteme (WS 2019/2020, Übung, 1 SWS, Jörn Thielecke et al.)

**Empfohlene Voraussetzungen:**

Keine formalen Voraussetzungen, grundlegende Kenntnisse erforderlich in: linearer Algebra, Signal- & Systemtheorie, Wahrscheinlichkeitstheorie

**Inhalt:**

**Hinweis:**

1. Mehrere Übungsstunden werden rechnergestützt (MATLAB) sein, um den Vorlesungsstoff durch eigene praktische Erfahrung zu vertiefen.
2. Eine Laborbesichtigung beim Fraunhofer Institut für Integrierte Schaltungen soll Ihnen Einblick in die einschlägigen Arbeiten zu GPS und Galileo geben.

**Inhalte:**

**1. Überblick: Signale und Systeme**

- Einführung
- GPS - Global Positioning System
- Galileo
- Satellitenergänzungssysteme: EGNOS, WAAS, LAAS
- Mathematische Grundlagen: Navigationssignale, Gold Codes, Cramer-Rao-Schranke für Laufzeitmessungen

**2. Grundlagen und Funktionsweise der Satellitenortung**

- Koordinatensysteme, Zeitsysteme und Orbits
- Ausbreitungsbedingungen und Fehlerquellen
- Positions-, Geschwindigkeits- und Zeitschätzung
- Hochgenaue Positionsschätzung mittels Trägerphase

**3. GNSS Empfänger**

- Signalkonditionierung
- Leistungsfähigkeit der GPS- und Galileo-Signale
- Releschleifen zur Signalverfolgung

**Lernziele und Kompetenzen:**

1. Eine Übersicht über die Möglichkeiten von GPS und Galileo soll Ihre Beurteilungsfähigkeit für neue Anwendungen schärfen.
2. Durch vertiefte Kenntnisse der Grundlagen, Funktionsweise und Fehlerquellen sollen Sie die gelösten Herausforderungen und die Grenzen von GPS und Galileo einschätzen lernen.
3. Sie sollen ein nachrichtentechnisches Verständnis für die Funktionsweise eines GPS-Empfängers erlangen.

**Literatur:**

1. Pratap Misra, Per Enge, „Global Positioning System“, Ganga-Jamuna Press, 2001
2. E.D. Kaplan, C.J. Hegarty, „Understanding GPS - Principles and Applications“ Artech House, 2. Auflage, 2006
3. Werner Mansfeld, "Satellitenortung und Navigation", Vieweg, 2004

### **Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

#### **[1] Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule | Vertiefungsrichtung Elektronik und Informationstechnik)

Dieses Modul ist daneben auch in den Studienfächern "Informatik (Master of Science)", "Mathematik (Bachelor of Science)" verwendbar.

---

### **Studien-/Prüfungsleistungen:**

Satellitengestützte Ortsbestimmung\_ (Prüfungsnummer: 652213)

(englische Bezeichnung: Global Navigation Satellite Systems\_)

Prüfungsleistung, Klausur, Dauer (in Minuten): 90

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Globale Navigationssatellitensysteme
- Übung zu Globale Navigationssatellitensysteme

weitere Erläuterungen:

Klausurergebnis: 100% der Modulnote Bei bestandener Prüfung wird die Note um eine Teilnotenstufe (z.B. von 2,0 auf 1,7) verbessert, wenn Sie:

1. Mindestens 75% der Hausaufgaben bestanden haben und 2. mindestens 75% der Rechnerübungen erfolgreich absolviert haben.

Eine Note besser als 1,0 wird nicht vergeben.

Erstablingung: WS 2019/2020, 1. Wdh.: SS 2020

1. Prüfer: Jörn Thielecke (100224)

---

### **Organisatorisches:**

Masterstudium, kann ins Bachelorstudium vorgezogen werden. (Wahl- oder Wahlpflichtfach)

---

**Modulbezeichnung:** **Satellitenkommunikation (SATKOM)** **5 ECTS**  
(Satellite Communications)

Modulverantwortliche/r: Albert Heuberger  
Lehrende: Christian Rohde

---

Startsemester: SS 2019	Dauer: 1 Semester	Turnus: jährlich (SS)
Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Deutsch

---

**Lehrveranstaltungen:**

Satellitenkommunikation (SS 2019, Vorlesung, 2 SWS, Christian Rohde)  
Übung Satellitenkommunikation (SS 2019, Übung, 2 SWS, Jochen Martin-Creuzburg)

---

**Empfohlene Voraussetzungen:**

Keine formalen Voraussetzungen

---

**Inhalt:**

Nach einem historischen Rückblick zur Entwicklung der Satellitenkommunikation werden die einzelnen Komponenten eines typischen Gesamtsystems (Boden- und Raumsegment) näher betrachtet. Hierzu zählt der prinzipielle Aufbau von Trägerraketen (am Beispiel Ariane 5), von Satelliten (Satellitenplattformen, Subsysteme, Nutzlasten), die meist genutzten Umlaufbahnen und die Kommunikationslinks zwischen Bodenstation und Satellit (Uplink) und Satellit und Empfänger (Downlink). Die Besonderheiten der Signalausbreitung und -übertragung über diese großen Entfernungen (verwendete Frequenzen, Signaldispersion, -dämpfung, weitere Störeinflüsse der Weltraumumgebung) werden erklärt und in Beispielen berechnet. Die Architektur transparenter und regenerativer Kommunikationseinheiten wird ausführlich an Beispielen kommerziell verfügbarer Transponder und Onboard-Prozessoren erklärt. Die Prinzipien moderner, standardisierter Verfahren zur Signalaufbereitung und Übertragung von Video-/Bild und Audiosignalen über Satellit (MPEG, DVB-S/-S2 Quellencodierung, Kanalcodierung und Modulation, Kanalzugriff) werden erläutert und diskutiert. Ein Exkurs auf die im Orbit und im kommerziellen Einsatz befindlichen Kommunikationssatelliten für TV- und Breitbandversorgung, von Mobilkommunikationssystemen und eine Vorstellung von Forschungsansätzen für zukünftige Satellitensysteme (Megakonstellationen, Next generation High Throughput Satellites (HTS) sowie die große Dienstvielfalt mithilfe von Satelliten runden die Vorlesung ab. Die in der Vorlesung behandelten physikalischen, elektro- und nachrichtentechnischen Zusammenhänge werden in den ergänzenden Übungen mit Rechenbeispielen vertieft.

**Gliederung der Vorlesung:**

1. Einführung
  - Hauptkomponenten (Space Segment, Ground Segment), Überblick: Satelliten, Launcher, Orbits, Aufgaben, genutzte Frequenzen, Standarddienste, Satellitennetzwerke der größten Satellitenbetreiber
2. Entwicklung der Satellitenkommunikation
  - Meilensteine, Organisationen, Nationale Programme
3. Orbits und Konstellationen
  - Keplersche Gesetze, Beschreibung von Umlaufbahnen, genutzte Orbits, Bodenspuren, erreichbare Abdeckung
4. Trägersysteme
  - Trägerraketen, Entwicklung, Anbietermarkt, Nutzlastfähigkeit, Startplätze, Startverlauf
5. Satellitenaufbau
  - Auswahl aktueller Satellitenplattformen, Satellitenaufbau, Plattformkomponenten, Montageschritte und Tests
6. Satellitennutzlast (Payload)
  - Komponenten, Industrielle Beispiele, Aufbau und Aufgaben der Payload, Transponderarchitekturen, Antennen
7. Signalausbreitung und Leistungsbilanz

- Signalausbreitung, Freiraumverluste, Signaldämpfung, Rauschen, Signal-Rausch-Verhältnis, Linkbudget
- 8. Weltraumumgebung
  - Weltraumumgebungsbedingungen, Einflüsse auf den Satelliten und die Elektronik der Nutzlast (SEE, TID, DD)
- 9., 10., 11. Quellencodierung, Kanalcodierung, Modulationsverfahren, Signalübertragung am Beispiel DVB-S/-S2
  - Audio-, Bild- und Videokompression, FEC, Interleaver, Modulation, Zugriffs-, Duplex-, Multiplexverfahren
- 12. Vielfalt der Kommunikationsdienste, Beispiele moderner, zellulärer Satellitenkommunikationssysteme
  - Iridium, Globalstar, Inmarsat, Thuraya, Orbcom, O3b, ...), kommende Megakonstellationen (OneWeb, SpaceX, ...)
- 13. Neue Märkte, neue Dienste

#### **Lernziele und Kompetenzen:**

Nach der Teilnahme an der Vorlesung "Satellitenkommunikation" verstehen die Studierenden die Funktionsweise moderner satellitengestützter Kommunikationssysteme und wissen um deren Bedeutung für die großflächige und breitbandige Verbreitung von Bild-, Video- und Audiosignalen einschließlich einer Vielfalt an Datendiensten über Satellit. Die theoretischen Grundlagen und Rechenübungen anhand konkreter Beispiele erlauben die prinzipielle Berechnung und Überprüfung grundlegender Leistungsdaten und Systemkenngrößen.

#### **Literatur:**

Skriptum zur Lehrveranstaltung

#### **Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

##### **[1] Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule | Vertiefungsrichtung Elektronik und Informationstechnik)

Dieses Modul ist daneben auch in den Studienfächern "Berufspädagogik Technik (Bachelor of Science)", "Berufspädagogik Technik (Master of Education)", "Computational Engineering (Master of Science with Honours)", "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Master of Science)", "Elektrotechnik, Elektronik und Informationstechnik (Bachelor of Science)", "Elektrotechnik, Elektronik und Informationstechnik (Master of Science)", "Informatik (Master of Science)", "Information and Communication Technology (Master of Science)", "Informations- und Kommunikationstechnik (Bachelor of Science)", "Informations- und Kommunikationstechnik (Master of Science)", "Mathematik (Bachelor of Science)", "Wirtschaftsingenieurwesen (Master of Science)" verwendbar.

#### **Studien-/Prüfungsleistungen:**

Satellitenkommunikation (Prüfungsnummer: 34601)

(englische Bezeichnung: Satellite Communication)

Prüfungsleistung, Klausur, Dauer (in Minuten): 90

Anteil an der Berechnung der Modulnote: 100%

Erstablesung: SS 2019, 1. Wdh.: WS 2019/2020

1. Prüfer: Albert Heuberger (100228)

---

**Modulbezeichnung:** Interaktive Computergraphik und Globale Beleuchtungsberechnung (InCG+GlobIllum) 10 ECTS  
(Interactive Computer Graphics and Global Illumination)

Modulverantwortliche/r: Marc Stamminger  
Lehrende: Marc Stamminger

---

Startsemester: SS 2019	Dauer: 1 Semester	Turnus: jährlich (SS)
Präsenzzeit: 120 Std.	Eigenstudium: 180 Std.	Sprache: Deutsch

---

**Lehrveranstaltungen:**

Interaktive Computergraphik (SS 2019, Vorlesung, 2 SWS, Frank Bauer)  
 Übungen zur Interaktiven Computergraphik (SS 2019, Übung, 2 SWS, Alexander Lier et al.)  
 Globale Beleuchtungsberechnung (SS 2019, Vorlesung, 2 SWS, Marc Stamminger)  
 Übungen zur Globalen Beleuchtungsberechnung (SS 2019, Übung, Marc Stamminger)

---

**Inhalt:**

Siehe Module "Interaktive Computergraphik" und "Globale Beleuchtung"

**Lernziele und Kompetenzen:**

Siehe Module "Interaktive Computergraphik" und "Globale Beleuchtung"

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

**[1] Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule)

---

**Studien-/Prüfungsleistungen:**

Interaktive Computergraphik und Globale Beleuchtungsberechnung (Prüfungsnummer: 698977)  
 (englische Bezeichnung: Interactive Computer Graphics and Global Illumination)

Prüfungsleistung, Portfolio, Dauer (in Minuten): 30

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Interaktive Computergraphik
- Übungen zur Interaktiven Computergraphik
- Globale Beleuchtungsberechnung
- Übungen zur Globalen Beleuchtungsberechnung

weitere Erläuterungen:

InCG Übung: 50% der schriftlichen Aufgaben, GlobIllum Übung: 50% der schriftlichen Aufgaben,  
 Modulnote durch mündliche Prüfung über 30 Minuten

Prüfungssprache: Deutsch oder Englisch

Erstablingung: SS 2019, 1. Wdh.: WS 2019/2020

1. Prüfer: Marc Stamminger (100173)

---



---

**Modulbezeichnung:** Interaktive Computergraphik (InCG) **5 ECTS**  
(Interactive Computer Graphics)

Modulverantwortliche/r: Marc Stamminger  
Lehrende: Marc Stamminger

---

Startsemester: SS 2019	Dauer: 1 Semester	Turnus: jährlich (SS)
Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Deutsch

---

**Lehrveranstaltungen:**

Interaktive Computergraphik (SS 2019, Vorlesung, 2 SWS, Frank Bauer)  
Übungen zur Interaktiven Computergraphik (SS 2019, Übung, 2 SWS, Alexander Lier et al.)

---

**Inhalt:**

In der Vorlesung werden Algorithmen der Interaktiven Computergraphik behandelt, die sich für eine hochparallele Ausführung auf spezieller Grafikkhardware (GPUs) eignen. Insbesondere sind dies:

- Bump- und Normalmaps zur Erhöhung des visuellen Dateilgrads
- Tessellierung auf Grafikkhardware
- Berechnung von Schatten, insbesondere auch Filterung von Schatten und Schatten von Flächenlichtquellen
- Vereinfachte Globale Beleuchtungsverfahren
- Tiefen- und Bewegungsunschärfe

**Lernziele und Kompetenzen:**

Die Studierenden

- klassifizieren Schattierungsverfahren
- bestimmen den Unterschied zwischen lokaler und globaler Beleuchtung und formulieren - Algorithmen für Ray Tracing und Radiosity
- ermitteln graphische Algorithmen zur Berechnung von Schatten für unterschiedlichen Lichtquellen
- veranschaulichen Methoden zur Tessellierung von virtuellen 3D Modellen auf der Graphikkhardware
- klassifizieren Algorithmen zur Simulation von feinen Oberflächenstrukturen zur Erhöhung der Rendering-Detailsgrad
- skizzieren interaktive Algorithmen zur Berechnung globaler Beleuchtungsverfahren
- erläutern Techniken zur Bestimmung von Tiefen- und Bewegungsschärfe

**Literatur:**

- Möller, Haines: "Real-Time Rendering"
  - Foley, van Dam, Feiner, Hughes: "Computer Graphics: Principles and Practice"
- 

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

[1] **Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule)

---

**Studien-/Prüfungsleistungen:**

Interaktive Computergraphik (Prüfungsnummer: 503330)

(englische Bezeichnung: Interactive Computer Graphics)

Prüfungsleistung, Portfolio

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Interaktive Computergraphik
- Übungen zur Interaktiven Computergraphik

weitere Erläuterungen:

50% der Punkte aus den Übungen, Note aus 30-minütiger mündlicher Prüfung

Erstablegung: SS 2019, 1. Wdh.: WS 2019/2020

1. Prüfer: Marc Stamminger (100173)



---

**Modulbezeichnung: Applied Visualization (AppVis)** **5 ECTS**  
 (Applied Visualization)

 Modulverantwortliche/r: Roberto Grosso  
 Lehrende: Roberto Grosso

---

Startsemester: SS 2019	Dauer: 1 Semester	Turnus: jährlich (SS)
Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Englisch

---

**Lehrveranstaltungen:**

 Applied Visualization (SS 2019, Vorlesung, 2 SWS, Roberto Grosso)  
 Tutorials to Applied Visualization (SS 2019, Übung, 2 SWS, Daniel Zint)

---

**Inhalt:**

Die Visualisierung beschäftigt sich mit allen Aspekten, die im Zusammenhang stehen mit der visuellen Aufbereitung der (oft großen) Datenmengen aus technisch-wissenschaftlichen Experimenten oder Simulationen zum Zwecke des tieferen Verständnisses und der einfacheren Präsentation komplexer Phänomene. Die Vorlesung gibt eine Einführung in die grundlegenden Algorithmen und Datenstrukturen, sowie einen Überblick über die verfügbaren Softwarewerkzeuge und verbreiteten Dateiformate. Behandelt werden u.a. folgende Themen:

- Visualisierungsszenarien
- Gitterstrukturen und Interpolation
- Verfahren für 2D Skalar- und Vektorfelder
- Verfahren für 3D Skalar- und Vektorfelder
- Verfahren für multivariate Daten
- Volumenvisualisierung mit Isoflächen
- Direktes Volume-Rendering

In der Übung werden die Vorlesungsinhalte eingeübt und vertieft.

Visualization includes all aspects related to the visual preparation of usually large data sets from technical or scientific experiments and simulation. For a better understanding and a meaningful representation of complex phenomena, methods from interactive computer graphics are applied. This lecture introduces basic algorithms and data structures and gives an overview of available software tools and common data formats.

The lecture covers the following topics:

- scenarios for visualization
- meshes and data representation
- methods for 2D scalar and vector fields
- methods for 3D scalar and vector fields
- methods for multivariate data
- volume rendering with iso-surfaces
- direct volume rendering

**Lernziele und Kompetenzen:**

Die Studierenden:

- verfügen über tieferes Verständnis der visuellen Aufarbeitung von großen Datenmengen aus technisch-wissenschaftlichen Experimenten oder Simulationen
- sind mit grundlegenden Algorithmen und Datenstrukturen der Visualisierung wissenschaftlicher Daten vertraut
- verfügen über einen Überblick über die verfügbaren Softwarewerkzeuge und verbreiteten Dateiformate
- sind fähig, einfachere Präsentation komplexer Phänomene mit erlernten Methoden selbständig vorzubereiten
- sind in der Lage, selbstständig einfache Computerprogramme für die Visualisierung anwendungsspezifischer Daten zu entwickeln.

Students

- have a deep understanding of the process of visual processing of large data sets from scientific experiments and simulations
- can explain and apply fundamentals algorithm and data structures of scientific visualization to common practical problems
- are familiar with standard software tools in the area of scientific data visualization
- can carry out simple research projects requiring methods for the visualization of scientific data
- are able to implement simple algorithms for the visualization of scientific data from common science and engineering applications

**Literatur:**

- M. Ward, G.G. Grinstein, D. Keim, Interactive Data Visualization: Foundations, Techniques, and Applications, Taylor & Francis, 2010
- AC. Telea, Data Visualization: Principles and Practice, AK Peters, 2008
- C.D. Hansen and C.R. Johnson, Visualization Handbook, Academic Press, 2004
- G.M. Nielson, H. Hagen, H.Müller, Scientific Visualization, IEEE Computer Society Press, Los Alamitos, 1997

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

**[1] Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule)

Dieses Modul ist daneben auch in den Studienfächern "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Bachelor of Science)", "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Master of Science)", "Informatik (Master of Science)", "Information and Communication Technology (Master of Science)", "Informations- und Kommunikationstechnik (Master of Science)", "Maschinenbau (Bachelor of Science)", "Maschinenbau (Master of Science)", "Mathematik (Bachelor of Science)", "Medizintechnik (Bachelor of Science)", "Medizintechnik (Master of Science)", "Physische Geographie (Bachelor of Science)" verwendbar.

---

**Studien-/Prüfungsleistungen:**

Applied Visualization (Prüfungsnummer: 37211)

(englische Bezeichnung: Applied Visualisation)

Prüfungsleistung, Klausur, Dauer (in Minuten): 90

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Applied Visualization
- Tutorials to Applied Visualization

weitere Erläuterungen:

Klausur in elektronischer Form mit einem Anteil im Antwort-Wahl-Verfahren

Erstabelleung: SS 2019, 1. Wdh.: WS 2019/2020

1. Prüfer: Roberto Grosso (100188)

---

**Modulbezeichnung:** **Applied Visualization (AppVis)** **5 ECTS**  
 (Applied Visualization)

Modulverantwortliche/r: Roberto Grosso  
 Lehrende: Roberto Grosso

---

Startsemester: SS 2019	Dauer: 1 Semester	Turnus: jährlich (SS)
Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Englisch

---

**Lehrveranstaltungen:**

Applied Visualization (SS 2019, Vorlesung, 2 SWS, Roberto Grosso)  
 Tutorials to Applied Visualization (SS 2019, Übung, 2 SWS, Daniel Zint)

---

**Inhalt:**

Die Visualisierung beschäftigt sich mit allen Aspekten, die im Zusammenhang stehen mit der visuellen Aufbereitung der (oft großen) Datenmengen aus technisch-wissenschaftlichen Experimenten oder Simulationen zum Zwecke des tieferen Verständnisses und der einfacheren Präsentation komplexer Phänomene. Die Vorlesung gibt eine Einführung in die grundlegenden Algorithmen und Datenstrukturen, sowie einen Überblick über die verfügbaren Softwarewerkzeuge und verbreiteten Dateiformate. Behandelt werden u.a. folgende Themen:

- Visualisierungsszenarien
- Gitterstrukturen und Interpolation
- Verfahren für 2D Skalar- und Vektorfelder
- Verfahren für 3D Skalar- und Vektorfelder
- Verfahren für multivariate Daten
- Volumenvisualisierung mit Isoflächen
- Direktes Volume-Rendering

In der Übung werden die Vorlesungsinhalte eingeübt und vertieft.

Visualization includes all aspects related to the visual preparation of usually large data sets from technical or scientific experiments and simulation. For a better understanding and a meaningful representation of complex phenomena, methods from interactive computer graphics are applied. This lecture introduces basic algorithms and data structures and gives an overview of available software tools and common data formats.

The lecture covers the following topics:

- scenarios for visualization
- meshes and data representation
- methods for 2D scalar and vector fields
- methods for 3D scalar and vector fields
- methods for multivariate data
- volume rendering with iso-surfaces
- direct volume rendering

**Lernziele und Kompetenzen:**

Die Studierenden:

- verfügen über tieferes Verständnis der visuellen Aufarbeitung von großen Datenmengen aus technisch-wissenschaftlichen Experimenten oder Simulationen
- sind mit grundlegenden Algorithmen und Datenstrukturen der Visualisierung wissenschaftlicher Daten vertraut
- verfügen über einen Überblick über die verfügbaren Softwarewerkzeuge und verbreiteten Dateiformate
- sind fähig, einfachere Präsentation komplexer Phänomene mit erlernten Methoden selbständig vorzubereiten
- sind in der Lage, selbstständig einfache Computerprogramme für die Visualisierung anwendungsspezifischer Daten zu entwickeln.

Students

- have a deep understanding of the process of visual processing of large data sets from scientific experiments and simulations
- can explain and apply fundamentals algorithm and data structures of scientific visualization to common practical problems
- are familiar with standard software tools in the area of scientific data visualization
- can carry out simple research projects requiring methods for the visualization of scientific data
- are able to implement simple algorithms for the visualization of scientific data from common science and engineering applications

**Literatur:**

- M. Ward, G.G. Grinstein, D. Keim, Interactive Data Visualization: Foundations, Techniques, and Applications, Taylor & Francis, 2010
- AC. Telea, Data Visualization: Principles and Practice, AK Peters, 2008
- C.D. Hansen and C.R. Johnson, Visualization Handbook, Academic Press, 2004
- G.M. Nielson, H. Hagen, H.Müller, Scientific Visualization, IEEE Computer Society Press, Los Alamitos, 1997

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

**[1] Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule | Vertiefungsrichtung Graphische Datenverarbeitung)

Dieses Modul ist daneben auch in den Studienfächern "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Bachelor of Science)", "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Master of Science)", "Informatik (Master of Science)", "Information and Communication Technology (Master of Science)", "Informations- und Kommunikationstechnik (Master of Science)", "Maschinenbau (Bachelor of Science)", "Maschinenbau (Master of Science)", "Mathematik (Bachelor of Science)", "Medizintechnik (Bachelor of Science)", "Medizintechnik (Master of Science)", "Physische Geographie (Bachelor of Science)" verwendbar.

**Studien-/Prüfungsleistungen:**

Applied Visualization (Prüfungsnummer: 37211)

(englische Bezeichnung: Applied Visualisation)

Prüfungsleistung, Klausur, Dauer (in Minuten): 90

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Applied Visualization
- Tutorials to Applied Visualization

weitere Erläuterungen:

Klausur in elektronischer Form mit einem Anteil im Antwort-Wahl-Verfahren

Erstablingung: SS 2019, 1. Wdh.: WS 2019/2020

1. Prüfer: Roberto Grosso (100188)

---

**Modulbezeichnung:** Computergraphik-VU (CG-VU) 5 ECTS  
(Computer Graphics)

Modulverantwortliche/r: Marc Stamminger  
Lehrende: Marc Stamminger

---

Startsemester: WS 2019/2020	Dauer: 1 Semester	Turnus: jährlich (WS)
Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Deutsch

---

**Lehrveranstaltungen:**

Computergraphik (WS 2019/2020, Vorlesung, 3 SWS, Marc Stamminger)  
Übungen zur Computergraphik (WS 2019/2020, Übung, 1 SWS, N.N.)

---

**Vorhergehende Module:**

Algorithmik kontinuierlicher Systeme

---

**Inhalt:**

Die Vorlesung gibt eine Einführung in die Computergraphik:

- Graphik Pipeline
- Clipping
- 3D Transformationen
- Hierarchische Display Strukturen
- Perspektive und Projektionen
- Visibilitätsbetrachtungen
- Rastergraphik und Scankonvertierung
- Farbmodelle
- Lokale und globale Beleuchtungsmodelle
- Schattierungsverfahren
- Ray Tracing und Radiosity
- Schatten und Texturen

Contents:

This lecture covers the following aspects of Computer Graphics:

- graphics pipeline
- clipping
- 3D transformations
- hierarchical display structures
- perspective transformations and projections
- visibility determination
- raster graphics and scan conversion
- color models
- local and global illumination models
- shading models
- ray tracing and radiosity
- shadows and textures

**Lernziele und Kompetenzen:**

Die Studierenden

- geben die unterschiedlichen Schritte der Graphik Pipeline wieder
- erklären die Funktionsweise der Clippingalgorithmen für Linien und Polygone
- beschreiben, charakterisieren und berechnen affine und perspektivische Transformationen in 3D und veranschaulichen die allgemeine Form der Transformationsmatrix in homogener Koordinaten
- skizzieren die Verfahren zur Tiefe- und Visibilitätsberechnung
- vergleichen die unterschiedlichen Farbmodelle der Computergraphik

- illustrieren und untersuchen die Datenstrukturen zur Beschreibung virtueller 3D Modelle und komplexer Szenen
- erläutern die Funktionsweise der Rasterisierung und Scankonvertierung in der Graphikpipeline
- lösen Aufgaben zu Beleuchtung und Texturierung von 3D virtuellen Modellen
- klassifizieren Schattierungsverfahren
- bestimmen den Unterschied zwischen lokaler und globaler Beleuchtung und formulieren Algorithmen für Ray Tracing und Radiosity

#### **Educational objectives and skills:**

Students should be able to

- describe the processing steps in the graphics pipeline
- explain clipping algorithms for lines and polygons
- explain, characterize and compute affine and perspective transformations in 2D and 3D, and provide an intuitive description of the general form of corresponding transformation matrices in homogeneous coordinates
- depict techniques to compute depth, occlusion and visibility
- compare the different color models
- describe data structures to represent 3D virtual models and complex scenes
- explain the algorithms for rasterization and scan conversion
- solve problems with shading and texturing of 3D virtual models
- classify different shadowing techniques
- explain the difference between local and global illumination techniques and formulate algorithms for ray tracing and radiosity

#### **Literatur:**

- P. Shirley: Fundamentals of Computer Graphics. AK Peters Ltd., 2002
- Hearn, M. P. Baker: Computer Graphics with OpenGL. Pearson
- Foley, van Dam, Feiner, Hughes: Computer Graphics - Principles and Practice
- Rauber: Algorithmen der Computergraphik
- Bungartz, Griebel, Zenger: Einführung in die Computergraphik
- Encarnaçã, Strasser, Klein: Computer Graphics

#### **Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

##### **[1] Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule | Vertiefungsrichtung Graphische Datenverarbeitung)

Dieses Modul ist daneben auch in den Studienfächern "Communications and Multimedia Engineering (Master of Science)", "Computational Engineering (Master of Science with Honours)", "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Bachelor of Science)", "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Master of Science)", "Digital Humanities (Master of Arts)", "Informatik (Master of Science)", "Information and Communication Technology (Master of Science)", "Informations- und Kommunikationstechnik (Bachelor of Science)", "Informations- und Kommunikationstechnik (Master of Science)", "Maschinenbau (Bachelor of Science)", "Maschinenbau (Master of Science)", "Mathematik (Bachelor of Science)", "Medizintechnik (Bachelor of Science)", "Medizintechnik (Master of Science)" verwendbar.

#### **Studien-/Prüfungsleistungen:**

Computergraphik (Vorlesung mit Übung) (Prüfungsnummer: 38201)

Prüfungsleistung, Klausur, Dauer (in Minuten): 60

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Computergraphik
- Übungen zur Computergraphik

weitere Erläuterungen:

Übung: 50% der schriftlichen Aufgaben, Modulnote durch Klausur über 60 Minuten



Erstablingung: WS 2019/2020, 1. Wdh.: SS 2020  
1. Prüfer: Marc Stamminger (100173)

---

**Bemerkungen:**

Vorlesungsunterlagen, Übungsblätter und die Klausur sind in englischer Sprache

---

**Modulbezeichnung:** **Computergraphik-VUP (CG-VUP)** **7.5 ECTS**  
(Computer Graphics)

Modulverantwortliche/r: Marc Stamminger  
Lehrende: Marc Stamminger

---

Startsemester: WS 2019/2020	Dauer: 1 Semester	Turnus: jährlich (WS)
Präsenzzeit: 90 Std.	Eigenstudium: 135 Std.	Sprache: Englisch

---

**Lehrveranstaltungen:**

Computergraphik (WS 2019/2020, Vorlesung, 3 SWS, Marc Stamminger)  
 Übungen zur Computergraphik (WS 2019/2020, Übung, 1 SWS, N.N.)  
 Vertiefte Übungen zur Computergraphik (WS 2019/2020, Übung, 2 SWS, N.N.)

---

**Vorhergehende Module:**

Algorithmik kontinuierlicher Systeme

---

**Inhalt:**

Die Vorlesung gibt eine Einführung in die Computergraphik:

- Graphik Pipeline
- Clipping
- 3D Transformationen
- Hierarchische Display Strukturen
- Perspektive und Projektionen
- Visibilitätsbetrachtungen
- Rastergraphik und Scankonvertierung
- Farbmodelle
- Lokale und globale Beleuchtungsmodelle
- Schattierungsverfahren
- Ray Tracing und Radiosity
- Schatten und Texturen

**Lernziele und Kompetenzen:**

Die Studierenden

- geben die unterschiedlichen Schritte der Graphik Pipeline wieder
- erklären die Funktionsweise der Clippingalgorithmen für Linien und Polygone
- beschreiben, charakterisieren und berechnen affine und perspektivische Transformationen in 3D und veranschaulichen die allgemeine Form der Transformationsmatrix in homogener Koordinaten
- skizzieren die Verfahren zur Tiefe- und Visibilitätsberechnung
- vergleichen die unterschiedlichen Farbmodelle der Computergraphik
- illustrieren und untersuchen die Datenstrukturen zur Beschreibung virtueller 3D Modelle und komplexer Szenen
- erläutern die Funktionsweise der Rasterisierung und Scankonvertierung in der Graphikpipeline
- implementieren 3D Transformationen mithilfe der Programmiersprache C++ und der graphischen Bibliothek OpenGL
- Implementieren Beleuchtungsmodelle und Texturierung von virtuellen 3D Objekten mithilfe der Programmiersprachen OpenGL und GLSL
- lösen Aufgaben zu Beleuchtung und Texturierung von 3D virtuellen Modellen
- klassifizieren Schattierungsverfahren
- bestimmen den Unterschied zwischen lokaler und globaler Beleuchtung und formulieren Algorithmen für Ray Tracing und Radiosity

**Educational objectives and skills:**

Students should be able to

- describe the processing steps in the graphics pipeline
- explain clipping algorithms for lines and polygons

- explain, characterize and compute affine and perspective transformations in 2D and 3D, and provide an intuitive description of the general form of corresponding transformation matrices in homogeneous coordinates
- depict techniques to compute depth, occlusion and visibility
- compare the different color models
- describe data structures to represent 3D virtual models and complex scenes
- explain the algorithms for rasterization and scan conversion
- solve problems with shading and texturing of 3D virtual models
- classify different shadowing techniques
- explain the difference between local and global illumination techniques and formulate algorithms for ray tracing and radiosity

#### Literatur:

- P. Shirley: Fundamentals of Computer Graphics. AK Peters Ltd., 2002
- Hearn, M. P. Baker: Computer Graphics with OpenGL. Pearson
- Foley, van Dam, Feiner, Hughes: Computer Graphics - Principles and Practice
- Rauber: Algorithmen der Computergraphik
- Bungartz, Griebel, Zenger: Einführung in die Computergraphik
- Encarnação, Strasser, Klein: Computer Graphics

---

#### Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

##### [1] Informatik (Bachelor of Science)

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule | Vertiefungsrichtung Graphische Datenverarbeitung)

Dieses Modul ist daneben auch in den Studienfächern "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Bachelor of Science)", "Informatik (Master of Science)", "Informations- und Kommunikationstechnik (Master of Science)", "Mathematik (Bachelor of Science)", "Medizintechnik (Bachelor of Science)", "Medizintechnik (Master of Science)" verwendbar.

---

#### Studien-/Prüfungsleistungen:

Computergraphik (Vorlesung mit Übung und Praktikum) (Prüfungsnummer: 321743)

Prüfungsleistung, Klausur, Dauer (in Minuten): 60

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Computergraphik
- Übungen zur Computergraphik
- Vertiefte Übungen zur Computergraphik

weitere Erläuterungen:

Übung: 50% der schriftlichen Aufgaben, Praktikum: 50% der Programmieraufgaben, Modulnote durch Klausur über 60 Minuten

Erstablingung: WS 2019/2020, 1. Wdh.: SS 2020

1. Prüfer: Marc Stamminger (100173)

---

#### Bemerkungen:

Vorlesungsunterlagen, Übungsblätter und die Klausur sind in englischer Sprache

**Modulbezeichnung: Geometrische Modellierung - VU (GM-VU)** **5 ECTS**  
(Geometric Modeling)

Modulverantwortliche/r: Marc Stamminger, Roberto Grosso

Lehrende: Marc Stamminger, Roberto Grosso

Startsemester: WS 2019/2020

Dauer: 1 Semester

Turnus: jährlich (WS)

Präsenzzeit: 60 Std.

Eigenstudium: 90 Std.

Sprache: Englisch

**Lehrveranstaltungen:**

Geometric Modeling (WS 2019/2020, Vorlesung, 3 SWS, Roberto Grosso)

Tutorials to Geometric Modeling (WS 2019/2020, Übung, 1 SWS, Daniel Zint et al.)

**Inhalt:**

Die Vorlesung beschäftigt sich mit Methoden zur Modellierung dreidimensionaler Oberflächen. Typische Einsatzgebiete sind der rechnerunterstützte Entwurf (CAD, z.B. im Automobil- oder Flugzeugbau), die Rekonstruktion von Flächen aus Sensordaten oder die Konstruktion glatter Interpolationsflächen. Behandelt werden u.a. folgende Themen:

- Polynomkurven
- Bezierkurven, rationale Bezierkurven
- B-Splines
- Tensorproduktflächen
- Bezier-Dreiecksflächen
- polygonale Flächen
- Subdivision-Verfahren

This lecture is concerned with different aspects of modelling three-dimensional curves and surfaces. Typical areas of application are computer-aided design (CAD), reconstruction of surfaces from sensor data (reverse engineering) and construction of smooth interpolants. The lecture covers the following topics:

- polynomial curves
- Bézier curves, rational Bézier curves
- B-splines
- tensor product surfaces
- triangular Bézier surfaces
- polyhedral surfaces

**Lernziele und Kompetenzen:**

Die Studierenden

- erklären die Begriffe Polynomial-, Bezierkurven und B-Splines
- klassifizieren und veranschaulichen die unterschiedlichen Auswertung- und Subdivision-Verfahren für Bezier-Kurven und B-Splines
- veranschaulichen und ermitteln die Eigenschaften von Bezierkurven, rationalen Bezierkurven und B-Splines
- beschreiben Tensorproduktflächen und skizzieren Auswertungsalgorithmen
- erklären polygonale Flächen und Subdivision-Verfahren und veranschaulichen ihre Unterschiede und Eigenschaften
- lernen gängige Datenstrukturen zur Darstellung polygonaler Flächen kennen
- wenden die Verfahren der Geometrischen Modellierung an unterschiedlichen Beispiele an
- berechnen Bezierkurven und B-Splines
- führen Subdivision-Verfahren aus

**Educational objectives and skills:**

Students should be able to

- explain the meaning of the terms Polynomial and Bezier curves and B-Splines

- classify and illustrate the different evaluation and subdivision methods for Bezier curves and B-Splines
- describe and establish the properties of Bezier curves, rational Bezier curves and B-Splines
- describe tensor product surfaces and illustrated evaluation algorithms
- explain polygonal surfaces and subdivision algorithms and depict their properties and differences
- get used with common data structures to represent polygonal surfaces
- apply geometric modeling algorithms to representative examples
- compute Bezier curves and B-Splines
- implement subdivision algorithms

#### Literatur:

- Hoschek, Lasser: Grundlagen der Geometrischen Datenverarbeitung
- Farin: Kurven und Flächen im Computer Aided Geometric Design
- de Boor: A Practical Guide to Splines
- Bartels, Beatty, Barsky: Splines for Use in Computer Graphics and Geometric Modeling
- Abramowski, Müller: Geometrisches Modellieren

---

#### Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

##### [1] Informatik (Bachelor of Science)

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule | Vertiefungsrichtung Graphische Datenverarbeitung)

Dieses Modul ist daneben auch in den Studienfächern "Informatik (Master of Science)", "Mathematik (Bachelor of Science)", "Medizintechnik (Master of Science)" verwendbar.

---

#### Studien-/Prüfungsleistungen:

Geometric Modeling (Vorlesung mit Übung) (Prüfungsnummer: 796399)

Prüfungsleistung, Klausur, Dauer (in Minuten): 90

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Geometric Modeling
- Tutorials to Geometric Modeling

weitere Erläuterungen:

Klausur in elektronischer Form mit einem Anteil im Antwort-Wahl-Verfahren

Erstablingung: WS 2019/2020, 1. Wdh.: SS 2020

1. Prüfer: Roberto Grosso (100188), 2. Prüfer: Marc Stamminger (100173)

---

**Modulbezeichnung:** Informationsvisualisierung (InfoVIS) 5 ECTS  
 (Information Visualization)

Modulverantwortliche/r: Roberto Grosso  
 Lehrende: Roberto Grosso

---

Startsemester: WS 2019/2020	Dauer: 1 Semester	Turnus: jährlich (WS)
Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Englisch

---

**Lehrveranstaltungen:**

Informationsvisualisierung (WS 2019/2020, Vorlesung, 2 SWS, Roberto Grosso)  
 Übung zur Informationsvisualisierung (WS 2019/2020, Übung, Roberto Grosso)

---

**Vorhergehende Module:**

Algorithmik kontinuierlicher Systeme

---

**Inhalt:**

Aufgrund der rasanten Entwicklung der Informationstechnologie sind wir mit einer noch nie dagewesenen Flut an Daten konfrontiert:

- 250.000.000 Fotos werden täglich auf Facebook geladen
- Weltweit finden 130.000.000 VISA-Transaktionen statt
- Täglich werden über 500 Millionen Tweets gesendet
- ...

Informationsvisualisierung befasst sich mit der graphischen Darstellung abstrakter Daten, die keine räumliche Struktur aufweisen. Die Visualisierung abstrakter Daten nutzt visuelle Metaphern und Interaktion, um Information aus den Daten zu extrahieren. Typische Anwendungsszenarien sind die Analyse von Finanztransaktionen oder sozialen Netzwerken, Bioinformatik, Geographie, Textanalyse oder Visualisierung von Software-Quellcode.

In dieser Vorlesung werden unterschiedliche Techniken vorgestellt, um verschieden Arten von Daten zu visualisieren. Insbesondere werden folgende Themen behandelt:

- Graphen und Netzwerke
- Dynamische Graphen
- Hierarchien und Bäume
- Multivariate Daten
- Time-Series Daten
- Textvisualisierung
- Visualisierung Biologischer Daten

**Lernziele und Kompetenzen:**

Die Studierenden:

- listen und identifizieren die unterschiedlichen Algorithmen der Informationsvisualisierung
- veranschaulichen die Methoden zur Visualisierung von Graphen und Netzwerke und bestimmen ihre Unterschiede
- klassifizieren Algorithmen zur Visualisierung multivariater Daten und erklären ihrer Funktionsweise
- erklären und charakterisieren Techniken für die Text-Visualisierung und veranschaulichen die Methoden zur Visualisierung zeitabhängiger Daten
- lernen Visualisierungswerkzeuge kennen und wenden diese zur Lösung praxisrelevanter Aufgaben der Informationsvisualisierung
- sind in der Lage, die vorgestellten Algorithmen der Informationsvisualisierung in JavaScript zu implementieren

**Literatur:**

- Robert Spence: Information Visualization: Design for Interaction
- Stuart K. Card, Jock Mackinlay, Ben Shneiderman: Readings in Information Visualization - Using Vision to Think
- Benjamin B. Bederson, Ben Shneiderman: The Craft of Information Visualization - Readings and Reflections

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

**[1] Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule | Vertiefungsrichtung Graphische Datenverarbeitung)

Dieses Modul ist daneben auch in den Studienfächern "Digital Humanities (Master of Arts)", "Informatik (Master of Science)", "International Information Systems (IIS) (Master of Science)", "Kunstgeschichte (Master of Arts)", "Mathematik (Bachelor of Science)" verwendbar.

---

**Studien-/Prüfungsleistungen:**

Informationsvisualisierung (Prüfungsnummer: 299892)

(englische Bezeichnung: Information Visualization)

Prüfungsleistung, Klausur, Dauer (in Minuten): 90

Anteil an der Berechnung der Modulnote: 100%

weitere Erläuterungen:

Klausur in elektronischer Form mit einem Anteil im Antwort-Wahl-Verfahren

Erstablingung: WS 2019/2020, 1. Wdh.: SS 2020

1. Prüfer: Roberto Grosso (100188)

---

---

**Modulbezeichnung:** **Visual Computing in Medicine (VCMed)** **5 ECTS**  
 (Visual Computing in Medicine)

Modulverantwortliche/r: Peter Hastreiter, Thomas Wittenberg  
 Lehrende: Peter Hastreiter, Thomas Wittenberg

---

Startsemester: WS 2019/2020	Dauer: 2 Semester	Turnus: jährlich (WS)
Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Englisch

---

**Lehrveranstaltungen:**

Visual Computing in Medicine 1 (WS 2019/2020, Vorlesung, 2 SWS, Peter Hastreiter et al.)  
 Visual Computing in Medicine 2 (SS 2020, Vorlesung, Thomas Wittenberg et al.)

---

**Vorhergehende Module:**

Algorithmik kontinuierlicher Systeme  
 Computergraphik-VU

---

**Inhalt:**

Die Flut und Komplexität medizinischer Bilddaten sowie die klinischen Anforderungen an Genauigkeit und Effizienz erfordern leistungsfähige wie auch robuste Konzepte der medizinischen Datenverarbeitung. Auf Grund der Vielfalt an Bildinformation und ihrer klinischen Relevanz spielt der Übergang von der Messung medizinischer Bilddaten (u.a. MRT, CT, PET) hin zur Analyse der Bildinhalte eine wichtige Rolle. Durch die visuelle Wiedergabe der abstrakten Daten können sowohl technische als auch medizinische Aspekte anschaulich und intuitiv verstanden werden. Aufbauend auf einem Regelkreis zur Verarbeitung medizinischer Bilddaten werden im ersten Teil (Visual Computing in Medicine I) die Eigenschaften medizinischer Bilddaten sowie grundlegende Methoden und Verfahren der medizinischen Bildanalyse und Visualisierung im Zusammenhang vermittelt. Beispiele aus der Praxis erläutern den Bezug zur medizinischen Anwendung. Darauf aufbauend werden im zweiten Teil (Visual Computing in Medicine II) konkrete Lösungsansätze für die Diagnose und Therapieplanung komplexer Krankheitsbilder erläutert. Es wird gezeigt, wie grundlegende Methoden ausgewählt und zu praktisch anwendbaren Gesamtkonzepten zusammengefasst werden. An Beispielen wird der Bezug zu Strategien und Anforderungen in der industriellen Entwicklung und klinischen Anwendung hergestellt. Ergänzend werden komplexe Methoden der medizinischen Bildanalyse und Visualisierung ausführlich besprochen.

The flood and complexity of medical image data as well as the clinical need for accuracy and efficiency require powerful and robust concepts of medical data processing. Due to the diversity of image information and their clinical relevance the transition from imaging to medical analysis and interpretation plays an important role. The visual representation of abstract data allows understanding both technical and medical aspects in a comprehensive and intuitive way. Based on a processing pipeline for medical image data an overview of the characteristics of medical image data as well as fundamental methods and procedures for medical image analysis and visualization is given. Examples of clinical practice show the relation to the medical application. Based on VCMed1 the lecture VCMed2 discusses practical approaches for the diagnosis and therapy planning of complex diseases. It will be shown how fundamental methods are selected and integrated to practically applicable concepts. Examples demonstrate the relation to strategies and requirements in clinical practice and the industrial development process. Additionally, complex methods of medical image analysis and visualization will be explained.

**Lernziele und Kompetenzen:**

**Visual Computing in Medicine I**

Die Studierenden

- erhalten einen Überblick zu Grundlagen und Unterschieden medizinischer Bildgebungsverfahren
- erwerben fundierte Kenntnisse über Gitterstrukturen, Datentypen und Formate medizinischer Bilddaten
- üben an Beispielen die Erkennung und Interpretation unterschiedlicher Bilddaten
- erwerben Kenntnisse zu Verfahren der Vorverarbeitung, Filterung und Interpolation medizinischer Bilddaten sowie zu grundlegenden Ansätzen der Segmentierung
- erlernen Prinzipien und Methoden der expliziten und impliziten Bildregistrierung und erhalten einen



Überblick zu wichtigen Verfahren der starren Registrierung

- erwerben fundierte Kenntnisse zu allen Aspekten der me-dizinischen Visualisierung (2D, 3D, 4D) von Skalar-, Vek-tor-, Tensor-daten
- erhalten an einfachen Beispielen einen ersten Eindruck, wie sich Visualisierung zur Steuerung von Bildanalysever-fahren und für die medizinische Diagnostik einsetzen lässt

The students

- get an overview of the basic principles and differences of medical imaging methods,
- acquire profound knowledge about grid structures, data types and formats of medical image data,
- use sample data to recognize and interpret different image data,
- acquire knowledge about methods of preprocessing, filtering and interpolation of medical image data as well as on basic approaches of segmentation,
- learn the principles and methods of explicit and implicit image registration and get an overview of important procedures of rigid registration,
- acquire profound knowledge about all aspects of medical visualization (2D, 3D, 4D) of scalar, vector, tensor data,
- get a first impression of how visualization can be used to control image analysis and medical diagnostics.

### Visual Computing in Medicine II

Die Studierenden

- erwerben aus Sicht der medizinischen Anwendung und konkreter Lösungsstrategien einen Einblick in komplexe Ansätze zur Bearbeitung wichtiger Krankheitsbilder
- lernen die Anforderungen an und die Verknüpfung von Methoden der medizinischen Bildanalyse und Visualisie-rung zur Bearbeitung kardiologischer, neurologischer, on-kologischer und strahlenthera-peutischer Fragestellungen
- erhalten einen Überblick zu komplexen Krankheitsbildern als Grundlage für effektive und effiziente Lösungen
- erwerben erweiterte Kenntnisse zur multimodalen Bildre-gistrierung mit nichtstarren Transformatio-nen
- erhalten vertieftes Wissen zu komplexen und aktuellen Themen der medizinischen Visualisierung (u.a. Integrati-onsverfahren, Transferfunktionen, Beschleunigungstech-niken mit Grafikhardware)

The students

- gain an insight into complex approaches to the treatment of important disease patterns from the point of view of medical application and specific solution strategies
- learn the requirements and the linking of methods of medical image analysis and visualization for the processing of cardiological, neurological, oncological and radiotherapeutic questions
- get an overview of complex disease pictures as a basis for effective and efficient solutions
- acquire advanced knowledge to process multimodal image data using advanced methods
- receive in-depth knowledge on complex and up-to-date topics of medical visualization (including integration procedures, transfer functions, acceleration techniques with graphics hardware)

### Literatur:

- B. Preim, C. Botha: Visual Computing for Medicine, Morgan Kaufmann Verlag, 2013
- B. Preim, D. Bartz: Visualization in Medicine - Theory, Algorithms, and Applications, Morgan Kaufmann Verlag, 2007
- H. Handels: Medizinische Bildverarbeitung, Bildanalyse, Mustererkennung und Visualisierung für die computergestützte ärztliche Diagnostik und Therapie, Vieweg und Teubner Verlag, 2009
- P.M. Schlag, S. Eulenstein, Th. Lange: Computerassistierte Chirurgie, Elsevier Verlag, 2010
- E. Neri, D. Caramella, C. Bartolozzi: Image Processing in Radiology, Springer Verlag, 2008

---

### Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

#### [1] Informatik (Bachelor of Science)

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule | Vertiefungsrichtung Graphische Datenverarbeitung)

Dieses Modul ist daneben auch in den Studienfächern "Informatik (Master of Science)", "Mathematik (Bachelor of Science)", "Medizintechnik (Bachelor of Science)", "Medizintechnik (Master of Science)" verwendbar.

---

**Studien-/Prüfungsleistungen:**

Visual Computing in Medicine (Prüfungsnummer: 44811)

(englische Bezeichnung: Visual Computing in Medicine)

Prüfungsleistung, mündliche Prüfung, Dauer (in Minuten): 30

Anteil an der Berechnung der Modulnote: 100%

Erstablingung: SS 2020, 1. Wdh.: WS 2020/2021

1. Prüfer: Thomas Wittenberg (100167)

---

---

**Modulbezeichnung:** **Verifikation digitaler Systeme (VdS)** **5 ECTS**  
 (Verification of Digital Systems)

Modulverantwortliche/r: Oliver Keszöcze  
 Lehrende: Oliver Keszöcze

---

Startsemester: SS 2019	Dauer: 1 Semester	Turnus: jährlich (WS)
Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Deutsch

---

**Lehrveranstaltungen:**

Verifikation digitaler Systeme (SS 2019, Vorlesung, 2 SWS, Oliver Keszöcze)  
 Übung zur Verifikation digitaler Systeme (SS 2019, Übung, 2 SWS, Oliver Keszöcze)

---

**Inhalt:**

Für den Entwurf eines digitalen Systems werden heute in der Industrie ebenso viele Verifikationsingenieure wie Designer benötigt. Trotzdem beansprucht die Verifikation heute bereits 70%-80% der gesamten Entwurfszeit. Neben konventionellen Verifikationserfahren wie der Simulation sind werden seit einigen Jahren sogenannte "formale Verifikationsmethoden" in heutigen Entwurfsflüssen eingesetzt. Der Umgang mit diesen Methoden stellt ein wichtiges neues Aufgabenfeld dar. Im Gegensatz zur Simulation beruht die formale Verifikation auf exakten mathematischen Methoden zum Nachweis funktionaler Schaltungseigenschaften. Dadurch können Entwurfsfehler frühzeitiger und mit höherer Zuverlässigkeit als bisher erkannt werden. Jedes System zur formalen Hardwareverifikation erfordert:

1. ein geeignetes Modell des zu verifizierenden Systems
2. eine Sprache zur Formulierung der zu verifizierenden Eigenschaften
3. eine Beweismethode.

Die Vorlesung behandelt diese drei Bereiche, vermittelt die grundlegenden Algorithmen und Konzepte moderner Werkzeuge für die formale Hardwareverifikation und erläutert deren Einsatz in der industriellen Praxis. Im Einzelnen werden in dieser Vorlesung die folgenden Punkte behandelt:

1. Modellierung digitaler Systeme
2. Unterschiede formaler und simulationsbasierter Verifikationsmethoden
3. Äquivalenzvergleich
4. Formale und simulationsbasierte Eigenschaftsprüfung
5. Assertions
6. Verifikation arithmetischer Schaltungen

**Lernziele und Kompetenzen:**

*Fachkompetenz*

*Verstehen*

- Die Studierenden erläutern die wesentlichen Techniken zur Verifikation digitaler Systeme, zeigen deren Vor- und Nachteile auf und vergleichen diese bezüglich Ihrer Mächtigkeit und Komplexität.

*Anwenden*

- Die Studierenden wenden Modellierungstechniken aus den Bereichen der Binären Entscheidungsdiagramme inkl. bekannter Erweiterungen (MDD, etc.) sowie der SAT-Löser auf Systembeschreibungen an.
  - Die Studierenden Verifikationstechniken aus den Bereichen der formalen Äquivalenz- und Eigenschaftsprüfung (Model Checking, Symbolic Model Checking, Bounded Model Checking) auf Systembeschreibungen an.
- 

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

**[1] Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule)

Dieses Modul ist daneben auch in den Studienfächern "Computational Engineering (Master of Science with Honours)", "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Bachelor of Science)", "Computa-

tional Engineering (Rechnergestütztes Ingenieurwesen) (Master of Science)", "Informatik (Master of Science)",  
"Mathematik (Bachelor of Science)" verwendbar.

---

**Studien-/Prüfungsleistungen:**

Verifikation digitaler Systeme (Prüfungsnummer: 22801)

Prüfungsleistung, Klausur, Dauer (in Minuten): 90

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Verifikation digitaler Systeme
- Übung zur Verifikation digitaler Systeme

weitere Erläuterungen:

Aufgrund geringer Teilnehmerzahlen kann die Prüfung auch als 30-minütige mündliche Prüfung erfolgen.

Erstablingung: SS 2019, 1. Wdh.: WS 2019/2020

1. Prüfer: Oliver Keszöcze (100421)

---

---

**Modulbezeichnung:** Cyber-Physical Systems (CPS) **5 ECTS**  
 (Cyber-Physical Systems)

Modulverantwortliche/r: Torsten Klie  
 Lehrende: Torsten Klie

---

Startsemester: SS 2019	Dauer: 1 Semester	Turnus: halbjährlich (WS+SS)
Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Deutsch

---

**Lehrveranstaltungen:**

Cyber-Physical Systems (SS 2019, Vorlesung, 2 SWS, Torsten Klie)  
 Übung zu Cyber-Physical Systems (SS 2019, Übung, 2 SWS, Torsten Klie)

---

**Inhalt:**

Klassische Computersysteme zeichnen sich durch eine strikte Trennung von realer und virtueller Welt aus. Moderne Steuerungssysteme, die z.B. in modernen Fahrzeugen verbaut sind und die aus einer Vielzahl von Sensoren und Aktoren bestehen, entsprechen diesem Bild nur sehr eingeschränkt. Diese Systeme, oft "Cyber-physical Systems (CPS)" genannt, erkennen ihre physische Umgebung, verarbeiten diese Informationen und können die physische Umwelt auch koordiniert beeinflussen. Hierzu ist eine starke Kopplung von physischem Anwendungsmodell und dem Computer-Steuerungsmodell nötig. Im Unterschied zu Eingebetteten Systemen bestehen CPS meist aus vielen vernetzten Komponenten, die sich selbständig untereinander koordinieren.

**Lernziele und Kompetenzen:**

- Was sind Cyber-physical Systems? (Definitionen, Abgrenzung zu eingebetteten Systemen, Ubiquitous Computing, etc.)
- Kontrolltheorie und Echtzeitanforderungen
- Selbstorganisationsprinzipien ("Self-X", Autonomie, Verhandlungen)
- Anwendungen für Cyber-physical Systems (Beispiele für existierende oder visionäre zukünftige Anwendungen im Bereich Verkehr, Medizintechnik, u.a.)
- Entwurfsmethoden für Cyber-physical Systems (Modellierung, Programmierung, Model-Integrated Development)

**Literatur:**

- Otto Föllinger Regelungstechnik. Hüthig 1992.
  - Rolf Isermann Mechatronische Systeme. Springer 2008.
  - Hilmar Jaschek und Holger Voos Grundkurs der Regelungstechnik. Oldenbourg 2010.
  - Wolfgang Schneider Praktische Regelungstechnik. Vieweg+Teubner 2008.
  - Wayne Wolf Computers as Components: Principles of Embedded Computing System Design. Elsevier 2008
- 

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

**[1] Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule)

Dieses Modul ist daneben auch in den Studienfächern "Computational Engineering (Master of Science with Honours)", "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Bachelor of Science)", "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Master of Science)", "Informatik (Master of Science)", "Informations- und Kommunikationstechnik (Bachelor of Science)", "Mathematik (Bachelor of Science)", "Mechatronik (Bachelor of Science)", "Mechatronik (Master of Science)", "Medizintechnik (Bachelor of Science)", "Medizintechnik (Master of Science)" verwendbar.

---

**Studien-/Prüfungsleistungen:**

Cyber-Physical Systems (Prüfungsnummer: 451696)

(englische Bezeichnung: Cyber-Physical Systems)

Prüfungsleistung, Klausur, Dauer (in Minuten): 90

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Cyber-Physical Systems
- Übung zu Cyber-Physical Systems

weitere Erläuterungen:

Die Note ergibt sich zu 100% aus der 90 minütigen Klausur.

Erstablingung: SS 2019, 1. Wdh.: WS 2019/2020

1. Prüfer: Torsten Klie (100126)

Mündliche Prüfung Cyber-Physical Systems (Prüfungsnummer: 44701)

(englische Bezeichnung: Oral Examination on Cyber-Physical Systems)

Prüfungsleistung, mündliche Prüfung, Dauer (in Minuten): 30

Anteil an der Berechnung der Modulnote: 100%

Erstablingung: SS 2019, 1. Wdh.: WS 2019/2020

1. Prüfer: Torsten Klie (100126)

---

---

**Modulbezeichnung:** Hardware-Software-Co-Design (HSCD-VU) 5 ECTS  
 (Hardware-Software-Co-Design)

Modulverantwortliche/r: Jürgen Teich  
 Lehrende: Jürgen Teich

---

Startsemester: SS 2019	Dauer: 1 Semester	Turnus: jährlich (SS)
Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Deutsch

---

**Lehrveranstaltungen:**

Hardware-Software-Co-Design (SS 2019, Vorlesung, 2 SWS, Jürgen Teich)  
 Übungen zu Hardware-Software-Co-Design (SS 2019, Übung, 2 SWS, Franz-Josef Streit et al.)

---

**Inhalt:**

1. Überblick und Vergleich von Architekturen und Komponenten in Hardware/Software-Systemen.
2. Aufbau eines Compilers und Codeoptimierungsverfahren für Hardware und Software
3. Hardware/Software-Partitionierung (Partitionierung komplexer Systeme, Schätzungsverfahren, Performanzanalyse, Codegenerierung)
4. Interfacesynthese (Kommunikationsarten, Synchronisation, Synthese)
5. Verifikation und Cosimulation
6. Tafelübungen

**Lernziele und Kompetenzen:**

Zahlreiche Realisierungen eingebetteter Systeme (z.B. Mobiltelefone, Faxgeräte, Industriesteuerungen) zeichnen sich durch kooperierende Hardware- und Softwarekomponenten aus. Die Popularität solcher Realisierungsformen lässt sich begründen durch 1) die steigende Vielfalt und Komplexität heterogener Systeme, 2) die Notwendigkeit, Entwurfs- und Testkosten zu senken und 3) Fortschritte in Schlüsseltechnologien (Mikroelektronik, formale Entwurfsmethoden). Zum Beispiel bieten Halbleiterhersteller kostengünstige ASICs an, die einen Mikrocontroller und benutzerspezifische Peripherie und Datenpfade auf einem Chip integrieren.

Die Synthese solcher Systeme wirft jedoch eine Reihe neuartiger Entwurfsprobleme auf, insbesondere 1) die Frage der Auswahl von Hardware- und Softwarekomponenten, 2) die Partitionierung einer Spezifikation in Hard- und Software, 3) die automatische Synthese von Interface- und Kommunikationsstrukturen und 4) die Verifikation und Cosimulation.

*Fachkompetenz*

*Wissen*

- Die Studierenden erhalten Einblick in ein aktuelles Forschungsgebiet.

*Verstehen*

- Die Studierenden verstehen Grundlagen des modernen Systementwurfs.
- Die Studierenden erklären Implementierungsalternativen für digitale Hardware/Software-Systeme.

*Anwenden*

- Die Studierenden wenden grundlegende Algorithmen an, zur Analyse und Optimierung von Hardware/Software-Systemen.

**Literatur:**

Teich, J.; Haubelt, C.: Digitale Hardware/Software-Systeme: Synthese und Optimierung; Springer, Berlin; Auflage: 2. erw. Aufl. (2. März 2007)  
 Teich, J.: Hardware/Software-Architekturen. Ergänzendes Skriptum zur Vorlesung.  
 Gajski, D.: Specification and Design of Embedded Systems. Prentice Hall, Englewood Cliffs, NJ, 1994.

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

[1] **Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule)

Dieses Modul ist daneben auch in den Studienfächern "Computational Engineering (Master of Science with Honours)", "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Bachelor of Science)", "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Master of Science)", "Information and Communication Technology (Master of Science)", "Informations- und Kommunikationstechnik (Bachelor of Science)", "Informations- und Kommunikationstechnik (Master of Science)", "Mathematik (Bachelor of Science)", "Mechatronik (Bachelor of Science)", "Mechatronik (Master of Science)", "Medizintechnik (Master of Science)" verwendbar.

---

### **Studien-/Prüfungsleistungen:**

Hardware-Software-Co-Design (Prüfungsnummer: 502509)

Prüfungsleistung, Klausur, Dauer (in Minuten): 90

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Hardware-Software-Co-Design
- Übungen zu Hardware-Software-Co-Design

weitere Erläuterungen:

Die Modulnote ergibt sich aus der Klausurnote.

Erstablingung: SS 2019, 1. Wdh.: WS 2019/2020

1. Prüfer: Jürgen Teich (100115)

---

### **Organisatorisches:**

Die Auswahl dieses Moduls schließt die Auswahl des Moduls „Hardware-Software-Co-Design mit erweiterter Übung (HSCD-VEU)“ aus.

### **Bemerkungen:**

auch für Computational Engineering



---

**Modulbezeichnung:** **Approximate Computing (APPROXC)** **5 ECTS**  
(Approximate Computing)

Modulverantwortliche/r: Oliver Keszöcze, Jürgen Teich

Lehrende: Oliver Keszöcze, Jürgen Teich

---

Startsemester: SS 2019

Dauer: 1 Semester

Turnus: jährlich (SS)

Präsenzzeit: 60 Std.

Eigenstudium: 90 Std.

Sprache: Englisch

---

**Lehrveranstaltungen:**

Approximate Computing (SS 2019, Vorlesung, 2 SWS, Oliver Keszöcze et al.)

---



---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

**[1] Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule)

---

**Studien-/Prüfungsleistungen:**

Approximate Computing (Prüfungsnummer: 965820)

(englische Bezeichnung: Approximate Computing)

Prüfungsleistung, mündliche Prüfung, Dauer (in Minuten): 30

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Approximate Computing

weitere Erläuterungen:

On default, an oral examination of duration 30 minutes will determine your grade. However, if a large number of students participate, your grade will be determined by a written examination of duration 60 minutes.

Prüfungssprache: Englisch

Erstablingung: SS 2019, 1. Wdh.: keine Angabe

1. Prüfer: Oliver Keszöcze (100421), 2. Prüfer: Jürgen Teich (100115)

---

---

**Modulbezeichnung:** Parallele Systeme mit erweiterter Übung (PSYS-VEU) 7.5 ECTS  
 (Parallel Systems with Extended Exercises)

Modulverantwortliche/r: Jürgen Teich

Lehrende: Jürgen Teich, Frank Hannig

---

Startsemester: SS 2019

Dauer: 1 Semester

Turnus: jährlich (SS)

Präsenzzeit: 90 Std.

Eigenstudium: 135 Std.

Sprache: Deutsch oder Englisch

---

**Lehrveranstaltungen:**

Parallele Systeme (SS 2019, Vorlesung, 2 SWS, Jürgen Teich et al.)

Erweiterte Übungen zu Parallele Systeme (SS 2019, Übung, 2 SWS, Marcel Brand et al.)

Übung zu Parallele Systeme (SS 2019, Übung, 2 SWS, Frank Hannig)

---

**Inhalt:**

Aktuelle PCs verfügen über Mehrkernprozessoren und Grafikkarten, die wiederum aus hunderten von einfachen Prozessoren bestehen können. Hierdurch wird ein hohes Maß an nebenläufiger Datenverarbeitung möglich, welche bis vor einigen Jahren nur in Großrechnern erreicht werden konnte. Die effiziente Ausnutzung dieser Parallelität bedarf allerdings mehr als nur mehrerer Prozessoren, insbesondere muss das zu lösende Problem Parallelverarbeitung erlauben. In dieser Vorlesung werden Eigenschaften unterschiedlicher paralleler Rechnerarchitekturen und Metriken zu deren Beurteilung behandelt. Weiterhin werden Modelle und Sprachen zum Programmieren paralleler Rechner eingeführt. Neben der Programmierung von allgemeinen Parallelrechnern werden Entwurfsmethoden (CAD) vorgestellt, wie man ausgehend von einer algorithmischen Problemstellung ein massiv paralleles Rechenfeld in VLSI herleiten kann, das genau dieses Problem optimal parallel berechnet. Solche Schaltungen spielen auf der Bit- bzw. Wortebene eine dominante Rolle (Arithmetik) sowie bei Problemen der Signal- und Bildverarbeitung (z.B. Filter).

Im Einzelnen werden behandelt:

1. Theorie der Parallelität (parallele Computermodelle, parallele Spezifikationsformen und -sprachen, Performanzmodelle und -berechnung)
2. Klassifikation paralleler und skalierbarer Rechnerarchitekturen (Multiprozessoren und Multicomputer, Vektorrechner, Datenflussmaschinen, VLSI-Rechenfelder)
3. Programmierbare System-on-Chip (SoC) und Mehrkern-Architekturen (Grafik-Prozessoren, Cell, etc.)
4. Programmierung paralleler Rechner (Sprachen und Modelle, Entwurfsmethoden und Compiler, Optimierung)
5. Massive Parallelität: Vom Algorithmus zur Schaltung
6. Praktische Übungen mit rechnergestützten Werkzeugen

*Today's PCs consist of multi-core processors and graphics cards that again comprise hundreds to thousands of simple processors. As a result of this, a very high degree of parallel data processing becomes possible, which was subjected to supercomputers a couple of years ago. The efficient exploitation of parallel processing requires not only multiple processors but also parallelism inherent in the problem to be solved. In this lecture, properties of different parallel computer architectures and corresponding quality metrics are examined. Further, models and parallel programming languages are introduced. In addition to programming general parallel computers, design methods (CAD) are presented that systematically transform an algorithmic problem description into a massive parallel processor array (VLSI), which can optimally execute the given problem in parallel. Such highly parallel circuits play an essential role at the bit level and circuit level (arithmetics) as well as in the case of signal processing and image processing (e.g., filter).*

*In detail, the following topics are covered:*

1. *Theory of parallelism (parallel models of computation, parallel specification and parallel languages, performance models)*
2. *Classification of parallel and scalable computer architectures (multi-processors and multi-computers, vector computers, data-flow machines, VLSI processor arrays)*
3. *Programmable System-on-Chip (SoC) and multi-core architectures (graphics processors, Cell, etc.)*

4. Programming of parallel computers (languages and models, design methods and compiler, optimization)
5. Massive parallelism: From algorithm to circuit
6. Practical training with computer-aided design tools

#### Lernziele und Kompetenzen:

Schwerpunkt der Vorlesung ist die Vermittlung von Grundlagen der parallelen Datenverarbeitung. / *The focus of this lecture are foundations of parallel data processing.*

Fachkompetenz

Verstehen

- Die Studierenden verstehen grundlegende Konzepte der parallelen Datenverarbeitung, sowohl theoretischer Art anhand von Modellen, als auch an Architekturbeispielen. / *The students become familiar with the fundamentals of parallel data processing, theoretic in the form of models as well as by architecture examples.*

Anwenden

- Die Studierenden setzen sich mit modernen eingebetteten parallelen Ein-Chip-Architekturen auseinander. / *The students get familiar with modern embedded parallel system-on-chip architectures.*
- Die Studierenden wenden grundlegende Performanzmodelle und Parallelisierungstechniken zur Analyse und Optimierung von parallelen Algorithmen und Architekturen an. / *The students exercise basic performance models and parallelization techniques for the analysis and optimization of parallel algorithms and architectures.*
- Die Studierenden setzen die Modellierung und den Entwurf von massiv-parallelen Prozessorfeldern in konkreten Aufgaben selbstständig um. / *In concrete tasks, the students apply independently the modeling and the design of massively parallel processors arrays.*
- Die Studierenden wenden das erlernte Wissen in Rechnerübungen an. / *The students apply their learned knowledge in hands-on computer exercises.*

#### Literatur:

Siehe Webseite: <https://www.cs12.tf.fau.de/lehre/lehveranstaltungen/vorlesungen/parallelsysteme/>

#### Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

##### [1] Informatik (Bachelor of Science)

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule)

#### Studien-/Prüfungsleistungen:

Parallele Systeme mit erweiterter Übung (Prüfungsnummer: 740665)

(englische Bezeichnung: Parallel Systems with Extended Exercises)

Prüfungsleistung, Klausur, Dauer (in Minuten): 90

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Parallele Systeme
- Erweiterte Übungen zu Parallele Systeme
- Übung zu Parallele Systeme

weitere Erläuterungen:

Mündliche (Dauer: 30 min) oder schriftliche (Dauer: 90 min) Prüfung + erfolgreiche Bearbeitung aller Übungsaufgaben in den erweiterten Übungen (verpflichtend). Die Modulnote ergibt sich aus der mündlichen oder schriftlichen Prüfung.

Prüfungssprache: Deutsch oder Englisch

Erstablesung: SS 2019, 1. Wdh.: WS 2019/2020, 2. Wdh.: SS 2020

1. Prüfer: Jürgen Teich (100115), 2. Prüfer: Frank Hannig (100192)

**Organisatorisches:**

Die Auswahl dieses Moduls schließt die Auswahl des Moduls „Parallele Systeme (PSYS-VU)“ aus.

**Bemerkungen:**

auch für Computational Engineering

---

**Modulbezeichnung:** Hardware-Software-Co-Design mit erweiterter Übung (HSCD-VEU) 7.5 ECTS  
 (Hardware-Software-Co-Design with Extended Exercises)

Modulverantwortliche/r: Jürgen Teich  
 Lehrende: Jürgen Teich

---

Startsemester: SS 2019	Dauer: 1 Semester	Turnus: jährlich (SS)
Präsenzzeit: 90 Std.	Eigenstudium: 135 Std.	Sprache: Deutsch oder Englisch

---

**Lehrveranstaltungen:**

Hardware-Software-Co-Design (SS 2019, Vorlesung, 2 SWS, Jürgen Teich)  
 Übungen zu Hardware-Software-Co-Design (SS 2019, Übung, 2 SWS, Franz-Josef Streit et al.)  
 Erweiterte Übungen zu Hardware-Software-Co-Design (SS 2019, Übung, 2 SWS, Franz-Josef Streit)

**Inhalt:**

1. Überblick und Vergleich von Architekturen und Komponenten in Hardware/Software-Systemen.
2. Aufbau eines Compilers und Codeoptimierungsverfahren für Hardware und Software
3. Hardware/Software-Partitionierung (Partitionierung komplexer Systeme, Schätzungsverfahren, Performanzanalyse, Codegenerierung)
4. Interfacesynthese (Kommunikationsarten, Synchronisation, Synthese)
5. Verifikation und Cosimulation
6. Tafelübungen
7. Demonstrationen mit rechnergestützten Entwurfswerkzeugen und praktische Übungen

**Lernziele und Kompetenzen:**

Zahlreiche Realisierungen eingebetteter Systeme (z.B. Mobiltelefone, Faxgeräte, Industriesteuerungen) zeichnen sich durch kooperierende Hardware- und Softwarekomponenten aus. Die Popularität solcher Realisierungsformen lässt sich begründen durch 1) die steigende Vielfalt und Komplexität heterogener Systeme, 2) die Notwendigkeit, Entwurfs- und Testkosten zu senken und 3) Fortschritte in Schlüsseltechnologien (Mikroelektronik, formale Entwurfsmethoden). Zum Beispiel bieten Halbleiterhersteller kostengünstige ASICs an, die einen Mikrocontroller und benutzerspezifische Peripherie und Datenpfade auf einem Chip integrieren.

Die Synthese solcher Systeme wirft jedoch eine Reihe neuartiger Entwurfsprobleme auf, insbesondere 1) die Frage der Auswahl von Hardware- und Softwarekomponenten, 2) die Partitionierung einer Spezifikation in Hard- und Software, 3) die automatische Synthese von Interface- und Kommunikationsstrukturen und 4) die Verifikation und Cosimulation.

*Fachkompetenz*

*Wissen*

- Die Studierenden erhalten Einblick in ein aktuelles Forschungsgebiet.

*Verstehen*

- Die Studierenden verstehen Grundlagen des modernen Systementwurfs.
- Die Studierenden erklären Implementierungsalternativen für digitale Hardware/Software-Systeme.

*Anwenden*

- Die Studierenden wenden grundlegende Algorithmen an, zur Analyse und Optimierung von Hardware/Software-Systemen.

*Sozialkompetenz*

- Die Studierenden benutzen aktuelle Entwurfswerkzeuge für die Spezifikation, Optimierung und Prototypisierung von Hardware/Software-Systemen bei der kooperativen Bearbeitung der erweiterten Übung in Gruppen.

**Literatur:**

Teich, J.; Haubelt, C.: Digitale Hardware/Software-Systeme: Synthese und Optimierung; Springer, Berlin; Auflage: 2. erw. Aufl. (2. März 2007)  
 Teich, J.: Hardware/Software-Architekturen. Ergänzendes Skriptum zur Vorlesung.

Gajski, D.: Specification and Design of Embedded Systems. Prentice Hall, Englewood Cliffs, NJ, 1994.

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

**[1] Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule)

Dieses Modul ist daneben auch in den Studienfächern "Computational Engineering (Master of Science with Honours)", "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Bachelor of Science)", "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Master of Science)", "Mechatronik (Bachelor of Science)", "Mechatronik (Master of Science)" verwendbar.

---

**Studien-/Prüfungsleistungen:**

Hardware-Software-Co-Design (Vorlesung mit erweiterter Übung) (Prüfungsnummer: 292952)

Prüfungsleistung, Portfolio

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Hardware-Software-Co-Design
- Übungen zu Hardware-Software-Co-Design
- Erweiterte Übungen zu Hardware-Software-Co-Design

weitere Erläuterungen:

Portfolio besteht aus: Klausur (90 min) und erfolgreicher Teilnahme an den erweiterten Übungen (verpflichtend)

Die Modulnote ergibt sich aus der Klausurnote.

Erstablingung: SS 2019, 1. Wdh.: WS 2019/2020

1. Prüfer: Jürgen Teich (100115)

---

**Organisatorisches:**

Die Auswahl dieses Moduls schließt die Auswahl des Moduls „Hardware-Software-Co-Design (HSCD-VU)“ aus.

**Bemerkungen:**

auch für Computational Engineering

---

**Modulbezeichnung: Parallele Systeme (PSYS-VU)**  
 (Parallel Systems)

**5 ECTS**

Modulverantwortliche/r: Jürgen Teich

Lehrende: Jürgen Teich, Frank Hannig

Startsemester: SS 2019

Dauer: 1 Semester

Turnus: jährlich (SS)

Präsenzzeit: 60 Std.

Eigenstudium: 90 Std.

Sprache: Deutsch oder Englisch

---

**Lehrveranstaltungen:**

Parallele Systeme (SS 2019, Vorlesung, 2 SWS, Jürgen Teich et al.)

Übung zu Parallele Systeme (SS 2019, Übung, 2 SWS, Frank Hannig)

---

**Inhalt:**

Aktuelle PCs verfügen über Mehrkernprozessoren und Grafikkarten, die wiederum aus hunderten von einfachen Prozessoren bestehen können. Hierdurch wird ein hohes Maß an nebenläufiger Datenverarbeitung möglich, welche bis vor einigen Jahren nur in Großrechnern erreicht werden konnte. Die effiziente Ausnutzung dieser Parallelität bedarf allerdings mehr als nur mehrerer Prozessoren, insbesondere muss das zu lösende Problem Parallelverarbeitung erlauben. In dieser Vorlesung werden Eigenschaften unterschiedlicher paralleler Rechnerarchitekturen und Metriken zu deren Beurteilung behandelt. Weiterhin werden Modelle und Sprachen zum Programmieren paralleler Rechner eingeführt. Neben der Programmierung von allgemeinen Parallelrechnern werden Entwurfsmethoden (CAD) vorgestellt, wie man ausgehend von einer algorithmischen Problemstellung ein massiv paralleles Rechenfeld in VLSI herleiten kann, das genau dieses Problem optimal parallel berechnet. Solche Schaltungen spielen auf der Bit- bzw. Wortebene eine dominante Rolle (Arithmetik) sowie bei Problemen der Signal- und Bildverarbeitung (z.B. Filter).

Im Einzelnen werden behandelt:

1. Theorie der Parallelität (parallele Computermodelle, parallele Spezifikationsformen und -sprachen, Performanzmodelle und -berechnung)
2. Klassifikation paralleler und skalierbarer Rechnerarchitekturen (Multiprozessoren und Multicomputer, Vektorrechner, Datenflussmaschinen, VLSI-Rechenfelder)
3. Programmierbare System-on-Chip (SoC) und Mehrkern-Architekturen (Grafik-Prozessoren, Cell, etc.)
4. Programmierung paralleler Rechner (Sprachen und Modelle, Entwurfsmethoden und Compiler, Optimierung)
5. Massive Parallelität: Vom Algorithmus zur Schaltung

*Today's PCs consist of multi-core processors and graphics cards that again comprise hundreds to thousands of simple processors. As a result of this, a very high degree of parallel data processing becomes possible, which was subjected to supercomputers a couple of years ago. The efficient exploitation of parallel processing requires not only multiple processors but also parallelism inherent in the problem to be solved. In this lecture, properties of different parallel computer architectures and corresponding quality metrics are examined. Further, models and parallel programming languages are introduced. In addition to programming general parallel computers, design methods (CAD) are presented that systematically transform an algorithmic problem description into a massive parallel processor array (VLSI), which can optimally execute the given problem in parallel. Such highly parallel circuits play an essential role at the bit level and circuit level (arithmetics) as well as in the case of signal processing and image processing (e.g., filter).*

*In detail, the following topics are covered:*

1. *Theory of parallelism (parallel models of computation, parallel specification and parallel languages, performance models)*
2. *Classification of parallel and scalable computer architectures (multi-processors and multi-computers, vector computers, data-flow machines, VLSI processor arrays)*
3. *Programmable System-on-Chip (SoC) and multi-core architectures (graphics processors, Cell, etc.)*
4. *Programming of parallel computers (languages and models, design methods and compiler, optimization)*

## 5. Massive parallelism: From algorithm to circuit

### Lernziele und Kompetenzen:

Schwerpunkt der Vorlesung ist die Vermittlung von Grundlagen der parallelen Datenverarbeitung. / *The focus of this lecture are foundations of parallel data processing.*

#### Fachkompetenz

##### Verstehen

- Die Studierenden verstehen grundlegende Konzepte der parallelen Datenverarbeitung, sowohl theoretischer Art anhand von Modellen, als auch an Architekturbeispielen. / *The students become familiar with the fundamentals of parallel data processing, theoretic in the form of models as well as by architecture examples.*

##### Anwenden

- Die Studierenden setzen sich mit modernen eingebetteten parallelen Ein-Chip-Architekturen auseinander. / *The students get familiar with modern embedded parallel system-on-chip architectures.*
- Die Studierenden wenden grundlegende Performanzmodelle und Parallelisierungstechniken zur Analyse und Optimierung von parallelen Algorithmen und Architekturen an. / *The students exercise basic performance models and parallelization techniques for the analysis and optimization of parallel algorithms and architectures.*
- Die Studierenden setzen die Modellierung und den Entwurf von massiv-parallelen Prozessorfeldern in konkreten Aufgaben selbstständig um. / *In concrete tasks, the students apply independently the modeling and the design of massively parallel processors arrays.*

### Literatur:

Siehe Webseite: <https://www.cs12.tf.fau.de/lehre/lehrveranstaltungen/vorlesungen/parallele-systeme/>

### Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

#### [1] Informatik (Bachelor of Science)

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule)

Dieses Modul ist daneben auch in den Studienfächern "Computational Engineering (Master of Science with Honours)", "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Bachelor of Science)", "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Master of Science)", "Informatik (Master of Science)", "Informations- und Kommunikationstechnik (Master of Science)", "Mathematik (Bachelor of Science)", "Mechatronik (Bachelor of Science)", "Mechatronik (Master of Science)", "Medizintechnik (Master of Science)" verwendbar.

### Studien-/Prüfungsleistungen:

Parallele Systeme (Prüfungsnummer: 967118)

(englische Bezeichnung: Parallel Systems)

Prüfungsleistung, Klausur, Dauer (in Minuten): 90

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Parallele Systeme
- Übung zu Parallele Systeme

weitere Erläuterungen:

Mündliche (Dauer: 30 min) oder schriftliche (Dauer: 90 min) Prüfung. Die Modulnote ergibt sich aus der Prüfung.

Prüfungssprache: Deutsch oder Englisch

Erstablingung: SS 2019, 1. Wdh.: WS 2019/2020, 2. Wdh.: SS 2020

1. Prüfer: Jürgen Teich (100115), 2. Prüfer: Frank Hannig (100192)



## Parallele Systeme (Prüfungsnummer: 35101)

(englische Bezeichnung: Parallel Systems)

Prüfungsleistung, Klausur, Dauer (in Minuten): 90

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Parallele Systeme
- Übung zu Parallele Systeme

weitere Erläuterungen:

Mündliche (Dauer: 30 min) oder schriftliche (Dauer: 90 min) Prüfung. Die Modulnote ergibt sich aus der Prüfung.

Prüfungssprache: Deutsch oder Englisch

Erstablingung: SS 2019, 1. Wdh.: WS 2019/2020

1. Prüfer: Jürgen Teich (100115), 2. Prüfer: Frank Hannig (100192)

---

### **Organisatorisches:**

Die Auswahl dieses Moduls schließt die Auswahl des Moduls „Parallele Systeme mit erweiterter Übung (PSYS-VEU)“ aus.

### **Bemerkungen:**

auch für Computational Engineering

**Modulbezeichnung:** Security in Embedded Hardware (SEH) 5 ECTS  
(Security in Embedded Hardware)

Modulverantwortliche/r: Jürgen Teich  
Lehrende: Jürgen Teich

Startsemester: SS 2019      Dauer: 1 Semester      Turnus: jährlich (SS)  
Präsenzzeit: 60 Std.      Eigenstudium: 90 Std.      Sprache: Deutsch oder Englisch

**Lehrveranstaltungen:**

Security in Embedded Hardware (SS 2019, Vorlesung, 2 SWS, Jürgen Teich)  
Übung zu Security in Embedded Hardware (SS 2019, Übung, 2 SWS, Jürgen Teich)

**Inhalt:**

Der Schutz eingebetteter Systeme gegenüber Angriffe Dritter auf gespeicherte Daten und Implementierungen, stellt eine immer wichtigere, jedoch auch durch zunehmende Vernetzung herausfordernde Aufgabe dar. Der Schutz der eingebetteten Systeme gegenüber bekannten als auch neueren ausgeklügelten Angriffsmöglichkeiten ist Gegenstand dieser Vorlesung. Es wird gezeigt, welche Angriffe existieren, welche Gegenmaßnahmen man ergreifen kann und wie man sichere eingebettete Systeme entwirft.

Einleitung und Motivation

- Was ist Security?
- Die Bedeutung von Security für zuverlässige Systeme
- Klassifikation von Angriffen
- Entwurf eingebetteter Systeme

Angriffsszenarien

- Beispiele von Angriffsszenarien
- Kryptographischer Algorithmen als Ziel von Angriffen

Angriffe durch Einschleusen von Code (Code Injection Attacks)

- Welche Arten von Code Injection-Angriffe gibt es?
- Gegenmaßnahmen

Invasive physikalische Angriffe (Invasive Physical Attacks)

- Microprobing
- Reverse Engineering
- Differential Fault Analysis
- Gegenmaßnahmen

Nichtinvasive softwarebasierte Angriffe (Non-Invasive Logical Attacks)

- Erlangen von nicht autorisiertem Zugriff
- Gegenmaßnahmen

Nichtinvasive physikalische Angriffe (Non-Invasive Physical Attacks)

- Abhören
- Seitenkanalangriffe
- Gegenmaßnahmen

**Lernziele und Kompetenzen:**

*Fachkompetenz*

*Wissen*

- Die Studierenden legen die entsprechenden Gegenmaßnahmen dar
- Die Studierenden nennen verschiedene Sicherheitseinrichtungen und -maßnahmen in eingebetteten Systemen

*Verstehen*

- Die Studierenden zeigen den Einfluss von Angriffen und deren Gegenmaßnahmen auf die Verlässlichkeit eines eingebetteten Systems auf
- Die Studierenden zeigen den zusätzlichen Aufwand (Fläche, Rechenzeit) von Sicherheitseinrichtungen auf

*Analysieren*

- Die Studierenden klassifizieren verschiedene Angriffstypen auf eingebettete Systeme

#### *Sozialkompetenz*

- Die Studierenden erarbeiten kooperativ in Gruppen Lösungskonzepte und implementieren diese gemeinsam

#### **Literatur:**

- Catherine H. Gebotys Security in Embedded Devices. Springer 2010.
- Benoit Badrignans et al. Security Trends for FPGAs. Springer 2011.
- Daniel Ziener Techniques for Increasing Security and Reliability of IP Cores Embedded in FPGA and ASIC Designs. Dr. Hut 2010.

---

#### **Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

##### **[1] Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule)

---

#### **Studien-/Prüfungsleistungen:**

Security in Embedded Hardware (Prüfungsnummer: 172338)

Prüfungsleistung, Klausur, Dauer (in Minuten): 60

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Security in Embedded Hardware
- Übung zu Security in Embedded Hardware

Prüfungssprache: Deutsch oder Englisch

Erstablingung: SS 2019, 1. Wdh.: WS 2019/2020 (nur für Wiederholer)

1. Prüfer: Jürgen Teich (100115)

---

---

**Modulbezeichnung:** Entwurf und Analyse eingebetteter Netzwerke des 5 ECTS  
**Automobilbaus (EAENA)**  
 (Design and Analysis of Embedded Networks in the Automotive Domain)

Modulverantwortliche/r: Jürgen Teich  
 Lehrende: Hritam Dutta, Fedor Smirnov

---

Startsemester: SS 2019	Dauer: 1 Semester	Turnus: jährlich (WS)
Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Deutsch

---

**Lehrveranstaltungen:**

Entwurf und Analyse eingebetteter Netzwerke des Automobilbaus (SS 2019, Vorlesung, 2 SWS, Hritam Dutta)

Übung zu Entwurf und Analyse eingebetteter Netzwerke des Automobilbaus (SS 2019, Übung, 2 SWS, Fedor Smirnov)

---

**Inhalt:**

Heutige Netzwerkarchitekturen bestehen aus bis 80 Steuergeräten, die über verschiedene Bussysteme vernetzt sind. Sowohl die Steuergeräte wie die Bussysteme sind durch eine hohe Diversität geprägt: Von kleinen kommunizierenden Sensoren mit 8-Bit  $\mu$ Controller und LIN-Interface, über bildverarbeitende Kameras mit CAN-Anschluss bis hin zu 32-Bit  $\mu$ Controllern in Gateways mit einem oder mehreren CAN-, FlexRay- und MOST-Bussen. Diese Vielzahl an Ressourcen, die für den Entwurf von Netzwerkarchitekturen bereit steht, spannt einen riesigen Entwurfsraum auf, der allerdings zahlreichen Randbedingungen unterliegt. Diese Randbedingungen sind in erster Linie durch die verschiedenen Disziplinen wie Maschinenbau, Design, Vertrieb, Wartung, etc. gegeben, die im Automobilbau aufeinander treffen. Um die verschiedenen sinnvollen Möglichkeiten aus Sicht eines Netzwerkentwicklers zu bewerten, spielen Analysemethoden zur frühzeitigen Bewertung von Netzwerken eine entscheidende Rolle. In dieser Vorlesung soll der Spagat zwischen Formeln und praktischer Anwendung gemeistert und gleichzeitig der aktuelle Stand der Forschung im Bereich der Analyse von Echtzeitsystemen vermittelt werden.

- Grundlagen des automobilen Systementwurfs
- Grundlagen der Kommunikationssysteme
- Analyse eingebetteter Netzwerke
- Eingebettete Rechnerstrukturen für Echtzeitsysteme
- Modellbasierte Software-Entwicklung und Echtzeitbetriebssysteme
- Analyse von Software für eingebettete Systeme

**Lernziele und Kompetenzen:**

*Fachkompetenz*

*Wissen*

- Die Studierenden nennen Methoden der Software-Analyse.

*Verstehen*

- Die Studierenden vergleichen grundlegende Kommunikationsarchitekturen des Automobilbaus und erörtern deren Vor- und Nachteile.

*Anwenden*

- Die Studierenden wenden Verfahren zur Analyse und Optimierung von Echtzeitsystemen im Automobil an.

**Literatur:**

- Hermann Kopetz, Real-Time Systems: Design Principles for Distributed Embedded Applications Kluwer International Series in Engineering & Computer Science, ISBN-10: 0792398947 ISBN-13: 978-0792398943
- Jörg Schäuffele und Thomas Zurawka, Automotive Software Engineering, Vieweg+Teubner, ISBN-10: 3834800511 ISBN-13: 978-3834800510
- Giorgio C. Buttazzo, Hard Real-time Computing Systems: Predictable Scheduling Algorithms and Applications, Springer, ISBN-10: 0387231374 ISBN-13: 978-0387231372

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

**[1] Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule)

---

**Studien-/Prüfungsleistungen:**

Entwurf und Analyse eingebetteter Netzwerke des Automobilbaus\_ (Prüfungsnummer: 491597)

(englische Bezeichnung: Design and Analysis of Embedded Networks in the Automotive Domain)

Prüfungsleistung, Klausur, Dauer (in Minuten): 90

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Entwurf und Analyse eingebetteter Netzwerke des Automobilbaus
- Übung zu Entwurf und Analyse eingebetteter Netzwerke des Automobilbaus

Erstablingung: SS 2019, 1. Wdh.: WS 2019/2020

1. Prüfer: Jürgen Teich (100115)

---

---

**Modulbezeichnung:** Cyber-Physical Systems (CPS) **5 ECTS**  
 (Cyber-Physical Systems)

Modulverantwortliche/r: Torsten Klie  
 Lehrende: Torsten Klie

---

Startsemester: WS 2019/2020	Dauer: 1 Semester	Turnus: halbjährlich (WS+SS)
Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Deutsch

---

**Lehrveranstaltungen:**

Cyber-Physical Systems (WS 2019/2020, Vorlesung, 2 SWS, Torsten Klie)  
 Übung zu Cyber-Physical Systems (WS 2019/2020, Übung, Torsten Klie)

---

**Inhalt:**

Klassische Computersysteme zeichnen sich durch eine strikte Trennung von realer und virtueller Welt aus. Moderne Steuerungssysteme, die z.B. in modernen Fahrzeugen verbaut sind und die aus einer Vielzahl von Sensoren und Aktoren bestehen, entsprechen diesem Bild nur sehr eingeschränkt. Diese Systeme, oft "Cyber-physical Systems (CPS)" genannt, erkennen ihre physische Umgebung, verarbeiten diese Informationen und können die physische Umwelt auch koordiniert beeinflussen. Hierzu ist eine starke Kopplung von physischem Anwendungsmodell und dem Computer-Steuerungsmodell nötig. Im Unterschied zu Eingebetteten Systemen bestehen CPS meist aus vielen vernetzten Komponenten, die sich selbständig untereinander koordinieren.

**Lernziele und Kompetenzen:**

- Was sind Cyber-physical Systems? (Definitionen, Abgrenzung zu eingebetteten Systemen, Ubiquitous Computing, etc.)
- Kontrolltheorie und Echtzeitanforderungen
- Selbstorganisationsprinzipien ("Self-X", Autonomie, Verhandlungen)
- Anwendungen für Cyber-physical Systems (Beispiele für existierende oder visionäre zukünftige Anwendungen im Bereich Verkehr, Medizintechnik, u.a.)
- Entwurfsmethoden für Cyber-physical Systems (Modellierung, Programmierung, Model-Integrated Development)

**Literatur:**

- Otto Föllinger Regelungstechnik. Hüthig 1992.
  - Rolf Isermann Mechatronische Systeme. Springer 2008.
  - Hilmar Jaschek und Holger Voos Grundkurs der Regelungstechnik. Oldenbourg 2010.
  - Wolfgang Schneider Praktische Regelungstechnik. Vieweg+Teubner 2008.
  - Wayne Wolf Computers as Components: Principles of Embedded Computing System Design. Elsevier 2008
- 

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

**[1] Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule | Vertiefungsrichtung Hardware-Software-Co-Design)

Dieses Modul ist daneben auch in den Studienfächern "Computational Engineering (Master of Science with Honours)", "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Bachelor of Science)", "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Master of Science)", "Informatik (Master of Science)", "Informations- und Kommunikationstechnik (Bachelor of Science)", "Mathematik (Bachelor of Science)", "Mechatronik (Bachelor of Science)", "Mechatronik (Master of Science)", "Medizintechnik (Bachelor of Science)", "Medizintechnik (Master of Science)" verwendbar.

---

**Studien-/Prüfungsleistungen:**

Cyber-Physical Systems (Prüfungsnummer: 44701)

Prüfungsleistung, mündliche Prüfung, Dauer (in Minuten): 30

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Cyber-Physical Systems
- Übung zu Cyber-Physical Systems

weitere Erläuterungen:

Die Note ergibt sich zu 100% aus der 30 mündlichen Prüfung.

Erstablingung: WS 2019/2020, 1. Wdh.: SS 2020

1. Prüfer: Torsten Klie (100126)

---

---

**Modulbezeichnung:** Cyber-Physical Systems (CPS) 5 ECTS  
 (Cyber-Physical Systems)

Modulverantwortliche/r: Torsten Klie  
 Lehrende: Torsten Klie

---

Startsemester: SS 2019	Dauer: 1 Semester	Turnus: halbjährlich (WS+SS)
Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Deutsch

---

**Lehrveranstaltungen:**

Cyber-Physical Systems (SS 2019, Vorlesung, 2 SWS, Torsten Klie)  
 Übung zu Cyber-Physical Systems (SS 2019, Übung, 2 SWS, Torsten Klie)

---

**Inhalt:**

Klassische Computersysteme zeichnen sich durch eine strikte Trennung von realer und virtueller Welt aus. Moderne Steuerungssysteme, die z.B. in modernen Fahrzeugen verbaut sind und die aus einer Vielzahl von Sensoren und Aktoren bestehen, entsprechen diesem Bild nur sehr eingeschränkt. Diese Systeme, oft "Cyber-physical Systems (CPS)" genannt, erkennen ihre physische Umgebung, verarbeiten diese Informationen und können die physische Umwelt auch koordiniert beeinflussen. Hierzu ist eine starke Kopplung von physischem Anwendungsmodell und dem Computer-Steuerungsmodell nötig. Im Unterschied zu Eingebetteten Systemen bestehen CPS meist aus vielen vernetzten Komponenten, die sich selbständig untereinander koordinieren.

**Lernziele und Kompetenzen:**

- Was sind Cyber-physical Systems? (Definitionen, Abgrenzung zu eingebetteten Systemen, Ubiquitous Computing, etc.)
- Kontrolltheorie und Echtzeitanforderungen
- Selbstorganisationsprinzipien ("Self-X", Autonomie, Verhandlungen)
- Anwendungen für Cyber-physical Systems (Beispiele für existierende oder visionäre zukünftige Anwendungen im Bereich Verkehr, Medizintechnik, u.a.)
- Entwurfsmethoden für Cyber-physical Systems (Modellierung, Programmierung, Model-Integrated Development)

**Literatur:**

- Otto Föllinger Regelungstechnik. Hüthig 1992.
  - Rolf Isermann Mechatronische Systeme. Springer 2008.
  - Hilmar Jaschek und Holger Voos Grundkurs der Regelungstechnik. Oldenbourg 2010.
  - Wolfgang Schneider Praktische Regelungstechnik. Vieweg+Teubner 2008.
  - Wayne Wolf Computers as Components: Principles of Embedded Computing System Design. Elsevier 2008
- 

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

[1] **Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule | Vertiefungsrichtung Hardware-Software-Co-Design)

Dieses Modul ist daneben auch in den Studienfächern "Computational Engineering (Master of Science with Honours)", "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Bachelor of Science)", "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Master of Science)", "Informatik (Master of Science)", "Informations- und Kommunikationstechnik (Bachelor of Science)", "Mathematik (Bachelor of Science)", "Mechatronik (Bachelor of Science)", "Mechatronik (Master of Science)", "Medizintechnik (Bachelor of Science)", "Medizintechnik (Master of Science)" verwendbar.

---

**Studien-/Prüfungsleistungen:**



Mündliche Prüfung Cyber-Physical Systems (Prüfungsnummer: 44701)

(englische Bezeichnung: Oral Examination on Cyber-Physical Systems)

Prüfungsleistung, mündliche Prüfung, Dauer (in Minuten): 30

Anteil an der Berechnung der Modulnote: 100%

Erstablingung: SS 2019, 1. Wdh.: WS 2019/2020

1. Prüfer: Torsten Klie (100126)

---

**Modulbezeichnung:** Cyber-Physical Systems (CPS)  
 (Cyber-Physical Systems)

**5 ECTS**

Modulverantwortliche/r: Torsten Klie  
 Lehrende: Torsten Klie

Startsemester: WS 2019/2020      Dauer: 1 Semester      Turnus: halbjährlich (WS+SS)  
 Präsenzzeit: 60 Std.      Eigenstudium: 90 Std.      Sprache: Deutsch

**Lehrveranstaltungen:**

Cyber-Physical Systems (WS 2019/2020, Vorlesung, 2 SWS, Torsten Klie)  
 Übung zu Cyber-Physical Systems (WS 2019/2020, Übung, Torsten Klie)

**Inhalt:**

Klassische Computersysteme zeichnen sich durch eine strikte Trennung von realer und virtueller Welt aus. Moderne Steuerungssysteme, die z.B. in modernen Fahrzeugen verbaut sind und die aus einer Vielzahl von Sensoren und Aktoren bestehen, entsprechen diesem Bild nur sehr eingeschränkt. Diese Systeme, oft "Cyber-physical Systems (CPS)" genannt, erkennen ihre physische Umgebung, verarbeiten diese Informationen und können die physische Umwelt auch koordiniert beeinflussen. Hierzu ist eine starke Kopplung von physischem Anwendungsmodell und dem Computer-Steuerungsmodell nötig. Im Unterschied zu Eingebetteten Systemen bestehen CPS meist aus vielen vernetzten Komponenten, die sich selbständig untereinander koordinieren.

**Lernziele und Kompetenzen:**

- Was sind Cyber-physical Systems? (Definitionen, Abgrenzung zu eingebetteten Systemen, Ubiquitous Computing, etc.)
- Kontrolltheorie und Echtzeitanforderungen
- Selbstorganisationsprinzipien ("Self-X", Autonomie, Verhandlungen)
- Anwendungen für Cyber-physical Systems (Beispiele für existierende oder visionäre zukünftige Anwendungen im Bereich Verkehr, Medizintechnik, u.a.)
- Entwurfsmethoden für Cyber-physical Systems (Modellierung, Programmierung, Model-Integrated Development)

**Literatur:**

- Otto Föllinger Regelungstechnik. Hüthig 1992.
- Rolf Isermann Mechatronische Systeme. Springer 2008.
- Hilmar Jaschek und Holger Voos Grundkurs der Regelungstechnik. Oldenbourg 2010.
- Wolfgang Schneider Praktische Regelungstechnik. Vieweg+Teubner 2008.
- Wayne Wolf Computers as Components: Principles of Embedded Computing System Design. Elsevier 2008

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

**[1] Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule | Vertiefungsrichtung Hardware-Software-Co-Design)

Dieses Modul ist daneben auch in den Studienfächern "Computational Engineering (Master of Science with Honours)", "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Bachelor of Science)", "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Master of Science)", "Informatik (Master of Science)", "Informations- und Kommunikationstechnik (Bachelor of Science)", "Mathematik (Bachelor of Science)", "Mechatronik (Bachelor of Science)", "Mechatronik (Master of Science)", "Medizintechnik (Bachelor of Science)", "Medizintechnik (Master of Science)" verwendbar.

**Studien-/Prüfungsleistungen:**

Mündliche Prüfung Cyber-Physical Systems (Prüfungsnummer: 636348)

(englische Bezeichnung: Oral Examination on Cyber-Physical Systems)

Prüfungsleistung, Klausur, Dauer (in Minuten): 30

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Cyber-Physical Systems
- Übung zu Cyber-Physical Systems

weitere Erläuterungen:

Die Note ergibt sich zu 100% aus der 90 minütigen Klausur.

Erstablingung: WS 2019/2020, 1. Wdh.: SS 2020

1. Prüfer: Torsten Klie (100126)

---

---

**Modulbezeichnung:** Eingebettete Systeme mit erweiterter Übung (ES-VEU) 7.5 ECTS  
 (Embedded Systems with Extended Exercises)

Modulverantwortliche/r: Jürgen Teich

Lehrende: Jürgen Teich, Frank Hannig

---

Startsemester: WS 2019/2020

Dauer: 1 Semester

Turnus: jährlich (WS)

Präsenzzeit: 90 Std.

Eigenstudium: 135 Std.

Sprache: Deutsch

---

### Lehrveranstaltungen:

Eingebettete Systeme (WS 2019/2020, Vorlesung, 2 SWS, Jürgen Teich et al.)

Übung zu Eingebettete Systeme (WS 2019/2020, Übung, 2 SWS, Peter Brand et al.)

Erweiterte Übungen zu Eingebettete Systeme (WS 2019/2020, Übung, Peter Brand et al.)

---

### Inhalt:

Schwerpunkt der Vorlesung ist der Entwurf und die Implementierung eingebetteter Systeme unter Einsatz formaler Methoden und rechnergestützter Entwurfsverfahren.

Unter eingebetteten Systemen versteht man Rechensysteme, die auf einen Anwendungsbereich zugeschnitten (z.B. mobile Kommunikationsgeräte, Chipkartensysteme, Industriesteuerungen, Unterhaltungselektronik, Medizintechnik) und in einen technischen Kontext eingebunden sind. Das große Interesse am systematischen Entwurf von heterogenen eingebetteten Systemen ist verursacht durch die steigende Vielfalt und Komplexität von Anwendungen für eingebettete Systeme, die Notwendigkeit, Entwurfs- und Testkosten zu senken sowie durch Fortschritte in Schlüsseltechnologien (Mikroelektronik, formale Methoden).

*The focus of the lecture is the design and implementation of embedded systems using formal methods and computer-aided design techniques.*

*Embedded systems are computing systems tailored for a particular application (e.g., mobile communication devices, smart card systems, industrial control, consumer electronics, medical technology) and integrated into a technical context. The keen interest in the systematic design of heterogeneous embedded systems is driven by the increasing diversity and complexity of embedded system applications, the need to reduce design and test costs, and advances in key technologies (microelectronics, formal methods).*

### Lernziele und Kompetenzen:

#### Fachkompetenz

##### Wissen

- Die Studierenden setzen sich mit einem aktuellen Forschungsgebiet auseinander. / *The students deal with a current field of research.*
- Die Studierenden lernen aktuelle Entwurfswerkzeuge für die Architektursynthese (Hardware) und Softwaresynthese kennen. / *The students learn about current design tools for architectural synthesis (hardware) and software synthesis.*

##### Verstehen

- Die Studierenden verstehen grundlegende Konzepte des Entwurfs eingebetteter Systeme. / *The students become familiar with the fundamental concepts of designing of embedded systems.*

##### Anwenden

- Die Studierenden wenden grundlegende Algorithmen an zur Analyse und Optimierung von Hardware-Architekturen und Echtzeit-Softwaresystemen. / *The students apply basic algorithms to analyze and optimize hardware architectures and real-time software systems.*
- Die Studierenden erfassen den Hardware/Software-Entwurf von Systemen mit harten Beschränkungen. / *The students understand the hardware/software design of hard-constrained systems.*

#### Sozialkompetenz

- Die Studierenden lernen aktuelle Entwurfswerkzeuge für die Architektursynthese (Hardware) und Softwaresynthese kennen bei der kooperativen Bearbeitung der erweiterten Übung in Gruppen. / *The students learn about current design tools for architectural synthesis (hardware) and software synthesis by processing the extended exercises in groups cooperatively.*

## Literatur:

- Buch zur Vorlesung
- Vorlesungsskript (Zugriff nur innerhalb des Uni-Netzwerks möglich)

---

## Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

### [1] Informatik (Bachelor of Science)

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule | Vertiefungsrichtung Hardware-Software-Co-Design)

Dieses Modul ist daneben auch in den Studienfächern "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Bachelor of Science)", "Informatik (Master of Science)", "Information and Communication Technology (Master of Science)", "Informations- und Kommunikationstechnik (Master of Science)", "Mathematik (Bachelor of Science)", "Medizintechnik (Bachelor of Science)", "Medizintechnik (Master of Science)" verwendbar.

---

## Studien-/Prüfungsleistungen:

Eingebettete Systeme (Vorlesung mit erweiterten Übungen) (Prüfungsnummer: 773774)

Prüfungsleistung, Portfolio

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Eingebettete Systeme
- Übung zu Eingebettete Systeme
- Erweiterte Übungen zu Eingebettete Systeme

weitere Erläuterungen:

Portfolio besteht aus: Klausur (90 min) und erfolgreicher Teilnahme an den erweiterten Übungen (verpflichtend)

Die Modulnote ergibt sich aus der Klausurnote.

Erstablingung: WS 2019/2020, 1. Wdh.: SS 2020

1. Prüfer: Jürgen Teich (100115)

---

## Organisatorisches:

Die Auswahl dieses Moduls schließt die Auswahl des Moduls „Eingebettete Systeme“ aus.

---

**Modulbezeichnung: Eingebettete Systeme (ES-VU)** **5 ECTS**  
(Embedded Systems)

Modulverantwortliche/r: Jürgen Teich  
Lehrende: Jürgen Teich

---

Startsemester: WS 2019/2020	Dauer: 1 Semester	Turnus: jährlich (WS)
Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Deutsch

---

**Lehrveranstaltungen:**

Eingebettete Systeme (WS 2019/2020, Vorlesung, 2 SWS, Jürgen Teich et al.)  
Übung zu Eingebettete Systeme (WS 2019/2020, Übung, 2 SWS, Peter Brand et al.)

---

**Inhalt:**

Schwerpunkt der Vorlesung ist der Entwurf und die Implementierung eingebetteter Systeme unter Einsatz formaler Methoden und rechnergestützter Entwurfsverfahren.

Unter eingebetteten Systemen versteht man Rechensysteme, die auf einen Anwendungsbereich zugeschnitten (z.B. mobile Kommunikationsgeräte, Chipkartensysteme, Industriesteuerungen, Unterhaltungselektronik, Medizintechnik) und in einen technischen Kontext eingebunden sind. Das große Interesse am systematischen Entwurf von heterogenen eingebetteten Systemen ist verursacht durch die steigende Vielfalt und Komplexität von Anwendungen für eingebettete Systeme, die Notwendigkeit, Entwurfs- und Testkosten zu senken sowie durch Fortschritte in Schlüsseltechnologien (Mikroelektronik, formale Methoden).

Content:

*The focus of the lecture is the design and implementation of embedded systems using formal methods and computer-aided design techniques.*

*Embedded systems are computing systems tailored for a particular application (e.g., mobile communication devices, smart card systems, industrial control, consumer electronics, medical technology) and integrated into a technical context. The keen interest in the systematic design of heterogeneous embedded systems is driven by the increasing diversity and complexity of embedded system applications, the need to reduce design and test costs, and advances in key technologies (microelectronics, formal methods).*

**Lernziele und Kompetenzen:**

*Fachkompetenz*

*Wissen*

- Die Studierenden setzen sich mit einem aktuellen Forschungsgebiet auseinander. / *The students deal with a current field of research.*

*Verstehen*

- Die Studierenden verstehen grundlegende Konzepte des Entwurfs eingebetteter Systeme. / *The students become familiar with the fundamental concepts of designing of embedded systems.*

*Anwenden*

- Die Studierenden wenden grundlegende Algorithmen an zur Analyse und Optimierung von Hardware-Architekturen und Echtzeit-Softwaresystemen. / *The students apply basic algorithms to analyze and optimize hardware architectures and real-time software systems.*
- Die Studierenden erfassen den Hardware/Software-Entwurf von Systemen mit harten Beschränkungen. / *The students understand the hardware/software design of hard-constrained systems.*

**Literatur:**

- Buch zur Vorlesung
- Vorlesungsskript (Zugriff nur innerhalb des Uni-Netzwerks möglich)

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

## [1] Informatik (Bachelor of Science)

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule | Vertiefungsrichtung Hardware-Software-Co-Design)

Dieses Modul ist daneben auch in den Studienfächern "Advanced Signal Processing & Communications Engineering (Master of Science)", "Computational Engineering (Master of Science with Honours)", "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Bachelor of Science)", "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Master of Science)", "Informatik (Master of Science)", "Information and Communication Technology (Master of Science)", "Informations- und Kommunikationstechnik (Bachelor of Science)", "Informations- und Kommunikationstechnik (Master of Science)", "Mathematik (Bachelor of Science)", "Mechatronik (Bachelor of Science)", "Medizintechnik (Bachelor of Science)", "Medizintechnik (Master of Science)", "Wirtschaftsinformatik (Bachelor of Science)" verwendbar.

---

### Studien-/Prüfungsleistungen:

Eingebettete Systeme (Vorlesung mit Übungen) (Prüfungsnummer: 604896)

Prüfungsleistung, Klausur, Dauer (in Minuten): 90

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Eingebettete Systeme
- Übung zu Eingebettete Systeme

weitere Erläuterungen:

Die Modulnote ergibt sich aus der Klausurnote.

Erstablingung: WS 2019/2020, 1. Wdh.: SS 2020

1. Prüfer: Jürgen Teich (100115)

---

### Organisatorisches:

Die Auswahl dieses Moduls schließt die Auswahl des Moduls „Eingebettete Systeme mit erweiterter Übung“ aus.

---

**Modulbezeichnung: Ereignisgesteuerte Systeme (EGS)** **5 ECTS**  
 (Discrete Event Systems)

Modulverantwortliche/r: Stefan Wildermann  
 Lehrende: Stefan Wildermann

---

Startsemester: WS 2019/2020	Dauer: 1 Semester	Turnus: jährlich (WS)
Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Deutsch

---

**Lehrveranstaltungen:**

Ereignisgesteuerte Systeme (WS 2019/2020, Vorlesung, 2 SWS, Stefan Wildermann)  
 Übung zu Ereignisgesteuerte Systeme (WS 2019/2020, Übung, 2 SWS, Stefan Wildermann et al.)

---

**Inhalt:**

Die rasante Entwicklung von Rechnertechnologien in den vergangenen Jahrzehnten hatte die Verbreitung neuer dynamischer und komplexer Systeme zu Folge. Wesentliche Charakteristika solcher Systeme sind Verteiltheit, Nebenläufigkeit und das asynchrone Auftreten diskreter Ereignisse. Der Prozess, neue Modelle und Methoden für ereignisgesteuerte Systeme zu entwickeln, ist vergleichsweise jung. Der Rechner selbst spielt hierbei eine entscheidende Rolle als Werkzeug für Systementwurf, Analyse und Steuerung. Die Vorlesung EGS hat zum Ziel, Modellierungs-, Simulations- und Entwurfsmethoden für verteilte und ereignisdiskrete Systeme zu vermitteln. Die Methoden werden dabei beispielhaft auf Anwendung aus den Bereichen Computernetzwerke, automatischen Produktionssysteme, komplexen Softwaresysteme und integrierte Steuerungs-, Kommunikations- und Informationssysteme angewendet. In diesem Kontext behandelt die Vorlesung daher die folgenden Themenbereiche:

- Eigenschaften komplexer Systeme
- Überblick über Systeme und Modelle
- Zeitfreie und zeitbehaftete Modelle
- Stochastische Modelle
- Umsetzung in Programmiersprachen
- Simulation-, Entwurfs- und Testverfahren auf der Basis der vorgestellten Modelle.

**Lernziele und Kompetenzen:**

*Fachkompetenz*

*Verstehen*

- Die Studierenden erläutern grundlegende Techniken zur Modellierung diskreter, ereignisgesteuerter Systeme, zeigen deren Vor- und Nachteile auf und vergleichen diese bezüglich Ihrer Mächtigkeit.

*Anwenden*

- Die Studierenden wenden Modellierungs- und Analysetechniken aus dem Bereich endlicher Automaten, Petri-Netze, Markov-Ketten auf komplexe Systeme an.
- Die Studierenden setzen die Modellierung und Analyse eines Systems mit einem konkreten Entwurfswerkzeug um.

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

**[1] Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule | Vertiefungsrichtung Hardware-Software-Co-Design)

Dieses Modul ist daneben auch in den Studienfächern "Informatik (Master of Science)", "Informations- und Kommunikationstechnik (Bachelor of Science)", "Mathematik (Bachelor of Science)", "Medizintechnik (Master of Science)" verwendbar.

---

**Studien-/Prüfungsleistungen:**



Vorlesung und Übung Ereignisgesteuerte Systeme (Prüfungsnummer: 35401)

(englische Bezeichnung: Lecture/Tutorial: Discrete Event Systems)

Prüfungsleistung, Klausur, Dauer (in Minuten): 90

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Ereignisgesteuerte Systeme
- Übung zu Ereignisgesteuerte Systeme

Erstablingung: WS 2019/2020, 1. Wdh.: SS 2020

1. Prüfer: Jürgen Teich (100115)

---

**Bemerkungen:**

auch für Computational Engineering und I&K

---

**Modulbezeichnung: Organic Computing (OC)** **5 ECTS**  
 (Organic Computing)

Modulverantwortliche/r: Rolf Wanka  
 Lehrende: Rolf Wanka

---

Startsemester: SS 2019	Dauer: 1 Semester	Turnus: jährlich (SS)
Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Deutsch oder Englisch

---

**Lehrveranstaltungen:**

Organic Computing (SS 2019, Vorlesung, 2 SWS, Rolf Wanka)  
 Übungen zu Organic Computing (SS 2019, Übung, 2 SWS, Alexander Raß)

---

**Inhalt:**

Unter Organic Computing (OC) versteht man den Entwurf und den Einsatz von selbst-organisierenden Systemen, die sich den jeweiligen Umgebungsbedürfnissen dynamisch anpassen. Diese Systeme zeichnen sich dadurch aus, dass sie die sog. Self-\*-Eigenschaft besitzen, d.h. sie sind selbst-konfigurierend, selbst-optimierend, selbst-heilend, selbst-schützend, selbst-erklärend, ...

Als Vorbild für solche technischen Systeme werden Strukturen und Methoden biologischer und anderer natürlicher Systeme gewählt.

**Lernziele und Kompetenzen:**

*Fachkompetenz*

*Wissen*

Lernende können Wissen abrufen und wiedergeben. Sie kennen konkrete Einzelheiten wie Begriffe, Definitionen, Fakten, Regeln, Theorien, Merkmale, Kriterien, Abläufe. Sie lernen den Begriff des Organic Computings von anderen Paradigmen zu unterscheiden.

*Verstehen*

Lernende können Beispiele anführen und Aufgabenstellungen interpretieren.

*Anwenden*

Lernende können ein neues Problem wie z.B. Ranking-Erstellung durch Transfer des Wissens lösen.

*Analysieren*

Lernende können ein Problem in einzelne Teile zerlegen und so die Struktur des Problems verstehen.

*Sozialkompetenz*

Fähigkeit und Bereitschaft, zielorientiert mit anderen zusammenzuarbeiten.

**Literatur:**

- Ch. Müller-Schloer, Ch. von der Malsburg, R. P. Würt. Organic Computing. Informatik-Spektrum, Band 27, Nummer 4, S. 332-336. (LINK)
  - I. C. Trelea. The particle swarm optimization algorithm: convergence analysis and parameter selection. Information Processing Letters 85 (2003) 317-325. (LINK)
  - J. M. Kleinberg. Authoritative sources in a hyperlinked environment. Journal of the ACM 46 (1999) 604-632. (LINK)
  - M. Dorigo. V. Maniezzo. A Colomi. Ant system: an autocatalytic optimizing process. Technical Report 91-016, Politecnico di Milano, 1991. (LINK)
  - A. Badr. A. Fahmy. A proof of convergence for Ant algorithms. Information Sciences 160 (2004) 267-279.
  - M. Clerc. J. Kennedy. The particle swarm - Explosion, stability, and convergence in a multidimensional complex space. IEEE Transactions on Evolutionary Computation 8 (2002) 58-73.
- 

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

**[1] Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule | Vertiefungsrichtung Hardware-Software-Co-Design)

## [2] Informatik (Bachelor of Science)

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule | Vertiefungsrichtung Theoretische Informatik)

Dieses Modul ist daneben auch in den Studienfächern "Computational Engineering (Master of Science with Honours)", "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Master of Science)", "Informatik (Master of Science)", "Information and Communication Technology (Master of Science)", "Informations- und Kommunikationstechnik (Master of Science)", "Mathematik (Bachelor of Science)", "Medizintechnik (Master of Science)" verwendbar.

---

### Studien-/Prüfungsleistungen:

Organic Computing (Prüfungsnummer: 39701)

(englische Bezeichnung: Organic Computing)

Prüfungsleistung, Klausur, Dauer (in Minuten): 90

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Organic Computing
- Übungen zu Organic Computing

Erstablingung: SS 2019, 1. Wdh.: WS 2019/2020

1. Prüfer: Rolf Wanka (100367)

---

### Bemerkungen:

Auch für CE

---

**Modulbezeichnung: Parallele Systeme (PSYS-VU)**  
 (Parallel Systems)

**5 ECTS**

Modulverantwortliche/r: Jürgen Teich

Lehrende: Jürgen Teich, Frank Hannig

Startsemester: SS 2019

Dauer: 1 Semester

Turnus: jährlich (SS)

Präsenzzeit: 60 Std.

Eigenstudium: 90 Std.

Sprache: Deutsch oder Englisch

---

**Lehrveranstaltungen:**

Parallele Systeme (SS 2019, Vorlesung, 2 SWS, Jürgen Teich et al.)

Übung zu Parallele Systeme (SS 2019, Übung, 2 SWS, Frank Hannig)

---

**Inhalt:**

Aktuelle PCs verfügen über Mehrkernprozessoren und Grafikkarten, die wiederum aus hunderten von einfachen Prozessoren bestehen können. Hierdurch wird ein hohes Maß an nebenläufiger Datenverarbeitung möglich, welche bis vor einigen Jahren nur in Großrechnern erreicht werden konnte. Die effiziente Ausnutzung dieser Parallelität bedarf allerdings mehr als nur mehrerer Prozessoren, insbesondere muss das zu lösende Problem Parallelverarbeitung erlauben. In dieser Vorlesung werden Eigenschaften unterschiedlicher paralleler Rechnerarchitekturen und Metriken zu deren Beurteilung behandelt. Weiterhin werden Modelle und Sprachen zum Programmieren paralleler Rechner eingeführt. Neben der Programmierung von allgemeinen Parallelrechnern werden Entwurfsmethoden (CAD) vorgestellt, wie man ausgehend von einer algorithmischen Problemstellung ein massiv paralleles Rechenfeld in VLSI herleiten kann, das genau dieses Problem optimal parallel berechnet. Solche Schaltungen spielen auf der Bit- bzw. Wortebene eine dominante Rolle (Arithmetik) sowie bei Problemen der Signal- und Bildverarbeitung (z.B. Filter).

Im Einzelnen werden behandelt:

1. Theorie der Parallelität (parallele Computermodelle, parallele Spezifikationsformen und -sprachen, Performanzmodelle und -berechnung)
2. Klassifikation paralleler und skalierbarer Rechnerarchitekturen (Multiprozessoren und Multicomputer, Vektorrechner, Datenflussmaschinen, VLSI-Rechenfelder)
3. Programmierbare System-on-Chip (SoC) und Mehrkern-Architekturen (Grafik-Prozessoren, Cell, etc.)
4. Programmierung paralleler Rechner (Sprachen und Modelle, Entwurfsmethoden und Compiler, Optimierung)
5. Massive Parallelität: Vom Algorithmus zur Schaltung

*Today's PCs consist of multi-core processors and graphics cards that again comprise hundreds to thousands of simple processors. As a result of this, a very high degree of parallel data processing becomes possible, which was subjected to supercomputers a couple of years ago. The efficient exploitation of parallel processing requires not only multiple processors but also parallelism inherent in the problem to be solved. In this lecture, properties of different parallel computer architectures and corresponding quality metrics are examined. Further, models and parallel programming languages are introduced. In addition to programming general parallel computers, design methods (CAD) are presented that systematically transform an algorithmic problem description into a massive parallel processor array (VLSI), which can optimally execute the given problem in parallel. Such highly parallel circuits play an essential role at the bit level and circuit level (arithmetics) as well as in the case of signal processing and image processing (e.g., filter).*

*In detail, the following topics are covered:*

1. *Theory of parallelism (parallel models of computation, parallel specification and parallel languages, performance models)*
2. *Classification of parallel and scalable computer architectures (multi-processors and multi-computers, vector computers, data-flow machines, VLSI processor arrays)*
3. *Programmable System-on-Chip (SoC) and multi-core architectures (graphics processors, Cell, etc.)*
4. *Programming of parallel computers (languages and models, design methods and compiler, optimization)*

## 5. Massive parallelism: From algorithm to circuit

### Lernziele und Kompetenzen:

Schwerpunkt der Vorlesung ist die Vermittlung von Grundlagen der parallelen Datenverarbeitung. / *The focus of this lecture are foundations of parallel data processing.*

#### Fachkompetenz

##### Verstehen

- Die Studierenden verstehen grundlegende Konzepte der parallelen Datenverarbeitung, sowohl theoretischer Art anhand von Modellen, als auch an Architekturbeispielen. / *The students become familiar with the fundamentals of parallel data processing, theoretic in the form of models as well as by architecture examples.*

##### Anwenden

- Die Studierenden setzen sich mit modernen eingebetteten parallelen Ein-Chip-Architekturen auseinander. / *The students get familiar with modern embedded parallel system-on-chip architectures.*
- Die Studierenden wenden grundlegende Performanzmodelle und Parallelisierungstechniken zur Analyse und Optimierung von parallelen Algorithmen und Architekturen an. / *The students exercise basic performance models and parallelization techniques for the analysis and optimization of parallel algorithms and architectures.*
- Die Studierenden setzen die Modellierung und den Entwurf von massiv-parallelen Prozessorfeldern in konkreten Aufgaben selbstständig um. / *In concrete tasks, the students apply independently the modeling and the design of massively parallel processors arrays.*

### Literatur:

Siehe Webseite: <https://www.cs12.tf.fau.de/lehre/lehrveranstaltungen/vorlesungen/parallele-systeme/>

### Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

#### [1] Informatik (Bachelor of Science)

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule | Vertiefungsrichtung Hardware-Software-Co-Design)

Dieses Modul ist daneben auch in den Studienfächern "Computational Engineering (Master of Science with Honours)", "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Bachelor of Science)", "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Master of Science)", "Informatik (Master of Science)", "Informations- und Kommunikationstechnik (Master of Science)", "Mathematik (Bachelor of Science)", "Mechatronik (Bachelor of Science)", "Mechatronik (Master of Science)", "Medizintechnik (Master of Science)" verwendbar.

### Studien-/Prüfungsleistungen:

Parallele Systeme (Prüfungsnummer: 35101)

(englische Bezeichnung: Parallel Systems)

Prüfungsleistung, Klausur, Dauer (in Minuten): 90

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Parallele Systeme
- Übung zu Parallele Systeme

weitere Erläuterungen:

Mündliche (Dauer: 30 min) oder schriftliche (Dauer: 90 min) Prüfung. Die Modulnote ergibt sich aus der Prüfung.

Prüfungssprache: Deutsch oder Englisch

Erstablingung: SS 2019, 1. Wdh.: WS 2019/2020

1. Prüfer: Jürgen Teich (100115), 2. Prüfer: Frank Hannig (100192)

**Organisatorisches:**

Die Auswahl dieses Moduls schließt die Auswahl des Moduls „Parallele Systeme mit erweiterter Übung (PSYS-VEU)“ aus.

**Bemerkungen:**

auch für Computational Engineering

---

**Modulbezeichnung:** Reconfigurable Computing (RC-VU) 5 ECTS  
 (Reconfigurable Computing)

Modulverantwortliche/r: Jürgen Teich  
 Lehrende: Jürgen Teich

---

Startsemester: WS 2019/2020	Dauer: 1 Semester	Turnus: jährlich (WS)
Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Englisch

---

**Lehrveranstaltungen:**

Reconfigurable Computing (WS 2019/2020, Vorlesung, 2 SWS, Jürgen Teich)  
 Exercises to Reconfigurable Computing (WS 2019/2020, Übung, 2 SWS, Jorge A. Echavarria)

---

**Inhalt:**

Content:

Reconfigurable (adaptive) computing is a novel yet important research field investigating the capability of hardware to adapt to changing computational requirements such as emerging standards, late design changes, and even to changing processing requirements arising at run-time. Reconfigurable computing thus benefits from a) the programmability of software similar to the Von Neumann computer and b) the speed and efficiency of parallel hardware execution.

The purpose of the course reconfigurable computing is to instruct students about the possibilities and rapidly growing interest in adaptive hardware and corresponding design techniques by providing them the necessary knowledge for understanding and designing reconfigurable hardware systems and studying applications benefiting from dynamic hardware reconfiguration.

After a general introduction about benefits and application ranges of reconfigurable (adaptive) computing in contrast to general-purpose and application-specific computing, the following topics will be covered:

- Reconfigurable computing systems: Introduction of available technology including fine grained look up table (LUT-) based reconfigurable systems such as field programmable gate arrays (FPGA) as well as newest coarse grained architectures and technology.
- Design and implementation: Algorithms and steps (design entry, functional simulation, logic synthesis, technology mapping, place and route, bit stream generation) to implement (map) algorithms to FPGAs. The main focus lies on logic synthesis algorithms for FPGAs, in particular LUT technology mapping.
- Temporal partitioning: techniques to reconfigure systems over time. Covered are the problems of mapping large circuits which do not fit one single device. Several temporal partitioning techniques are studied and compared.
- Temporal placement: Techniques and algorithms to exploit the possibility of partial and dynamic (run-time) hardware reconfiguration. Here, OS-like services are needed that optimize the allocation and scheduling of modules at run-time.
- On-line communication: Modules dynamically placed at run-time on a given device need to communicate as well as transport data off-chip. State-of-the-art techniques are introduced how modules can communicate data at run-time including bus-oriented as well as network-on-a-chip (NoC) approaches.
- Designing reconfigurable applications on Xilinx Virtex FPGAs: In this part, the generation of partial bitstreams for components to be placed at run-time on Xilinx FPGAs is introduced and discussed including newest available tool flows.
- Applications: This section presents applications benefiting from dynamic hardware reconfiguration. It covers the use of reconfigurable systems including rapid prototyping, reconfigurable supercomputers, reconfigurable massively parallel computers and studies important application domains such as distributed arithmetic, signal processing, network packet processing, control design, and cryptography.

**Lernziele und Kompetenzen:**

Learning objectives and competencies:

*Fachkompetenz*

### Wissen

- The students know to exploit run-time reconfigurable design methodologies for adaptive applications.

### Verstehen

- The students understand the mapping steps, and optimization algorithms.
- The students classify different types and kinds of reconfigurable hardware technologies available today.
- The students clarify pros and cons of reconfigurable computing technology.
- The students summarize applications benefiting from reconfigurable computing.
- The students describe the design of circuits and systems-on-a-chip (SoC) on FPGAs.

### Literatur:

- The Hamburg VHDL Archive (see Documentation link for free books)
- Interactive VHDL Tutorial with 150 examples from ALDEC
- Easy FPGA tutorials, projects and boards
- Xilinx WebPack ISE and Modelsim MXE (free FPGA synthesis tool and free VHDL simulator)
- Symphone EDA free VHDL simulator (select FREE Edition license)
- Icarus open-source Verilog simulator

### Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

#### [1] Informatik (Bachelor of Science)

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule | Vertiefungsrichtung Hardware-Software-Co-Design)

Dieses Modul ist daneben auch in den Studienfächern "Advanced Signal Processing & Communications Engineering (Master of Science)", "Informatik (Master of Science)", "Informations- und Kommunikationstechnik (Master of Science)", "Mathematik (Bachelor of Science)", "Mechatronik (Bachelor of Science)", "Mechatronik (Master of Science)" verwendbar.

### Studien-/Prüfungsleistungen:

Reconfigurable Computing (Lecture with Exercises) (Prüfungsnummer: 741941)

Prüfungsleistung, Portfolio

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Reconfigurable Computing
- Exercises to Reconfigurable Computing

weitere Erläuterungen:

Oral examination (Duration: 30 min) + Successful completion of all tasks in the exercises (mandatory)

Final grade of the module is determined by the oral examination.

Erstablingung: WS 2019/2020, 1. Wdh.: SS 2020

1. Prüfer: Jürgen Teich (100115)

### Organisatorisches:

Selection of this module prohibits the selection of the module "Reconfigurable Computing with Extended Exercises (RC-VEU)" by the student.

### Bemerkungen:

Reconfigurable computing is an interdisciplinary field of research between computer science and electrical engineering on a 4 SWS (4 hours/week) basis. Lecture and Exercises will give 5 ECTS, the FPGA & VHDL labs 2.5 ECTS.



---

**Modulbezeichnung:** Reconfigurable Computing with Extended Exercises (RC-VEU) 7.5 ECTS  
 (Reconfigurable Computing with Extended Exercises)

Modulverantwortliche/r: Jürgen Teich  
 Lehrende: Jürgen Teich

---

Startsemester: WS 2019/2020	Dauer: 1 Semester	Turnus: jährlich (WS)
Präsenzzeit: 90 Std.	Eigenstudium: 135 Std.	Sprache: Englisch

---

**Lehrveranstaltungen:**

Reconfigurable Computing (WS 2019/2020, Vorlesung, 2 SWS, Jürgen Teich)  
 Exercises to Reconfigurable Computing (WS 2019/2020, Übung, 2 SWS, Jorge A. Echavarria)  
 Extended Exercises to Reconfigurable Computing (WS 2019/2020, Übung, Martin Letras)

---

**Inhalt:**

Content:

Reconfigurable (adaptive) computing is a novel yet important research field investigating the capability of hardware to adapt to changing computational requirements such as emerging standards, late design changes, and even to changing processing requirements arising at run-time. Reconfigurable computing thus benefits from a) the programmability of software similar to the Von Neumann computer and b) the speed and efficiency of parallel hardware execution.

The purpose of the course reconfigurable computing is to instruct students about the possibilities and rapidly growing interest in adaptive hardware and corresponding design techniques by providing them the necessary knowledge for understanding and designing reconfigurable hardware systems and studying applications benefiting from dynamic hardware reconfiguration.

After a general introduction about benefits and application ranges of reconfigurable (adaptive) computing in contrast to general-purpose and application-specific computing, the following topics will be covered:

- Reconfigurable computing systems: Introduction of available technology including fine grained look up table (LUT-) based reconfigurable systems such as field programmable gate arrays (FPGA) as well as newest coarse grained architectures and technology.
- Design and implementation: Algorithms and steps (design entry, functional simulation, logic synthesis, technology mapping, place and route, bit stream generation) to implement (map) algorithms to FPGAs. The main focus lies on logic synthesis algorithms for FPGAs, in particular LUT technology mapping.
- Temporal partitioning: techniques to reconfigure systems over time. Covered are the problems of mapping large circuits which do not fit one single device. Several temporal partitioning techniques are studied and compared.
- Temporal placement: Techniques and algorithms to exploit the possibility of partial and dynamic (run-time) hardware reconfiguration. Here, OS-like services are needed that optimize the allocation and scheduling of modules at run-time.
- On-line communication: Modules dynamically placed at run-time on a given device need to communicate as well as transport data off-chip. State-of-the-art techniques are introduced how modules can communicate data at run-time including bus-oriented as well as network-on-a-chip (NoC) approaches.
- Designing reconfigurable applications on Xilinx Virtex FPGAs: In this part, the generation of partial bitstreams for components to be placed at run-time on Xilinx FPGAs is introduced and discussed including newest available tool flows.
- Applications: This section presents applications benefiting from dynamic hardware reconfiguration. It covers the use of reconfigurable systems including rapid prototyping, reconfigurable supercomputers, reconfigurable massively parallel computers and studies important application domains such as distributed arithmetic, signal processing, network packet processing, control design, and cryptography.

**Lernziele und Kompetenzen:**

Learning objectives and competencies:

*Fachkompetenz*

*Wissen*

- The students know to exploit run-time reconfigurable design methodologies for adaptive applications.

*Verstehen*

- The students understand the mapping steps, and optimization algorithms.
- The students classify different types and kinds of reconfigurable hardware technologies available today.
- The students clarify pros and cons of reconfigurable computing technology.
- The students summarize applications benefiting from reconfigurable computing.

*Anwenden*

- The students apply design tools for implementation of circuits and systems-on-a-chip (SoC) on FPGAs during practical training.

*Sozialkompetenz*

- The students perform group work in small teams during practical training.

**Literatur:**

- The Hamburg VHDL Archive (see Documentation link for free books)
- Interactive VHDL Tutorial with 150 examples from ALDEC
- Easy FPGA tutorials, projects and boards
- Xilinx WebPack ISE and Modelsim MXE (free FPGA synthesis tool and free VHDL simulator)
- Symphone EDA free VHDL simulator (select FREE Edition license)
- Icarus open-source Verilog simulator

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

**[1] Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule | Vertiefungsrichtung Hardware-Software-Co-Design)

Dieses Modul ist daneben auch in den Studienfächern "Informatik (Master of Science)", "Information and Communication Technology (Master of Science)", "Informations- und Kommunikationstechnik (Master of Science)", "Mathematik (Bachelor of Science)", "Mechatronik (Bachelor of Science)", "Mechatronik (Master of Science)", "Medizintechnik (Master of Science)" verwendbar.

**Studien-/Prüfungsleistungen:**

Reconfigurable Computing (Lecture with Extended Exercises) (Prüfungsnummer: 110334)

Prüfungsleistung, Portfolio

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Reconfigurable Computing
- Exercises to Reconfigurable Computing
- Extended Exercises to Reconfigurable Computing

weitere Erläuterungen:

Oral examination (Duration: 30 min) + Successfull attendance of the extended exercises (mandatory)

+ Successfull completion of all tasks in the exercises (mandatory)

Final grade of the module is determined by the oral examination.

Erstablingung: WS 2019/2020, 1. Wdh.: SS 2020

1. Prüfer: Jürgen Teich (100115)

**Organisatorisches:**

Selection of this module prohibits the selection of the module "Reconfigurable Computing (RC-VU)" by the student.

**Bemerkungen:**

Reconfigurable computing is an interdisciplinary field of research between computer science and electrical engineering on a 4 SWS (4 hours/week) basis. Lecture and Exercises will give 5 ECTS, the FPGA & VHDL labs 2.5 ECTS.

---

**Modulbezeichnung:** **Verifikation digitaler Systeme (VdS)** **5 ECTS**  
 (Verification of Digital Systems)

Modulverantwortliche/r: Oliver Keszöcze  
 Lehrende: Oliver Keszöcze, Oliver Keszöcze

---

Startsemester: WS 2019/2020	Dauer: 1 Semester	Turnus: jährlich (WS)
Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Deutsch

---

**Lehrveranstaltungen:**

Verifikation digitaler Systeme (WS 2019/2020, Vorlesung, 2 SWS, Oliver Keszöcze)  
 Übung zur Verifikation digitaler Systeme (WS 2019/2020, Übung, 2 SWS, Oliver Keszöcze)

---

**Inhalt:**

Für den Entwurf eines digitalen Systems werden heute in der Industrie ebenso viele Verifikationsingenieure wie Designer benötigt. Trotzdem beansprucht die Verifikation heute bereits 70%-80% der gesamten Entwurfszeit. Neben konventionellen Verifikationserfahren wie der Simulation sind werden seit einigen Jahren sogenannte "formale Verifikationsmethoden" in heutigen Entwurfsflüssen eingesetzt. Der Umgang mit diesen Methoden stellt ein wichtiges neues Aufgabenfeld dar. Im Gegensatz zur Simulation beruht die formale Verifikation auf exakten mathematischen Methoden zum Nachweis funktionaler Schaltungseigenschaften. Dadurch können Entwurfsfehler frühzeitiger und mit höherer Zuverlässigkeit als bisher erkannt werden. Jedes System zur formalen Hardwareverifikation erfordert:

1. ein geeignetes Modell des zu verifizierenden Systems
2. eine Sprache zur Formulierung der zu verifizierenden Eigenschaften
3. eine Beweismethode.

Die Vorlesung behandelt diese drei Bereiche, vermittelt die grundlegenden Algorithmen und Konzepte moderner Werkzeuge für die formale Hardwareverifikation und erläutert deren Einsatz in der industriellen Praxis. Im Einzelnen werden in dieser Vorlesung die folgenden Punkte behandelt:

1. Modellierung digitaler Systeme
2. Unterschiede formaler und simulationsbasierter Verifikationsmethoden
3. Äquivalenzvergleich
4. Formale und simulationsbasierte Eigenschaftsprüfung
5. Assertions
6. Verifikation arithmetischer Schaltungen

**Lernziele und Kompetenzen:**

*Fachkompetenz*

*Verstehen*

- Die Studierenden erläutern die wesentlichen Techniken zur Verifikation digitaler Systeme, zeigen deren Vor- und Nachteile auf und vergleichen diese bezüglich Ihrer Mächtigkeit und Komplexität.

*Anwenden*

- Die Studierenden wenden Modellierungstechniken aus den Bereichen der Binären Entscheidungsdiagramme inkl. bekannter Erweiterungen (MDD, etc.) sowie der SAT-Löser auf Systembeschreibungen an.
- Die Studierenden Verifikationstechniken aus den Bereichen der formalen Äquivalenz- und Eigenschaftsprüfung (Model Checking, Symbolic Model Checking, Bounded Model Checking) auf Systembeschreibungen an.

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

[1] **Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule | Vertiefungsrichtung Hardware-Software-Co-Design)

Dieses Modul ist daneben auch in den Studienfächern "Computational Engineering (Master of Science with Honours)", "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Bachelor of Science)", "Computa-

tional Engineering (Rechnergestütztes Ingenieurwesen) (Master of Science)", "Informatik (Master of Science)",  
"Mathematik (Bachelor of Science)" verwendbar.

---

**Studien-/Prüfungsleistungen:**

Verifikation digitaler Systeme (Prüfungsnummer: 22801)

Prüfungsleistung, mündliche Prüfung, Dauer (in Minuten): 30

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Verifikation digitaler Systeme
- Übung zur Verifikation digitaler Systeme

Erstablingung: WS 2019/2020, 1. Wdh.: SS 2020

1. Prüfer: Jürgen Teich (100115)

---

**Modulbezeichnung:** Cryptocurrencies (CryptoCur) 5 ECTS  
(Cryptocurrencies)

Modulverantwortliche/r: Dominique Schröder  
Lehrende: Dominique Schröder

Startsemester: SS 2019                      Dauer: 1 Semester                      Turnus: jährlich (WS)  
Präsenzzeit: 60 Std.                      Eigenstudium: 90 Std.                      Sprache: Englisch

**Lehrveranstaltungen:**

CryptoCurrencies (SS 2019, Vorlesung mit Übung, 4 SWS, Dominique Schröder)

**Empfohlene Voraussetzungen:**

Only basic knowledge in cryptography or computer security is assumed. It would be ideal if you have done a cryptography and/or security class at FAU or during your previous studies somewhere else. Motivated students without this background are also welcome to the course. However, we would expect some more spirited efforts from you.

**Inhalt:**

A preliminary list of topics covered are the following (subject to change):

- Cryptographic e-cash
- Blockchain
- Bitcoin Protocol
- Nakamoto Consensus
- Bitcoin Community and Politics
- Alternative Mining Approaches
- Alternative Coins: Ethereum, Anonymous Payments, Zero-Knowledge Cash. . .
- Other state-of-the-art topic related to cryptocurrencies

**Lernziele und Kompetenzen:**

*Fachkompetenz*

*Verstehen*

Die Studierenden kennen die wesentlichen Eigenschaften digitaler Währungen und können diese auch vergleichen.

*Analysieren*

Die Studierenden können digitale Währungen untersuchen und überprüfen, ob diese die grundlegenden Eigenschaften einer digitalen Währung erfüllen.

*Evaluiieren (Beurteilen)*

Die Studierenden können unterschiedliche digitalen Währung miteinander vergleichen und je nach Anwendung einen geeigneten Kandidaten ermitteln.

**Literatur:**

Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction by Arvind Narayanan , Joseph Bonneau, Edward Felten, Andrew Miller, Steven Goldfeder

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

[1] **Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule)

**Studien-/Prüfungsleistungen:**

Cryptocurrencies (Prüfungsnummer: 566245)

(englische Bezeichnung: Cryptocurrencies)

Prüfungsleistung, mündliche Prüfung, Dauer (in Minuten): 30

Anteil an der Berechnung der Modulnote: 100%

weitere Erläuterungen:

The grade consists to 35% of grades for exercise sheets and to 65% of the grade for a programming

project (to be defined) developed by the end of the course (in groups of at most 3 students).  
Prüfungssprache: Deutsch oder Englisch

Erstablingung: SS 2019, 1. Wdh.: keine Angabe  
1. Prüfer: Dominique Schröder (100450)

---

---

**Modulbezeichnung:** **Software Reverse Engineering V+Ü (RE)** **5 ECTS**  
 (Software Reverse Engineering)

Modulverantwortliche/r: Tilo Müller  
 Lehrende: Tilo Müller

---

Startsemester: SS 2019	Dauer: 1 Semester	Turnus: halbjährlich (WS+SS)
Präsenzzeit: 50 Std.	Eigenstudium: 100 Std.	Sprache: Deutsch

---

**Lehrveranstaltungen:**

Software Reverse Engineering (SS 2019, Vorlesung, 2 SWS, Tilo Müller)  
 Software Reverse Engineering - Übung (SS 2019, Übung, 2 SWS, Anatoli Kalysch et al.)

---

**Inhalt:**

Ein regelmäßiger Untersuchungsgegenstand der angewandten IT-Sicherheit ist unbekanntes Software, deren Funktionsweise analysiert werden soll bspw. um Schwachstellen aufzudecken oder um das Verhalten von Malware zu analysieren. Während "Reverse Engineering" im allgemeinen den Vorgang bezeichnet Konstruktionselemente und Strukturen aus einem bestehenden System zurückzugewinnen, bezeichnet "Software Reverse Engineering" speziell die Rückgewinnung von Software-Anforderungen und Intentionen aus Binärcode. Im Gegensatz zu "Software Engineering", bei dem es um die Übersetzung von Anforderungen in Code geht, muss hier also der umgekehrte Weg gegangen werden. Die Rückgewinnung von Informationen kann die Erstellung von äquivalenten Quell- oder Pseudocode beinhalten, hat darüberhinaus aber stets den Anspruch ein Programm auf höherer Ebene "zu verstehen". Im Rahmen der Vorlesung werden grundlegende Konzepte sowie fortgeschrittene Techniken und Tools aus dem Bereich Software Reverse Engineering vorgestellt. Ziel der Veranstaltung ist ein Verständnis von statischen und dynamischen Methoden zur Analyse von Binärprogrammen, indem typische Werkzeuge wie Disassembler (bspw. IDA Pro) und Debugger (bspw. OllyDbg) zum Einsatz kommen. Es wird auch verdeutlicht, mit welchen Schwierigkeiten Software Reverse Engineering zu kämpfen hat und wo seine Grenzen liegen, bspw. im Bereich automatisierter Decompiler. Ebenso werden Techniken angesprochen um sich gegen Software Reverse Engineering zu schützen, bspw. Code-Verschleierung (Obfuscation) gegen statische Analyse und Anti-Debugging Techniken gegen dynamische Analyse. Außerdem wird vorgestellt wie sich Schwachstellen in Binärprogrammen aufspüren lassen und wie sich diese trotz aktueller Schutzmechanismen (ASLR und DEP) ausnutzen lassen.

**Lernziele und Kompetenzen:**

Die Vorlesung vermittelt Kompetenzen im Bereich "Software Reverse Engineering", insbesondere zur Analyse von Schadsoftware und um Schwachstellen in Binärprogrammen aufzudecken. Das Lernziel dieser Veranstaltung ist, dass die Teilnehmer ausführbare Programme unter Windows und Linux mit entsprechenden Werkzeugen analysieren können.

**Literatur:**

- "Reversing: Secrets of Reverse Engineering" von Eldad Eilam, Wiley Publishing Inc., Indianapolis, 2005
- "The IDA Pro Book: An Unofficial Guide to the World's Most Popular Disassembler" von Chris Eagle, No Starch Press, San Francisco, 2011
- "Microsoft Windows Internals: Fifth Edition" von Mark E. Russinovich und David Solomon, Microsoft Press, Washington, 2009
- "Surreptitious Software: Obfuscation, Watermarking, and Tamperproofing for Software Protection" von Christian Collberg und Jasvir Nagra, Addison-Wesley Professional, 2009
- "Buffer Overflows und Format-String-Schwachstellen: Funktionsweisen, Exploits und Gegenmaßnahmen" von Tobias Klein, Dpunkt Verlag, 2003

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

[1] **Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule)

---



**Studien-/Prüfungsleistungen:**

Software Reverse Engineering V+Ü (Prüfungsnummer: 890193)

Prüfungsleistung, Portfolio

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Software Reverse Engineering
- Software Reverse Engineering - Übung

weitere Erläuterungen:

schriftliche Prüfung (Klausur) + erfolgreiche Teilnahme an den Übungen (verpflichtend) + erfolgreiche Bearbeitung aller Übungsaufgaben (verpflichtend)

Erstablingung: SS 2019, 1. Wdh.: WS 2019/2020, 2. Wdh.: SS 2020

1. Prüfer: Tilo Müller (100000)

---

---

**Modulbezeichnung:** Forensische Informatik (ForensInf) 5 ECTS  
(Forensic Computing)

Modulverantwortliche/r: Felix Freiling

Lehrende: Felix Freiling, Christian Moch, Stefan Vömel

---

Startsemester: SS 2019

Dauer: 1 Semester

Turnus: jährlich (SS)

Präsenzzeit: 60 Std.

Eigenstudium: 90 Std.

Sprache: Deutsch

---

**Lehrveranstaltungen:**

Forensische Informatik (SS 2019, Vorlesung mit Übung, 2 SWS, Felix Freiling)

Forensische Informatik - Übung (SS 2019, Übung, 2 SWS, Ralph Palutke)

---

**Inhalt:**

Forensische Informatik befasst sich mit der Sammlung, Aufbereitung und Analyse digitaler Beweismittel zur Verwendung vor Gericht. Ausgangspunkt ist jeweils der Verdacht auf einen Computereintrich oder eine Straftat, die mit Hilfe von digitalen Geräten vorgenommen worden ist. Diese Vorlesung gibt einen Überblick über die Methoden der forensischen Informatik aus einer wissenschaftlichen Perspektive. Der Schwerpunkt liegt auf der Analyse von Dateisystemen. Ziel der Vorlesung ist nicht die Ausbildung von Forensik-Praktikern, sondern die Vermittlung von Kenntnissen, die es einem erlauben, Forschung im Bereich Computerforensik zu betreiben. Voraussichtliche Themen:

- Definition forensische Informatik
- Der forensische Prozess und seine wissenschaftliche Fundierung
- Rechtliche Rahmenbedingungen
- Sichern von Festplatten
- Analyse verschiedener Dateisysteme (FAT32, NTFS, Ext2/Ext3)
- Tools

**Lernziele und Kompetenzen:**

Die Studierenden können Termini und Methoden der digitalen Forensik in die Entwicklung der forensischen Wissenschaften einordnen.

Die Studierenden können die wesentlichen Datenstrukturen verschiedener Dateisysteme erklären. Sie können die für forensische Zwecke wesentlichen Datenstrukturen lokalisieren und geeignete Werkzeuge zu ihrer Analyse auswählen und anwenden.

Die Studierenden können digitale Spuren konkreter Fallkonstellationen durch Anwendung von Werkzeugen rekonstruieren, analysieren, interpretieren und dokumentieren. Sie lernen ihre Untersuchungsergebnisse zu präsentieren und gegenüber kritischen Nachfragen zu verteidigen.

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

[1] Informatik (Bachelor of Science)

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule)

---

**Studien-/Prüfungsleistungen:**

Forensische Informatik (Vorlesung mit Übung) (Prüfungsnummer: 792501)

Prüfungsleistung, mündliche Prüfung, Dauer (in Minuten): 30

Anteil an der Berechnung der Modulnote: 100%

Erstablingung: SS 2019, 1. Wdh.: WS 2019/2020

1. Prüfer: Felix Freiling (100128)

---

**Modulbezeichnung:** Security in Embedded Hardware (SEH) 5 ECTS  
(Security in Embedded Hardware)

Modulverantwortliche/r: Jürgen Teich  
Lehrende: Jürgen Teich

Startsemester: SS 2019 Dauer: 1 Semester Turnus: jährlich (SS)  
Präsenzzeit: 60 Std. Eigenstudium: 90 Std. Sprache: Deutsch oder Englisch

**Lehrveranstaltungen:**

Security in Embedded Hardware (SS 2019, Vorlesung, 2 SWS, Jürgen Teich)  
Übung zu Security in Embedded Hardware (SS 2019, Übung, 2 SWS, Jürgen Teich)

**Inhalt:**

Der Schutz eingebetteter Systeme gegenüber Angriffe Dritter auf gespeicherte Daten und Implementierungen, stellt eine immer wichtigere, jedoch auch durch zunehmende Vernetzung herausfordernde Aufgabe dar. Der Schutz der eingebetteten Systeme gegenüber bekannten als auch neueren ausgeklügelten Angriffsmöglichkeiten ist Gegenstand dieser Vorlesung. Es wird gezeigt, welche Angriffe existieren, welche Gegenmaßnahmen man ergreifen kann und wie man sichere eingebettete Systeme entwirft.

Einleitung und Motivation

- Was ist Security?
- Die Bedeutung von Security für zuverlässige Systeme
- Klassifikation von Angriffen
- Entwurf eingebetteter Systeme

Angriffsszenarien

- Beispiele von Angriffsszenarien
- Kryptographischer Algorithmen als Ziel von Angriffen

Angriffe durch Einschleusen von Code (Code Injection Attacks)

- Welche Arten von Code Injection-Angriffe gibt es?
- Gegenmaßnahmen

Invasive physikalische Angriffe (Invasive Physical Attacks)

- Microprobing
- Reverse Engineering
- Differential Fault Analysis
- Gegenmaßnahmen

Nichtinvasive softwarebasierte Angriffe (Non-Invasive Logical Attacks)

- Erlangen von nicht autorisiertem Zugriff
- Gegenmaßnahmen

Nichtinvasive physikalische Angriffe (Non-Invasive Physical Attacks)

- Abhören
- Seitenkanalangriffe
- Gegenmaßnahmen

**Lernziele und Kompetenzen:**

*Fachkompetenz*

*Wissen*

- Die Studierenden legen die entsprechenden Gegenmaßnahmen dar
- Die Studierenden nennen verschiedene Sicherheitseinrichtungen und -maßnahmen in eingebetteten Systemen

*Verstehen*

- Die Studierenden zeigen den Einfluss von Angriffen und deren Gegenmaßnahmen auf die Verlässlichkeit eines eingebetteten Systems auf
- Die Studierenden zeigen den zusätzlichen Aufwand (Fläche, Rechenzeit) von Sicherheitseinrichtungen auf

*Analysieren*

- Die Studierenden klassifizieren verschiedene Angriffstypen auf eingebettete Systeme

*Sozialkompetenz*

- Die Studierenden erarbeiten kooperativ in Gruppen Lösungskonzepte und implementieren diese gemeinsam

**Literatur:**

- Catherine H. Gebotys Security in Embedded Devices. Springer 2010.
- Benoit Badrignans et al. Security Trends for FPGAs. Springer 2011.
- Daniel Ziener Techniques for Increasing Security and Reliability of IP Cores Embedded in FPGA and ASIC Designs. Dr. Hut 2010.

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

**[1] Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule)

---

**Studien-/Prüfungsleistungen:**

Security in Embedded Hardware (Prüfungsnummer: 172338)

Prüfungsleistung, Klausur, Dauer (in Minuten): 60

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Security in Embedded Hardware
- Übung zu Security in Embedded Hardware

Prüfungssprache: Deutsch oder Englisch

Erstablingung: SS 2019, 1. Wdh.: WS 2019/2020 (nur für Wiederholer)

1. Prüfer: Jürgen Teich (100115)

---

---

**Modulbezeichnung:** **Angewandte IT-Sicherheit (ApplITSec)** **5 ECTS**  
 (Applied IT Security)

Modulverantwortliche/r: Felix Freiling  
 Lehrende: Felix Freiling

---

Startsemester: WS 2019/2020	Dauer: 1 Semester	Turnus: jährlich (WS)
Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Deutsch

---

**Lehrveranstaltungen:**

Angewandte IT-Sicherheit (WS 2019/2020, Vorlesung, 2 SWS, Felix Freiling et al.)  
 Angewandte IT-Sicherheit - Übung (WS 2019/2020, Übung, 2 SWS, Felix Freiling et al.)

---

**Inhalt:**

Die Vorlesung gibt einen einführenden Überblick über Konzepte und Methoden der IT-Sicherheit und eignet sich als Einstieg in das Vertiefungsgebiet "IT-Sicherheit" an der FAU. Themen (unter anderem): IT-Sicherheit vs. physische Sicherheit, Identifizierung und Authentifizierung, grundlegende Muster von Unsicherheit in technischen Systemen, grundlegende Abwehrmechanismen, ausgewählte Beispiele aus dem Bereich Systemsicherheit, Netzwerksicherheit und Softwaresicherheit.

In der Übung werden die Themen der Veranstaltung beispielhaft eingeübt. Themen (unter anderem): "lock picking", "social engineering", ausnutzen von Softwareschwachstellen.

**Lernziele und Kompetenzen:**

Teilnehmer erwerben einen Überblick über Konzepte und Methoden aus dem Bereich der IT-Sicherheit und können diese im Kontext der Informatik und der Lebenswirklichkeit anhand von Beispielen einordnen und erläutern. Die Studierenden können die wichtigsten Arten von Softwareschwachstellen in Programmen erkennen und benennen. Sie können außerdem erläutern, wie man diese Schwachstellen ausnutzt und welche technischen und organisatorischen Maßnahmen geeignet sind, diese Schwachstellen zu vermeiden. Die Studierenden lernen, die Wirksamkeit von IT-Sicherheitsmechanismen im gesellschaftlichen Kontext und in Kenntnis professioneller Strukturen der Cyberkriminalität aus technischen, ethischen und rechtlichen Perspektiven zu bewerten.

**Literatur:**

Dieter Gollmann: Computer Security. 3. Auflage, Wiley, 2010. Joachim Biskup: Security in Computing Systems. Springer, 2008. Weitere Literatur wird in der Vorlesung bekanntgegeben.

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

**[1] Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule | Vertiefungsrichtung IT-Sicherheit)

Dieses Modul ist daneben auch in den Studienfächern "Informatik (Master of Science)", "Mathematik (Bachelor of Science)", "Mechatronik (Bachelor of Science)", "Mechatronik (Master of Science)", "Medizintechnik (Bachelor of Science)", "Medizintechnik (Master of Science)" verwendbar.

---

**Studien-/Prüfungsleistungen:**

Angewandte IT-Sicherheit (Vorlesung mit Übungen) (Prüfungsnummer: 326311)

(englische Bezeichnung: Applied IT Security (Lecture with Exercises))

Prüfungsleistung, Klausur, Dauer (in Minuten): 90

Anteil an der Berechnung der Modulnote: 100%

Erstablegung: WS 2019/2020, 1. Wdh.: SS 2020

1. Prüfer: Felix Freiling (100128)

---

---

**Modulbezeichnung:** Einführung in die IT-Sicherheit (EinfITSec) 5 ECTS  
 (Introduction to IT Security)

Modulverantwortliche/r: Felix Freiling  
 Lehrende: Felix Freiling

---

Startsemester: WS 2019/2020	Dauer: 1 Semester	Turnus: jährlich (WS)
Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Deutsch

---

**Lehrveranstaltungen:**

Angewandte IT-Sicherheit (WS 2019/2020, Vorlesung, 2 SWS, Felix Freiling et al.)  
 Einführung in die IT-Sicherheit - Übung (WS 2019/2020, Übung, 2 SWS, Felix Freiling et al.)

---

**Inhalt:**

Die Vorlesung gibt einen einführenden Überblick über Konzepte und Methoden der IT-Sicherheit und eignet sich als Einstieg in das Vertiefungsgebiet "IT-Sicherheit" an der FAU. Themen (unter anderem): IT-Sicherheit vs. physische Sicherheit, Identifizierung und Authentifizierung, grundlegende Muster von Unsicherheit in technischen Systemen, grundlegende Abwehrmechanismen, ausgewählte Beispiele aus dem Bereich Systemsicherheit, Netzwerksicherheit und Softwaresicherheit. In der Übung werden die Themen der Veranstaltung beispielhaft eingeübt. Themen (unter anderem): "lock picking", "social engineering", ausnutzen von Softwareschwachstellen.

**Lernziele und Kompetenzen:**

Teilnehmer erwerben einen Überblick über Konzepte und Methoden aus dem Bereich der IT-Sicherheit und können diese im Kontext der Informatik und der Lebenswirklichkeit anhand von Beispielen einordnen und erläutern. Die Studierenden können die wichtigsten Arten von Softwareschwachstellen in Programmen erkennen und benennen. Sie können außerdem erläutern, wie man diese Schwachstellen ausnutzt und welche technischen und organisatorischen Maßnahmen geeignet sind, diese Schwachstellen zu vermeiden. Die Studierenden lernen, die Wirksamkeit von IT-Sicherheitsmechanismen im gesellschaftlichen Kontext und in Kenntnis professioneller Strukturen der Cyberkriminalität aus technischen, ethischen und rechtlichen Perspektiven zu bewerten.

**Literatur:**

Dieter Gollmann: Computer Security. 3. Auflage, Wiley, 2010. Joachim Biskup: Security in Computing Systems. Springer, 2008. Weitere Literatur wird in der Vorlesung bekanntgegeben.

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

**[1] Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule | Vertiefungsrichtung IT-Sicherheit)

Dieses Modul ist daneben auch in den Studienfächern "Informatik (Master of Science)", "Information and Communication Technology (Master of Science)", "Informations- und Kommunikationstechnik (Master of Science)", "International Information Systems (IIS) (Master of Science)", "Mathematik (Bachelor of Science)", "Mechatronik (Bachelor of Science)", "Mechatronik (Master of Science)", "Medizintechnik (Bachelor of Science)", "Medizintechnik (Master of Science)", "Wirtschaftsinformatik (Bachelor of Science)" verwendbar.

---

**Studien-/Prüfungsleistungen:**

Einführung in die IT-Sicherheit (ApplITSec-Ü) (Prüfungsnummer: 46311)

(englische Bezeichnung: Applied IT security (ApplITSec))

Prüfungsleistung, Klausur, Dauer (in Minuten): 60

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Angewandte IT-Sicherheit
- Einführung in die IT-Sicherheit - Übung

Prüfungssprache: Deutsch oder Englisch

Erstablingung: WS 2019/2020, 1. Wdh.: SS 2020

1. Prüfer: Felix Freiling (100128)

---

---

**Modulbezeichnung:** Security and Privacy in Pervasive Computing (SecPriPC) 5 ECTS  
 (Security and Privacy in Pervasive Computing)

Modulverantwortliche/r: Zinaida Benenson  
 Lehrende: Zinaida Benenson

---

Startsemester: WS 2019/2020	Dauer: 1 Semester	Turnus: jährlich (WS)
Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Deutsch

---

**Lehrveranstaltungen:**

Die erste Übung findet in der ersten Vorlesungswoche statt.

Die Modulsprache ist Deutsch, Folien sind auf Englisch.

This module will be held in German, slides are in English.

Security and Privacy in Pervasive Computing (WS 2019/2020, Vorlesung, 2 SWS, N.N.)

Security and Privacy in Pervasive Computing - Übung (WS 2019/2020, Übung, 2 SWS, Zinaida Benenson et al.)

---

**Empfohlene Voraussetzungen:**

Basic knowledge in the area of IT security and privacy, such as security goals (CIA), symmetric and asymmetric cryptography principles, PKI, basic functionality of SSL is required. This knowledge can be acquired through the attendance of the module "Applied IT Security" (Angewandte IT Sicherheit) or similar modules.

**Vorhergehende Module:**

Angewandte IT-Sicherheit

---

**Inhalt:**

Pervasive Computing, also called Ubiquitous Computing, is a computing paradigm that comprises billions computing devices integrated into everyday objects and connected into a global communication network that is orders of magnitude larger than the Internet today. These devices measure environmental characteristics, exchange information about their surroundings and interact with people in many different ways, such that sometimes people may be even unaware that they are using computers. The era of pervasive computing has already started and moves on rapidly, integrating the Internet, smartphones, wearable computing devices (such as Google glass or Apple Watch), smart grid, home automation, intelligent cars and smart cities.

In this course we look at the visions and current scenarios of Pervasive Computing from the security and privacy point of view. We consider security mechanisms and privacy concerns of the present-day technologies, such as smartphone operating systems, GSM/UMTS, WLAN, Bluetooth, ZigBee, RFID, and also of present and envisioned systems and services such vehicular networks, sensor networks, location-based services and augmented reality.

The exercise comprises (1) practical tasks on specific attacks, such as eavesdropping on WiFi or ZigBee communication, and (2) guest talks on selected topics, for example, NFC security. For practical exercises, students will be divided into groups, and each group will have to execute the tasks in our lab and write a report about their work for each task. Further details will be communicated in the first exercise.

**Lernziele und Kompetenzen:**

The students achieve the **main module goals** if they are able to:

- recognize existing and future computing systems as pervasive through analysis of their conceptual design and development, deployment and actual usage
- critically appraise pervasive computing systems for typical security- and privacy-related concerns and weaknesses in design, deployment and usage
- choose appropriate techniques and policies for securing pervasive computing systems
- choose appropriate techniques and policies for addressing privacy issues in pervasive computing systems

**Literatur:**

- Frank Stajano: Security for Ubiquitous Computing.
- John Krumm (Ed.): Ubiquitous Computing Fundamentals



- Adam Greenfield: Everyware

Other books and papers will be presented during the lecture.

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

**[1] Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule | Vertiefungsrichtung IT-Sicherheit)

Dieses Modul ist daneben auch in den Studienfächern "Informatik (Master of Science)", "Information and Communication Technology (Master of Science)", "Informations- und Kommunikationstechnik (Master of Science)", "Mathematik (Bachelor of Science)" verwendbar.

---

**Studien-/Prüfungsleistungen:**

Security and Privacy in Pervasive Computing (Prüfungsnummer: 327615)

(englische Bezeichnung: Security and Privacy in Pervasive Computing)

Prüfungsleistung, mündliche Prüfung, Dauer (in Minuten): 30

Anteil an der Berechnung der Modulnote: 100%

Erstablingung: WS 2019/2020, 1. Wdh.: SS 2020

1. Prüfer: Zinaida Benenson (100191)

---

**Modulbezeichnung:** Fahrzeugkommunikation (FzK) 5 ECTS  
(Vehicular Networks)

Modulverantwortliche/r: Kai-Steffen Jens Hielscher  
Lehrende: Kai-Steffen Jens Hielscher

Startsemester: SS 2019	Dauer: 1 Semester	Turnus: jährlich (SS)
Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Deutsch

**Lehrveranstaltungen:**

Fahrzeugkommunikation (SS 2019, Vorlesung, 2 SWS, Kai-Steffen Jens Hielscher)  
Fahrzeugkommunikation Übungen (SS 2019, Übung, 2 SWS, Kai-Steffen Jens Hielscher)

**Empfohlene Voraussetzungen:**

Rechnerkommunikation

**Inhalt:**

Die Bedeutung von Elektronik und Software ist in der Fahrzeugtechnik stark gestiegen, gleichzeitig stellen die komplexen Entwicklungsprozesse in der Automobilindustrie hohe Anforderungen an Berufseinsteiger. Absolventen benötigen daher zunehmend spezialisierte Kenntnisse aus den Themenbereichen Elektronik, Software und Vernetzung von Fahrzeugen. Um diesen Anforderungen Rechnung zu tragen, wurde am Department Informatik ein spezieller Studienschwerpunkt „Informatik in der Fahrzeugtechnik“ im Studiengang Informatik eingerichtet.

Die Vorlesung „Fahrzeugkommunikation“ [FzK] ist in zwei Blöcke gegliedert: Zunächst führt die Vorlesung in die fahrzeuginterne Vernetzung ein, behandelt Vernetzungsarchitekturen, Bussysteme und Steuergeräte, Fahrerassistenzfunktionen, Multimedia und Systemarchitekturen. Den Abschluss bilden Betrachtungen zu Security und Safety in der fahrzeuginternen Vernetzung. Als zweiten Block gibt die Vorlesung einen Überblick über Themen der externen Kommunikation von und mit Fahrzeugen, behandelt Systemarchitekturen zur Vernetzung von Fahrzeugen untereinander und mit Infrastruktur, Medienzugriffsverfahren verbreiteter Lösungen, Protokolle von Verkehrsinformationssystemen. Sie schließt wiederum mit Betrachtungen zu Safety und Security - erweitert um die in diesem Themengebiet besonders relevanten Fragestellungen rund um die Wahrung der Privatsphäre von Nutzern.

Die Vorlesung wird für einen Abschluss mit Studienschwerpunkt „Informatik in der Fahrzeugtechnik“ anerkannt. Sie setzt Grundkenntnisse in Rechnerkommunikation voraus und richtet sich so schwerpunktmäßig an Studierende aus Informatik, IuK und CE ab dem 5. Semester.

Auszug Interne Vernetzung:

- Interne Steuerung: ECU-ECU, Safety
- Bussysteme (CAN, LIN, FlexRay, MOST, ...)
- HW-, SW- Architekturen von Steuergeräten
- Security & Safety

Auszug Externe Vernetzung

- Car-2-X-Kommunikation
- Topologien, Architekturen
- Medienzugriff: Wifi, WAVE/DSRC
- Safety Anwendungen
- Security & Privacy

**Lernziele und Kompetenzen:**

Die Vorlesung „Fahrzeugkommunikation“ [FzK] ist in zwei Blöcke gegliedert: Zunächst führt die Vorlesung in die fahrzeuginterne Vernetzung ein, behandelt Vernetzungsarchitekturen, Bussysteme und Steuergeräte, Fahrerassistenzfunktionen, Multimedia und Systemarchitekturen. Den Abschluss bilden Betrachtungen zu Security und Safety in der fahrzeuginternen Vernetzung. Als zweiten Block gibt die Vorlesung einen Überblick über Themen der externen Kommunikation von und mit Fahrzeugen, behandelt Systemarchitekturen zur Vernetzung von Fahrzeugen untereinander und mit Infrastruktur, Medienzugriffsverfahren verbreiteter Lösungen, Protokolle von Verkehrsinformationssystemen. Sie schließt wiederum mit Betrachtungen zu Safety und Security - erweitert um die in diesem Themengebiet beson-

ders relevanten Fragestellungen rund um die Wahrung der Privatsphäre von Nutzern. Ziel der Vorlesung ist es, Wissen anhand von Beispielen und Interaktion mit den Studierenden zu vermitteln.

Im Rahmen der Übung werden Protokolle aus der internen Fahrzeugkommunikation implementiert, analysiert und anschliessend evaluiert. Anschließend soll Wissen über die Simulation von Netzwerken durch selbständiges Arbeiten vermittelt werden. Es folgt eine Analyse von aktuellen Lösungen aus dem Bereich der Verkehrssimulation, die im nächsten Schritt mit der Netzwerksimulation gekoppelt wird, um Funknetzwerkprotokolle und -anwendungen mit der Mobilität von Fahrzeugen zu implementieren und evaluieren. Darüber hinaus erstellen die Studierenden eigenständig Lösungen für Anforderungen aus dem Bereich Car-2-X-Kommunikation, bewerten diese auf ihre praktische Anwendbarkeit hin und entwerfen in Kleingruppen Verbesserungsvorschläge dazu.

Das Modul wird für einen Abschluss mit Studienschwerpunkt „Informatik in der Fahrzeugtechnik“ anerkannt. Sie setzt Grundkenntnisse in Rechnerkommunikation voraus und richtet sich so schwerpunktmäßig an Studierende aus Informatik, IuK und CE ab dem 5. Semester.

---

### **Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

[1] **Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule)

Dieses Modul ist daneben auch in den Studienfächern "International Information Systems (IIS) (Master of Science)" verwendbar.

---

### **Studien-/Prüfungsleistungen:**

Fahrzeugkommunikation (Prüfungsnummer: 716033)

(englische Bezeichnung: Vehicular Networks)

Prüfungsleistung, Portfolio

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Fahrzeugkommunikation
- Fahrzeugkommunikation Übungen

weitere Erläuterungen:

Die Modulprüfung besteht aus:

- unbenotete Studienleistung, zu erwerben durch erfolgreiche Teilnahme an den Übungen
- Klausur von 90 Minuten oder mündliche Prüfung bei weniger als 20 Teilnehmern

Erstablingung: SS 2019, 1. Wdh.: WS 2019/2020 (nur für Wiederholer)

1. Prüfer: Kai-Steffen Jens Hielscher (100125)

---

---

**Modulbezeichnung: Smart Grids und Elektromobilität (SGEM)** **5 ECTS**  
 (Smart Grids und Elektromobilität)

Modulverantwortliche/r: Marco Pruckner  
 Lehrende: Marco Pruckner

---

Startsemester: SS 2019	Dauer: 1 Semester	Turnus: halbjährlich (WS+SS)
Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Deutsch

---

**Lehrveranstaltungen:**

Smart Grids und Elektromobilität (SS 2019, Vorlesung, 2 SWS, Marco Pruckner)  
 Übungen zu Smart Grids und Elektromobilität (SS 2019, Übung, 2 SWS, Marco Pruckner)

---

**Inhalt:**

Dieser Kurs dient als Einführung in den neuen multi-disziplinären Bereich Smart Grids. Dabei wird ein Überblick zu den wesentlichen neuen Aspekten moderner Technologien in Erzeugung, Übertragung und Verteilung gegeben. Zudem werden relevante Aspekte der Kommunikationstechnologien wiedergegeben, die zur intelligenten Koordinierung zwischen den verschiedenen Einheiten innerhalb des Stromnetzes eingesetzt werden. Neben einer Einführung der Konzepte der verschiedenen Komponenten des Smart Grids, einschließlich der Integration erneuerbarer Energien, Energiespeicher und Demand Side Management wird die Elektromobilität und die Integration von Elektrofahrzeugen in das Stromnetz einen weiteren Schwerpunkt dieses Kurses bilden.

Die Vorlesung behandelt zentrale Fragestellungen zu Smart Grids und Elektromobilität:

- Vision und Strategie für die elektrischen Netze der Zukunft
- Smarte Erzeugung elektrischer Energie
- Aufbau und Betrieb von Übertragungs- und Verteilungsnetzen
- Informations- und Kommunikationstechnologien als Rückgrat des Smart Grids
- Technologien für das Laden von Elektrofahrzeugen
- Zusammenspiel von Elektrofahrzeugen und dem Energiesystem
- Umweltaspekte, einschließlich CO<sub>2</sub>-Emissionen

**Lernziele und Kompetenzen:**

Die Studierenden

- unterscheiden Probleme und Herausforderungen die mit den künftigen Stromnetzen verbunden sind
  - erfassen die Vorteile und die Anwendungsmöglichkeiten, die beim Einsatz von IKT im Stromnetz möglich sind, wie z.B. Demand Response
  - analysieren verschiedene Probleme, die beim Einsatz von IKT im Stromnetz entstehen, z.B. Datenschutzprobleme
  - erlernen verschiedene Technologien zur Erzeugung und Verteilung von Energie kennen
  - kennen die Rolle von Elektromobilität sowie die speziellen Herausforderungen zur Integration von Elektrofahrzeugen in das bestehende bzw. zukünftige Stromnetz
- 

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

**[1] Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule)

Dieses Modul ist daneben auch in den Studienfächern "International Information Systems (IIS) (Master of Science)" verwendbar.

---

**Studien-/Prüfungsleistungen:**

Smart Grid (Prüfungsnummer: 623734)

(englische Bezeichnung: Smart Grid)

Prüfungsleistung, Portfolio

Anteil an der Berechnung der Modulnote: 100%

weitere Erläuterungen:

Die Modulprüfung besteht aus:

- unbenotete Studienleistung, zu erwerben durch erfolgreiche Teilnahme an den Übungen
- schriftliche Klausur (90 Minuten) oder mündl. Prüfung (30 Min. bei weniger als 20 Teilnehmern)

Erstablingung: SS 2019, 1. Wdh.: WS 2019/2020

1. Prüfer: Marco Pruckner (100381)

---

**Modulbezeichnung:** Fahrzeugkommunikation (FzK-VL) 2.5 ECTS  
(Vehicular Networks)

Modulverantwortliche/r: Kai-Steffen Jens Hielscher  
Lehrende: Kai-Steffen Jens Hielscher

Startsemester: SS 2019	Dauer: 1 Semester	Turnus: jährlich (SS)
Präsenzzeit: 30 Std.	Eigenstudium: 45 Std.	Sprache: Deutsch

**Lehrveranstaltungen:**

Fahrzeugkommunikation (SS 2019, Vorlesung, 2 SWS, Kai-Steffen Jens Hielscher)

**Inhalt:**

Die Bedeutung von Elektronik und Software ist in der Fahrzeugtechnik stark gestiegen, gleichzeitig stellen die komplexen Entwicklungsprozesse in der Automobilindustrie hohe Anforderungen an Berufseinsteiger. Absolventen benötigen daher zunehmend spezialisierte Kenntnisse aus den Themenbereichen Elektronik, Software und Vernetzung von Fahrzeugen. Um diesen Anforderungen Rechnung zu tragen, wurde am Department Informatik ein spezieller Studienschwerpunkt „Informatik in der Fahrzeugtechnik“ im Studiengang Informatik eingerichtet. Die Vorlesung „Fahrzeugkommunikation“ [FzK] ist in zwei Blöcke gegliedert: Zunächst führt die Vorlesung in die fahrzeuginterne Vernetzung ein, behandelt Vernetzungsarchitekturen, Bussysteme und Steuergeräte, Fahrerassistenzfunktionen, Multimedia und Systemarchitekturen. Den Abschluss bilden Betrachtungen zu Security und Safety in der fahrzeuginternen Vernetzung. Als zweiten Block gibt die Vorlesung einen Überblick über Themen der externen Kommunikation von und mit Fahrzeugen, behandelt Systemarchitekturen zur Vernetzung von Fahrzeugen untereinander und mit Infrastruktur, Medienzugriffsverfahren verbreiteter Lösungen, Protokolle von Verkehrsinformationssystemen. Sie schließt wiederum mit Betrachtungen zu Safety und Security - erweitert um die in diesem Themengebiet besonders relevanten Fragestellungen rund um die Wahrung der Privatsphäre von Nutzern. Die Vorlesung wird für einen Abschluss mit Studienschwerpunkt „Informatik in der Fahrzeugtechnik“ anerkannt. Sie setzt Grundkenntnisse in Rechnerkommunikation voraus und richtet sich so schwerpunktmäßig an Studierende aus Informatik, IuK und CE ab dem 5. Semester.

Auszug Interne Vernetzung:

- Interne Steuerung: ECU-ECU, Safety
- Bussysteme (CAN, LIN, FlexRay, MOST, ...)
- HW-, SW- Architekturen von Steuergeräten
- Security & Safety

Auszug Externe Vernetzung

- Car-2-X-Kommunikation
- Topologien, Architekturen
- Medienzugriff: Wifi, WAVE/DSRC
- Safety Anwendungen
- Security & Privacy

**Lernziele und Kompetenzen:**

Die Vorlesung „Fahrzeugkommunikation“ [FzK] ist in zwei Blöcke gegliedert: Zunächst führt die Vorlesung in die fahrzeuginterne Vernetzung ein, behandelt Vernetzungsarchitekturen, Bussysteme und Steuergeräte, Fahrerassistenzfunktionen, Multimedia und Systemarchitekturen. Den Abschluss bilden Betrachtungen zu Security und Safety in der fahrzeuginternen Vernetzung. Als zweiten Block gibt die Vorlesung einen Überblick über Themen der externen Kommunikation von und mit Fahrzeugen, behandelt Systemarchitekturen zur Vernetzung von Fahrzeugen untereinander und mit Infrastruktur, Medienzugriffsverfahren verbreiteter Lösungen, Protokolle von Verkehrsinformationssystemen. Sie schließt wiederum mit Betrachtungen zu Safety und Security - erweitert um die in diesem Themengebiet besonders relevanten Fragestellungen rund um die Wahrung der Privatsphäre von Nutzern. Ziel der Vorlesung ist es, Wissen anhand von Beispielen und Interaktion mit den Studierenden zu vermitteln.

Das Modul wird für einen Abschluss mit Studienschwerpunkt „Informatik in der Fahrzeugtechnik“ anerkannt. Sie setzt Grundkenntnisse in Rechnerkommunikation voraus und richtet sich so schwerpunktmäßig an Studierende aus Informatik, IuK und CE ab dem 5. Semester.

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

**[1] Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule)

Dieses Modul ist daneben auch in den Studienfächern "International Information Systems (IIS) (Master of Science)" verwendbar.

---

**Studien-/Prüfungsleistungen:**

Fahrzeugkommunikation (Prüfungsnummer: 730178)

(englische Bezeichnung: Vehicular Networks)

Prüfungsleistung, Portfolio

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Fahrzeugkommunikation

weitere Erläuterungen:

Die Modulprüfung besteht aus einer Klausur von 90 Minuten oder mündlicher Prüfung bei weniger als 20 Teilnehmern

Prüfungssprache: Deutsch oder Englisch

Erstablingung: SS 2019, 1. Wdh.: WS 2019/2020

1. Prüfer: Kai-Steffen Jens Hielscher (100125)

---

---

**Modulbezeichnung: Human Computer Interaction (HCI)**  
 (Human Computer Interaction)

**5 ECTS**

 Modulverantwortliche/r: Björn Eskofier  
 Lehrende: Björn Eskofier

Startsemester: SS 2019

Dauer: 1 Semester

Turnus: jährlich (SS)

Präsenzzeit: 60 Std.

Eigenstudium: 90 Std.

Sprache: Deutsch oder Englisch

---

**Lehrveranstaltungen:**

Human Computer Interaction (SS 2019, Vorlesung, 3 SWS, Björn Eskofier)

Human Computer Interaction Exercises (SS 2019, Übung, 1 SWS, Markus Wirth)

---

**Inhalt:**

Die Vorlesung vermittelt Konzepte, Prinzipien, Modelle, Methoden und Techniken für die effektive Entwicklung von benutzerfreundlichen Mensch-Computer-Schnittstellen. Das Thema moderner Benutzungsschnittstellen wird dabei für klassische Computer aber auch für mobile Geräte, eingebettete Systeme, Automobile und intelligente Umgebungen betrachtet.

Die folgenden Themen werden in der Vorlesung behandelt:

- Einführung in die Grundlagen der Mensch-Computer-Interaktion, historische Entwicklung
- Entwurfsprinzipien und Modelle für moderne Benutzungsschnittstellen und interaktive Systeme
- Informationsverarbeitung des Menschen, Wahrnehmung, Motorik, Eigenschaften und Fähigkeiten des Benutzers
- Interaktionskonzepte und -stile, Metaphern, Normen, Regeln und Style Guides
- Ein- und Ausgabegeräte, Entwurfsraum für interaktive Systeme
- Analyse-, Entwurfs- und Entwicklungsmethoden und -werkzeuge für Benutzungsschnittstellen
- Prototypische Realisierung und Implementierung von interaktiven Systemen, Werkzeuge
- Architekturen für interaktive Systeme, User Interface Toolkits und Komponenten
- Akzeptanz, Evaluationsmethoden und Qualitätssicherung

Contents: Aim of the lecture is to teach basic knowledge of concepts, principles, models, methods and techniques for developing highly user-friendly Human Computer Interfaces. Beyond traditional computer system the topic of modern user interfaces is also discussed in the context of automobile and intelligent environments, mobile devices and embedded systems.

This lecture addresses the following topics:

- Introduction to the basics of Human Computer Interaction
- Design principles and models for modern user interfaces and interactive systems
- Information processing of humans, perception, motor skills, properties and skills of the users
- Interaction concepts, metaphors, standards, norms and style guides
- In- and output devices, design space for interactive systems
- Analysis-, design- and development methodologies and tools for easy to use user interfaces
- Prototypic implementation of interactive systems
- Architectures for interactive systems, User Interface Toolkits and components
- Acceptance, evaluation methods and quality assurance

**Lernziele und Kompetenzen:**

- Studierende entwickeln ein Verständnis für Modelle, Methoden und Konzepte der Mensch-Computer-Interaktion.
- Sie lernen verschiedene Ansätze für den Entwurf, die Entwicklung und Bewertung von Benutzungsschnittstellen kennen und verstehen deren Vor- und Nachteile.
- Die Teilnahme an der Veranstaltung versetzt Studierende in die Lage, einen Entwicklungsprozess in der Mensch-Computer-Interaktion zu verstehen und umzusetzen.
- Sie werden weiterhin in die Lage versetzt, dies vor dem Hintergrund der Informationsverarbeitungsfähigkeit, Wahrnehmung und Motorik des Benutzers zu gestalten.
- Schlussendlich werden Methoden der Evaluation sowie Akzeptanz- und Qualitätssicherung erlangt.

Learning Objectives and Competences:



- Students develop an understanding for models, methods and concepts in the field of Human-Computer Interaction.
- They learn different approaches for designing, developing and evaluating User Interfaces and their advantages and disadvantages.
- Joining the course enables students to understand and execute a development process in the area of Human-Computer Interaction.
- Student will be able to do an UI evaluation by learning basics about Informationprocessing, perception and motoric skills of the user.
- Additionally, appropriate evaluation method as well as acceptance and quality assurance aspects will be learned.

---

### **Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

#### **[1] Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule)

Dieses Modul ist daneben auch in den Studienfächern "International Information Systems (IIS) (Master of Science)" verwendbar.

---

### **Studien-/Prüfungsleistungen:**

Human Computer Interaction (Prüfungsnummer: 645618)

(englische Bezeichnung: Human Computer Interaction)

Prüfungsleistung, Klausur, Dauer (in Minuten): 90

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Human Computer Interaction
- Human Computer Interaction Exercises

weitere Erläuterungen:

90-minütige schriftliche Klausur über den Stoff der Vorlesung und der Übungen. Bei weniger als 25 Teilnehmern erfolgt die Prüfung nach vorheriger Ankündigung mündlich (Dauer: 30 Minuten).

Prüfungssprache: Deutsch oder Englisch

Erstablingung: SS 2019, 1. Wdh.: WS 2019/2020

1. Prüfer: Björn Eskofier (100122)

---

### **Organisatorisches:**

Folien zur Vorlesung und Organisation über Studon.

Organisation and slides via StudOn.

---

**Modulbezeichnung:** **Automotive Systems & Software Engineering (ASSE)** **5 ECTS**  
 (Automotive Systems & Software Engineering)

Modulverantwortliche/r: Reinhard German  
 Lehrende: Christian Allmann

---

Startsemester: SS 2019	Dauer: 1 Semester	Turnus: jährlich (SS)
Präsenzzeit: 40 Std.	Eigenstudium: 110 Std.	Sprache: Deutsch

---

**Lehrveranstaltungen:**

Automotive Systems & Software Engineering (SS 2019, Vorlesung, 4 SWS, Christian Allmann et al.)

---

**Inhalt:**

Allgemein:

- Überblick Systeme und Anwendungen in der automotive Welt
- Embedded Automotive Systems: Abgrenzung Echtzeitsysteme; Definition Steuergeräte, Sensoren, Aktuatoren; Definition verteilte, vernetzte Funktionen; Betriebssysteme, Kommunikationsschnittstellen
- Entwicklungsprozesse für automotive Anwendungen: Allgemeine Vorgehensmodelle; Vorgehensmodelle in der Elektronikentwicklung; Kooperationsmodelle OEM-Zulieferer; Lieferantenmanagement

System und Software:

- Softwareentwicklungsprozess
- Anforderungsmanagement: Lastenhefte - Aufbau/Handhabung/Tools; Erhebung und Analyse von Anforderungen; Systembeschreibungen (u.a. UML, SysML); (Semi-)Formale Beschreibungen
- Modellbasierte Entwicklung: Rapid Prototyping; Modellierungstechniken; Automatische Codegenerierung; Tools (Simulink, Target Link); Autosar
- Test und Diagnose: SW-Test; Integrationsstufen; SIL - PIL - HIL-Test; On- Offboard Diagnose
- Virtuelle Entwicklung von Elektronik
- Begleitende Prozesse: Projektmanagement; Systemsicherheit (ISO WD 26262); Konfigurationsmanagement; Qualitätsmanagement; Variantenmanagement
- Architektur: Architekturmodelle (u.a. EAST-ADL); Bussysteme - Typen & Eigenschaften; SW- & HW-Architektur; Auslegung und Bewertung

Anwendung:

- Überblick der Anwendungsdomänen: Infotainment/ Fahrerassistenz/ Karosserie
- Systemauslegung von Fahrerassistenzsystemen: Überblick Sensoren; Umwelt und Umfeldmodelle; Bildverarbeitung; Konzeption, Erprobung, Umsetzung; Tools (ADTF)
- Alternative Antriebskonzepte: Elektronikumfänge; Betriebsstrategie
- Fahrdynamische Systeme: x-by-wire Technologien

**Lernziele und Kompetenzen:**

- Vertiefung der Themen aus „Entwicklung und Test von verteilten, eingebetteten Systemen im Bereich Automotive“WS, Dr. Hehn
- Herausforderungen in der System- und Softwareentwicklung aus Sicht eines OEMs kennenlernen
- Bedeutung des Requirements Engineering in Vor- & Serienentwicklung verstehen und die Methoden anwenden können
- Herausforderungen in der Entwicklung von Fahrerassistenzsystemen erkennen und die Konzepte nachvollziehen können
- Entwicklung eines eigenen FAS durchführen und prüfen
- Erfahrung im Umgang mit typischen Entwicklungstools
- Absicherung der Systementwicklung am HIL
- Neue Ideen und Themen im Bereich automotive Entwicklung verstehen

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

**[1] Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule)

**Studien-/Prüfungsleistungen:**

Automotive Systems & Software Engineering (Prüfungsnummer: 313638)

(englische Bezeichnung: Automotive Systems & Software Engineering)

Prüfungsleistung, Klausur, Dauer (in Minuten): 90

Anteil an der Berechnung der Modulnote: 100%

Erstablesung: SS 2019, 1. Wdh.: WS 2019/2020

1. Prüfer: Anatoli Djanatliev (100361)

---

**Modulbezeichnung:** Dienstgüte von Kommunikationssystemen (DKS) **5 ECTS**  
(Quality of Service of Communication Systems)

Modulverantwortliche/r: Reinhard German  
Lehrende: Reinhard German

Startsemester: SS 2019	Dauer: 1 Semester	Turnus: jährlich (SS)
Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Englisch

**Lehrveranstaltungen:**

Quality of Service of Communication Systems / Dienstgüte von Kommunikationssystemen (SS 2019, Vorlesung, 2 SWS, Kai-Steffen Jens Hielscher)  
Dienstgüte von Kommunikationssystemen Übungen (SS 2019, Übung, 2 SWS, Lisa Maile)

**Empfohlene Voraussetzungen:**

Rechnerkommunikation, Kommunikationssysteme, Programmierkenntnisse in Java

**Inhalt:**

Zunächst wird der Begriff der Dienstgüte (Quality-of-Service, QoS) eingegrenzt und es werden die wichtigsten Ansätze zur Erzielung von Dienstgüte besprochen und in ausgewählten Netztechnologien untersucht. Dann werden unterschiedliche Methoden vorgestellt, mit denen Systeme bezüglich ihrer Dienstgüte bewertet und ausgelegt werden können:

- Netzplanung und -optimierung,
- stochastische Analyse (Markow-Ketten, Warteschlangen),
- Netzwerksimulation,
- deterministische Analyse mit Network Calculus zur Ermittlung von Dienstgütegarantien
- Messung (HW-, SW-, Hybrid-Monitoring, Benchmarks).

Alle Methoden werden an Beispielen demonstriert.

Contents:

We introduce the term quality-of-service (QoS), discuss important approaches to achieve certain degrees of QoS, and show how the implementation in computer networks. Then a number of methodologies to assess and design systems with respect to their QoS:

- network planning and optimization,
- network simulation,
- stochastic analysis (Markov chains, non-Markovian models, queuing systems),
- deterministic analysis with network calculus to determine QoS guarantees
- measurements (hardware, software, and hybrid monitoring, benchmarks).

All methods are illustrated by examples.

**Lernziele und Kompetenzen:**

Die Studierenden erwerben

- Kenntnisse in Methoden zur Modellierung und Bewertung von quantitativen, nicht-funktionalen Eigenschaften von vernetzten Systemen
- Kenntnisse in Mechanismen von vernetzten Systemen zur Erzielung von Dienstgüte

Learning targets and competences:

The students get

- experience in methods to model and evaluate quantitative, non-functional properties of computer networks and related systems
- knowledge of mechanisms of computer networks to achieve quality-of-service

**Literatur:**

- Kurose, Ross. Computer Networking: A Top-Down Approach Featuring the Internet. 6th Ed., Addison Wesley, 2013
- W. Stallings. Data and Computer Communications, 10th ed., Pearson Education, 2014
- W. Stallings. Foundations of Modern Networking: SDN, NFV, QoE, IoT, and Cloud, Pearson Education, 2016

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

**[1] Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule)

Dieses Modul ist daneben auch in den Studienfächern "Information and Communication Technology (Master of Science)", "Informations- und Kommunikationstechnik (Master of Science)", "International Information Systems (IIS) (Master of Science)" verwendbar.

---

**Studien-/Prüfungsleistungen:**

Dienstgüte von Kommunikationssystemen (Vorlesung und Übung) (Prüfungsnummer: 472330)

(englische Bezeichnung: Quality of Service of Communication Systems (Lecture and Exercise))

Prüfungsleistung, Portfolio

weitere Erläuterungen:

Die Modulprüfung besteht aus:

- unbenotete Studienleistung, zu erwerben durch erfolgreiche Teilnahme an den Übungen
- Klausur von 90 Minuten oder mündliche Prüfung bei weniger als 20 Teilnehmern

Erstablingung: SS 2019, 1. Wdh.: WS 2019/2020

1. Prüfer: Kai-Steffen Jens Hielscher (100125)

---

**Organisatorisches:**

Alles Material ist in Englisch, die Vorlesung und die Übungen werden in Englisch gehalten.

---

**Modulbezeichnung:** **Kommunikationssysteme-VÜ (KS-VÜ)** **5 ECTS**  
 (Communication Systems LeEx)

Modulverantwortliche/r: Reinhard German  
 Lehrende: Reinhard German

---

Startsemester: WS 2019/2020	Dauer: 1 Semester	Turnus: jährlich (WS)
Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Deutsch

---

**Lehrveranstaltungen:**

Kommunikationssysteme (WS 2019/2020, Vorlesung, 2 SWS, Reinhard German)  
 Übungen zu Kommunikationssysteme (WS 2019/2020, Übung, 2 SWS, Alexander Brummer et al.)

---

**Inhalt:**

Aus Rechnerkommunikation ist der grundlegende Aufbau von IP-basierten Netzen bekannt, Inhalt von Kommunikationssysteme sind weitere Netztechnologien wie Leitungsvermittlung (ISDN, Sonet/SDH) und Netze mit virtueller Leitungsvermittlung (ATM, MPLS), Multimediakommunikation über paketvermittelte Netze (Streaming, RTP, H.323, SIP, Multicast), Dienstgüte in paketvermittelten Netzen (Integrated Services, RSVP, Differentiated Services, Active Queue Management, Policing, Scheduling), drahtlose und mobile Kommunikation (GSM, UMTS, Wimax, WLAN, Bluetooth, ZigBee u.a. Sensornetze, Mobile IP) sowie Kommunikation in der Fahrzeug- und Automatisierungstechnik. Weiterhin werden Verfahren zum Systemdesign behandelt: Spezifikation von Architekturen und Protokollen (SDL, MSC, ASN.1, UML), Analyseverfahren, Simulation, Messung, Test. In der Übung werden praktische Aufgaben im Labor durchgeführt: ein Labor enthält mehrere IP-Router, Switches und Rechner, IP-Telefone und Telefonie-Software für VoIP, es werden verschiedene Konfigurationen eingestellt und getestet, ein weiteres Labor besteht aus eingebetteten Geräten mit CAN-Bus zur Kontrolle von Aktoren und Sensoren, die mittels Java programmiert werden.

Contents:

Based on the course computer communications the architecture of IP networks is known. Contents of this course will be additional networking technologies such as circuit switching (telephony, SONET/SDH/WDM) and networks with virtual circuit switching (ATM, MPLS) as well as network virtualization (SDN, NFV), multimedia communications over packet switched networks (streaming, RT, SIP, multicast), quality-of-service in packet switched networks (integrated services, RSVP, differentiated services, active queue management, policing, scheduling), wireless and mobile communications (GSM, UMTS, LTE, 5G, Wimax, WLAN, Bluetooth, sensor networks such as ZigBee). Industrial communication will also be a topic. In the tutorial practical tasks are performed in the laboratory: One laboratory contains several IP routers, switches and computers, IP phones and VoIP telephone software with which various configurations are set up and tested. Another laboratory consists of embedded devices with a CAN bus for controlling actuators and sensors that are programmed using Java.

**Lernziele und Kompetenzen:**

- Kenntnisse über Technologien bei der Leitungs- und Paketvermittlung in leitungsgebundenen und drahtlosen/mobilen Netzen
- Kenntnisse über Systemdesign
- praktische Erfahrung in der Konfiguration eines IP-Switch-Router-Netzes mit Multimedieverkehr sowie in der Programmierung vernetzter eingebetteter Systeme

Learning targets and competences:

- Knowledge of technologies in circuit and packet switching in wireline and wireless / mobile networks
- Knowledge of network system design
- Practical experience in configuring an IP switch router network with multimedia traffic as well as programming interconnected embedded systems

**Literatur:**

Lehrbuch: Kurose, Ross, "Computer Networking: A Top-Down Approach Featuring the Internet", 4th Ed., Addison Wesley, 2007

---

## Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

### [1] Informatik (Bachelor of Science)

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule | Vertiefungsrichtung Kommunikationssysteme)

Dieses Modul ist daneben auch in den Studienfächern "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Bachelor of Science)", "Informatik (Master of Science)", "Information and Communication Technology (Master of Science)", "Informations- und Kommunikationstechnik (Bachelor of Science)", "Informations- und Kommunikationstechnik (Master of Science)", "International Information Systems (IIS) (Master of Science)", "Mathematik (Bachelor of Science)", "Medizintechnik (Bachelor of Science)" verwendbar.

---

## Studien-/Prüfungsleistungen:

Kommunikationssysteme (Vorlesung mit Übungen) (Prüfungsnummer: 39501)

(englische Bezeichnung: Communication Systems LeEx)

Prüfungsleistung, Klausur, Dauer (in Minuten): 90

Anteil an der Berechnung der Modulnote: 100%

weitere Erläuterungen:

Die Modulprüfung besteht aus:

- unbenotete Studienleistung, zu erwerben durch erfolgreiche Teilnahme an den Übungen
- Klausur von 90 Minuten oder mündliche Prüfung bei weniger als 20 Teilnehmern

Erstablingung: WS 2019/2020, 1. Wdh.: SS 2020

1. Prüfer: Reinhard German (100165)

---

---

**Modulbezeichnung:** Modellierung, Optimierung und Simulation von 5 ECTS  
**Energiesystemen (MOSES)**

(Modeling, Optimization and Simulation of Energy Systems)

Modulverantwortliche/r: Marco Pruckner

Lehrende: Marco Pruckner

---

Startsemester: WS 2019/2020

Dauer: 1 Semester

Turnus: jährlich (WS)

Präsenzzeit: 60 Std.

Eigenstudium: 90 Std.

Sprache: Deutsch

---

**Lehrveranstaltungen:**

Modellierung, Optimierung und Simulation von Energiesystemen (WS 2019/2020, Vorlesung, 2 SWS, Marco Pruckner)

Übungen zu Modellierung, Optimierung und Simulation von Energiesystemen (WS 2019/2020, Übung, 2 SWS, N.N.)

---

**Inhalt:**

In der Vorlesung Modellierung, Optimierung und Simulation von Energiesystemen werden systemtechnische Planungs- und Analysemethoden behandelt, die zur Lösung komplexer und interdisziplinärer Entscheidungsaufgaben in der Energiewirtschaft eingesetzt werden. Dabei werden die wichtigsten Methoden und Verfahren anhand praktischer Fragestellungen (z.B. Ausbau erneuerbarer Energien, Zunahme der Elektromobilität) aus der energiepolitischen Planung vermittelt und die Bewältigung technisch-ökonomischer Probleme verdeutlicht.

Übersicht der Vorlesungsinhalte:

- Einführung in die Energiewirtschaft und Systemtechnik
- Systemtechnische Methoden der Energieplanung
  - o Datenanalyse (Regressionsanalysen, Clusteranalyse, Zeitreihenanalyse)
  - o Mathematische Optimierung (Lineare Programmierung, Gemischt-ganzzahlige lineare Programmierung, dynamische Optimierung)
  - o Simulation (Diskrete Ereignissimulation, System Dynamics, agentenbasierte Simulation)
  - o Input-Output-Analyse, Gleichgewichtsmodelle
  - o Behandlung von Unsicherheiten
- Energiemodelle
  - o Energienachfragemodelle
  - o Kraftwerkseinsatzmodelle
  - o Kraftwerksausbaumodelle
  - o Modelle für Energieversorgungsmodelle

Zu den eingesetzten Tools zählen die Statistiksoftware R, AnyLogic und IpSolve. Vorkenntnisse im Umgang mit diesen Werkzeugen ist nicht zwingend erforderlich. In den Übungen werden Einführungen in die genannten Softwarepakete gegeben.

**Lernziele und Kompetenzen:**

Die Studierenden

- unterscheiden Probleme und Herausforderungen, die mit dem Energieumstieg verbunden sind,
- erfassen die Vorteile und die Anwendungsmöglichkeiten computergestützter Planungsmethoden im Energiebereich,
- analysieren verschiedene Problemstellungen und setzen Lösungen dafür um,
- erlernen verschiedene Methoden der Datenanalyse, Optimierung und Simulation.

**Literatur:**

Vorlesungsskript wird ausgegeben

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:



## [1] Informatik (Bachelor of Science)

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule | Vertiefungsrichtung Kommunikationssysteme)

Dieses Modul ist daneben auch in den Studienfächern "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Master of Science)", "Elektrotechnik, Elektronik und Informationstechnik (Bachelor of Science)", "Energietechnik (Master of Science)", "Informatik (Master of Science)", "Information and Communication Technology (Master of Science)", "Informations- und Kommunikationstechnik (Master of Science)", "International Information Systems (IIS) (Master of Science)", "Maschinenbau (Master of Science)", "Mathematik (Bachelor of Science)" verwendbar.

---

### Studien-/Prüfungsleistungen:

Modellierung, Optimierung und Simulation von Energiesystemen (Prüfungsnummer: 858896)

(englische Bezeichnung: Modeling, Optimization and Simulation of Energy Systems)

Prüfungsleistung, Portfolio

Anteil an der Berechnung der Modulnote: 100%

weitere Erläuterungen:

Die Modulprüfung besteht aus:

- unbenotete Studienleistung, zu erwerben durch erfolgreiche Teilnahme an den Übungen
- schriftliche Klausur (90 Minuten) oder mündl. Prüfung (30 Min. bei weniger als 30 Teilnehmern)

Prüfungssprache: Deutsch

Erstablingung: WS 2019/2020, 1. Wdh.: SS 2020

1. Prüfer: Marco Pruckner (100381)

---

---

**Modulbezeichnung:** Künstliche Intelligenz II (KI II) 7.5 ECTS  
 (Artificial Intelligence II)

Modulverantwortliche/r: Michael Kohlhase  
 Lehrende: Michael Kohlhase

---

Startsemester: SS 2019	Dauer: 1 Semester	Turnus: jährlich (SS)
Präsenzzeit: 90 Std.	Eigenstudium: 135 Std.	Sprache: Deutsch oder Englisch

---

**Lehrveranstaltungen:**

Künstliche Intelligenz II (SS 2019, Vorlesung, 4 SWS, Michael Kohlhase)  
 Übungen zu Künstliche Intelligenz II (SS 2019, Übung, 2 SWS, Dennis Müller)

---

**Inhalt:**

Dieser Kurs beschäftigt sich mit den Grundlagen der Künstlichen Intelligenz (KI), insbesondere mit Techniken des Schliessens unter Unsicherheit, des maschinellen Lernens und dem Sprachverstehen. Der Kurs baut auf der Vorlesung Künstliche Intelligenz I vom Wintersemester auf und führt diese weiter.

**Lernziele und Kompetenzen:**

Fach- Lern- bzw. Methodenkompetenz

- Wissen: Die Studierenden lernen grundlegende Repräsentationsformalismen und Algorithmen der Künstlichen Intelligenz kennen.
- Anwenden: Die Konzepte werden an Beispielen aus der realen Welt angewandt (Übungsaufgaben).
- Analyse: Die Studierenden lernen über die Modellierung in der Maschine menschliche Intelligenzleistungen besser einzuschätzen.

Sozialkompetenz

- Die Studierenden arbeiten in Kleingruppen zusammen um kleine Projekte zu bewältigen.

This course covers the foundations of Artificial Intelligence (AI), in particular reasoning under uncertainty, machine learning and (if there is time) natural language understanding. This course builds on the course Artificial Intelligence I from the preceding winter semester and continues it. Learning Goals and Competencies Technical, Learning, and Method Competencies

- Knowledge: The students learn foundational representations and algorithms in AI.
- Application: The concepts learned are applied to examples from the real world (homeworks ).
- Analysis: By modeling human cognitive abilities, students learn to assess and understand human intelligence better.
- Social Competencies: Students work in small groups to solve the and machine learning challenge/competition.

**Literatur:**

Die Vorlesung folgt weitgehend dem Buch Stuart Russell und Peter Norvig: Artificial Intelligence: A Modern Approach. Prentice Hall, 3rd edition, 2009.

Deutsche Ausgabe: Stuart Russell und Peter Norvig: Künstliche Intelligenz: Ein Moderner Ansatz. Pearson-Studium, 2004 (Übersetzung der 2. Auflage).

ISBN: 978-3-8273-7089-1.

Literature The course follows the following textbook: Stuart Russell and Peter Norvig: Artificial Intelligence: A Modern Approach. Prentice Hall, 3rd edition, 2009.

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

[1] **Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule)

---

**Studien-/Prüfungsleistungen:**

Künstliche Intelligenz II (Prüfungsnummer: 532733)

(englische Bezeichnung: Artificial Intelligence II)

Prüfungsleistung, Klausur, Dauer (in Minuten): 90

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Künstliche Intelligenz II
- Übungen zu Künstliche Intelligenz II

weitere Erläuterungen:

Die Prüfung besteht aus einer Klausur (90 Minuten). Auf Basis der Bewertungen der abgegebenen Übungsaufgaben können bis zu 10 % Bonuspunkte erworben werden, die zu dem Ergebnis einer bestandenen Klausur hinzugerechnet werden.

Prüfungssprache: Deutsch

Erstablingung: SS 2019, 1. Wdh.: WS 2019/2020 (nur für Wiederholer)

1. Prüfer: Michael Kohlhasse (100451)

---

---

**Modulbezeichnung:** Logik-Basierte Wissensrepräsentation für 5 ECTS  
**Mathematisch/Technisches Wissen (KRMT)**  
 (Logic-based knowledge representation for mathematic/technical knowledge)

Modulverantwortliche/r: Michael Kohlhase  
 Lehrende: Michael Kohlhase

---

Startsemester: SS 2019	Dauer: 1 Semester	Turnus: jährlich (SS)
Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Deutsch oder Englisch

---

**Lehrveranstaltungen:**

Logik-Basierte Wissensrepräsentation für Mathematisch/Technisches Wissen (SS 2019, Vorlesung mit Übung, 4 SWS, Michael Kohlhase et al.)

---

**Inhalt:**

Grundlagen der Mathematik, Modulare Formalisierung in Theoriegraphen, Narrative Strukturen in informellen mathematisch/technischen Dokumenten, Formalisierung von Logiksprachen in Metalogiken.

**Lernziele und Kompetenzen:**

- Fachkompetenz
- Wissen

Die Studierenden beherrschen moderne, sehr expressive Formalismen (Syntax, Semantik, Beweistheorie) zur Repräsentation komplexen Wissens. Sie können diese selbst in Metalogiken repräsentieren und eingeschränkt neue Formalismen entwickeln.

- Anwenden

Die Studierenden formalisieren komplexe mathematisch/technische Sachverhalte in jeweils geeigneten Meta-Sprachen und setzen diese durch Interpretationsabbildungen in Verbindung.

- Analysieren

Die Studierenden analysieren die innere Struktur komplexer Objekte, Konzepte, und Modelle. Sie wählen für eine zu repräsentierende mathematisch/technische Domäne geeignete Formalismen aus.

- Lern- bzw. Methodenkompetenz Die Studierenden erarbeiten selbständig formale Repräsentationen und Beweise.
- Sozialkompetenz Die Studierenden arbeiten in Kleingruppen erfolgreich zusammen.

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

**[1] Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule)

---

**Studien-/Prüfungsleistungen:**

Logik-Basierte Wissensrepräsentation für Mathematisch/Technisches Wissen (Prüfungsnummer: 221236)

(englische Bezeichnung: Logic-based knowledge representation for mathematic/technical knowledge)

Prüfungsleistung, Portfolio

Anteil an der Berechnung der Modulnote: 100%

weitere Erläuterungen:

Die Modulnote setzt sich zu 50% aus dem Ergebnis einer 20-minütigen mündlichen Prüfung am Semesterende und zu 50% aus der Bewertung der Leistungen aus dem Übungsbetrieb zusammen.

Prüfungssprache: Deutsch

Erstablingung: SS 2019, 1. Wdh.: WS 2019/2020

1. Prüfer: Michael Kohlhase (100451)

---

---

**Modulbezeichnung:** Künstliche Intelligenz I (KI I) 7.5 ECTS  
(Artificial Intelligence I)

Modulverantwortliche/r: Michael Kohlhase  
Lehrende: Michael Kohlhase

---

Startsemester: WS 2019/2020	Dauer: 1 Semester	Turnus: jährlich (WS)
Präsenzzeit: 90 Std.	Eigenstudium: 135 Std.	Sprache: Deutsch und Englisch

---

**Lehrveranstaltungen:**

Künstliche Intelligenz I (WS 2019/2020, Vorlesung, 4 SWS, Michael Kohlhase)  
Übungen zu Künstliche Intelligenz I (WS 2019/2020, Übung, 2 SWS, Katja Bercic)

---

**Inhalt:**

Dieser Kurs beschäftigt sich mit den Grundlagen der Künstlichen Intelligenz (KI), insbesondere formale Wissensrepräsentation, Heuristische Suche, Automatisches Planen und Schliessen unter Unsicherheit.

—  
This course covers the foundations of Artificial Intelligence (AI), in particular symbolic techniques based on search and inference.

**Lernziele und Kompetenzen:**

- Wissen: Die Studierenden lernen grundlegende Repräsentationsformalismen und Algorithmen der Künstlichen Intelligenz kennen.
- Anwenden: Die Konzepte werden an Beispielen aus der realen Welt angewandt (Übungsaufgaben).
- Analyse: Die Studierenden lernen die über die modellierung in der Maschine menschliche Intelligenzleistungen besser einzuschätzen.

Sozialkompetenz

- Die Studierenden arbeiten in Kleingruppen zusammen um kleine Projekte zu bewältigen

—  
**Learning Goals and Competencies**

Technical, Learning, and Method Competencies

- Knowledge: The students learn foundational representations and algorithms in AI.
- Application: The concepts learned are applied to examples from the real world (homeworks).
- Analysis: By modeling human cognitive abilities, students learn to assess and understand human intelligence better.
- Social Competences: Students work in small groups to solve an AI game-play challenge/competition (Kalah).

**Literatur:**

Die Vorlesung folgt weitgehend dem Buch

Stuart Russell und Peter Norvig: Artificial Intelligence: A Modern Approach. Prentice Hall, 3rd edition, 2009.

Deutsche Ausgabe:

Stuart Russell und Peter Norvig: Künstliche Intelligenz: Ein Moderner Ansatz. Pearson-Studium, 2004 (Übersetzung der 2. Auflage). ISBN: 978-3-8273-7089-1.

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

**[1] Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule | Vertiefungsrichtung Künstliche Intelligenz)

Dieses Modul ist daneben auch in den Studienfächern "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Bachelor of Science)", "Informatik (Master of Science)", "Mathematik (Bachelor of Science)", "Medizintechnik (Master of Science)" verwendbar.

---

**Studien-/Prüfungsleistungen:**

Künstliche Intelligenz I (Prüfungsnummer: 894856)

(englische Bezeichnung: Artificial Intelligence I)

Prüfungsleistung, Portfolio

Anteil an der Berechnung der Modulnote: 100%

weitere Erläuterungen:

Die Prüfung besteht aus einer Klausur (90 Minuten). Auf Basis der Bewertungen der abgegebenen Übungsaufgaben können bis zu 10 % Bonuspunkte erworben werden, die zu dem Ergebnis einer bestandenen Klausur hinzugerechnet werden

Prüfungssprache: Englisch

Erstablingung: WS 2019/2020, 1. Wdh.: SS 2020

1. Prüfer: Michael Kohlhase (100451)

---

**Modulbezeichnung:** **Logik-Basierte Sprachverarbeitung (LBS)** **5 ECTS**  
(Logic-based speech representation)

Modulverantwortliche/r: Michael Kohlhase

Lehrende: Michael Kohlhase, Dennis Müller

Startsemester: WS 2019/2020	Dauer: 1 Semester	Turnus: jährlich (WS)
Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Deutsch oder Englisch

**Lehrveranstaltungen:**

Logik-Basierte Sprachverarbeitung (WS 2019/2020, Vorlesung mit Übung, 4 SWS, Michael Kohlhase et al.)

**Inhalt:**

Grundlagen der logikbasierten Sprachverarbeitung, Syntax, Semantik-Konstruktion, und Semantische Verarbeitung natürlicher Sprache im Grammatical Framework (GF)

**Lernziele und Kompetenzen:**

*Fachkompetenz*

*Wissen*

Die Studierenden beherrschen moderne, sehr expressive Formalismen zur Syntaktisch/Semantischen Analyse und Bedeutungsrepräsentation natürlicher Sprache. Sie können eingeschränkt neue Formalismen entwickeln.

*Anwenden*

Die Studierenden entwickeln Grammatiken und Bedeutungsrepräsentationen im Meta-Framework GF und setzen diese durch Interpretationsabbildungen in Verbindung.

*Analysieren*

Die Studierenden analysieren die innere Struktur natürlichsprachlicher Phrasen und Sätze. Sie wählen für eine zu repräsentierenden Phänomene geeignete Formalismen aus.

*Lern- bzw. Methodenkompetenz*

Die Studierenden erarbeiten selbständig formale syntaktische und semantische Repräsentationen für Natürliche Sprache

*Sozialkompetenz*

Die Studierenden arbeiten in Kleingruppen erfolgreich zusammen.

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

**[1] Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule | Vertiefungsrichtung Künstliche Intelligenz)

Dieses Modul ist daneben auch in den Studienfächern "Informatik (Master of Science)", "Mathematik (Bachelor of Science)" verwendbar.

**Studien-/Prüfungsleistungen:**

Logik-Basierte Sprachverarbeitung (Prüfungsnummer: 675137)

(englische Bezeichnung: Logic-based speech representation)

Prüfungsleistung, Portfolio

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Logik-Basierte Sprachverarbeitung

weitere Erläuterungen:

Die Modulnote setzt sich zu 50% aus dem Ergebnis einer 20-minütigen mündlichen Prüfung am Semesterende und zu 50% aus der Bewertung der Leistungen aus dem Übungsbetrieb zusammen.

Prüfungssprache: Deutsch

Erstablingung: WS 2019/2020, 1. Wdh.: SS 2020

1. Prüfer: Michael Kohlhasse (100451)

---



---

**Modulbezeichnung:** **Ontologien im Semantic Web (OntoSWeb)** **7.5 ECTS**  
 (Ontologies for the Semantic Web)

Modulverantwortliche/r: Lutz Schröder  
 Lehrende: Lutz Schröder

---

Startsemester: WS 2019/2020	Dauer: 1 Semester	Turnus: jährlich (SS)
Präsenzzeit: 60 Std.	Eigenstudium: 165 Std.	Sprache: Deutsch

---

**Lehrveranstaltungen:**

Ontologien im Semantic Web (WS 2019/2020, Vorlesung mit Übung, 4 SWS, Lutz Schröder)

---

**Inhalt:**

- Algorithmen für Aussagenlogik
- Tableaukalküle
- Anfänge der (endlichen) Modelltheorie
- Modal- und Beschreibungslogiken
- Ontologieentwurf

**Lernziele und Kompetenzen:**

*Fachkompetenz*

*Wissen*

Die Studierenden geben Definitionen der Syntax und Semantik verschiedener Wissensrepräsentationssprachen wieder und legen wesentliche Eigenschaften hinsichtlich Entscheidbarkeit, Komplexität und Ausdrucksstärke dar.

*Anwenden*

Die Studierenden wenden Deduktionsalgorithmen auf Beispielformeln an. Sie stellen einfache Ontologien auf und führen anhand der diskutierten Techniken Beweise elementarer logischer Metaeigenschaften.

*Analysieren*

Die Studierenden klassifizieren Logiken nach grundlegenden Eigenschaften wie Ausdrucksstärke und Komplexität. Sie wählen für ein gegebenes Anwendungsproblem geeignete Formalismen aus.

*Lern- bzw. Methodenkompetenz*

Die Studierenden erarbeiten selbständig formale Beweise.

*Sozialkompetenz*

Die Studierenden arbeiten in Kleingruppen erfolgreich zusammen.

**Literatur:**

- M Krötzsch, F Simancik, I Horrocks; A description logic primer, arXiv, 2012
  - F. Baader et al. (ed.): The Description Logic Handbook, Cambridge University Press, 2003
  - M. Huth, M. Ryan: Logic in Computer Science, Cambridge University Press, 2004
  - L. Libkin: Elements of Finite Model Theory, Springer, 2004
- 

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

[1] **Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule | Vertiefungsrichtung Künstliche Intelligenz)

[2] **Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule | Vertiefungsrichtung Theoretische Informatik)

Dieses Modul ist daneben auch in den Studienfächern "Informatik (Master of Science)", "Mathematik (Bachelor of Science)" verwendbar.

---

**Studien-/Prüfungsleistungen:**

Ontologien im Semantic Web (Prüfungsnummer: 806144)

Prüfungsleistung, mündliche Prüfung, Dauer (in Minuten): 30

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Ontologien im Semantic Web

weitere Erläuterungen:

In die Note der mündliche Prüfung gehen auch die Übungsleistungen ein.

Prüfungssprache: Deutsch oder Englisch

Erstablingung: WS 2019/2020, 1. Wdh.: SS 2020 (nur für Wiederholer), 2. Wdh.: WS 2020/2021  
(nur für Wiederholer)

1. Prüfer: Lutz Schröder (100190)

---

---

**Modulbezeichnung:** **Multimedia-Technik (MMT)** **5 ECTS**  
(Multimedia Technology)

Modulverantwortliche/r: Klaus Meyer-Wegener  
Lehrende: Klaus Meyer-Wegener

---

Startsemester: SS 2019	Dauer: 1 Semester	Turnus: jährlich (SS)
Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Deutsch

---

**Lehrveranstaltungen:**

Multimedia-Technik (SS 2019, Vorlesung, 2 SWS, Klaus Meyer-Wegener)  
Übungen zu Multimedia-Technik (SS 2019, Übung, 2 SWS, Demian Vöhringer)

---

**Empfohlene Voraussetzungen:**

siehe Lehrveranstaltungsbeschreibung

---

**Inhalt:**

siehe Lehrveranstaltungsbeschreibung

**Lernziele und Kompetenzen:**

siehe Lehrveranstaltungsbeschreibung

*Fachkompetenz*

*Wissen*

Die Studierenden können:

- die Begriffe Multimedia, Medienobjekt, Multimedia-Objekt, Mediendaten definieren;
- Gefahren und Nutzen von Multimedia auflisten;
- die wichtigsten Anwendungsbereiche von Multimedia nennen;
- die wichtigsten Bestandteile und Schichten eines Multimedia-Systems wiedergeben;
- die nachrichtentechnischen Grundlagen von Multimedia wiedergeben;
- die wesentlichen Eigenschaften der Huffman- und Lempel-Ziv-Codierung auflisten;
- typische Multimedia-Hardware nennen und charakterisieren;
- die Arten der Darstellung von Texten in Rechnern unterscheiden;
- die Grundbegriffe der Typografie erklären;
- Wohlgeformtheit und Gültigkeit von XML-Dokumenten definieren;
- die wichtigsten Farbmodelle und Farbräume beschreiben;
- Zweck und Inhalt von Farbprofilen nennen;
- Dateiformate für Rasterbilder aufzählen;
- die Schritte der JPEG-2000-Codierung nennen
- das Wesen der HDR-Fotografie beschreiben;
- Anwendungsbereiche und Grundidee von Image Based Lighting wiedergeben
- die Bestandteile von SVG benennen;
- eine Abgrenzung von SVG zu VRML und X3D vornehmen;
- analoge Video-Technik definieren;
- digitale Video-Technik abgrenzen;
- die Digitalisierung von Audio wiedergeben;
- den Effekt der Audio-Maskierung darlegen;
- das Vorgehen bei der MP3-Komprimierung angeben;
- die Bestandteile von HTML5 auflisten.

*Verstehen*

Die Studierenden können:

- die Struktur von Mediendaten beschreiben;
- Grundlegende Eigenschaften von Codierverfahren und deren Zusammenhang mit der Entropie erläutern;
- Probleme beim falschen Einsatz von Farbräumen aufzeigen;
- Möglichkeiten zur Strukturierung von Text klassifizieren;
- Zweck und Aufbau von DTD und XML-Schema erläutern

- Funktionsweise der Filterung durch Faltung erläutern;
- Möglichkeiten zur Visualisierung von Datenmengen skizzieren;
- Unterschiede zwischen Raster- und Vektorgrafik erläutern;
- Funktionsweise und Eigenschaften der MPEG-Videocodierung schildern;
- Blockgrößen bei der MP3-Audiocodierung erläutern.

#### *Anwenden*

Die Studierenden können:

- Quantisierer aufstellen;
- die Entropie von Quellen berechnen;
- Daten mit der Huffman und Lempel-Ziv-Codierung codieren;
- Schriftarten für bestimmte Anwendungszwecke auswählen;
- Gammakorrekturen berechnen;
- Faltungsfilter für Bild- und Audiodateien implementieren;
- Microformats einsetzen;
- DTDs und XML-Schemas erstellen;
- XML-Dokumente mit XSLT umformen;
- einfache SVG-Grafiken von Hand erstellen;
- einfache 3D-Objekte in X3D von Hand erstellen;
- HTML5-Neuerungen einsetzen.

#### *Evaluieren (Beurteilen)*

Die Studierenden können:

- das geeignete Rasterbildformat für einen Einsatzzweck wählen;
- die Eignung einer bestimmten Abtastrate für gegebene Signale beurteilen.

#### *Lern- bzw. Methodenkompetenz*

Die Studierenden können

- sich selbst unter Einsatz von Fachliteratur über Multimedia-Technik informieren

#### **Literatur:**

siehe Lehrveranstaltungsbeschreibung

#### **Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

##### **[1] Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule)

#### **Studien-/Prüfungsleistungen:**

Multimedia-Technik (Prüfungsnummer: 345938)

Prüfungsleistung, Klausur mit MultipleChoice, Dauer (in Minuten): 60

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Multimedia-Technik
- Übungen zu Multimedia-Technik

Erstablingung: SS 2019, 1. Wdh.: WS 2019/2020 (nur für Wiederholer)

1. Prüfer: Klaus Meyer-Wegener (100153)

---

**Modulbezeichnung: Human Computer Interaction (HCI)** **5 ECTS**  
 (Human Computer Interaction)

Modulverantwortliche/r: Björn Eskofier  
 Lehrende: Björn Eskofier

---

Startsemester: SS 2019	Dauer: 1 Semester	Turnus: jährlich (SS)
Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Deutsch oder Englisch

---

**Lehrveranstaltungen:**

Human Computer Interaction (SS 2019, Vorlesung, 3 SWS, Björn Eskofier)  
 Human Computer Interaction Exercises (SS 2019, Übung, 1 SWS, Markus Wirth)

---

**Inhalt:**

Die Vorlesung vermittelt Konzepte, Prinzipien, Modelle, Methoden und Techniken für die effektive Entwicklung von benutzerfreundlichen Mensch-Computer-Schnittstellen. Das Thema moderner Benutzungsschnittstellen wird dabei für klassische Computer aber auch für mobile Geräte, eingebettete Systeme, Automobile und intelligente Umgebungen betrachtet.

Die folgenden Themen werden in der Vorlesung behandelt:

- Einführung in die Grundlagen der Mensch-Computer-Interaktion, historische Entwicklung
- Entwurfsprinzipien und Modelle für moderne Benutzungsschnittstellen und interaktive Systeme
- Informationsverarbeitung des Menschen, Wahrnehmung, Motorik, Eigenschaften und Fähigkeiten des Benutzers
- Interaktionskonzepte und -stile, Metaphern, Normen, Regeln und Style Guides
- Ein- und Ausgabegeräte, Entwurfsraum für interaktive Systeme
- Analyse-, Entwurfs- und Entwicklungsmethoden und -werkzeuge für Benutzungsschnittstellen
- Prototypische Realisierung und Implementierung von interaktiven Systemen, Werkzeuge
- Architekturen für interaktive Systeme, User Interface Toolkits und Komponenten
- Akzeptanz, Evaluationsmethoden und Qualitätssicherung

Contents: Aim of the lecture is to teach basic knowledge of concepts, principles, models, methods and techniques for developing highly user-friendly Human Computer Interfaces. Beyond traditional computer system the topic of modern user interfaces is also discussed in the context of automobile and intelligent environments, mobile devices and embedded systems.

This lecture addresses the following topics:

- Introduction to the basics of Human Computer Interaction
- Design principles and models for modern user interfaces and interactive systems
- Information processing of humans, perception, motor skills, properties and skills of the users
- Interaction concepts, metaphors, standards, norms and style guides
- In- and output devices, design space for interactive systems
- Analysis-, design- and development methodologies and tools for easy to use user interfaces
- Prototypic implementation of interactive systems
- Architectures for interactive systems, User Interface Toolkits and components
- Acceptance, evaluation methods and quality assurance

**Lernziele und Kompetenzen:**

- Studierende entwickeln ein Verständnis für Modelle, Methoden und Konzepte der Mensch-Computer-Interaktion.
- Sie lernen verschiedene Ansätze für den Entwurf, die Entwicklung und Bewertung von Benutzungsschnittstellen kennen und verstehen deren Vor- und Nachteile.
- Die Teilnahme an der Veranstaltung versetzt Studierende in die Lage, einen Entwicklungsprozess in der Mensch-Computer-Interaktion zu verstehen und umzusetzen.
- Sie werden weiterhin in die Lage versetzt, dies vor dem Hintergrund der Informationsverarbeitungsfähigkeit, Wahrnehmung und Motorik des Benutzers zu gestalten.
- Schlussendlich werden Methoden der Evaluation sowie Akzeptanz- und Qualitätssicherung erlangt.

Learning Objectives and Competences:

- Students develop an understanding for models, methods and concepts in the field of Human-Computer Interaction.
- They learn different approaches for designing, developing and evaluating User Interfaces and their advantages and disadvantages.
- Joining the course enables students to understand and execute a development process in the area of Human-Computer Interaction.
- Student will be able to do an UI evaluation by learning basics about Informationprocessing, perception and motoric skills of the user.
- Additionally, appropriate evaluation method as well as acceptance and quality assurance aspects will be learned.

---

### Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

#### [1] Informatik (Bachelor of Science)

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule)

Dieses Modul ist daneben auch in den Studienfächern "International Information Systems (IIS) (Master of Science)" verwendbar.

---

### Studien-/Prüfungsleistungen:

Human Computer Interaction (Prüfungsnummer: 645618)

(englische Bezeichnung: Human Computer Interaction)

Prüfungsleistung, Klausur, Dauer (in Minuten): 90

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Human Computer Interaction
- Human Computer Interaction Exercises

weitere Erläuterungen:

90-minütige schriftliche Klausur über den Stoff der Vorlesung und der Übungen. Bei weniger als 25 Teilnehmern erfolgt die Prüfung nach vorheriger Ankündigung mündlich (Dauer: 30 Minuten).

Prüfungssprache: Deutsch oder Englisch

Erstablingung: SS 2019, 1. Wdh.: WS 2019/2020

1. Prüfer: Björn Eskofier (100122)

---

### Organisatorisches:

Folien zur Vorlesung und Organisation über Studon.

Organisation and slides via StudOn.

---

**Modulbezeichnung:** Gestalterische Grundlagen der Medien (GdMed) 5 ECTS  
(Design Foundations of Media)

Modulverantwortliche/r: Wolfgang Keller  
Lehrende: Wolfgang Keller

---

Startsemester: WS 2019/2020	Dauer: 1 Semester	Turnus: jährlich (WS)
Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Deutsch

---

**Lehrveranstaltungen:**

Gestalterische Grundlagen der Medien (WS 2019/2020, Vorlesung mit Übung, 4 SWS, Wolfgang Keller)

---

**Inhalt:**

siehe Lehrveranstaltungsbeschreibung

**Lernziele und Kompetenzen:**

Neben gestalterischen Grundlagen und Entwicklungsstufen wird auch eine praxisnahe Perspektive vermittelt. Die Studierenden werden mit einen sinnvollen Arbeitsablauf von der Idee bis zum fertigen Produkt bekannt gemacht.

**Literatur:**

siehe Lehrveranstaltungsbeschreibung

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

**[1] Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule)

---

**Studien-/Prüfungsleistungen:**

Gestalterische Grundlagen der Medien (Prüfungsnummer: 588261)

Prüfungsleistung, Portfolio

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Gestalterische Grundlagen der Medien

weitere Erläuterungen:

Abgabe einer Hausarbeit

Erstablingung: WS 2019/2020, 1. Wdh.: keine Wdh.

1. Prüfer: Klaus Meyer-Wegener (100153)

---

---

**Modulbezeichnung:** Music Processing Analysis - Lecture & Exercise (MPA-LE) 5 ECTS  
 (Music Processing Analysis - Lecture & Exercise)

Modulverantwortliche/r: Meinard Müller  
 Lehrende: Meinard Müller

---

Startsemester: WS 2019/2020	Dauer: 1 Semester	Turnus: jährlich (WS)
Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Englisch

---

**Lehrveranstaltungen:**

Music Processing Analysis (WS 2019/2020, Vorlesung, 2 SWS, Meinard Müller)  
 Music Processing Analysis - Exercise (WS 2019/2020, Übung, 2 SWS, Meinard Müller)

---

**Empfohlene Voraussetzungen:**

In this course, we discuss a number of current research problems in music processing or music information retrieval (MIR) covering aspects from information science and digital signal processing. We provide the necessary background information and give numerous motivating examples so that no specialized knowledge is required. However, the students should have a solid mathematical background. The lecture is accompanied by readings from textbooks or the research literature. Furthermore, the students are required to experiment with the presented algorithms using MATLAB.

---

**Inhalt:**

Music signals possess specific acoustic and structural characteristics that are not shared by spoken language or audio signals from other domains. In fact, many music analysis tasks only become feasible by exploiting suitable music-specific assumptions. In this course, we study feature design principles that have been applied to music signals to account for the music-specific aspects. In particular, we discuss various musically expressive feature representations that refer to musical dimensions such as harmony, rhythm, timbre, or melody. Furthermore, we highlight the practical and musical relevance of these feature representations in the context of current music analysis and retrieval tasks. Here, our general goal is to show how the development of music-specific signal processing techniques is of fundamental importance for tackling otherwise infeasible music analysis problems.

**Lernziele und Kompetenzen:**

*Fachkompetenz*  
*Verstehen*

- Die Studierenden stellen zentrale Aufgabenstellungen der Musikverarbeitung in eigenen Worten dar und skizzieren Lösungsansätze.
- Die Studierenden verstehen die Eigenschaften von unterschiedlichen Darstellungsformen von Musik.
- Die Studierenden interpretieren Signaleigenschaften anhand von Visualisierungen (Exercise).

*Anwenden*

- Die Studierenden wenden grundlegende Algorithmen zur Analyse und zum Vergleich von Musiksignalen an.
- Die Studierenden können voraussagen, wie sich unterschiedliche musikalische Eigenschaften bei der Signalanalyse auswirken.
- Die Studierenden implementieren Algorithmen zur Analyse, zum Vergleich und zur inhaltsbasierten Suche von Musiksignalen (Exercise).

*Analysieren*

- Die Studierenden beobachten und diskutieren die Bedeutung und Auswirkung von Parametern bei der Musikanalyse.
- Die Studierenden stellen unterschiedliche Verfahren bei der Analyse von Periodizitäten gegenüber.
- Die Studierenden analysieren und erforschen Eigenschaften von Musiksignalen mittels automatisierter Methoden (Exercise).
- Die Studierenden klassifizieren und strukturieren Musikdaten mittels Lernverfahren (Exercise).



### *Evaluieren (Beurteilen)*

- Die Studierenden hinterfragen Annahmen, die implizit bei der Verwendung von Analysemethoden gemacht werden.
- Die Studierenden schätzen ein, wann Methoden bei der Analyse von gewissen Musiksignalen funktionieren könnten und wann sie typischerweise versagen.
- Die Studierenden evaluieren automatisierte Methoden mittels geeigneter Evaluationsmaße unter Verwendung von manuell erstellten Annotationen (Exercise).

### *Erschaffen*

- Die Studierenden entwerfen neue Algorithmen zur Musikanalyse durch Modifikation und Integration bestehender Software (Exercise).

### *Lern- bzw. Methodenkompetenz*

- Die Studierenden bereiten sich auf die Vorlesung anhand ausgewählter Literatur vor.
- Die Studierenden hinterfragen bestehende Ansätze hinsichtlich ihrer Anwendbarkeit in der Praxis.
- Die Studierenden beachten Fragen der Effizienz bei den diskutierten Algorithmen.
- Die Studierenden entwickeln praktische Lösungswege für Problem in der Musikverarbeitung (Exercise)

### *Selbstkompetenz*

- Die Studierenden hinterfragen ihr Verständnis von dem Gelernten anhand von Übungsaufgaben.
- Die Studierenden formulieren Fragen und stellen diese in der Vorlesung an den Dozenten und die Zuhörerschaft.
- Die Studierenden nutzen Verbesserungshinweise des Betreuers und der Tutoren zur Verbesserung ihrer Lernstrategien (Exercise).

### *Sozialkompetenz*

- Die Studierenden organisieren selbständig Lerngruppen, in denen der Stoff diskutiert und vertieft wird.
- Die Studierenden simulieren mit ihren Kommilitonen mündliche Prüfungen.
- Die Studierenden entwickeln und implementieren Software im Team (Exercise).
- Die Studierenden geben Kommilitonen im Rahmen ihrer Zusammenarbeit wertschätzendes Feedback (Exercise).

## **Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

### [1] **Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule | Vertiefungsrichtung Medieninformatik)

### [2] **Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule | Vertiefungsrichtung Mustererkennung)

Dieses Modul ist daneben auch in den Studienfächern "Informatik (Master of Science)", "Mathematik (Bachelor of Science)" verwendbar.

## **Studien-/Prüfungsleistungen:**

Music Processing Analysis - Lecture & Exercise (Prüfungsnummer: 639119)

(englische Bezeichnung: Music Processing Analysis - Lecture & Exercise)

Prüfungsleistung, mündliche Prüfung, Dauer (in Minuten): 30

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Music Processing Analysis
- Music Processing Analysis - Exercise

weitere Erläuterungen:

Eine erfolgreiche Teilnahme an den Übungen ist Voraussetzung für die Zulassung zur Prüfung.

Erstablingung: WS 2019/2020, 1. Wdh.: SS 2020  
1. Prüfer: Meinard Müller (100297)

---

---

**Modulbezeichnung:** **Interventional Medical Image Processing (Online-Kurs) (IMIP)** **5 ECTS**  
 (Interventional Medical Image Processing (online course))

Modulverantwortliche/r: Andreas Maier  
 Lehrende: Andreas Maier

---

Startsemester: SS 2019	Dauer: 1 Semester	Turnus: halbjährlich (WS+SS)
Präsenzzeit: k.A. Std.	Eigenstudium: 150 Std.	Sprache: Englisch

---

**Lehrveranstaltungen:**

Medical Image Processing for Interventional Applications (VHB-Kurs) (SS 2019, Vorlesung, 4 SWS, Julian Hoßbach et al.)

---

**Vorhergehende Module:**

Diagnostic Medical Image Processing (VHB-Kurs)

---

**Inhalt:**

English Version:

This lecture focuses on recent developments in image processing driven by medical applications. All algorithms are motivated by practical problems. The mathematical tools required to solve the considered image processing tasks will be introduced.

The lecture starts with an overview on preprocessing algorithms such as scatter correction for x-ray images, edge detection, super-resolution and edge-preserving noise reduction. The second chapter describes automatic image analysis using feature descriptors, key point detection, and segmentation using bottom-up algorithms such as the random walker or top-down approaches such as active shape models. Furthermore, the lecture covers geometric calibration algorithms for single view calibration, epipolar geometry, and factorization. The last part of the lecture covers non-rigid registration based on variational methods and motion-compensated image reconstruction.

Deutsche Version:

Die Vorlesung ist auf die jüngsten Entwicklungen in der Verarbeitung von medizinischen Bildern ausgerichtet. Alle Algorithmen werden durch praktische Probleme motiviert. Die mathematischen Werkzeuge, die für die Bildverarbeitungsaufgaben benötigt werden, werden eingeführt.

Die Vorlesung beginnt mit einem Überblick über Vorverarbeitungsalgorithmen, wie zum Beispiel Streustrahlkorrektur für Röntgenbilder, Kantenerkennung, Superresolution und kantenerhaltende Rauschunterdrückung. Das zweite Kapitel beschreibt die automatische Bildanalyse mit Merkmalsdeskriptoren, Punkterkennung und Segmentierung mit Bottom-up-Algorithmen wie dem Random-Walker oder Top-Down-Ansätzen wie aktiven Formmodellen. Darüber hinaus deckt die Vorlesung auch geometrische Kalibrierungsalgorithmen zur Einzelansicht-Kalibrierung, Epipolargeometrie und Faktorisierung ab. Der letzte Teil der Vorlesung deckt nicht-starre Registrierung auf der Grundlage von Variationsmethoden und bewegungskompensierter Bildrekonstruktion ab.

**Lernziele und Kompetenzen:**

English Version:

The participants

- summarize the contents of the lecture.
- apply pre-processing algorithms such as scatter correction and edge-preserving filtering.
- extract information from images automatically by image analysis methods such as key point detectors and segmentation algorithms.
- calibrate projection geometries for single images and image sequences using the described methods.
- develop non-rigid registration methods using variational calculus and different regularizers.
- adopt algorithms to new domains by appropriate modifications.

Deutsche Version:

Die Teilnehmer

- fassen die Inhalte der Vorlesung zusammen.
- wenden Vorverarbeitungsalgorithmen wie Streustrahlkorrektur und kantenerhaltende Filterung an.

- extrahieren automatisch Informationen aus Bildern, indem sie Bildanalyseverfahren wie Punktdetektoren und Segmentierungsalgorithmen verwenden.
- kalibrieren Projektionsgeometrien für Einzelbilder und Bildsequenzen mit den beschriebenen Methoden.
- entwickeln nicht-starre Registrierungsmethoden mit Hilfe von Variationsrechnung und unterschiedlichen Regularisierungen.
- wenden Algorithmen auf neue Modalitäten durch entsprechende Änderungen im Algorithmus an.

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

**[1] Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule)

Dieses Modul ist daneben auch in den Studienfächern "Advanced Optical Technologies (Master of Science)", "Communications and Multimedia Engineering (Master of Science)", "Computational Engineering (Master of Science with Honours)", "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Master of Science)", "Informatik (Master of Science)", "Information and Communication Technology (Master of Science)", "Informations- und Kommunikationstechnik (Master of Science)", "Mathematik (Bachelor of Science)", "Medizintechnik (Bachelor of Science)", "Medizintechnik (Master of Science)" verwendbar.

---

**Studien-/Prüfungsleistungen:**

Interventional Medical Image Processing (VHB-Kurs) (Prüfungsnummer: 41401)

Prüfungsleistung, Klausur, Dauer (in Minuten): 60

Anteil an der Berechnung der Modulnote: 100% Prüfungssprache: Englisch

Erstablingung: SS 2019, 1. Wdh.: WS 2019/2020

1. Prüfer: Andreas Maier (100196)

---

---

**Modulbezeichnung: Human Computer Interaction (HCI)** **5 ECTS**  
 (Human Computer Interaction)

Modulverantwortliche/r: Björn Eskofier  
 Lehrende: Björn Eskofier

---

Startsemester: SS 2019	Dauer: 1 Semester	Turnus: jährlich (SS)
Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Deutsch oder Englisch

---

**Lehrveranstaltungen:**

Human Computer Interaction (SS 2019, Vorlesung, 3 SWS, Björn Eskofier)  
 Human Computer Interaction Exercises (SS 2019, Übung, 1 SWS, Markus Wirth)

---

**Inhalt:**

Die Vorlesung vermittelt Konzepte, Prinzipien, Modelle, Methoden und Techniken für die effektive Entwicklung von benutzerfreundlichen Mensch-Computer-Schnittstellen. Das Thema moderner Benutzungsschnittstellen wird dabei für klassische Computer aber auch für mobile Geräte, eingebettete Systeme, Automobile und intelligente Umgebungen betrachtet.

Die folgenden Themen werden in der Vorlesung behandelt:

- Einführung in die Grundlagen der Mensch-Computer-Interaktion, historische Entwicklung
- Entwurfsprinzipien und Modelle für moderne Benutzungsschnittstellen und interaktive Systeme
- Informationsverarbeitung des Menschen, Wahrnehmung, Motorik, Eigenschaften und Fähigkeiten des Benutzers
- Interaktionskonzepte und -stile, Metaphern, Normen, Regeln und Style Guides
- Ein- und Ausgabegeräte, Entwurfsraum für interaktive Systeme
- Analyse-, Entwurfs- und Entwicklungsmethoden und -werkzeuge für Benutzungsschnittstellen
- Prototypische Realisierung und Implementierung von interaktiven Systemen, Werkzeuge
- Architekturen für interaktive Systeme, User Interface Toolkits und Komponenten
- Akzeptanz, Evaluationsmethoden und Qualitätssicherung

Contents: Aim of the lecture is to teach basic knowledge of concepts, principles, models, methods and techniques for developing highly user-friendly Human Computer Interfaces. Beyond traditional computer system the topic of modern user interfaces is also discussed in the context of automobile and intelligent environments, mobile devices and embedded systems.

This lecture addresses the following topics:

- Introduction to the basics of Human Computer Interaction
- Design principles and models for modern user interfaces and interactive systems
- Information processing of humans, perception, motor skills, properties and skills of the users
- Interaction concepts, metaphors, standards, norms and style guides
- In- and output devices, design space for interactive systems
- Analysis-, design- and development methodologies and tools for easy to use user interfaces
- Prototypic implementation of interactive systems
- Architectures for interactive systems, User Interface Toolkits and components
- Acceptance, evaluation methods and quality assurance

**Lernziele und Kompetenzen:**

- Studierende entwickeln ein Verständnis für Modelle, Methoden und Konzepte der Mensch-Computer-Interaktion.
- Sie lernen verschiedene Ansätze für den Entwurf, die Entwicklung und Bewertung von Benutzungsschnittstellen kennen und verstehen deren Vor- und Nachteile.
- Die Teilnahme an der Veranstaltung versetzt Studierende in die Lage, einen Entwicklungsprozess in der Mensch-Computer-Interaktion zu verstehen und umzusetzen.
- Sie werden weiterhin in die Lage versetzt, dies vor dem Hintergrund der Informationsverarbeitungsfähigkeit, Wahrnehmung und Motorik des Benutzers zu gestalten.
- Schlussendlich werden Methoden der Evaluation sowie Akzeptanz- und Qualitätssicherung erlangt.

Learning Objectives and Competences:

- Students develop an understanding for models, methods and concepts in the field of Human-Computer Interaction.
- They learn different approaches for designing, developing and evaluating User Interfaces and their advantages and disadvantages.
- Joining the course enables students to understand and execute a development process in the area of Human-Computer Interaction.
- Student will be able to do an UI evaluation by learning basics about Informationprocessing, perception and motoric skills of the user.
- Additionally, appropriate evaluation method as well as acceptance and quality assurance aspects will be learned.

---

### Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

#### [1] Informatik (Bachelor of Science)

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule)

Dieses Modul ist daneben auch in den Studienfächern "International Information Systems (IIS) (Master of Science)" verwendbar.

---

### Studien-/Prüfungsleistungen:

Human Computer Interaction (Prüfungsnummer: 645618)

(englische Bezeichnung: Human Computer Interaction)

Prüfungsleistung, Klausur, Dauer (in Minuten): 90

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Human Computer Interaction
- Human Computer Interaction Exercises

weitere Erläuterungen:

90-minütige schriftliche Klausur über den Stoff der Vorlesung und der Übungen. Bei weniger als 25 Teilnehmern erfolgt die Prüfung nach vorheriger Ankündigung mündlich (Dauer: 30 Minuten).

Prüfungssprache: Deutsch oder Englisch

Erstablingung: SS 2019, 1. Wdh.: WS 2019/2020

1. Prüfer: Björn Eskofier (100122)

---

### Organisatorisches:

Folien zur Vorlesung und Organisation über Studon.

Organisation and slides via StudOn.

---

**Modulbezeichnung:** Diagnostic Medical Image Processing (VHB-Kurs) (DMIP-VHB)  
 (Diagnostic Medical Image Processing (VHB course)) 5 ECTS

Modulverantwortliche/r: Andreas Maier  
 Lehrende: Andreas Maier

---

Startsemester: SS 2019	Dauer: 1 Semester	Turnus: halbjährlich (WS+SS)
Präsenzzeit: k.A. Std.	Eigenstudium: 150 Std.	Sprache: Englisch

---

**Lehrveranstaltungen:**

Medical Image Processing for Diagnostic Applications (VHB-Kurs) (SS 2019, Vorlesung, 4 SWS, Julian Hoßbach et al.)

---

**Empfohlene Voraussetzungen:**

Ingenieurmathematik

---

**Inhalt:**

English version:

The contents of the lecture comprise basics about medical imaging modalities and acquisition hardware. Furthermore, details on acquisition-dependent preprocessing are covered for image intensifiers, flat-panel detectors, and MR. The fundamentals of 3D reconstruction from parallel-beam to cone-beam reconstruction are also covered. In the last chapter, rigid registration for image fusion is explained. In the exercises, algorithms that were presented in the lecture are implemented in Java.

Deutsche Version:

Die Inhalte der Vorlesung umfassen Grundlagen der medizinischen Bildverarbeitung und Aufnahmeprinzipien. Darüber hinaus werden Details der Vorverarbeitung für Bildverstärker, Flachpaneldetektoren und MR erklärt. Die Grundlagen der Rekonstruktion von Parallelstrahl bis hin zur Kegelstrahl-Tomographie werden ebenfalls behandelt. Im letzten Kapitel wird starre Registrierung für Bildfusion erläutert. In den Übungen werden Algorithmen aus der Vorlesung in Java implementiert.

**Lernziele und Kompetenzen:**

English Version: The participants

- understand the challenges in interdisciplinary work between engineers and medical practitioners.
- develop understanding of algorithms and math for diagnostic medical image processing.
- learn that creative adaptation of known algorithms to new problems is key for their future career.
- develop the ability to adapt algorithms to different problems.
- are able to explain algorithms and concepts of the lecture to other engineers.

Deutsche Version: Die Teilnehmer

- verstehen die Herausforderungen in der interdisziplinären Arbeit zwischen Ingenieuren und Ärzten.
- entwickeln Verständnis für Algorithmen und Mathematik der diagnostischen medizinischen Bildverarbeitung.
- erfahren, dass kreative Adaption von bekannten Algorithmen auf neue Probleme der Schlüssel für ihre berufliche Zukunft ist.
- entwickeln die Fähigkeit Algorithmen auf verschiedene Probleme anzupassen.
- sind in der Lage, Algorithmen und Konzepte der Vorlesung anderen Studenten der Technischen Fakultät zu erklären.

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

[1] **Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule)

Dieses Modul ist daneben auch in den Studienfächern "Advanced Optical Technologies (Master of Science)", "Communications and Multimedia Engineering (Master of Science)", "Computational Engineering (Master of Science with Honours)", "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Bachelor of Science)"

Science)", "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Master of Science)", "Informatik (Master of Science)", "Information and Communication Technology (Master of Science)", "Informations- und Kommunikationstechnik (Master of Science)", "Mathematik (Bachelor of Science)", "Medizintechnik (Bachelor of Science)", "Medizintechnik (Master of Science)" verwendbar.

---

**Studien-/Prüfungsleistungen:**

Diagnostic Medical Image Processing (Prüfungsnummer: 41501)

(englische Bezeichnung: Diagnostic Medical Image Processing)

Prüfungsleistung, Klausur, Dauer (in Minuten): 60

Anteil an der Berechnung der Modulnote: 100% Prüfungssprache: Englisch

Erstablingung: SS 2019, 1. Wdh.: WS 2019/2020

1. Prüfer: Andreas Maier (100196)

---



---

**Modulbezeichnung:** IT-Modernisierung (IT-Modern)**5 ECTS**  
(IT Modernization)

Modulverantwortliche/r: Peter Wilke  
Lehrende: Peter Wilke

---

Startsemester: SS 2019	Dauer: 1 Semester	Turnus: jährlich (SS)
Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Deutsch

---

**Lehrveranstaltungen:**

IT-Modernisierung (SS 2019, Vorlesung, 4 SWS, Peter Wilke)

---

**Inhalt:**

IT-Modernisierung beschäftigt sich mit dem Ersatz alter Software- und/oder Hardware. Software im kommerziellen Bereich hat eine typische Lebensdauer von über 25 Jahren, damit ist klar, dass diese keine der momentan oder zukünftig zur Verfügung stehenden Möglichkeiten nutzt oder nutzen kann, denn "damals" waren Single-CPU's der Standard und Vernetzung unbekannt.

Durch das hohe Investitionsvolumen ist eine Neu-Programmierung praktisch immer wirtschaftlich nicht sinnvoll und technisch oft unmöglich, da gar nicht genügend Programmierer zur Verfügung stehen. Die Software hat aber einen hohen Reifegrad erreicht, so dass sich die Frage stellt, ob man diese nicht automatisiert auf neue Technologien umstellen kann.

Dieses Modul beleuchtet nun exemplarisch, auf welchen Feldern Bedarf besteht, wie der Stand der Technik ist, und welche zukünftigen Fragestellungen sich abzeichnen.

Die Studierenden werden durch Übungsaufgaben mit den "alten" Programmiersprachen wie Cobol, Assembler, Fortran uä. vertraut gemacht, und bearbeiten selbstständig kleine Migrations-Aufgaben.

Als Dozenten werden erfahrene Spezialisten aus der Industrie über ihre Fragestellungen und Ansätze berichten.

Momentane Planung (Stand 2015Apr08, Themen nicht zwingend in dieser Reihenfolge):

- Einleitung
- Überblick
- Aufbau (Architektur) eines Rechenzentrums
- DB2 unter z/OS
- Exkursion DATEV Rechenzentrum
- RZ Konsolidierung
- Server Konsolidierung
- Cobol Grundlagen, RD/z, TSO/ISPF, JCL
- System z Hardware Grundlagen
- Java am Host
- Mainframe Programmierung
- Legacy-Anwendungen in einer Cloud-Architektur, CICS Modernisierung
- Internationalisierung: Unicode im Rechenzentrum
- Praxisbericht IT-Betrieb (Aufgabenstellung)
- Infrastrukturen-Modernisierung
- Praxisbericht IT-Betrieb

**Lernziele und Kompetenzen:**

*Fachkompetenz*

*Analysieren*

Lernende können ein Problem in einzelne Teile zerlegen und so die Struktur des Problems verstehen; sie können Widersprüche aufdecken, Zusammenhänge erkennen und Folgerungen ableiten und zwischen Fakten und Interpretationen unterscheiden.

*Evaluiieren (Beurteilen)*

Lernende können evidenzbasierte, qualitative und quantitative Urteile zu Sachverhalten anhand von Kriterien anstellen.

*Lern- bzw. Methodenkompetenz*

Fähigkeit und Bereitschaft zur Anwendung bestimmter Lern- und Arbeitsmethoden, die zur Entwicklung der anderen Kompetenzen, insbesondere der Fachkompetenz nötig sind.

**Selbstkompetenz**

Fähigkeit und Bereitschaft, sich weiterzuentwickeln und das eigene Leben eigenständig und verantwortlich im jeweiligen sozialen, kulturellen bzw. beruflichen Kontext zu gestalten.

**Sozialkompetenz**

Fähigkeit und Bereitschaft, zielorientiert mit anderen zusammenzuarbeiten, ihre Interessen und sozialen Situationen zu erfassen, sich mit ihnen rational und verantwortungsbewusst auseinanderzusetzen und zu verständigen sowie die Arbeits- und Lebenswelt mitzugestalten.

**Literatur:**

wird gerade zusammengestellt ...

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

**[1] Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule)

Dieses Modul ist daneben auch in den Studienfächern "International Information Systems (IIS) (Master of Science)" verwendbar.

---

**Studien-/Prüfungsleistungen:**

IT-Modernisierung (Prüfungsnummer: 716516)

(englische Bezeichnung: IT Modernization)

Prüfungsleistung, mündliche Prüfung, Dauer (in Minuten): 30 Minuten

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- IT-Modernisierung

Prüfungssprache: Deutsch oder Englisch

Erstablingung: SS 2019, 1. Wdh.: WS 2019/2020

1. Prüfer: Peter Wilke (100155)

---

**Organisatorisches:**

Bei Interessen bitte Anmeldung per E-Mail an

<mailto:Peter.Wilke@FAU.DE>

Sie werden dann über alles Aktuelle auf dem Laufenden gehalten.

Momentan planen wir die Veranstaltung Mo 10:00-14:00 Uhr. Dies gibt uns die Möglichkeit ca. 1 Stunde Mittagspause zu machen, die konkrete Zeit und Dauer der Pause richtet sich nach dem Stoff.

In der Mittagspause besteht die Gelegenheit, mit den Dozenten zu plaudern...

Einige der Themen werden als Fallstudien bearbeitet. D.h. der Dozent stellt eine IT-Modernisierungssituation dar, die Hörer müssen dann eine Lösung erarbeiten, entweder direkt in der Vorlesung oder als Hausaufgabe. Die Gruppen tragen dann ihre Lösung vor und dann wird mit der tatsächlich umgesetzten Lösung verglichen.

Prüfungsleistung: mdl. Prüfung von ca. 30min Dauer

Anrechenbar für folgende Säulen:

- anwendungsorientierte Säule
- softwareorientierte Säule

---

**Modulbezeichnung:** **Biomedizinische Signalanalyse (BioSig)** **5 ECTS**  
 (Biomedical Signal Analysis)

Modulverantwortliche/r: Björn Eskofier  
 Lehrende: Felix Kluge, Björn Eskofier

---

Startsemester: WS 2019/2020	Dauer: 1 Semester	Turnus: jährlich (WS)
Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Englisch

---

**Lehrveranstaltungen:**

Biomedizinische Signalanalyse (WS 2019/2020, Vorlesung, 2 SWS, Felix Kluge et al.)  
 Biomedizinische Signalanalyse Übung (WS 2019/2020, Übung, 2 SWS, N.N.)

---

**Inhalt:**

Im Rahmen der Vorlesung werden (a) die Grundlagen der Generation von wichtigen Biosignalen im menschlichen Körper, (b) die Messung von Biosignalen und (c) Methoden zur Analyse von Biosignalen erläutert und dargestellt.

Behandelte Biosignale sind unter anderem Aktionspotential (AP), Elektrokardiogramm (EKG), Elektromyogramm (EMG), Elektroenzephalogramm (EEG), oder Mechanomyogramm (MMG). Bei der Messung liegt der Fokus beispielsweise auf der Messtechnik oder der korrekten Sensor- bzw. Elektrodenanbringung. Im größten Teil der Vorlesung, Analyse von Biosignalen, werden Konzepte zur Filterung für die Artefaktreduktion, der Wavelet Analyse, der Ereigniserkennung und der Wellenformanalyse behandelt. Zum Schluss wird einen Einblick in überwachte Verfahren der Mustererkennung gegeben.

The lecture content explains and outlines (a) basics for the generation of important biosignals of the human body, (b) measurement of biosignals, and (c) methods for biosignals analysis.

Considered biosignals are among others action potential (AP), electrocardiogram (ECG), electromyogram (EMG), electroencephalogram (EEG), or mechanomyogram (MMG). The focus during the measurement part is for example the measurement technology or the correct sensor and electrode placement. The main part of the lecture is the analysis part. In this part, concepts like filtering for artifact reduction, wavelet analysis, event detection or waveform analysis are covered. In the end, an insight into pattern recognition methods is gained.

**Lernziele und Kompetenzen:**

**Die Studierenden können nach erfolgreichem Abschluss des Kurses**

1. *Wissen*
  - die Entstehung, Messung und Charakteristika der wichtigsten Biosignale des menschlichen Körpers wiedergeben
2. *Verstehen*
  - die wesentlichen Ursachen von Artefakten in Biosignalen erklären
  - Zusammenhänge zwischen der Entstehung der Biosignale des menschlichen Körper und dem gemessenen Signal erklären
  - Messmethoden der wichtigsten Biosignale erklären
  - Filteroperationen zur Eliminierung von Artefakten erläutern
  - bekannte Algorithmen der Verarbeitung bestimmter Biosignal erklären (z.B. Pan Tompkins für EKG)
  - typische Komponenten und ihre Bedeutung in einer generischen Signalanalyse Kette erläutern
  - die Struktur und Funktionsweise von Systemen zur maschinellen Klassifikation einfacher Muster darstellen
3. *Anwenden*
  - Signalcharakteristiken im Zeit- und Frequenzbereich bestimmen
  - Algorithmen der Biosignalverarbeitung anwenden und in Python implementieren
  - Filteroperationen zur Eliminierung von Artefakten anwenden und in Python implementieren
  - Methoden selbstständig auf interdisziplinäre Fragestellungen der Medizin und der Ingenieurwissenschaften anwenden
  - das Ergebnis von typischen Filteroperationen abschätzen
4. *Analysieren*
  - Filtercharakteristika von Schaltkreisen ableiten

- Algorithmen der Biosignalverarbeitung vergleichen
- Klassifikationsprobleme in Python lösen
- Typische Artefakte in Biosignalen erkennen und Lösungsstrategien vorschlagen

#### 5. *Evaluieren*

- Biosignale mit medizinischen Normalwerten vergleichen und im medizinischen Kontext evaluieren
- Klassifikationsergebnisse beurteilen
- die Bedeutung der Biosignalverarbeitung für die Medizintechnik diskutieren
- Probleme in Gruppen kooperativ und verantwortlich lösen und in der Übungsgruppe bzw. im Forum diskutieren

### After completion of the course students are able to

#### 1. *Remembering*

- reproduce the generation, measurement, and characteristics of important biosignals of the human body

#### 2. *Understanding*

- explain the causes of artifacts in biosignals
- explain relations between the generation of biosignals and the measured signal
- explain methods for the measurement of important biosignals
- explain filter operations for the reduction of artifacts
- explain algorithms for the analysis of important biosignals (e.g. Pan Tompkins for EKG)
- explain typical components and their importance in the signal analysis chain
- explain structure and functioning of systems for machine learning and pattern recognition

#### 3. *Applying*

- determine signal characteristics in the time and frequency domain
- apply and implement algorithms for signal analysis in Python
- implement filter operations for the reduction of artifacts in Python
- estimate the result of filter operations
- apply methods to interdisciplinary problems in medicine and medical engineering

#### 4. *Analyzing*

- derive filter characteristics from electric circuits
- compare signal analysis algorithms
- solve classification problems in Python
- recognize typical artifacts in biosignals and propose solutions for their reduction

#### 5. *Evaluating*

- compare biosignals with medical norm values and evaluate them in a medical context
- evaluate classification results
- discuss the importance of biomedical signal analysis for medical engineering
- solve and discuss problems in groups cooperatively in the group exercises and the online forum

### Literatur:

- R.M. Rangayyan, Biomedical Signal Analysis: A case-study approach. 1st ed., 2002, New York, NY: John Wiley & Sons.
- E.N. Bruce, Biomedical Signal Processing and Signal Modeling. 1st ed., 2001, New York, NY: John Wiley & Sons.

---

### Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

#### [1] Informatik (Bachelor of Science)

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule | Vertiefungsrichtung Mustererkennung)

#### [2] Informatik (Bachelor of Science)

(Po-Vers. 2009s | Nebenfach | Nebenfach Medizin (Medizinische Informatik))

Dieses Modul ist daneben auch in den Studienfächern "Communications and Multimedia Engineering (Master of Science)", "Elektrotechnik, Elektronik und Informationstechnik (Bachelor of Science)", "Elektrotechnik, Elektronik und Informationstechnik (Master of Science)", "Informatik (Master of Science)", "Mathematik (Bachelor

of Science)", "Medizintechnik (Bachelor of Science)", "Medizintechnik (Master of Science)" verwendbar.

---

**Studien-/Prüfungsleistungen:**

Biomedizinische Signalanalyse (Prüfungsnummer: 30701)

(englische Bezeichnung: Biomedical Signal Analysis)

Prüfungsleistung, Klausur, Dauer (in Minuten): 90

Anteil an der Berechnung der Modulnote: 100%

weitere Erläuterungen:

90-minütige schriftliche Klausur über den Stoff der Vorlesung und der Übungen

Erstablingung: WS 2019/2020, 1. Wdh.: SS 2020

1. Prüfer: Björn Eskofier (100122)

---

---

**Modulbezeichnung:** Diagnostic Medical Image Processing (VHB-Kurs) (DMIP-VHB)  
(Diagnostic Medical Image Processing (VHB course)) 5 ECTS

Modulverantwortliche/r: Andreas Maier  
Lehrende: Andreas Maier

---

Startsemester: WS 2019/2020	Dauer: 1 Semester	Turnus: halbjährlich (WS+SS)
Präsenzzeit: k.A. Std.	Eigenstudium: 150 Std.	Sprache: Englisch

---

**Lehrveranstaltungen:**

Medical Image Processing for Diagnostic Applications (VHB-Kurs) (WS 2019/2020, Vorlesung, 4 SWS, Julian Hoßbach et al.)

---

**Empfohlene Voraussetzungen:**

Ingenieurmathematik

---

**Inhalt:**

English version:

The contents of the lecture comprise basics about medical imaging modalities and acquisition hardware. Furthermore, details on acquisition-dependent preprocessing are covered for image intensifiers, flat-panel detectors, and MR. The fundamentals of 3D reconstruction from parallel-beam to cone-beam reconstruction are also covered. In the last chapter, rigid registration for image fusion is explained. In the exercises, algorithms that were presented in the lecture are implemented in Java.

Deutsche Version:

Die Inhalte der Vorlesung umfassen Grundlagen der medizinischen Bildverarbeitung und Aufnahmeprinzipien. Darüber hinaus werden Details der Vorverarbeitung für Bildverstärker, Flachpaneldetektoren und MR erklärt. Die Grundlagen der Rekonstruktion von Parallelstrahl bis hin zur Kegelstrahl-Tomographie werden ebenfalls behandelt. Im letzten Kapitel wird starre Registrierung für Bildfusion erläutert. In den Übungen werden Algorithmen aus der Vorlesung in Java implementiert.

**Lernziele und Kompetenzen:**

English Version: The participants

- understand the challenges in interdisciplinary work between engineers and medical practitioners.
- develop understanding of algorithms and math for diagnostic medical image processing.
- learn that creative adaptation of known algorithms to new problems is key for their future career.
- develop the ability to adapt algorithms to different problems.
- are able to explain algorithms and concepts of the lecture to other engineers.

Deutsche Version: Die Teilnehmer

- verstehen die Herausforderungen in der interdisziplinären Arbeit zwischen Ingenieuren und Ärzten.
- entwickeln Verständnis für Algorithmen und Mathematik der diagnostischen medizinischen Bildverarbeitung.
- erfahren, dass kreative Adaption von bekannten Algorithmen auf neue Probleme der Schlüssel für ihre berufliche Zukunft ist.
- entwickeln die Fähigkeit Algorithmen auf verschiedene Probleme anzupassen.
- sind in der Lage, Algorithmen und Konzepte der Vorlesung anderen Studenten der Technischen Fakultät zu erklären.

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

[1] **Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule | Vertiefungsrichtung Mustererkennung)

Dieses Modul ist daneben auch in den Studienfächern "Advanced Optical Technologies (Master of Science)", "Communications and Multimedia Engineering (Master of Science)", "Computational Engineering (Master

of Science with Honours)", "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Bachelor of Science)", "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Master of Science)", "Informatik (Master of Science)", "Information and Communication Technology (Master of Science)", "Informations- und Kommunikationstechnik (Master of Science)", "Mathematik (Bachelor of Science)", "Medizintechnik (Bachelor of Science)", "Medizintechnik (Master of Science)" verwendbar.

---

**Studien-/Prüfungsleistungen:**

Diagnostic Medical Image Processing (Prüfungsnummer: 41501)

(englische Bezeichnung: Diagnostic Medical Image Processing)

Prüfungsleistung, Klausur, Dauer (in Minuten): 60

Anteil an der Berechnung der Modulnote: 100% Prüfungssprache: Englisch

Erstablingung: WS 2019/2020, 1. Wdh.: SS 2020

1. Prüfer: Andreas Maier (100196)

---

---

**Modulbezeichnung:** **Interventional Medical Image Processing (Online-Kurs) (IMIP)** **5 ECTS**  
 (Interventional Medical Image Processing (online course))

Modulverantwortliche/r: Andreas Maier  
 Lehrende: Andreas Maier

---

Startsemester: WS 2019/2020	Dauer: 1 Semester	Turnus: halbjährlich (WS+SS)
Präsenzzeit: k.A. Std.	Eigenstudium: 150 Std.	Sprache: Englisch

---

**Lehrveranstaltungen:**

Medical Image Processing for Interventional Applications (VHB-Kurs) (WS 2019/2020, Vorlesung, 4 SWS, Julian Hoßbach et al.)

---

**Vorhergehende Module:**

Diagnostic Medical Image Processing (VHB-Kurs)

---

**Inhalt:**

English Version:

This lecture focuses on recent developments in image processing driven by medical applications. All algorithms are motivated by practical problems. The mathematical tools required to solve the considered image processing tasks will be introduced.

The lecture starts with an overview on preprocessing algorithms such as scatter correction for x-ray images, edge detection, super-resolution and edge-preserving noise reduction. The second chapter describes automatic image analysis using feature descriptors, key point detection, and segmentation using bottom-up algorithms such as the random walker or top-down approaches such as active shape models. Furthermore, the lecture covers geometric calibration algorithms for single view calibration, epipolar geometry, and factorization. The last part of the lecture covers non-rigid registration based on variational methods and motion-compensated image reconstruction.

Deutsche Version:

Die Vorlesung ist auf die jüngsten Entwicklungen in der Verarbeitung von medizinischen Bildern ausgerichtet. Alle Algorithmen werden durch praktische Probleme motiviert. Die mathematischen Werkzeuge, die für die Bildverarbeitungsaufgaben benötigt werden, werden eingeführt.

Die Vorlesung beginnt mit einem Überblick über Vorverarbeitungsalgorithmen, wie zum Beispiel Streustrahlkorrektur für Röntgenbilder, Kantenerkennung, Superresolution und kantenerhaltende Rauschunterdrückung. Das zweite Kapitel beschreibt die automatische Bildanalyse mit Merkmalsdeskriptoren, Punkterkennung und Segmentierung mit Bottom-up-Algorithmen wie dem Random-Walker oder Top-Down-Ansätzen wie aktiven Formmodellen. Darüber hinaus deckt die Vorlesung auch geometrische Kalibrierungsalgorithmen zur Einzelansicht-Kalibrierung, Epipolargeometrie und Faktorisierung ab. Der letzte Teil der Vorlesung deckt nicht-starre Registrierung auf der Grundlage von Variationsmethoden und bewegungskompensierter Bildrekonstruktion ab.

**Lernziele und Kompetenzen:**

English Version:

The participants

- summarize the contents of the lecture.
- apply pre-processing algorithms such as scatter correction and edge-preserving filtering.
- extract information from images automatically by image analysis methods such as key point detectors and segmentation algorithms.
- calibrate projection geometries for single images and image sequences using the described methods.
- develop non-rigid registration methods using variational calculus and different regularizers.
- adopt algorithms to new domains by appropriate modifications.

Deutsche Version:

Die Teilnehmer

- fassen die Inhalte der Vorlesung zusammen.
- wenden Vorverarbeitungsalgorithmen wie Streustrahlkorrektur und kantenerhaltende Filterung an.



- extrahieren automatisch Informationen aus Bildern, indem sie Bildanalyseverfahren wie Punktdetektoren und Segmentierungsalgorithmen verwenden.
- kalibrieren Projektionsgeometrien für Einzelbilder und Bildsequenzen mit den beschriebenen Methoden.
- entwickeln nicht-starre Registrierungsmethoden mit Hilfe von Variationsrechnung und unterschiedlichen Regularisierungen.
- wenden Algorithmen auf neue Modalitäten durch entsprechende Änderungen im Algorithmus an.

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

**[1] Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule | Vertiefungsrichtung Mustererkennung)

Dieses Modul ist daneben auch in den Studienfächern "Advanced Optical Technologies (Master of Science)", "Communications and Multimedia Engineering (Master of Science)", "Computational Engineering (Master of Science with Honours)", "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Master of Science)", "Informatik (Master of Science)", "Information and Communication Technology (Master of Science)", "Informations- und Kommunikationstechnik (Master of Science)", "Mathematik (Bachelor of Science)", "Medizintechnik (Bachelor of Science)", "Medizintechnik (Master of Science)" verwendbar.

---

**Studien-/Prüfungsleistungen:**

Interventional Medical Image Processing (VHB-Kurs) (Prüfungsnummer: 41401)

Prüfungsleistung, Klausur, Dauer (in Minuten): 60

Anteil an der Berechnung der Modulnote: 100% Prüfungssprache: Englisch

Erstablingung: WS 2019/2020, 1. Wdh.: SS 2020

1. Prüfer: Andreas Maier (100196)

---

---

**Modulbezeichnung:** Maschinelles Lernen für Zeitreihen (MLTS) 5 ECTS  
 (Machine Learning for Time Series)

Modulverantwortliche/r: Björn Eskofier, Oliver Amft  
 Lehrende: Björn Eskofier, Oliver Amft

---

Startsemester: WS 2019/2020	Dauer: 1 Semester	Turnus: jährlich (WS)
Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Englisch

---

**Lehrveranstaltungen:**

Maschinelles Lernen für Zeitreihen (WS 2019/2020, Vorlesung, 2 SWS, Björn Eskofier et al.)  
 Maschinelles Lernen für Zeitreihen Übung (WS 2019/2020, Übung, 2 SWS, Nooshin Haji Ghassemi et al.)

---

**Vorhergehende Module:**

Pattern Analysis  
 Introduction to Pattern Recognition  
 Pattern Recognition

---

**Inhalt:**

Die Vorlesung vermittelt Konzepte des Maschinellen Lernens speziell im Hinblick auf Anwendungen bei Zeitreihen. Es handelt sich hier um eine Spezialisierungsvorlesung, eine erfolgreiche Absolvierung der Vorlesungen „IntroPR“ und/oder „Pattern Recognition“/„Pattern Analysis“ wird empfohlen. Konzepte, die in „IntroPR“ vermittelt werden, werden hier als Grundwissen vorausgesetzt. Die folgenden Themen werden in der Vorlesung behandelt:

- Ein Überblick über die Anwendungsgebiete der Zeitreihenanalyse
- Methodische Grundlagen des Maschinellen Lernens (ML) für die Analyse von Zeitreihen, beispielsweise Gauß-Prozesse, Monte-Carlo Sampling und Deep Learning
- Design, Implementierung und Evaluation von ML Methoden, um Probleme in Zeitreihen zu adressieren
- Arbeitstechniken in bekannten Toolboxes zur Implementierung von relevanten Methoden, beispielsweise Tensorflow/Keras

**Content**

Aim of the lecture is to teach Machine learning (ML) methods for a variety of time series applications. The following topics will be covered:

- An overview of applications of time series analysis
- Fundamentals of Machine learning (ML) methods, such as Gaussian processes, Monte Carlo sampling methods and deep learning, for time series analysis
- Design, implementation and evaluation of ML methods in order to address time series problems
- Working with widely-used toolboxes that can be used for implementation of ML methods, such as Tensorflow or Keras

**Lernziele und Kompetenzen:**

- Die Studierenden sollen ein Verständnis für Zeitreihenprobleme und deren Lösungen in Applikationsgebieten der Industrie, Medizin, dem Finanzwesen, etc. entwickeln
- Die Studierenden erlernen Konzepte des Maschinellen Lernens im Allgemeinen und deren Anwendung auf Zeitreihen im Besonderen
- Die Studierenden erlernen die Charakteristika von Zeitreihendaten und werden zur Entwicklung und Implementierung von ML-Methoden angeleitet, um solche Daten in konkreten Fragestellungen zu modellieren, manipulieren und vorherzusagen.

**Learning Objectives**

- Students develop an understanding of concepts of time series problems and their wide applications in industry, medicine, finance, etc.
- Students learn concepts of machine learning (ML) methods in general and tackling time series problems in particular

- Students understand the characteristics of time series data and will be capable of developing and implementing ML methods to model, predict and manipulate such data in concrete problems

**Studon:** [https://www.studon.fau.de/studon/goto.php?target=crs\\_2202195](https://www.studon.fau.de/studon/goto.php?target=crs_2202195)

**Literatur:**

- Machine Learning: A Probabilistic Perspective, Kevin Murphy, MIT press, 2012
- The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Trevor Hastie, Robert Tibshirani, Jerome Friedman, Springer, 2009
- Deep Learning, Ian Goodfellow and Yoshua Bengio and Aaron Courville, MIT Press, 2016
- Reinforcement Learning: An Introduction, Richard S. Sutton and Andrew G. Barto, MIT press, 1998

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

**[1] Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule | Vertiefungsrichtung Mustererkennung)

Dieses Modul ist daneben auch in den Studienfächern "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Master of Science)", "Elektrotechnik, Elektronik und Informationstechnik (Bachelor of Science)", "Elektrotechnik, Elektronik und Informationstechnik (Master of Science)", "Informatik (Master of Science)", "Mathematik (Bachelor of Science)", "Medizintechnik (Master of Science)" verwendbar.

---

**Studien-/Prüfungsleistungen:**

Maschinelles Lernen für Zeitreihen (Prüfungsnummer: 428256)

(englische Bezeichnung: Machine Learning for Time Series)

Prüfungsleistung, mündliche Prüfung, Dauer (in Minuten): 30

Anteil an der Berechnung der Modulnote: 100% Prüfungssprache: Deutsch oder Englisch

Erstablingung: WS 2019/2020, 1. Wdh.: SS 2020

1. Prüfer: Björn Eskofier (100122)

---

---

**Modulbezeichnung:** Maschinelles Lernen für Zeitreihen Deluxe (MLTS+) 7.5 ECTS  
 (Machine Learning for Time Series)

Modulverantwortliche/r: Björn Eskofier, Oliver Amft  
 Lehrende: Björn Eskofier, Oliver Amft

---

Startsemester: WS 2019/2020	Dauer: 1 Semester	Turnus: jährlich (WS)
Präsenzzeit: 90 Std.	Eigenstudium: 135 Std.	Sprache: Englisch

---

**Lehrveranstaltungen:**

Maschinelles Lernen für Zeitreihen (WS 2019/2020, Vorlesung, 2 SWS, Björn Eskofier et al.)  
 Maschinelles Lernen für Zeitreihen Übung (WS 2019/2020, Übung, 2 SWS, Nooshin Haji Ghassemi et al.)  
 Maschinelles Lernen für Zeitreihen Laborprojekt (WS 2019/2020, Praktikum, 2 SWS, Nooshin Haji Ghassemi)

---

**Vorhergehende Module:**

Pattern Analysis  
 Introduction to Pattern Recognition  
 Pattern Recognition

---

**Inhalt:**

Die Vorlesung vermittelt Konzepte des Maschinellen Lernens speziell im Hinblick auf Anwendungen bei Zeitreihen. Es handelt sich hier um eine Spezialisierungsvorlesung, eine erfolgreiche Absolvierung der Vorlesungen „IntroPR“ und/oder „Pattern Recognition“/„Pattern Analysis“ wird empfohlen. Konzepte, die in „IntroPR“ vermittelt werden, werden hier als Grundwissen vorausgesetzt. Die folgenden Themen werden in der Vorlesung behandelt:

- Ein Überblick über die Anwendungsgebiete der Zeitreihenanalyse
- Methodische Grundlagen des Maschinellen Lernens (ML) für die Analyse von Zeitreihen, beispielsweise Gauß-Prozesse, Monte-Carlo Sampling und Deep Learning
- Design, Implementierung und Evaluation von ML Methoden, um Probleme in Zeitreihen zu adressieren
- Arbeitstechniken in bekannten Toolboxes zur Implementierung von relevanten Methoden, beispielsweise Tensorflow/Keras

**Content**

Aim of the lecture is to teach Machine learning (ML) methods for a variety of time series applications. The following topics will be covered:

- An overview of applications of time series analysis
- Fundamentals of Machine learning (ML) methods, such as Gaussian processes, Monte Carlo sampling methods and deep learning, for time series analysis
- Design, implementation and evaluation of ML methods in order to address time series problems
- Working with widely-used toolboxes that can be used for implementation of ML methods, such as Tensorflow or Keras

**Lernziele und Kompetenzen:**

- Die Studierenden sollen ein Verständnis für Zeitreihenprobleme und deren Lösungen in Applikationsgebieten der Industrie, Medizin, dem Finanzwesen, etc. entwickeln
- Die Studierenden erlernen Konzepte des Maschinellen Lernens im Allgemeinen und deren Anwendung auf Zeitreihen im Besonderen
- Die Studierenden erlernen die Charakteristika von Zeitreihendaten und werden zur Entwicklung und Implementierung von ML-Methoden angeleitet, um solche Daten in konkreten Fragestellungen zu modellieren, manipulieren und vorherzusagen.

**Learning Objectives**

- Students develop an understanding of concepts of time series problems and their wide applications in industry, medicine, finance, etc.

- Students learn concepts of machine learning (ML) methods in general and tackling time series problems in particular
- Students understand the characteristics of time series data and will be capable of developing and implementing ML methods to model, predict and manipulate such data in concrete problems

**Studon:** [https://www.studon.fau.de/studon/goto.php?target=crs\\_2202195](https://www.studon.fau.de/studon/goto.php?target=crs_2202195)

**Literatur:**

- Machine Learning: A Probabilistic Perspective, Kevin Murphy, MIT press, 2012
- The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Trevor Hastie, Robert Tibshirani, Jerome Friedman, Springer, 2009
- Deep Learning, Ian Goodfellow and Yoshua Bengio and Aaron Courville, MIT Press, 2016
- Reinforcement Learning: An Introduction, Richard S. Sutton and Andrew G. Barto, MIT press, 1998

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

**[1] Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule | Vertiefungsrichtung Mustererkennung)

Dieses Modul ist daneben auch in den Studienfächern "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Master of Science)", "Elektrotechnik, Elektronik und Informationstechnik (Master of Science)", "Informatik (Master of Science)", "Mathematik (Bachelor of Science)", "Medizintechnik (Bachelor of Science)", "Medizintechnik (Master of Science)" verwendbar.

---

**Studien-/Prüfungsleistungen:**

Maschinelles Lernen für Zeitreihen Deluxe (Prüfungsnummer: 482355)

(englische Bezeichnung: Machine Learning for Time Series Deluxe)

Prüfungsleistung, mündliche Prüfung, Dauer (in Minuten): 30

Anteil an der Berechnung der Modulnote: 100% Prüfungssprache: Deutsch oder Englisch

Erstablingung: WS 2019/2020, 1. Wdh.: SS 2020

1. Prüfer: Björn Eskofier (100122)

---

---

**Modulbezeichnung:** Pattern Recognition (PR) 5 ECTS  
(Pattern Recognition)

Modulverantwortliche/r: Andreas Maier

Lehrende: Elmar Nöth, Sebastian Käßler

---

Startsemester: WS 2019/2020

Dauer: 1 Semester

Turnus: jährlich (WS)

Präsenzzeit: 60 Std.

Eigenstudium: 90 Std.

Sprache: Englisch

---

**Lehrveranstaltungen:**

Pattern Recognition (WS 2019/2020, Vorlesung, 3 SWS, Elmar Nöth)

Pattern Recognition Exercises (WS 2019/2020, Übung, 1 SWS, Stephan Seitz et al.)

---

**Empfohlene Voraussetzungen:**

- Well grounded in probability calculus, linear algebra/matrix calculus
  - The attendance of our bachelor course 'Introduction to Pattern Recognition' is not required but certainly helpful.
  - Gute Kenntnisse in Wahrscheinlichkeitsrechnung und Linearer Algebra/Matrizenrechnung
  - Der Besuch der Bachelor-Vorlesung 'Introduction to Pattern Recognition' ist zwar keine Voraussetzung, aber sicherlich von Vorteil.
- 

**Inhalt:**

Mathematical foundations of machine learning based on the following classification methods:

- Bayesian classifier
- Logistic Regression
- Naive Bayes classifier
- Discriminant Analysis
- norms and norm dependent linear regression
- Rosenblatt's Perceptron
- unconstraint and constraint optimization
- Support Vector Machines (SVM)
- kernel methods
- Expectation Maximization (EM) Algorithm and Gaussian Mixture Models (GMMs)
- Independent Component Analysis (ICA)
- Model Assessment
- AdaBoost

Mathematische Grundlagen der maschinellen Klassifikation am Beispiel folgender Klassifikatoren:

- Bayes-Klassifikator
- Logistische Regression
- Naiver Bayes-Klassifikator
- Diskriminanzanalyse
- Normen und normabhängige Regression
- Rosenblatts Perzeptron
- Optimierung ohne und mit Nebenbedingungen
- Support Vector Maschines (SVM)
- Kernelmethoden
- Expectation Maximization (EM)-Algorithmus und Gaußsche Mischverteilungen (GMMs)
- Analyse durch unabhängige Komponenten
- Modellbewertung
- AdaBoost

**Lernziele und Kompetenzen:**

Die Studenten

- verstehen die Struktur von Systemen zur maschinellen Klassifikation einfacher Muster
- erläutern die mathematischen Grundlagen ausgewählter maschineller Klassifikatoren

- wenden Klassifikatoren zur Lösung konkreter Klassifikationsproblem an
- beurteilen unterschiedliche Klassifikatoren in Bezug auf ihre Eignung
- verstehen in der Programmiersprache Python geschriebene Lösungen von Klassifikationsproblemen und Implementierungen von Klassifikatoren

#### Students

- understand the structure of machine learning systems for simple patterns
- explain the mathematical foundations of selected machine learning techniques
- apply classification techniques in order to solve given classification tasks
- evaluate various classifiers with respect to their suitability to solve the given problem
- understand solutions of classification problems and implementations of classifiers written in the programming language Python

#### Literatur:

- Richard O. Duda, Peter E. Hart, David G. Stock: Pattern Classification, 2nd edition, John Wiley&Sons, New York, 2001
- Trevor Hastie, Robert Tibshirani, Jerome Friedman: The Elements of Statistical Learning - Data Mining, Inference, and Prediction, 2nd edition, Springer, New York, 2009
- Christopher M. Bishop: Pattern Recognition and Machine Learning, Springer, New York, 2006

#### Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

##### [1] Informatik (Bachelor of Science)

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule | Vertiefungsrichtung Mustererkennung)

Dieses Modul ist daneben auch in den Studienfächern "Advanced Optical Technologies (Master of Science)", "Advanced Signal Processing & Communications Engineering (Master of Science)", "Communications and Multimedia Engineering (Master of Science)", "Computational Engineering (Master of Science with Honours)", "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Master of Science)", "Digital Humanities (Master of Arts)", "Informatik (Master of Science)", "Information and Communication Technology (Master of Science)", "Informations- und Kommunikationstechnik (Master of Science)", "International Information Systems (IIS) (Master of Science)", "Mathematik (Bachelor of Science)", "Medizintechnik (Master of Science)" verwendbar.

#### Studien-/Prüfungsleistungen:

Pattern Recognition (Prüfungsnummer: 41301)

(englische Bezeichnung: Pattern Recognition)

Prüfungsleistung, mündliche Prüfung, Dauer (in Minuten): 30

Anteil an der Berechnung der Modulnote: 100%

weitere Erläuterungen:

30-minütige mündliche Prüfung über den Stoff der Vorlesung und der Tafelübung

30 minute oral exam about the lecture and the corresponding exercises

Prüfungssprache: Deutsch oder Englisch

Erstablingung: WS 2019/2020, 1. Wdh.: SS 2020

1. Prüfer: Elmar Nöth (100183)

---

**Modulbezeichnung:** Pattern Recognition Deluxe (PR) 7.5 ECTS  
 (Pattern Recognition Deluxe)

Modulverantwortliche/r: Andreas Maier  
 Lehrende: Elmar Nöth, Sebastian Käßler

---

Startsemester: WS 2019/2020	Dauer: 1 Semester	Turnus: jährlich (WS)
Präsenzzeit: 80 Std.	Eigenstudium: 145 Std.	Sprache: Englisch

---

**Lehrveranstaltungen:**

Pattern Recognition (WS 2019/2020, Vorlesung, 3 SWS, Elmar Nöth)  
 Pattern Recognition Exercises (WS 2019/2020, Übung, 1 SWS, Stephan Seitz et al.)  
 Pattern Recognition Programming (WS 2019/2020, Übung, 2 SWS, Lina Felsner et al.)

---

**Empfohlene Voraussetzungen:**

- Well grounded in probability calculus, linear algebra/matrix calculus
- A pattern recognition system consists of the following steps: sensor data acquisition, pre-processing, feature extraction, and classification. Our bachelor course 'Introduction to Pattern Recognition' focuses on the first three steps; this course on the final step of the pipeline, i.e. classification/machine learning. Knowledge about feature extraction is not required for studying the mathematical foundations of machine learning, but it is certainly helpful to get a better understanding of the whole picture.
- Gute Kenntnisse in Wahrscheinlichkeitsrechnung und Linearer Algebra/Matrizenrechnung
- Ein Mustererkennungssystem besteht aus den folgenden Verarbeitungsstufen: Aufnahme der Sensordaten, Vorverarbeitung, Merkmalsextraktion und Klassifikation. Unsere Bachelor-Vorlesung 'Introduction to Pattern Recognition' behandelt hauptsächlich die ersten drei Stufen, während diese Veranstaltung sich mit den mathematischen Grundlagen der Klassifikation/des maschinellen Lernens beschäftigt. Wissen über die Merkmalsextraktion ist für das Verständnis der mathematischen Grundlagen der automatischen Klassifikation zwar nicht notwendig, aber es hilft sicherlich, das Gesamtbild besser zu verstehen.

**Vorhergehende Module:**

Introduction to Pattern Recognition Deluxe  
 Introduction to Pattern Recognition

---

**Inhalt:**

Mathematical foundations of machine learning based on the following classification methods:

- Bayesian classifier
- Logistic Regression
- Naive Bayes classifier
- Discriminant Analysis
- norms and norm dependent linear regression
- Rosenblatt's Perceptron
- unconstraint and constraint optimization
- Support Vector Machines (SVM)
- kernel methods
- Expectation Maximization (EM) Algorithm and Gaussian Mixture Models (GMMs)
- Independent Component Analysis (ICA)
- Model Assessment
- AdaBoost

Mathematische Grundlagen der maschinellen Klassifikation am Beispiel folgender Klassifikatoren:

- Bayes-Klassifikator
- Logistische Regression
- Naiver Bayes-Klassifikator
- Diskriminanzanalyse



- Normen und normabhängige Regression
- Rosenblatts Perzeptron
- Optimierung ohne und mit Nebenbedingungen
- Support Vector Machines (SVM)
- Kernelmethoden
- Expectation Maximization (EM)-Algorithmus und Gaußsche Mischverteilungen (GMMs)
- Analyse durch unabhängige Komponenten
- Modellbewertung
- AdaBoost

### Lernziele und Kompetenzen:

Die Studierenden

- verstehen die Struktur von Systemen zur maschinellen Klassifikation einfacher Muster
- erläutern die mathematischen Grundlagen ausgewählter maschineller Klassifikatoren
- wenden Klassifikatoren zur Lösung konkreter Klassifikationsprobleme an
- beurteilen unterschiedliche Klassifikatoren in Bezug auf ihre Eignung
- lösen selbständig Klassifikationsprobleme und schreiben eigene Implementierungen von Klassifikatoren in der Programmiersprache Python

Students

- understand the structure of machine learning systems for simple patterns
- explain the mathematical foundations of selected machine learning techniques
- apply classification techniques in order to solve given classification tasks
- evaluate various classifiers with respect to their suitability to solve the given problem
- solve classification problems on their own and write their own implementations of classifiers in the programming language Python

### Literatur:

- Richard O. Duda, Peter E. Hart, David G. Stock: Pattern Classification, 2nd edition, John Wiley&Sons, New York, 2001
- Trevor Hastie, Robert Tibshirani, Jerome Friedman: The Elements of Statistical Learning - Data Mining, Inference, and Prediction, 2nd edition, Springer, New York, 2009
- Christopher M. Bishop: Pattern Recognition and Machine Learning, Springer, New York, 2006

### Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

#### [1] Informatik (Bachelor of Science)

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule | Vertiefungsrichtung Mustererkennung)

Dieses Modul ist daneben auch in den Studienfächern "Communications and Multimedia Engineering (Master of Science)", "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Master of Science)", "Informatik (Master of Science)", "Information and Communication Technology (Master of Science)", "Informations- und Kommunikationstechnik (Master of Science)", "Mathematik (Bachelor of Science)", "Medizintechnik (Master of Science)" verwendbar.

### Studien-/Prüfungsleistungen:

Pattern Recognition Deluxe (Prüfungsnummer: 456863)

(englische Bezeichnung: Pattern Recognition Deluxe)

Prüfungsleistung, mündliche Prüfung, Dauer (in Minuten): 30

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Pattern Recognition
- Pattern Recognition Exercises

weitere Erläuterungen:

30-minütige mündliche Prüfung über den Stoff der Vorlesung und der Übungen. Voraussetzung ist die erfolgreiche Bearbeitung der Programmieraufgaben.

30 minute oral exam about the lecture and the exercises. It is required to successfully complete the programming tasks.

Prüfungssprache: Deutsch oder Englisch

Erstablingung: WS 2019/2020, 1. Wdh.: SS 2020

1. Prüfer: Elmar Nöth (100183)

---

---

**Modulbezeichnung:** Programmierung und Architekturen von Cluster-Rechnern (PS-PAC) 7.5 ECTS  
 (Programming and Architecture of Compute Clusters)

Modulverantwortliche/r: Michael Philippsen  
 Lehrende: Dietmar Fey, Thorsten Blaß

---

Startsemester: SS 2019	Dauer: 1 Semester	Turnus: jährlich (SS)
Präsenzzeit: 90 Std.	Eigenstudium: 135 Std.	Sprache: Deutsch oder Englisch

---

**Lehrveranstaltungen:**

Programmierung und Architekturen von Cluster-Rechnern (SS 2019, Vorlesung, 4 SWS, Thorsten Blaß et al.)  
 Übung zu Programmierung und Architekturen von Cluster-Rechnern (SS 2019, Übung, 2 SWS, Thorsten Blaß)

---

**Inhalt:**

There are many problems where a single computer cannot compute a solution in reasonable time. In this case we must use multiple computers working in concert (a cluster). If a researcher wants to use a cluster at another location (because the current cluster is 'full'), then the researcher needs a way to safely run jobs in some other location. This is 'grid computing'. Similarly, a smartphone may not suffice, and we must let the computation run elsewhere (potentially in parallel). If many smartphones need remote services, we need many remote machines, which is again a cluster. This is called 'cloud-computing'. All of the above are topics of this lecture.

In this lecture we will:

- examine use cases
- examine ways to use a cluster, cloud, or grid.
- examine how to optimize applications for clusters (load balancing, message passing optimizations)
- examine message passing libraries, both from how to use them to how they are implemented.
- examine how high-end networking hardware works and how it is programmed
- study various cloud and grid computing systems
- look into the future where clouds and clusters are augmented with compute accelerators (exascale computing).

In the exercises you will:

- program a scalable application that runs on a cluster (there are a number of projects to choose from).
- learn how to achieve scalability,
- learn how to use a cluster practically, and
- learn how to interface with a cluster.

**Lernziele und Kompetenzen:**

The students

- program a scalable application that runs on a cluster
- optimize their implementation in order to achieve scalability
- use a cluster practically and configure it to satisfy their needs

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

**[1] Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule)

Dieses Modul ist daneben auch in den Studienfächern "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Bachelor of Science)", "Informatik (Master of Science)", "Information and Communication Technology (Master of Science)", "Informations- und Kommunikationstechnik (Master of Science)", "Mathematik (Bachelor of Science)" verwendbar.

---

**Studien-/Prüfungsleistungen:**

Programmierung und Architekturen von Cluster-Rechnern (Prüfungsnummer: 399289)

(englische Bezeichnung: Programming and Architecture of Compute Clusters)

Prüfungsleistung, mündliche Prüfung, Dauer (in Minuten): 30

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Programmierung und Architekturen von Cluster-Rechnern
- Übung zu Programmierung und Architekturen von Cluster-Rechnern

Erstablesung: SS 2019, 1. Wdh.: WS 2019/2020

1. Prüfer: Michael Philippsen (100113)

1. Prüfer: Dietmar Fey (100101)

---

**Organisatorisches:**

The time and date of the exercise class will be determined later in cooperation. No exercise class the first two weeks, however. The choice of language used in the class (English or German) is by request of the students.

---

**Modulbezeichnung:** Analyse und Design objektorientierter Softwaresysteme mit UML (PSWT-OOAD) **5 ECTS**  
 (Analysis and Design of Object Oriented Software Systems with UML)

Modulverantwortliche/r: Detlef Kips  
 Lehrende: Detlef Kips, Martin Jung

---

Startsemester: SS 2019	Dauer: 1 Semester	Turnus: jährlich (SS)
Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Deutsch

---

**Lehrveranstaltungen:**

Analyse und Design objektorientierter Softwaresysteme mit der Unified Modeling Language (UML) (SS 2019, Vorlesung, 2 SWS, Detlef Kips et al.)

Übungen zu Analyse und Design objektorientierter Softwaresysteme mit der Unified Modeling Language (UML) (SS 2019, Übung, 2 SWS, Ralf Ellner)

---

**Inhalt:**

Die sogenannte "Unified Modeling Language" (UML) ist der seit Jahren weltweit akzeptierte Notationsstandard für die Modellierung komplexer Softwaresysteme. Mit einem reichhaltigen Repertoire an graphischen und textuellen Ausdrucksmöglichkeiten bietet die UML ihren Anwendern die Möglichkeit, die Anforderungen an die Zielsoftware, ihre statischen bzw. dynamischen Systemeigenschaften sowie die gewählte Softwarearchitektur halbformal zu spezifizieren, im Team darüber zu kommunizieren und große Teile des Programmcodes aus den spezifizierten Systemmodellen zu generieren.

Ziel dieser Vorlesung ist es, die Studierenden mit Syntax und Semantik der UML vertraut zu machen und zu demonstrieren, wie die UML im Rahmen eines "typischen" Softwareentwicklungsprozesses angewendet werden kann. Zu diesem Zweck werden die verschiedenen Diagrammtypen und Notationselemente der UML schrittweise eingeführt und anhand eines durchgängigen Anwendungsbeispiels im Rahmen eines konkreten Vorgehensmodells über alle Entwicklungsphasen hinweg eingesetzt.

**Lernziele und Kompetenzen:**

Im Rahmen dieser Veranstaltung sollen die Studierenden insbesondere die Kompetenz erwerben,

- die syntaktische Struktur und die Semantik vorgegebener UML-Modelle bzw. Modellausschnitte zu analysieren und zu erläutern
- verschiedene Sprachelemente der UML (und ggf. deren Kombination) im Hinblick auf ihre Eignung zur Abbildung charakteristischer Modellierungsprobleme im Rahmen eines Softwareentwicklungsprozesses zu bewerten, auszuwählen und anzuwenden
- die Struktur und Systematik des UML-Metamodells zu erläutern und die UML mit geeigneten Metamodellierungskonzepten auf spezifische Anwendungskontexte anzupassen
- zu einer gegebenen Anforderungsdefinition im Rahmen einer systematischen Analyse- und Entwurfsmethodik ein integriertes UML-Systemmodell zu erstellen
- den Funktionsumfang eines UML-basierten Modellierungswerkzeugs zu bewerten, ein geeignetes Werkzeug auszuwählen und sicher anzuwenden.

**Literatur:**

- Rumbaugh, J.; Booch, G.; Jacobson, I.: The Unified Modeling Language Reference Manual, Addison-Wesley, 2004
- Hitz, M.; Kappel, G.; Kapsammer, E.; Retschitzegger, W.: UML @ work , 3., aktualisierte und überarbeitete Auflage, dpunkt-Verlag, 2005
- Winter, M.: Methodische objektorientierte Softwareentwicklung, dpunkt-Verlag, 2005
- Störrle, H.: UML 2 erfolgreich einsetzen, Addison-Wesley, 2007
- Rumpe, B.: Modellierung mit UML: Sprache, Konzepte und Methodik, Springer-Verlag, 2. Auflage, 2011
- Seidl, M., Brandsteidl, M., Huemer, C., Kappek, G.: UML@classroom - Eine Einführung in die objektorientierte Modellierung, dpunkt-Verlag, 2012
- Rupp, C.; Queins, S., et al. UML 2 glasklar: Praxiswissen für die UML-Modellierung, Carl Hanser Verlag, 2012

Die einschlägige Originalliteratur zur UML findet man auf der Website der Object Management Group ( <http://www.omg.org/spec/UML> ).

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

**[1] Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule)

Dieses Modul ist daneben auch in den Studienfächern "International Information Systems (IIS) (Master of Science)" verwendbar.

---

**Studien-/Prüfungsleistungen:**

Analyse und Design objektorientierter Softwaresysteme mit der Unified Modeling Language (UML)  
(Prüfungsnummer: 510375)

Prüfungsleistung, mündliche Prüfung, Dauer (in Minuten): 30

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Analyse und Design objektorientierter Softwaresysteme mit der Unified Modeling Language (UML)
- Übungen zu Analyse und Design objektorientierter Softwaresysteme mit der Unified Modeling Language (UML)

Erstablingung: SS 2019, 1. Wdh.: SS 2020

1. Prüfer: Detlef Kips (100178)

---

**Modulbezeichnung:** Testen von Softwaresystemen (PSWT-TSWS) 5 ECTS  
(Testing of Software Systems)

Modulverantwortliche/r: Norbert Oster

Lehrende: Norbert Oster, Klaudia Dussa-Zieger

Startsemester: SS 2019

Dauer: 1 Semester

Turnus: jährlich (SS)

Präsenzzeit: 60 Std.

Eigenstudium: 90 Std.

Sprache: Deutsch

**Lehrveranstaltungen:**

Testen von Softwaresystemen (SS 2019, Vorlesung, 4 SWS, Klaudia Dussa-Zieger et al.)

**Inhalt:**

- Test-Terminologie und Software-Qualität nach ISO/IEC 9126
- Fundamentaler Testprozess
- Teststufen im Softwarelebenszyklus
- Statischer Test: Reviews
- Erfahrungsbasiertes Testen
- Black-Box-Testverfahren: Äquivalenzklassen-/Grenzwerttest, Zustandsbezogener Test, Entscheidungstabellentest
- Statische Analyse: Daten- und Kontrollflussanomalien
- White-Box-Testverfahren: Kontrollflussbasiert, Datenflussbasiert, Bedingungsüberdeckung
- Mutationstest
- Testmanagement: Planung, Kostenschätzung, Überwachung, Risikobewertung, Priorisierung, Fehlermanagement
- Formale Verifikation: Theorem Proving und Model Checking
- Quantitative Zuverlässigkeitsbewertung

**Lernziele und Kompetenzen:**

Die Studierenden

- unterscheiden die verschiedenen Fachbegriffe aus der Test-Domäne
- erläutern die unterschiedlichen Aspekte der Software-Qualität
- beschreiben den fundamentalen Testprozess und seine einzelnen Teilaufgaben
- erläutern die wichtigsten Aspekte der Testpsychologie und entscheiden auf dieser Basis z.B. über das Testteam
- beschreiben die typischen Teststufen und ordnen diese den Phasen im Softwarelebenszyklus zu
- stellen die Unterschiede zwischen dynamischem Test, Review und statischer Analyse heraus
- unterscheiden verschiedene Review-Arten und veranschaulichen deren typische Arbeitsschritte und Rollen
- differenzieren unterschiedliche Formen erfahrungsbasierten Testens
- wenden das Verfahren der Äquivalenzklassenbildung an und ermitteln entsprechende Testfälle für den Grenzwerttest
- entwickeln Entscheidungstabellen für beliebige Testaufgaben und bestimmen die entsprechenden Testfälle
- erläutern typische Daten-/Kontrollflussanomalien an selbst-gewählten Beispielen
- unterscheiden verschiedene kontrollfluss-, datenfluss und bedingungsorientierte Testkriterien
- wenden die grundlegenden White-Box-Testkriterien an und leiten entsprechende Testfälle ab
- begründen die Ordnung der White-Box-Überdeckungskriterien in ihrer Subsumptionshierarchie
- erläutern das Konzept des Mutationstests zur quantitativen Bewertung der Testgüte
- gestalten einen konkreten Testprozess aus der Sicht eines Testmanagers
- beschreiben die wichtigsten Facetten des Fehlermanagements
- erläutern den Unterschied zwischen Theorem Proving und Model Checking und skizzieren das jeweilige Vorgehen
- wenden Theorem Proving auf sequentiellen Code an und skizzieren den Beweis der Interferenzfreiheit bei Nebenläufigkeit
- beschreiben Voraussetzungen, Annahmen und Vorgehen bei verschiedenen Arten der quantitativen

#### Zuverlässigkeitsbewertung

- entscheiden je nach Art des Softwareprodukts welche Art der quantitativen Zuverlässigkeitsbewertung zulässig ist
- nennen die wichtigsten Normen und Standards sowie deren typische Anforderungen
- bewerten und benutzen Werkzeuge für verschiedene Testaufgaben

#### Literatur:

- Spillner, Andreas; Linz, Tilo: Basiswissen Softwaretest, dpunkt-Verlag
- Liggesmeyer, Peter: Software-Qualität, Spektrum Verlag
- Spillner, Andreas; Roßner, Thomas; Winter, Mario; Linz, Tilo: Praxiswissen Softwaretest - Testmanagement, dpunkt-Verlag
- Lyu, Michael R.: Handbook of Software Reliability Engineering, McGraw-Hill

---

#### Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

##### [1] Informatik (Bachelor of Science)

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule)

Dieses Modul ist daneben auch in den Studienfächern "International Information Systems (IIS) (Master of Science)" verwendbar.

---

#### Studien-/Prüfungsleistungen:

Testen von Softwaresystemen (Prüfungsnummer: 189989)

Prüfungsleistung, mündliche Prüfung, Dauer (in Minuten): 30

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Testen von Softwaresystemen

Erstablingung: SS 2019, 1. Wdh.: WS 2019/2020

1. Prüfer: Norbert Oster (100120)

---



---

**Modulbezeichnung:** Grundlagen des Übersetzerbaus (PS-UE1) 7.5 ECTS  
 (Principles of Compiler Construction)

Modulverantwortliche/r: Michael Philippsen  
 Lehrende: Michael Philippsen

---

Startsemester: WS 2019/2020	Dauer: 1 Semester	Turnus: jährlich (WS)
Präsenzzeit: 50 Std.	Eigenstudium: 175 Std.	Sprache: Deutsch

---

**Lehrveranstaltungen:**

Grundlagen des Übersetzerbaus (WS 2019/2020, Vorlesung, 2 SWS, Patrick Kreutzer et al.)  
 Übungen zu Grundlagen des Übersetzerbaus (WS 2019/2020, Übung, 2 SWS, Patrick Kreutzer et al.)

---

**Inhalt:**

Auf den ersten Blick erscheint es wenig sinnvoll, sich mit Übersetzerbau zu beschäftigen. Andere Themen scheinen wesentlich näher an der direkten Anwendbarkeit in der industriellen Praxis. Der erste Blick täuscht:

- Übersetzer gehören wohl zu den am gründlichsten studierten mittelgroßen sequentiellen Software-Systemen. Man kann viel aus den Erfahrungen lernen, die im Laufe der Jahre gesammelt wurden.
- In den Übungen, die die Vorlesung begleiten, werden Sie selbst einen (kleinen) Übersetzer entwickeln.
- Für viele Teilnehmer wird dieses Projekt das erste größere Software-Projekt sein. Viele der Algorithmen aus dem Grundstudium werden angewendet.
- Bei jedem von Ihnen verwendeten Übersetzer gehen Sie in der Regel davon aus, dass richtiger Coder erzeugt wird. In der Vorlesung erfahren Sie, wie das geforderte hohe Maß an Korrektheit und Zuverlässigkeit erreicht wird.
- Sie erlangen ein Verständnis für Konzepte von Programmiersprachen und verstehen, welcher Maschinen-Code aus Sprachkonstrukten gemacht wird. Mit diesem Wissen im Hinterkopf verbessern Sie Ihre Fähigkeit, gute und effiziente Programme zu schreiben.
- Übersetzer werden nicht nur für Programmiersprachen benötigt. Spezielle Übersetzer braucht man in vielen Bereichen des täglichen Informatik-Lebens z.B. zur Textformatierung, für Programmtransformationen, für aspektorientiertes Programmieren, für die Verarbeitung von XML, ...
- Es gehört zu einer Ingenieur-Ausbildung, in der Lage zu sein, diejenigen Werkzeuge selbst zu fertigen, die man verwendet. Für Informatiker gehört daher ein Verständnis vom Innenleben eines Übersetzers zum Rüstzeug.

Themen der Vorlesung:

- Übersetzungsprinzipien für imperative Sprachen
- Struktur eines Übersetzers
- Symbolentschlüssler und Zerteiler (Scanner und Parser)
- Abstrakter Syntaxbaum
- Symboltabellen, Umgebungen
- Attributgrammatiken
- Semantische Analyse, Typprüfung
- Automatische Speicherbereinigung
- Code-Erzeugung
- Register Optimization

Themen der Übung:

- In den Tafelübungen werden die in der Vorlesung vorgestellten Themen anhand von Übungsaufgaben angewendet und vertieft.
- Zusätzlich implementieren die Studierenden in Zweiergruppen einen eigenen Übersetzer für eine Beispielprogrammiersprache. Ziel ist es, am Ende einen funktionsfähigen Übersetzer zu haben. Dazu wird ein Grundgerüst eines Übersetzers für den Einstieg vorgegeben.

**Lernziele und Kompetenzen:**

Die Studierenden

- nennen die typischen Aufgaben und Datenstrukturen eines Übersetzers
- erläutern das Konzept des Bootstrapping

- beschreiben Struktur und Arbeitsweise eines Abtasters (Scanner) und zeigen Grenzen und Problemfälle auf
- wenden Grammatiken zur Konstruktion von Zerteilern (Parser) an
- kennen die Komplexität eines Zerteilers für Java
- beschreiben die wichtigsten Aufgaben der semantischen Analyse und wenden diese am Beispiel verschiedener Programmiersprachen (insbesondere Java) an
- skizzieren typische AST-Transformationen am Beispiel des Java-Übersetzers
- veranschaulichen die Grundzüge der Java-Kellermaschine und die zugehörige Transformation von Quell- zu Byte-Code
- analysieren die Unterschiede zwischen Programmiersprachen hinsichtlich Felder und Verbundstrukturen
- erläutern die Verwendung von Stapel- und Kellerspeicher bei der Programmausführung
- kennen verschiedene Maschineninstruktionssätze
- optimieren die Registerverwendung vor der Generierung von Maschinencode
- wenden das Verfahren von Graham & Glanville zur Erzeugung von Maschinencode an
- erkennen Grenzen der Optimierung bei der Code-Generierung und analysieren alternative Strategien
- beschreiben den Unterschied zwischen statischer und dynamischer Ablaufplanung
- untersuchen Besonderheiten des Übersetzerbaus für objekt-orientierte Sprachen
- ergänzen einen vorgegebenen Abtaster und abstrakten Syntaxbaum, um alle Sprachkonstrukte einer Beispielsprache zu unterstützen
- implementieren Konstantenfaltung, den Aufbau der Symboltabelle und Typprüfung auf dem abstrakten Syntaxbaum
- erzeugen Zwischencode aus dem abstrakten Syntaxbaum
- bilden Kontrollstrukturen auf Sprünge ab
- veranschaulichen die Adressierung von (mehrdimensionalen) Feldern
- entwickeln Konventionen für Funktionsaufrufe und den Aufbau des Stacks
- berechnen Offsets fuer Variablen auf dem Stack.
- implementieren eine einfache Registervergabe.
- kennen Details verschiedener Prozessorarchitekturen
- generieren Maschinencode für mindestens eine Prozessorarchitektur
- implementieren eine Laufzeitbibliothek
- wenden Debugging für maschinennahen Code an

#### Literatur:

- „Modern Compiler Implementation in Java“, A.W. Appel, Cambridge University Press, 1998
- „Drachenbuch“, A. Aho, R. Sethi, J. Ullmann: Compilers - Principles, Techniques and Tools, Addison-Wesley, 2006
- „Modern Compiler Design“, Grune, Bal, Jacobs, Langendoen, Wiley, 2002

#### Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

##### [1] Informatik (Bachelor of Science)

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule | Vertiefungsrichtung Programmiersysteme)

Dieses Modul ist daneben auch in den Studienfächern "Computational Engineering (Master of Science with Honours)", "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Master of Science)", "Informatik (Master of Science)", "Information and Communication Technology (Master of Science)", "Informations- und Kommunikationstechnik (Master of Science)", "Mathematik (Bachelor of Science)" verwendbar.

#### Studien-/Prüfungsleistungen:

Grundlagen des Übersetzerbaus (Prüfungsnummer: 42401)

Prüfungsleistung, mündliche Prüfung, Dauer (in Minuten): 30

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Grundlagen des Übersetzerbaus
- Übungen zu Grundlagen des Übersetzerbaus

weitere Erläuterungen:

Die Note ergibt sich aus einer 30minütigen mündlichen Prüfung. Voraussetzung zur Teilnahme an der mündl. Prüfung ist die erfolgreiche Bearbeitung der Übungsaufgaben.

Erstablingung: WS 2019/2020, 1. Wdh.: SS 2020

1. Prüfer: Michael Philippsen (100113)

---

---

**Modulbezeichnung:** Programmierung und Architekturen von Cluster-Rechnern (PS-PAC) 7.5 ECTS  
 (Programming and Architecture of Compute Clusters)

Modulverantwortliche/r: Michael Philippsen  
 Lehrende: Dietmar Fey, Thorsten Blaß

---

Startsemester: SS 2019	Dauer: 1 Semester	Turnus: jährlich (SS)
Präsenzzeit: 90 Std.	Eigenstudium: 135 Std.	Sprache: Deutsch oder Englisch

---

**Lehrveranstaltungen:**

Programmierung und Architekturen von Cluster-Rechnern (SS 2019, Vorlesung, 4 SWS, Thorsten Blaß et al.)

Übung zu Programmierung und Architekturen von Cluster-Rechnern (SS 2019, Übung, 2 SWS, Thorsten Blaß)

---

**Inhalt:**

There are many problems where a single computer cannot compute a solution in reasonable time. In this case we must use multiple computers working in concert (a cluster). If a researcher wants to use a cluster at another location (because the current cluster is 'full'), then the researcher needs a way to safely run jobs in some other location. This is 'grid computing'. Similarly, a smartphone may not suffice, and we must let the computation run elsewhere (potentially in parallel). If many smartphones need remote services, we need many remote machines, which is again a cluster. This is called 'cloud-computing'. All of the above are topics of this lecture.

In this lecture we will:

- examine use cases
- examine ways to use a cluster, cloud, or grid.
- examine how to optimize applications for clusters (load balancing, message passing optimizations)
- examine message passing libraries, both from how to use them to how they are implemented.
- examine how high-end networking hardware works and how it is programmed
- study various cloud and grid computing systems
- look into the future where clouds and clusters are augmented with compute accelerators (exascale computing).

In the exercises you will:

- program a scalable application that runs on a cluster (there are a number of projects to choose from).
- learn how to achieve scalability,
- learn how to use a cluster practically, and
- learn how to interface with a cluster.

**Lernziele und Kompetenzen:**

The students

- program a scalable application that runs on a cluster
- optimize their implementation in order to achieve scalability
- use a cluster practically and configure it to satisfy their needs

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

**[1] Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule | Vertiefungsrichtung Programmiersysteme)

**[2] Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule | Vertiefungsrichtung Rechnerarchitektur)

Dieses Modul ist daneben auch in den Studienfächern "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Bachelor of Science)", "Informatik (Master of Science)", "Information and Communication Tech-

nology (Master of Science)", "Informations- und Kommunikationstechnik (Master of Science)", "Mathematik (Bachelor of Science)" verwendbar.

---

### **Studien-/Prüfungsleistungen:**

Programmierung und Architekturen von Cluster-Rechnern (Prüfungsnummer: 399289)

(englische Bezeichnung: Programming and Architecture of Compute Clusters)

Prüfungsleistung, mündliche Prüfung, Dauer (in Minuten): 30

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Programmierung und Architekturen von Cluster-Rechnern
- Übung zu Programmierung und Architekturen von Cluster-Rechnern

Erstablingung: SS 2019, 1. Wdh.: WS 2019/2020

1. Prüfer: Michael Philippsen (100113)

1. Prüfer: Dietmar Fey (100101)

---

### **Organisatorisches:**

The time and date of the exercise class will be determined later in cooperation. No exercise class the first two weeks, however. The choice of language used in the class (English or German) is by request of the students.

---

**Modulbezeichnung:** CPU Entwurf mit VHDL (CPU) 7.5 ECTS  
 (CPU Design with VHDL)

Modulverantwortliche/r: Dietmar Fey  
 Lehrende: Marc Reichenbach

---

Startsemester: SS 2019	Dauer: 1 Semester	Turnus: jährlich (SS)
Präsenzzeit: 90 Std.	Eigenstudium: 135 Std.	Sprache: Deutsch

---

**Lehrveranstaltungen:**

  CPU Entwurf mit VHDL (SS 2019, Vorlesung, 4 SWS, Marc Reichenbach)  
   Übungen zu CPU Entwurf mit VHDL (SS 2019, Übung, 2 SWS, Philipp Holzinger et al.)

---

**Empfohlene Voraussetzungen:**

keine

---

**Inhalt:**

Die Vorlesung wird einen Einblick in die wichtigsten Strukturen eines Rechners und insbesondere die der CPU geben. Wie sind die Register, die ALU, die Caches usw. aufgebaut, wie sind sie miteinander verschaltet? Wie werden Pipelines gebildet?

Um die Vorlesung verstehen und die Übungsaufgaben lösen zu können, werden VHDL-Kenntnisse benötigt. Diese werden in der Vorlesung vermittelt.

Im ersten Teil der Vorlesung geht es darum, aus einfachen Und-, Oder- und Inverter-Gattern größere Strukturen aufzubauen. So werden beispielsweise die Register und die Komponenten der ALU (Addierer, Multiplizierer, Shifter usw.) nach und nach aufgebaut. Aus den Einzelteilen lassen sich dann wiederum die Registerbänke, die ALU, die Speichereinheit usw. zusammensetzen.

Mit guten Rechenwerken allein ist eine CPU noch nicht wirklich schnell. Es müssen schnell genug Instruktionen gelesen und dekodiert sowie die Operanden geholt werden können. Nach dem schnellen Verarbeiten in den Rechenwerken müssen die Ergebnisse dann wiederum schnell in die Zielregister bzw. Ziel-Speicherzellen kopiert werden. Das wichtigste Verfahren, um hierbei Performance zu gewinnen, ist das Pipelining.

Leider gibt es eine Reihe von Bedingungen, die erfüllt sein müssen, dass Pipelining wirklich gut funktioniert. So dürfen keine Daten-Abhängigkeiten im Code vorhanden sein, müssen Sprungziele rechtzeitig bekannt sein und dürfen bestimmte Ressourcen nicht mehrfach gleichzeitig genutzt werden. Mittels Forwarding, Sprungvorhersage und Caches können die Probleme gemildert werden.

Mit den so gewonnenen Fähigkeiten sind dann auch Multi-Core- und Multi-Threading-Architekturen verständlich.

**Lernziele und Kompetenzen:**

Die Studierenden verstehen den inneren Aufbau moderner CPUs (Pipeline, Registerbänke, ALU, Caches, Memory-Management-Unit, Segmentierungseinheit, ...) und können selbst performante CPUs aus einfachen Basis-Schaltungen aufbauen.

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

**[1] Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule)

Dieses Modul ist daneben auch in den Studienfächern "Informatik (1. Staatsprüfung für das Lehramt an Gymnasien)" verwendbar.

---

**Studien-/Prüfungsleistungen:**

CPU Entwurf mit VHDL (Prüfungsnummer: 211243)

(englische Bezeichnung: CPU Design with VHDL)

Prüfungsleistung, mündliche Prüfung, Dauer (in Minuten): 30

Anteil an der Berechnung der Modulnote: 100%

Erstablingung: SS 2019, 1. Wdh.: WS 2019/2020

1. Prüfer: Marc Reichenbach (100455)

---

---

**Modulbezeichnung:** Architectures of Supercomputers / Architekturen von Superrechnern (ArchSup) 5 ECTS  
 (Architectures of Supercomputers)

Modulverantwortliche/r: Johannes Hofmann  
 Lehrende: Johannes Hofmann, Dietmar Fey

---

Startsemester: WS 2019/2020	Dauer: 1 Semester	Turnus: jährlich (WS)
Präsenzzeit: 120 Std.	Eigenstudium: 30 Std.	Sprache: Englisch

---

**Lehrveranstaltungen:**

Architectures of Supercomputers / Architekturen von Superrechnern (WS 2019/2020, Vorlesung, 2 SWS, Johannes Hofmann)

Exercises Architectures of Supercomputers / Übungen Architekturen von Superrechnern (WS 2019/2020, Übung, 2 SWS, Johannes Hofmann)

---

**Inhalt:**

- Principles of computer and processor architectures
- Modern processor architectures
- Homogeneous and heterogeneous multi/many-core processors
- Parallel computer architectures
- Classification and principles of coupling parallel computers
- High speed networks in supercomputers
- Examples of supercomputers
- Programming of supercomputers

**Lernziele und Kompetenzen:**

*Fachkompetenz*

*Wissen*

Lernende können die Funktionsweise moderner in Superrechnern eingesetzter Prozessoren wiedergeben. Sie erkennen die besonderen Probleme im Zusammenhang mit dem Energieverbrauch und der Programmierung von Superrechnern.

*Verstehen*

Lernende können die Unterschiede bei der Kopplung paralleler Prozesse darstellen. Sie können Parallelrechner hinsichtlich ihrer Speicheranbindung und den grundlegenden Verarbeitungsprinzipien klassifizieren.

*Anwenden*

Lernenden sind in der Lage ein eigenes technisches oder mathematisches Problem zur Lösung auf einem Supercomputer umzusetzen und auszuführen. Anhand der in der Vorlesung gezeigten Beispiele sind sie in der Lage, Herausforderungen beim Auffinden von Flaschenhälsen zu verallgemeinern und für ihr konkretes Problem anzuwenden.

*Analysieren*

Lernende sind in der Lage, ihre Problemstellungen, z.B. naturwissenschaftliche oder technische Simulationsexperimente, hinsichtlich der Rechen- und Speicheranforderungen für einen Supercomputer geeignet für die Architektur zu charakterisieren.

*Evaluieren (Beurteilen)*

Lernende können mithilfe der aufgezeigten Methodiken zur Leistungsmesung von Parallelrechnern unterschiedliche Rechnerarchitekturen bewerten und für ihre Problemstellung die passende Architektur auswählen.

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

**[1] Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule)

Dieses Modul ist daneben auch in den Studienfächern "Computational Engineering (Master of Science with



Honours)", "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Bachelor of Science)", "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Master of Science)" verwendbar.

---

**Studien-/Prüfungsleistungen:**

Architecture of Supercomputers (Prüfungsnummer: 44601)

(englische Bezeichnung: Architecture of Supercomputers)

Prüfungsleistung, Klausur, Dauer (in Minuten): 90

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Architectures of Supercomputers / Architekturen von Superrechnern
- Exercises Architectures of Supercomputers / Übungen Architekturen von Superrechnern

Prüfungssprache: Englisch

Erstablingung: WS 2019/2020, 1. Wdh.: SS 2020

1. Prüfer: Dietmar Fey (100101)

---

---

**Modulbezeichnung:** **FPGA-Online Basic Course with VHDL (FPGAonline)** **5 ECTS**  
(FPGA-Online Basic Course with VHDL)

Modulverantwortliche/r: Marc Reichenbach

Lehrende: Marc Reichenbach, Dietmar Fey

---

Startsemester: SS 2019	Dauer: 1 Semester	Turnus: halbjährlich (WS+SS)
Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Deutsch

---

**Lehrveranstaltungen:**

FPGA-Online Basic Course with VHDL (SS 2019, Vorlesung mit Übung, 4 SWS, Dietmar Fey et al.)

---

**Inhalt:**

- Einführung in die FPGA-Technologie
- Grundlagen der Schaltungsentwicklung
- Grundlagen VHDL
- Einführung in den Hardware-Designflow
- Schaltungs-Spezifikation mit FSMs
- Kommunikation mit externen Komponenten
- Rechenschaltungen

**Lernziele und Kompetenzen:**

Die Studierenden:

können die in der Veranstaltung vorgestellten Speichertechnologien, für programmierbare Logik, erklären

sind in der Lage diese Speichertechnologien auch hinsichtlich nicht-funktionaler Eigenschaften zu vergleichen

haben den grundlegenden Aufbau, sowie die grundlegende Funktionsweise von FPGAs verstanden und können diese in eigenen Worten erklären

erläutern Anwendungsbereich und Aufbau von Hardwarebeschreibungssprachen am Beispiel VHDL.

erklären die Abbildung von Hardwarebeschreibungssprachen auf programmierbare Logikeinheiten.

sind in der Lage einfache boolesche Funktionen, einfache Automaten sowie einfache arithmetische Schaltungen umzusetzen sowie auf einem FPGA zu erproben.

erfassen zeitliche Aspekte integrierter Schaltungen und erproben diese mit Hilfe von Simulationen.

implementieren einfache Schaltungen zur Nutzung von Peripherie Komponenten.

untersuchen Zusammenhänge nicht-funktionaler Eigenschaften (Timing vs. Area vs. Energy) in integrierten Schaltungen.

**Literatur:**

KESEL, F. and BARTHOLOMÄ, R., 2009. Entwurf von digitalen Schaltungen und Systemen mit HDLs und FPGAs. Oldenbourg Verlag.

ASHENDEN, P. J., 2008. The Designer's Guide to VHDL. Morgan Kaufmann.

REICHARDT, J. and SCHWARZ, B., 2009. VHDL-Synthese. Oldenbourg Verlag.

CHU, P.P., 2008. FPGA Prototyping by VHDL Examples. John Wiley & Sons.

KILTS, S., 2007. Advanced FPGA Design. John Wiley & Sons.

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

[1] **Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule)

---

**Studien-/Prüfungsleistungen:**

FPGA-Online Basic Course with VHDL (Prüfungsnummer: 414335)

Prüfungsleistung, mündliche Prüfung, Dauer (in Minuten): 30

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- FPGA-Online Basic Course with VHDL

Erstablingung: SS 2019, 1. Wdh.: WS 2019/2020

1. Prüfer: Marc Reichenbach (100455)

---

**Organisatorisches:**

Dies ist ein Einsteigerkurs. Keine speziellen Voraussetzungen benötigt.

Dieser Kurs ist ein Online Kurs. Lehrinhalte werden in Form von ausgearbeiteten Inhalten online zur Verfügung gestellt und sollen sich im Selbststudium angeeignet werden. Übungsaufgaben werden ebenfalls online gestellt. Die entwickelten Lösungen können in unserem online-remote Labor getestet werden.

Betreuung während des Kurses findet durch unsere Übungsleiter und Tutoren, in Form von E-Mails, Forum, Chat und Skype statt.

!!! Wichtig: Die Anmeldung zu diesem Kurs erfolgt über die Webseite der Virtuellen Hochschule Bayern (VHB) unter <http://www.vhb.org> !!!

**Bemerkungen:**

Vorlesungsmaterialien Englisch, Kommunikation mit Betreuern Deutsch

---

**Modulbezeichnung:** CPU Entwurf mit VHDL (Schwerpunkt Prozessorentwurf) (CPU-woVHDL) 5 ECTS  
 (CPU Design with VHDL (Focus on CPUs))

Modulverantwortliche/r: Dietmar Fey  
 Lehrende: Marc Reichenbach

---

Startsemester: SS 2019	Dauer: 1 Semester	Turnus: jährlich (SS)
Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Deutsch

---

**Lehrveranstaltungen:**

CPU Entwurf mit VHDL (SS 2019, Vorlesung, 4 SWS, Marc Reichenbach)  
 Übungen zu CPU Entwurf mit VHDL (SS 2019, Übung, 2 SWS, Philipp Holzinger et al.)

---

**Empfohlene Voraussetzungen:**

keine

---

**Inhalt:**

Die Vorlesung wird einen Einblick in die wichtigsten Strukturen eines Rechners und insbesondere die der CPU geben. Wie sind die Register, die ALU, die Caches usw. aufgebaut, wie sind sie miteinander verschaltet? Wie werden Pipelines gebildet?

Um die Vorlesung verstehen und die Übungsaufgaben lösen zu können, werden VHDL-Kenntnisse benötigt. Diese werden in der Vorlesung vermittelt.

Im ersten Teil der Vorlesung geht es darum, aus einfachen Und-, Oder- und Inverter-Gattern größere Strukturen aufzubauen. So werden beispielsweise die Register und die Komponenten der ALU (Addierer, Multiplizierer, Shifter usw.) nach und nach aufgebaut. Aus den Einzelteilen lassen sich dann wiederum die Registerbänke, die ALU, die Speichereinheit usw. zusammensetzen.

Mit guten Rechenwerken allein ist eine CPU noch nicht wirklich schnell. Es müssen schnell genug Instruktionen gelesen und dekodiert sowie die Operanden geholt werden können. Nach dem schnellen Verarbeiten in den Rechenwerken müssen die Ergebnisse dann wiederum schnell in die Zielregister bzw. Ziel-Speicherzellen kopiert werden. Das wichtigste Verfahren, um hierbei Performance zu gewinnen, ist das Pipelining.

Leider gibt es eine Reihe von Bedingungen, die erfüllt sein müssen, dass Pipelining wirklich gut funktioniert. So dürfen keine Daten-Abhängigkeiten im Code vorhanden sein, müssen Sprungziele rechtzeitig bekannt sein und dürfen bestimmte Ressourcen nicht mehrfach gleichzeitig genutzt werden. Mittels Forwarding, Sprungvorhersage und Caches können die Probleme gemildert werden.

Mit den so gewonnenen Fähigkeiten sind dann auch Multi-Core- und Multi-Threading-Architekturen verständlich.

**Lernziele und Kompetenzen:**

Die Studierenden verstehen den inneren Aufbau moderner CPUs (Pipeline, Registerbänke, ALU, Caches, Memory-Management-Unit, Segmentierungseinheit, ...) und können selbst performante CPUs aus einfachen Basis-Schaltungen aufbauen.

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

[1] **Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule)

---

**Studien-/Prüfungsleistungen:**

CPU Entwurf mit VHDL (Schwerpunkt Prozessorentwurf) (Prüfungsnummer: 169383)

(englische Bezeichnung: CPU Design with VHDL (Focus on CPUs))

Prüfungsleistung, mündliche Prüfung, Dauer (in Minuten): 30

Anteil an der Berechnung der Modulnote: 100%

Erstablesung: SS 2019, 1. Wdh.: WS 2019/2020

1. Prüfer: Marc Reichenbach (100455)



---

**Modulbezeichnung:** Heterogene Rechnerarchitekturen Online (HETRON) 5 ECTS  
 (Heterogeneous Computing Architectures Online)

Modulverantwortliche/r: Dietmar Fey, Marc Reichenbach

Lehrende: Marc Reichenbach, Thomas Heller, Johannes Hofmann

---

Startsemester: SS 2019	Dauer: 1 Semester	Turnus: halbjährlich (WS+SS)
Präsenzzeit: k.A. Std.	Eigenstudium: 150 Std.	Sprache: Englisch

---

**Lehrveranstaltungen:**

Heterogene Rechnerarchitekturen Online (SS 2019, Vorlesung, Marc Reichenbach et al.)

**Inhalt:**

Whereas heterogeneous architectures and parallel computing has filled an academic niche in the past it has become now a commodity technique with the rising of multi-core processors and programmable graphic cards. Even FPGAs play a role hereby in a certain extent due to their increasing importance as accelerator hardware what is clearly observable in the scientific community. However, on one side parallel hardware like multi-core and GPUs are now available nearly for everybody and not only for a selected selection of people, who have access to a parallel supercomputer. On the other side the knowledge about programming of this commodity hardware, and we mean here in particular hardware-orientated programming in order to squeeze out all offered GFlops and TFlops of such hardware, is still missing as well as the knowledge about the architecture details. To overcome this lack we offer this course HETRON.

The e-learning course HETRON for the exploitation of parallel and heterogeneous computer architectures) focuses on two main topics which are closely related to each other. This concerns on one side the benefits of using different kinds of multi-core processors and parallel architectures built-up on base of these multicore processors. These architectures differ among each other in the number and in the complexity of its single processing nodes. We distinguish between systems consisting of a large number of simpler, so called fine-grained, processor cores vs. systems consisting of a smaller number of more complex, so called coarse-grained, processor cores. On the other side we lay our focus on that we want to do with these different heterogeneous parallel architectures, namely the execution of parallel programs. Of course this requires the use of parallel programming languages and environments, like CUDA or OpenMP. However, besides these questions of using the right syntax and the right compiler switches to optimize a parallel program it is a pre-requisite to understand how parallel computing really works. This refers (i) to the comprehension which basic mechanisms of parallel computing exist, (ii) where are the limits of getting more performance with parallel computing and (iii) in what context stand these mechanisms to heterogeneous architectures. In other words it handles the question which architecture is the best one for a certain parallelization technique. To teach these three topics, is one main goal we pursuit with the course HETRON, and of course, this more fundamental basics of heterogeneous and parallel computing have to be proven by means of concrete application examples to deepen the acquired knowledge about heterogeneous architectures and parallel computing principles.

**Lernziele und Kompetenzen:**

Die Studierenden ...

...verstehen die Notwendigkeit sowie grundlegende Anwendungsfälle für heterogene Rechnerarchitekturen.

...können den grundlegenden Aufbau und das Zusammenspiel der Komponenten heterogener Rechnerarchitekturen erklären. ...erläutern grundsätzliche Parallelisierungsprinzipien wie Amdahls Law, High-Performance- und High-Throughput-Computing sowie Parallelisierungsstrategien. ...können einfache Programme mit Hilfe der vermittelten Parallelisierungsprinzipien (Amdahls Law, High-Performance- und High-Throughput-Computing) analysieren und entsprechende Parallelisierungsstrategien entwickeln.

...erklären den Aufbau sowie Stärken und Schwächen von verschiedenen Architekturen wie CPUs, GPUs, Many-Core Prozessoren und FPGAs.

...implementieren ausgewählte Anwendungsbeispiele (SHA256 Algorithmus, Ising-Modell und Fast-Fourier-Transformation) auf oben genannte Architekturen.

...erforschen und bewerten verschiedener Parallelisierungstechniken in Abhängigkeit der Anwendung

und der Architektur.

...erläutern die Grundlagen des Grid- und Cloud-Computings

...sind in der Lage parallele Berechnungen (SHA256) im Grid umzusetzen.

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

**[1] Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule)

---

**Studien-/Prüfungsleistungen:**

Heterogene Rechnerarchitekturen Online (Prüfungsnummer: 275245)

(englische Bezeichnung: Heterogeneous Computing Architectures Online)

Prüfungsleistung, mündliche Prüfung, Dauer (in Minuten): 30

Anteil an der Berechnung der Modulnote: 100% Prüfungssprache: Deutsch oder Englisch

Erstablingung: SS 2019, 1. Wdh.: WS 2019/2020

1. Prüfer: Marc Reichenbach (100455)

---

---

**Modulbezeichnung:** CPU Entwurf mit VHDL (Schwerpunkt VHDL) (CPU-VHDL) 5 ECTS  
 (CPU Design with VHDL (Focus on VHDL))

Modulverantwortliche/r: Dietmar Fey  
 Lehrende: Marc Reichenbach

---

Startsemester: SS 2019	Dauer: 1 Semester	Turnus: jährlich (SS)
Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Deutsch

---

**Lehrveranstaltungen:**

CPU Entwurf mit VHDL (SS 2019, Vorlesung, 4 SWS, Marc Reichenbach)  
 Übungen zu CPU Entwurf mit VHDL (SS 2019, Übung, 2 SWS, Philipp Holzinger et al.)

---

**Empfohlene Voraussetzungen:**

keine

---

**Inhalt:**

Die Vorlesung wird einen Einblick in die wichtigsten Strukturen eines Rechners und insbesondere die der CPU geben. Wie sind die Register, die ALU, die Caches usw. aufgebaut, wie sind sie miteinander verschaltet? Wie werden Pipelines gebildet?

Um die Vorlesung verstehen und die Übungsaufgaben lösen zu können, werden VHDL-Kenntnisse benötigt. Diese werden in der Vorlesung vermittelt.

Im ersten Teil der Vorlesung geht es darum, aus einfachen Und-, Oder- und Inverter-Gattern größere Strukturen aufzubauen. So werden beispielsweise die Register und die Komponenten der ALU (Addierer, Multiplizierer, Shifter usw.) nach und nach aufgebaut. Aus den Einzelteilen lassen sich dann wiederum die Registerbänke, die ALU, die Speichereinheit usw. zusammensetzen.

Mit guten Rechenwerken allein ist eine CPU noch nicht wirklich schnell. Es müssen schnell genug Instruktionen gelesen und dekodiert sowie die Operanden geholt werden können. Nach dem schnellen Verarbeiten in den Rechenwerken müssen die Ergebnisse dann wiederum schnell in die Zielregister bzw. Ziel-Speicherzellen kopiert werden. Das wichtigste Verfahren, um hierbei Performance zu gewinnen, ist das Pipelining.

Leider gibt es eine Reihe von Bedingungen, die erfüllt sein müssen, dass Pipelining wirklich gut funktioniert. So dürfen keine Daten-Abhängigkeiten im Code vorhanden sein, müssen Sprungziele rechtzeitig bekannt sein und dürfen bestimmte Ressourcen nicht mehrfach gleichzeitig genutzt werden. Mittels Forwarding, Sprungvorhersage und Caches können die Probleme gemildert werden.

Mit den so gewonnenen Fähigkeiten sind dann auch Multi-Core- und Multi-Threading-Architekturen verständlich.

**Lernziele und Kompetenzen:**

Die Studierenden verstehen den inneren Aufbau moderner CPUs (Pipeline, Registerbänke, ALU, Caches, Memory-Management-Unit, Segmentierungseinheit, ...) und können selbst performante CPUs aus einfachen Basis-Schaltungen aufbauen.

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

[1] **Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule)

---

**Studien-/Prüfungsleistungen:**

CPU Entwurf mit VHDL (Schwerpunkt VHDL) (Prüfungsnummer: 436348)

(englische Bezeichnung: CPU Design with VHDL (Focus on VHDL))

Prüfungsleistung, mündliche Prüfung, Dauer (in Minuten): 30

Anteil an der Berechnung der Modulnote: 100%

Erstablingung: SS 2019, 1. Wdh.: WS 2019/2020

1. Prüfer: Marc Reichenbach (100455)





---

**Modulbezeichnung:** Programmierung und Architekturen von Cluster-Rechnern (PS-PAC) 7.5 ECTS  
 (Programming and Architecture of Compute Clusters)

Modulverantwortliche/r: Michael Philippsen  
 Lehrende: Dietmar Fey, Thorsten Blaß

---

Startsemester: SS 2019	Dauer: 1 Semester	Turnus: jährlich (SS)
Präsenzzeit: 90 Std.	Eigenstudium: 135 Std.	Sprache: Deutsch oder Englisch

---

**Lehrveranstaltungen:**

Programmierung und Architekturen von Cluster-Rechnern (SS 2019, Vorlesung, 4 SWS, Thorsten Blaß et al.)  
 Übung zu Programmierung und Architekturen von Cluster-Rechnern (SS 2019, Übung, 2 SWS, Thorsten Blaß)

---

**Inhalt:**

There are many problems where a single computer cannot compute a solution in reasonable time. In this case we must use multiple computers working in concert (a cluster). If a researcher wants to use a cluster at another location (because the current cluster is 'full'), then the researcher needs a way to safely run jobs in some other location. This is 'grid computing'. Similarly, a smartphone may not suffice, and we must let the computation run elsewhere (potentially in parallel). If many smartphones need remote services, we need many remote machines, which is again a cluster. This is called 'cloud-computing'. All of the above are topics of this lecture.

In this lecture we will:

- examine use cases
- examine ways to use a cluster, cloud, or grid.
- examine how to optimize applications for clusters (load balancing, message passing optimizations)
- examine message passing libraries, both from how to use them to how they are implemented.
- examine how high-end networking hardware works and how it is programmed
- study various cloud and grid computing systems
- look into the future where clouds and clusters are augmented with compute accelerators (exascale computing).

In the exercises you will:

- program a scalable application that runs on a cluster (there are a number of projects to choose from).
- learn how to achieve scalability,
- learn how to use a cluster practically, and
- learn how to interface with a cluster.

**Lernziele und Kompetenzen:**

The students

- program a scalable application that runs on a cluster
- optimize their implementation in order to achieve scalability
- use a cluster practically and configure it to satisfy their needs

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

**[1] Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule)

Dieses Modul ist daneben auch in den Studienfächern "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Bachelor of Science)", "Informatik (Master of Science)", "Information and Communication Technology (Master of Science)", "Informations- und Kommunikationstechnik (Master of Science)", "Mathematik (Bachelor of Science)" verwendbar.

---

**Studien-/Prüfungsleistungen:**

Programmierung und Architekturen von Cluster-Rechnern (Prüfungsnummer: 399289)

(englische Bezeichnung: Programming and Architecture of Compute Clusters)

Prüfungsleistung, mündliche Prüfung, Dauer (in Minuten): 30

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Programmierung und Architekturen von Cluster-Rechnern
- Übung zu Programmierung und Architekturen von Cluster-Rechnern

Erstablesung: SS 2019, 1. Wdh.: WS 2019/2020

1. Prüfer: Michael Philippsen (100113)

1. Prüfer: Dietmar Fey (100101)

---

**Organisatorisches:**

The time and date of the exercise class will be determined later in cooperation. No exercise class the first two weeks, however. The choice of language used in the class (English or German) is by request of the students.

---

**Modulbezeichnung:** **FPGA-Online Basic Course with VHDL (FPGAonline)** **5 ECTS**  
 (FPGA-online basic course with VHDL)

Modulverantwortliche/r: Dietmar Fey

Lehrende: Dietmar Fey, Michael Schmidt, Marc Reichenbach

---

Startsemester: WS 2019/2020	Dauer: 1 Semester	Turnus: halbjährlich (WS+SS)
Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Deutsch

---

**Lehrveranstaltungen:**

FPGA-Online Basic Course with VHDL (WS 2019/2020, Vorlesung mit Übung, 4 SWS, Dietmar Fey et al.)

---

**Inhalt:**

- Einführung in die FPGA-Technologie
- Grundlagen der Schaltungsentwicklung
- Grundlagen VHDL
- Einführung in den Hardware-Designflow
- Schaltungs-Spezifikation mit FSMs
- Kommunikation mit externen Komponenten
- Rechenschaltungen

**Lernziele und Kompetenzen:**

Die Studierenden:

können die in der Veranstaltung vorgestellten Speichertechnologien, für programmierbare Logik, erklären

sind in der Lage diese Speichertechnologien auch hinsichtlich nicht-funktionaler Eigenschaften zu vergleichen

haben den grundlegenden Aufbau, sowie die grundlegende Funktionsweise von FPGAs verstanden und können diese in eigenen Worten erklären

erläutern Anwendungsbereich und Aufbau von Hardwarebeschreibungssprachen am Beispiel VHDL.

erklären die Abbildung von Hardwarebeschreibungssprachen auf programmierbare Logikeinheiten.

sind in der Lage einfache boolesche Funktionen, einfache Automaten sowie einfache arithmetische Schaltungen umzusetzen sowie auf einem FPGA zu erproben.

erfassen zeitliche Aspekte integrierter Schaltungen und erproben diese mit Hilfe von Simulationen.

implementieren einfache Schaltungen zur Nutzung von Peripherie Komponenten.

untersuchen Zusammenhänge nicht-funktionaler Eigenschaften (Timing vs. Area vs. Energy) in integrierten Schaltungen.

**Literatur:**

KESEL, F. and BARTHOLOMÄ, R., 2009. Entwurf von digitalen Schaltungen und Systemen mit HDLs und FPGAs. Oldenbourg Verlag.

ASHENDEN, P. J., 2008. The Designer's Guide to VHDL. Morgan Kaufmann.

REICHARDT, J. and SCHWARZ, B., 2009. VHDL-Synthese. Oldenbourg Verlag.

CHU, P.P., 2008. FPGA Prototyping by VHDL Examples. John Wiley & Sons.

KILTS, S., 2007. Advanced FPGA Design. John Wiley & Sons.

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

**[1] Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule | Vertiefungsrichtung Rechnerarchitektur)

Dieses Modul ist daneben auch in den Studienfächern "Mathematik (Bachelor of Science)" verwendbar.

---

**Studien-/Prüfungsleistungen:**

FPGA-Online Basic Course with VHDL (Prüfungsnummer: 520692)

(englische Bezeichnung: FPGA-online basic course with VHDL)

Prüfungsleistung, mündliche Prüfung, Dauer (in Minuten): 30

Anteil an der Berechnung der Modulnote: 100%

Erstablingung: WS 2019/2020, 1. Wdh.: SS 2020

1. Prüfer: Marc Reichenbach (100455)

---

**Organisatorisches:**

Dies ist ein Einsteigerkurs. Keine speziellen Voraussetzungen benötigt.

Dieser Kurs ist ein Online Kurs. Lehrinhalte werden in Form von ausgearbeiteten Inhalten online zur Verfügung gestellt und sollen sich im Selbststudium angeeignet werden. Übungsaufgaben werden ebenfalls online gestellt. Die entwickelten Lösungen können in unserem online-remote Labor getestet werden.

Betreuung während des Kurses findet durch unsere Übungsleiter und Tutoren, in Form von E-Mails, Forum, Chat und Skype statt.

!!!Wichtig: Die Anmeldung zu diesem Kurs erfolgt über die Webseite der Virtuellen Hochschule Bayern (VHB) unter <http://www.vhb.org> ab 01.10.2011 bis 17.10.2011!!!

**Bemerkungen:**

Vorlesungsmaterialien Englisch, Kommunikation mit Betreuern Deutsch

---

**Modulbezeichnung:** Heterogene Rechnerarchitekturen Online (HETRON) **5 ECTS**  
 (Heterogeneous Computing Architectures Online)

Modulverantwortliche/r: Dietmar Fey, Marc Reichenbach

Lehrende: Marc Reichenbach, Thomas Heller, Johannes Hofmann

---

Startsemester: WS 2019/2020	Dauer: 1 Semester	Turnus: halbjährlich (WS+SS)
Präsenzzeit: k.A. Std.	Eigenstudium: 150 Std.	Sprache: Englisch

---

**Lehrveranstaltungen:**

Heterogene Rechnerarchitekturen Online (WS 2019/2020, Vorlesung, Marc Reichenbach et al.)

---

**Inhalt:**

Whereas heterogeneous architectures and parallel computing has filled an academic niche in the past it has become now a commodity technique with the rising of multi-core processors and programmable graphic cards. Even FPGAs play a role hereby in a certain extent due to their increasing importance as accelerator hardware what is clearly observable in the scientific community. However, on one side parallel hardware like multi-core and GPUs are now available nearly for everybody and not only for a selected selection of people, who have access to a parallel supercomputer. On the other side the knowledge about programming of this commodity hardware, and we mean here in particular hardware-orientated programming in order to squeeze out all offered GFlops and TFlops of such hardware, is still missing as well as the knowledge about the architecture details. To overcome this lack we offer this course HETRON.

The e-learning course HETRON for the exploitation of parallel and heterogeneous computer architectures) focuses on two main topics which are closely related to each other. This concerns on one side the benefits of using different kinds of multi-core processors and parallel architectures built-up on base of these multicore processors. These architectures differ among each other in the number and in the complexity of its single processing nodes. We distinguish between systems consisting of a large number of simpler, so called fine-grained, processor cores vs. systems consisting of a smaller number of more complex, so called coarse-grained, processor cores. On the other side we lay our focus on that we want to do with these different heterogeneous parallel architectures, namely the execution of parallel programs. Of course this requires the use of parallel programming languages and environments, like CUDA or OpenMP. However, besides these questions of using the right syntax and the right compiler switches to optimize a parallel program it is a pre-requisite to understand how parallel computing really works. This refers (i) to the comprehension which basic mechanisms of parallel computing exist, (ii) where are the limits of getting more performance with parallel computing and (iii) in what context stand these mechanisms to heterogeneous architectures. In other words it handles the question which architecture is the best one for a certain parallelization technique. To teach these three topics, is one main goal we pursuit with the course HETRON, and of course, this more fundamental basics of heterogeneous and parallel computing have to be proven by means of concrete application examples to deepen the acquired knowledge about heterogeneous architectures and parallel computing principles.

**Lernziele und Kompetenzen:**

Die Studierenden ...

...verstehen die Notwendigkeit sowie grundlegende Anwendungsfälle für heterogene Rechnerarchitekturen.

...können den grundlegenden Aufbau und das Zusammenspiel der Komponenten heterogener Rechnerarchitekturen erklären. ...erläutern grundsätzliche Parallelisierungsprinzipien wie Amdahls Law, High-Performance- und High-Throughput-Computing sowie Parallelisierungsstrategien. ...können einfache Programme mit Hilfe der vermittelten Parallelisierungsprinzipien (Amdahls Law, High-Performance- und High-Throughput-Computing) analysieren und entsprechende Parallelisierungsstrategien entwickeln.

...erklären den Aufbau sowie Stärken und Schwächen von verschiedenen Architekturen wie CPUs, GPUs, Many-Core Prozessoren und FPGAs.

...implementieren ausgewählte Anwendungsbeispiele (SHA256 Algorithmus, Ising-Modell und Fast-Fourier-Transformation) auf oben genannte Architekturen.

...erforschen und bewerten verschiedener Parallelisierungstechniken in Abhängigkeit der Anwendung

und der Architektur.

...erläutern die Grundlagen des Grid- und Cloud-Computings

...sind in der Lage parallele Berechnungen (SHA256) im Grid umzusetzen.

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

**[1] Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule | Vertiefungsrichtung Rechnerarchitektur)

Dieses Modul ist daneben auch in den Studienfächern "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Bachelor of Science)", "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Master of Science)", "Informatik (Master of Science)", "Mathematik (Bachelor of Science)", "Medizintechnik (Master of Science)" verwendbar.

---

**Studien-/Prüfungsleistungen:**

Heterogene Rechnerarchitekturen Online (Prüfungsnummer: 275245)

(englische Bezeichnung: Heterogeneous Computing Architectures Online)

Prüfungsleistung, mündliche Prüfung, Dauer (in Minuten): 30

Anteil an der Berechnung der Modulnote: 100% Prüfungssprache: Deutsch oder Englisch

Erstablesung: WS 2019/2020, 1. Wdh.: SS 2020

1. Prüfer: Dietmar Fey (100101)

---

---

**Modulbezeichnung:** Rechnerarchitektur (Vorlesung mit Übung und Rechnerübung) (RA) 7.5 ECTS

(Computer Architecture (Lecture with Exercise and practical course))

Modulverantwortliche/r: Dietmar Fey

Lehrende: Dietmar Fey

---

Startsemester: WS 2019/2020

Dauer: 1 Semester

Turnus: jährlich (WS)

Präsenzzeit: 90 Std.

Eigenstudium: 135 Std.

Sprache: Deutsch

---

### Lehrveranstaltungen:

Rechnerarchitektur (WS 2019/2020, Vorlesung, 2 SWS, Dietmar Fey)

Übungen zu Rechnerarchitektur (WS 2019/2020, Übung, 2 SWS, Sebastian Rachuj et al.)

Rechnerübungen zu Rechnerarchitektur (WS 2019/2020, Übung, 2 SWS, Christian Widerspick)

---

### Inhalt:

Die Vorlesung baut auf die in den Grundlagen der Rechnerarchitektur und -organisation vermittelten Inhalte auf und setzt diese mit weiterführenden Themen fort. Es werden zunächst grundlegende fortgeschrittene Techniken bei Pipelineverarbeitung und Cachezugriffen in modernen Prozessoren und Parallelrechnern behandelt. Ferner wird die Architektur von Spezialprozessoren, z.B. DSPs und Embedded Prozessoren behandelt. Es wird aufgezeigt, wie diese Techniken in konkreten Architekturen (Intel Nehalem, GPGPU, Cell BE, TMS320 DSP, Embedded Prozessor ZPU) verwendet werden. Zur Vorlesung werden eine Tafel- und eine Rechnerübung angeboten, durch deren erfolgreiche Beteiligung abgestuft mit der Vorlesung 5 bzw. 7,5 ECTS erworben werden können. In den Tafelübungen werden die in der Vorlesung vermittelten Techniken durch zu lösende Aufgaben vertieft. In der Rechnerübung soll u.a. ein einfacher Vielkern-Prozessor auf Basis des ZPU-Prozessors mit Simulationswerkzeugen aufgebaut werden. Im Einzelnen werden folgende Themen behandelt:

- Organisationsaspekte von CISC und RISC-Prozessoren
- Behandlung von Hazards in Pipelines
- Fortgeschrittene Techniken der dynamischen Sprungvorhersage
- Fortgeschrittenen Cachetechniken, Cache-Kohärenz
- Ausnutzen von Cacheeffekten
- Architekturen von Digitalen Signalprozessoren
- Architekturen homogener und heterogener Multikern-Prozessoren (Intel Corei7, Nvidia GPUs, Cell BE)
- Architektur von Parallelrechnern (Clusterrechner, Superrechner)
- Effiziente Hardware-nahe Programmierung von Multikern-Prozessoren (OpenMP, SSE, CUDA, OpenCL)
- Leistungsmodellierung und -analyse von Multikern-Prozessoren (Roofline-Modell)

### Lernziele und Kompetenzen:

#### *Fachkompetenz*

##### *Wissen*

Lernende können Wissen abrufen und wiedergeben. Sie können konkrete Einzelheiten wie Begriffe, Definitionen, Fakten, und Abläufe in einem Prozessor darlegen.

##### *Verstehen*

Lernende können Beispiele für Rechnerarchitekturen anführen, sie sind in der Lage, Schaubilder von Prozessoren zu interpretieren und die Abläufe in eigenen Worten zu beschreiben.

##### *Anwenden*

Lernende können beim Erstellen eigener Programme durch Transfer des Wissens über Interna von Prozessorarchitekturen Optimierungen hinsichtlich des Laufzeitverhaltens vornehmen.

##### *Analysieren*

Lernende können zwischen verschiedenen Varianten von Lösungen einer Prozessorarchitektur klassifizieren, die Gründe für durchgeführte Entwurfsentscheidungen erschließen, Unterscheide gegenüberstellen und gegeneinander bewerten.



*Lern- bzw. Methodenkompetenz*

Lernende erwerben die Fähigkeit selbstständig Testprogramme zum Bewerten der Leistungsfähigkeit eines Prozessors zu erstellen.

**Literatur:**

- Patterson/Hennessy: Computer Organization und Design
- Hennessy/Patterson: Computer Architecture - A Quantitative Approach
- Stallings: Computer Organization and Architecture
- Märtin: Rechnerarchitekturen

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

**[1] Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule | Vertiefungsrichtung Rechnerarchitektur)

Dieses Modul ist daneben auch in den Studienfächern "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Bachelor of Science)", "Informatik (Master of Science)", "Information and Communication Technology (Master of Science)", "Informations- und Kommunikationstechnik (Master of Science)", "Mathematik (Bachelor of Science)" verwendbar.

---

**Studien-/Prüfungsleistungen:**

Rechnerarchitektur (Vorlesung mit Übung und Rechnerübung) (Prüfungsnummer: 333815)

Prüfungsleistung, Portfolio

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Rechnerarchitektur
- Übungen zu Rechnerarchitektur
- Rechnerübungen zu Rechnerarchitektur

weitere Erläuterungen:

30-minütige mündliche Prüfung + Teilnahme an der Tafelübung und Rechnerübung (verpflichtend)

Erstablingung: WS 2019/2020, 1. Wdh.: SS 2020

1. Prüfer: Dietmar Fey (100101)

---

**Organisatorisches:**

Schein durch Kolloquium

---

**Modulbezeichnung: Rechnerarchitektur (RA)** **5 ECTS**  
 (Computer Architecture)

Modulverantwortliche/r: Dietmar Fey

Lehrende: Dietmar Fey

Startsemester: WS 2019/2020

Dauer: 1 Semester

Turnus: jährlich (WS)

Präsenzzeit: 60 Std.

Eigenstudium: 90 Std.

Sprache: Deutsch

**Lehrveranstaltungen:**

Rechnerarchitektur (WS 2019/2020, Vorlesung, 2 SWS, Dietmar Fey)

Übungen zu Rechnerarchitektur (WS 2019/2020, Übung, 2 SWS, Sebastian Rachuj et al.)

Rechnerübungen zu Rechnerarchitektur (WS 2019/2020, Übung, 2 SWS, Christian Widerspick)

**Inhalt:**

Die Vorlesung baut auf die in den Grundlagen der Rechnerarchitektur und -organisation vermittelten Inhalte auf und setzt diese mit weiterführenden Themen fort. Es werden zunächst grundlegende fortgeschrittene Techniken bei Pipelineverarbeitung und Cachezugriffen in modernen Prozessoren und Parallelrechnern behandelt. Ferner wird die Architektur von Spezialprozessoren, z.B. DSPs und Embedded Prozessoren behandelt. Es wird aufgezeigt, wie diese Techniken in konkreten Architekturen (Intel Nehalem, GPGPU, Cell BE, TMS320 DSP, Embedded Prozessor ZPU) verwendet werden. Zur Vorlesung werden eine Tafel- und eine Rechnerübung angeboten, durch deren erfolgreiche Beteiligung abgestuft mit der Vorlesung 5 bzw. 7,5 ECTS erworben werden können. In den Tafelübungen werden die in der Vorlesung vermittelten Techniken durch zu lösende Aufgaben vertieft. In der Rechnerübung soll u.a. ein einfacher Vielkern-Prozessor auf Basis des ZPU-Prozessors mit Simulationswerkzeugen aufgebaut werden. Im Einzelnen werden folgende Themen behandelt:

- Organisationsaspekte von CISC und RISC-Prozessoren
- Behandlung von Hazards in Pipelines
- Fortgeschrittene Techniken der dynamischen Sprungvorhersage
- Fortgeschrittenen Cachetechniken, Cache-Kohärenz
- Ausnutzen von Cacheeffekten
- Architekturen von Digitalen Signalprozessoren
- Architekturen homogener und heterogener Multikern-Prozessoren (Intel Corei7, Nvidia GPUs, Cell BE)
- Architektur von Parallelrechnern (Clusterrechner, Superrechner)
- Effiziente Hardware-nahe Programmierung von Multikern-Prozessoren (OpenMP, SSE, CUDA, OpenCL)
- Leistungsmodellierung und -analyse von Multikern-Prozessoren (Roofline-Modell)

**Lernziele und Kompetenzen:**
*Fachkompetenz*
*Wissen*

Lernende können Wissen abrufen und wiedergeben. Sie kennen konkrete Einzelheiten wie Begriffe, Definitionen, Fakten, und Abläufe in einem Prozessor darlegen.

*Verstehen*

Lernende können Beispiele für Rechnerarchitekturen anführen, sie sind in der Lage, Schaubilder von Prozessoren zu interpretieren und die Abläufe in eigenen Worten zu beschreiben.

*Anwenden*

Lernende können beim Erstellen eigener Programme durch Transfer des Wissens über Interna von Prozessorarchitekturen Optimierungen hinsichtlich des Laufzeitverhaltens vornehmen.

*Analysieren*

Lernende können zwischen verschiedenen Varianten von Lösungen einer Prozessorarchitektur klassifizieren, die Gründe für durchgeführte Entwurfsentscheidungen erschließen, Unterscheide gegenüberstellen und gegeneinander bewerten.

*Lern- bzw. Methodenkompetenz*

Lernende erwerben die Fähigkeit selbstständig Testprogramme zum Bewerten der Leistungsfähigkeit eines Prozessors zu erstellen.

**Literatur:**

- Patterson/Hennessy: Computer Organization und Design
- Hennessy/Patterson: Computer Architecture - A Quantitative Approach
- Stallings: Computer Organization and Architecture
- Märtin: Rechnerarchitekturen

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

**[1] Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule | Vertiefungsrichtung Rechnerarchitektur)

Dieses Modul ist daneben auch in den Studienfächern "Computational Engineering (Master of Science with Honours)", "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Bachelor of Science)", "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Master of Science)", "Informatik (Master of Science)", "Information and Communication Technology (Master of Science)", "Informations- und Kommunikationstechnik (Bachelor of Science)", "Informations- und Kommunikationstechnik (Master of Science)", "Mathematik (Bachelor of Science)" verwendbar.

---

**Studien-/Prüfungsleistungen:**

Rechnerarchitektur (Vorlesung mit Übung) (Prüfungsnummer: 798810)

Prüfungsleistung, Portfolio

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Rechnerarchitektur
- Rechnerübungen zu Rechnerarchitektur

weitere Erläuterungen:

30-minütige mündliche Prüfung + Teilnahme an der Tafelübung bzw. Rechnerübung (verpflichtend)

Erstablingung: WS 2019/2020, 1. Wdh.: SS 2020

1. Prüfer: Dietmar Fey (100101)

---

---

**Modulbezeichnung:** Virtuelle Maschinen - V+EÜ (VM-VEU) 7.5 ECTS  
 (Virtual Machines - V+EÜ)

Modulverantwortliche/r: Volkmar Sieh  
 Lehrende: Volkmar Sieh

---

Startsemester: WS 2019/2020	Dauer: 1 Semester	Turnus: jährlich (WS)
Präsenzzeit: 90 Std.	Eigenstudium: 135 Std.	Sprache: Deutsch

---

**Lehrveranstaltungen:**

Virtuelle Maschinen (WS 2019/2020, Vorlesung, 2 SWS, Volkmar Sieh)  
 Übungen zu Virtuelle Maschinen (WS 2019/2020, Übung, 2 SWS, Volkmar Sieh)  
 Erweiterte Übungen zu Virtuelle Maschinen (WS 2019/2020, Übung, 2 SWS, Volkmar Sieh)

---

**Inhalt:**

Vorge stellt werden verschiedene Virtualisierungs-Ansätze:

- Emulation
- Just-In-Time-Compiler
- Para-Virtualisierung
- Bibliotheks-basierte Virtualisierung
- OS-Virtualisierung

**Lernziele und Kompetenzen:**

Studierende, die das Modul erfolgreich abgeschlossen haben:

- erläutern verschiedene Motivationen für den Einsatz von VMs
- unterscheiden verschiedene VMs
- klassifizieren verschiedene Ziele unterschiedlicher VMs (z.B. Performance, Konfigurierbarkeit, Genauigkeit, ...)
- hinterfragen verschiedene Simulationansätze für MMUs
- erstellen virtuelle Komponenten und Busse
- strukturieren Callbacks und entsprechendes Forwarding und Caching
- unterscheiden zwischen Architektur, Chip und Komponente
- klassifizieren unterschiedliche Just-In-Time-Compiler-Ansätze
- erzeugen JIT Code aus vorgefertigten Code-Teilen
- bewerten unterschiedliche JIT-Code-Optimierungen
- erläutern Probleme bei der JIT-Code-Invalidierung
- nennen JIT Probleme mit Exceptions/Interrupts sowie berechnete Sprünge und Return-Instruktionen
- unterscheiden verschiedene JIT Cache-Verwaltungen
- beschreiben Möglichkeiten der Fehlerinjektion durch VMs
- entwickeln ein an JIT angepasstes virtuelles "Hardware"-Design
- erläutern die Java-VM Instruktionssatz-Architektur
- nutzen Hardware-basierte Virtualisierung
- entwickeln Verfahren zum Ausfiltern bestimmter Befehle
- erläutern Probleme der Speicherverwaltung bei HW-basierter Virtualisierung
- nutzen User-Mode-Emulation zur Paravirtualisierung
- diskutieren Möglichkeiten von Debuggern für die Umleitung von System-Calls und die Ausfilterung von Befehlen
- nutzen einen Hypervisor zur Paravirtualisierung
- unterscheiden verschiedene Ansätze zur Geräteverwaltung in paravirtualisierten Systemen
- erläutern Betriebssystem-basierte Virtualisierung
- entwickeln unterschiedliche Bibliotheks-basierte Virtualisierungen
- erläutern Probleme beim Speicher-Layout bei Bibliotheks-basierte Virtualisierung
- konzipieren Personalities für Bibliotheks-basierte Virtualisierungen
- beurteilen Probleme bei der korrekten Zeit-Simulation
- nennen Ideen für die dynamische Anpassung der Zeit-Simulation

- klassifizieren bekannte VMs (z.B. VICE, FAUmachine, QEMU, Bochs, JVM, KVM, User-Mode-Linux, Xen, VServer, Wine)
- diskutieren in der Gruppe Vor- und Nachteile von bestimmten VM-Ansätzen
- entwickeln selbst CPU-Emulationen
- entwickeln selbst Geräte-Emulationen
- verteilen Implementierungsaufgaben in ihrer Gruppe
- erstellen Zeitpläne für Implementierungen für sich und ihre Gruppe

---

### Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

#### [1] Informatik (Bachelor of Science)

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule | Vertiefungsrichtung Rechnerarchitektur)

#### [2] Informatik (Bachelor of Science)

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule | Vertiefungsrichtung Verteilte Systeme und Betriebssysteme)

Dieses Modul ist daneben auch in den Studienfächern "Informatik (Master of Science)", "Informations- und Kommunikationstechnik (Master of Science)", "Mathematik (Bachelor of Science)" verwendbar.

---

### Studien-/Prüfungsleistungen:

Virtuelle Maschinen (Vorlesung mit Übung und Laborübung) (Prüfungsnummer: 462793)

Prüfungsleistung, Portfolio

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Virtuelle Maschinen
- Übungen zu Virtuelle Maschinen
- Erweiterte Übungen zu Virtuelle Maschinen

weitere Erläuterungen:

30-minütige mündliche Prüfung + erfolgreiche Ausführung von Übungsaufgaben (verpflichtend)

Erstablegung: WS 2019/2020, 1. Wdh.: SS 2020

1. Prüfer: Volkmar Sieh (100548)

---

---

**Modulbezeichnung:** Virtuelle Maschinen - V+Ü (VM-VU) 5 ECTS  
 (Virtual Machines - V+Ü)

Modulverantwortliche/r: Volkmar Sieh  
 Lehrende: Volkmar Sieh

---

Startsemester: WS 2019/2020	Dauer: 1 Semester	Turnus: jährlich (WS)
Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Deutsch

---

**Lehrveranstaltungen:**

Virtuelle Maschinen (WS 2019/2020, Vorlesung, 2 SWS, Volkmar Sieh)  
 Übungen zu Virtuelle Maschinen (WS 2019/2020, Übung, 2 SWS, Volkmar Sieh)

---

**Inhalt:**

Vorge stellt werden verschiedene Virtualisierungs-Ansätze:

- Emulation
- Just-In-Time-Compiler
- Para-Virtualisierung
- Bibliotheks-basierte Virtualisierung
- OS-Virtualisierung

**Lernziele und Kompetenzen:**

Studierende, die das Modul erfolgreich abgeschlossen haben:

- erläutern verschiedene Motivationen für den Einsatz von VMs
- unterscheiden verschiedene VMs
- klassifizieren verschiedene Ziele unterschiedlicher VMs (z.B. Performance, Konfigurierbarkeit, Genauigkeit, ...)
- hinterfragen verschiedene Simulationansätze für MMUs
- erstellen virtuelle Komponenten und Busse
- strukturieren Callbacks und entsprechendes Forwarding und Caching
- unterscheiden zwischen Architektur, Chip und Komponente
- klassifizieren unterschiedliche Just-In-Time-Compiler-Ansätze
- erzeugen JIT Code aus vorgefertigten Code-Teilen
- bewerten unterschiedliche JIT-Code-Optimierungen
- erläutern Probleme bei der JIT-Code-Invalidierung
- nennen JIT Probleme mit Exceptions/Interrupts sowie berechnete Sprünge und Return-Instruktionen
- unterscheiden verschiedene JIT Cache-Verwaltungen
- beschreiben Möglichkeiten der Fehlerinjektion durch VMs
- entwickeln ein an JIT angepasstes virtuelles "Hardware"-Design
- erläutern die Java-VM Instruktionssatz-Architektur
- nutzen Hardware-basierte Virtualisierung
- entwickeln Verfahren zum Ausfiltern bestimmter Befehle
- erläutern Probleme der Speicherverwaltung bei HW-basierter Virtualisierung
- nutzen User-Mode-Emulation zur Paravirtualisierung
- diskutieren Möglichkeiten von Debuggern für die Umleitung von System-Calls und die Ausfilterung von Befehlen
- nutzen einen Hypervisor zur Paravirtualisierung
- unterscheiden verschiedene Ansätze zur Geräteverwaltung in paravirtualisierten Systemen
- erläutern Betriebssystem-basierte Virtualisierung
- entwickeln unterschiedliche Bibliotheks-basierte Virtualisierungen
- erläutern Probleme beim Speicher-Layout bei Bibliotheks-basierte Virtualisierung
- konzipieren Personalities für Bibliotheks-basierte Virtualisierungen
- beurteilen Probleme bei der korrekten Zeit-Simulation
- nennen Ideen für die dynamische Anpassung der Zeit-Simulation

- klassifizieren bekannte VMs (z.B. VICE, FAUmachine, QEMU, Bochs, JVM, KVM, User-Mode-Linux, Xen, VServer, Wine)
- diskutieren in der Gruppe Vor- und Nachteile von bestimmten VM-Ansätzen
- untersuchen CPU-Emulationen
- untersuchen Geräte-Emulationen

---

### **Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

**[1] Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule | Vertiefungsrichtung Rechnerarchitektur)

**[2] Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule | Vertiefungsrichtung Verteilte Systeme und Betriebssysteme)

Dieses Modul ist daneben auch in den Studienfächern "Informatik (Master of Science)", "Mathematik (Bachelor of Science)" verwendbar.

---

### **Studien-/Prüfungsleistungen:**

Virtuelle Maschinen (Vorlesung mit Übungen) (Prüfungsnummer: 202041)

Prüfungsleistung, mündliche Prüfung, Dauer (in Minuten): 30

Anteil an der Berechnung der Modulnote: 100%

Erstablingung: WS 2019/2020, 1. Wdh.: SS 2020

1. Prüfer: Volkmar Sieh (100548)

---

**Modulbezeichnung:** **Software Architecture (PROJ 5-ECTS) (OSS-ARCH-PROJ)** **5 ECTS**  
(Software Architecture (PROJ 5-ECTS))

Modulverantwortliche/r: Martin Jung, Dirk Riehle

Lehrende: Martin Jung, Dirk Riehle

Startsemester: SS 2019

Dauer: 1 Semester

Turnus: jährlich (SS)

Präsenzzeit: k.A. Std.

Eigenstudium: 150 Std.

Sprache: Deutsch

**Lehrveranstaltungen:**

Software Architecture (PROJ) (SS 2019, Sonstige Lehrveranstaltung, 2 SWS, Martin Jung et al.)

**Inhalt:**

This course teaches students concepts, methods, and tools of software architecture.

Die Architektur eines Softwaresystems beschreibt die wesentlichen Komponenten des Systems, ihre Beziehungen und Struktur, sowie das Verhalten und die Dynamik der Beziehungen und Struktur dieser Komponenten. Dieser Kurs vermittelt in einer Vorlesung zunächst die folgenden Aspekte von Softwarearchitektur:

- Grundlegende Bausteine und ihre Beziehungen
- Softwarearchitekturbeschreibungssprachen
- Softwarearchitekturstile und -muster
- Bibliotheken, Rahmenwerke und Plattformen
- Formale sowie de-facto Industriestandards
- Die Softwarearchitekturen von Beispielsystemen
- Nicht technische Kriterien in der Architektur
- Werkzeuge für Softwarearchitekten
- Vorgehensmodelle der Softwarearchitektur
- Architekturgetriebene Entwicklung
- Die Rolle und Funktion der Softwarearchitektin

Studierende können eine oder beide von zwei Komponenten wählen:

- VUE (Vorlesung + Übungen), 4 SWS, 5 ECTS
- PROJ (kleines Projekt), 2 SWS, 5 ECTS. Die Projekte werden von unseren Industriepartnern bereitgestellt. Hier dokumentieren, analysieren und bewerten Studierende die Softwarearchitektur eines realen Softwaresystems. Diese verschiedenen Aspekte werden im Laufe des Semesters inkrementell abgearbeitet und am Ende dem Industriepartner in einer Präsentation vorgestellt.

Der Unterricht findet als 3h-Block während der Vorlesungszeit statt. Der Zeitplan befindet sich hier: <http://goo.gl/ZXJjg> . Der Zeitplan enthält auch einen Link auf den zur Veranstaltung gehörigen StudOn Kurs. Bitte registrieren Sie sich auf StudOn sobald wie möglich.

**Lernziele und Kompetenzen:**

- Ganzheitliches Verständnis des Konzepts "Softwarearchitektur"
- Befähigung zur Bewertung, Auswahl und Konstruktion problemangemessener Architekturen
- Kenntnis architekturgetriebener Entwicklungsmethodik und entsprechender Werkzeuge
- Kenntnis der typischen Verantwortlichkeiten und der Methodik eines Softwarearchitekten

**Literatur:**

- <http://goo.gl/ou7mja>

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

**[1] Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule)

Dieses Modul ist daneben auch in den Studienfächern "International Information Systems (IIS) (Master of Science)" verwendbar.

**Studien-/Prüfungsleistungen:**



Software Architecture (PROJ 5-ECTS) (Prüfungsnummer: 386409)

(englische Bezeichnung: Software Architecture (PROJ 5-ECTS))

Prüfungsleistung, Portfolio

Anteil an der Berechnung der Modulnote: 100%

weitere Erläuterungen:

- Unterricht
- Hausaufgaben
- Mündliche Prüfung
- Projektarbeit

Erstablingung: SS 2019, 1. Wdh.: keine Angabe

1. Prüfer: Martin Jung (100803), 2. Prüfer: Dirk Riehle (100100)

---

---

**Modulbezeichnung:** IT-Modernisierung (IT-Modern)**5 ECTS**  
(IT Modernization)

Modulverantwortliche/r: Peter Wilke  
Lehrende: Peter Wilke

---

Startsemester: SS 2019	Dauer: 1 Semester	Turnus: jährlich (SS)
Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Deutsch

---

**Lehrveranstaltungen:**

IT-Modernisierung (SS 2019, Vorlesung, 4 SWS, Peter Wilke)

---

**Inhalt:**

IT-Modernisierung beschäftigt sich mit dem Ersatz alter Software- und/oder Hardware. Software im kommerziellen Bereich hat eine typische Lebensdauer von über 25 Jahren, damit ist klar, dass diese keine der momentan oder zukünftig zur Verfügung stehenden Möglichkeiten nutzt oder nutzen kann, denn "damals" waren Single-CPU's der Standard und Vernetzung unbekannt.

Durch das hohe Investitionsvolumen ist eine Neu-Programmierung praktisch immer wirtschaftlich nicht sinnvoll und technisch oft unmöglich, da gar nicht genügend Programmierer zur Verfügung stehen. Die Software hat aber einen hohen Reifegrad erreicht, so dass sich die Frage stellt, ob man diese nicht automatisiert auf neue Technologien umstellen kann.

Dieses Modul beleuchtet nun exemplarisch, auf welchen Feldern Bedarf besteht, wie der Stand der Technik ist, und welche zukünftigen Fragestellungen sich abzeichnen.

Die Studierenden werden durch Übungsaufgaben mit den "alten" Programmiersprachen wie Cobol, Assembler, Fortran uä. vertraut gemacht, und bearbeiten selbstständig kleine Migrations-Aufgaben.

Als Dozenten werden erfahrene Spezialisten aus der Industrie über ihre Fragestellungen und Ansätze berichten.

Momentane Planung (Stand 2015Apr08, Themen nicht zwingend in dieser Reihenfolge):

- Einleitung
- Überblick
- Aufbau (Architektur) eines Rechenzentrums
- DB2 unter z/OS
- Exkursion DATEV Rechenzentrum
- RZ Konsolidierung
- Server Konsolidierung
- Cobol Grundlagen, RD/z, TSO/ISPF, JCL
- System z Hardware Grundlagen
- Java am Host
- Mainframe Programmierung
- Legacy-Anwendungen in einer Cloud-Architektur, CICS Modernisierung
- Internationalisierung: Unicode im Rechenzentrum
- Praxisbericht IT-Betrieb (Aufgabenstellung)
- Infrastrukturen-Modernisierung
- Praxisbericht IT-Betrieb

**Lernziele und Kompetenzen:**

*Fachkompetenz*

*Analysieren*

Lernende können ein Problem in einzelne Teile zerlegen und so die Struktur des Problems verstehen; sie können Widersprüche aufdecken, Zusammenhänge erkennen und Folgerungen ableiten und zwischen Fakten und Interpretationen unterscheiden.

*Evaluiieren (Beurteilen)*

Lernende können evidenzbasierte, qualitative und quantitative Urteile zu Sachverhalten anhand von Kriterien anstellen.

*Lern- bzw. Methodenkompetenz*

Fähigkeit und Bereitschaft zur Anwendung bestimmter Lern- und Arbeitsmethoden, die zur Entwicklung der anderen Kompetenzen, insbesondere der Fachkompetenz nötig sind.

#### *Selbstkompetenz*

Fähigkeit und Bereitschaft, sich weiterzuentwickeln und das eigene Leben eigenständig und verantwortlich im jeweiligen sozialen, kulturellen bzw. beruflichen Kontext zu gestalten.

#### *Sozialkompetenz*

Fähigkeit und Bereitschaft, zielorientiert mit anderen zusammenzuarbeiten, ihre Interessen und sozialen Situationen zu erfassen, sich mit ihnen rational und verantwortungsbewusst auseinanderzusetzen und zu verständigen sowie die Arbeits- und Lebenswelt mitzugestalten.

#### **Literatur:**

wird gerade zusammengestellt ...

---

#### **Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

##### [1] Informatik (Bachelor of Science)

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule)

Dieses Modul ist daneben auch in den Studienfächern "International Information Systems (IIS) (Master of Science)" verwendbar.

---

#### **Studien-/Prüfungsleistungen:**

IT-Modernisierung (Prüfungsnummer: 716516)

(englische Bezeichnung: IT Modernization)

Prüfungsleistung, mündliche Prüfung, Dauer (in Minuten): 30 Minuten

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- IT-Modernisierung

Prüfungssprache: Deutsch oder Englisch

Erstablingung: SS 2019, 1. Wdh.: WS 2019/2020

1. Prüfer: Peter Wilke (100155)

---

#### **Organisatorisches:**

Bei Interessen bitte Anmeldung per E-Mail an

<mailto:Peter.Wilke@FAU.DE>

Sie werden dann über alles Aktuelle auf dem Laufenden gehalten.

Momentan planen wir die Veranstaltung Mo 10:00-14:00 Uhr. Dies gibt uns die Möglichkeit ca. 1 Stunde Mittagspause zu machen, die konkrete Zeit und Dauer der Pause richtet sich nach dem Stoff.

In der Mittagspause besteht die Gelegenheit, mit den Dozenten zu plaudern...

Einige der Themen werden als Fallstudien bearbeitet. D.h. der Dozent stellt eine IT-Modernisierungssituation dar, die Hörer müssen dann eine Lösung erarbeiten, entweder direkt in der Vorlesung oder als Hausaufgabe. Die Gruppen tragen dann ihre Lösung vor und dann wird mit der tatsächlich umgesetzten Lösung verglichen.

Prüfungsleistung: mdl. Prüfung von ca. 30min Dauer

Anrechenbar für folgende Säulen:

- anwendungsorientierte Säule
- softwareorientierte Säule

---

**Modulbezeichnung:** Analyse und Design objektorientierter Softwaresysteme mit UML (PSWT-OOAD) **5 ECTS**  
 (Analysis and Design of Object Oriented Software Systems with UML)

Modulverantwortliche/r: Detlef Kips  
 Lehrende: Detlef Kips, Martin Jung

---

Startsemester: SS 2019	Dauer: 1 Semester	Turnus: jährlich (SS)
Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Deutsch

---

**Lehrveranstaltungen:**

Analyse und Design objektorientierter Softwaresysteme mit der Unified Modeling Language (UML) (SS 2019, Vorlesung, 2 SWS, Detlef Kips et al.)

Übungen zu Analyse und Design objektorientierter Softwaresysteme mit der Unified Modeling Language (UML) (SS 2019, Übung, 2 SWS, Ralf Ellner)

---

**Inhalt:**

Die sogenannte "Unified Modeling Language" (UML) ist der seit Jahren weltweit akzeptierte Notationsstandard für die Modellierung komplexer Softwaresysteme. Mit einem reichhaltigen Repertoire an graphischen und textuellen Ausdrucksmöglichkeiten bietet die UML ihren Anwendern die Möglichkeit, die Anforderungen an die Zielsoftware, ihre statischen bzw. dynamischen Systemeigenschaften sowie die gewählte Softwarearchitektur halbformal zu spezifizieren, im Team darüber zu kommunizieren und große Teile des Programmcodes aus den spezifizierten Systemmodellen zu generieren.

Ziel dieser Vorlesung ist es, die Studierenden mit Syntax und Semantik der UML vertraut zu machen und zu demonstrieren, wie die UML im Rahmen eines "typischen" Softwareentwicklungsprozesses angewendet werden kann. Zu diesem Zweck werden die verschiedenen Diagrammtypen und Notationselemente der UML schrittweise eingeführt und anhand eines durchgängigen Anwendungsbeispiels im Rahmen eines konkreten Vorgehensmodells über alle Entwicklungsphasen hinweg eingesetzt.

**Lernziele und Kompetenzen:**

Im Rahmen dieser Veranstaltung sollen die Studierenden insbesondere die Kompetenz erwerben,

- die syntaktische Struktur und die Semantik vorgegebener UML-Modelle bzw. Modellausschnitte zu analysieren und zu erläutern
- verschiedene Sprachelemente der UML (und ggf. deren Kombination) im Hinblick auf ihre Eignung zur Abbildung charakteristischer Modellierungsprobleme im Rahmen eines Softwareentwicklungsprozesses zu bewerten, auszuwählen und anzuwenden
- die Struktur und Systematik des UML-Metamodells zu erläutern und die UML mit geeigneten Metamodellierungskonzepten auf spezifische Anwendungskontexte anzupassen
- zu einer gegebenen Anforderungsdefinition im Rahmen einer systematischen Analyse- und Entwurfsmethodik ein integriertes UML-Systemmodell zu erstellen
- den Funktionsumfang eines UML-basierten Modellierungswerkzeugs zu bewerten, ein geeignetes Werkzeug auszuwählen und sicher anzuwenden.

**Literatur:**

- Rumbaugh, J.; Booch, G.; Jacobson, I.: The Unified Modeling Language Reference Manual, Addison-Wesley, 2004
- Hitz, M.; Kappel, G.; Kapsammer, E.; Retschitzegger, W.: UML @ work , 3., aktualisierte und überarbeitete Auflage, dpunkt-Verlag, 2005
- Winter, M.: Methodische objektorientierte Softwareentwicklung, dpunkt-Verlag, 2005
- Störrle, H.: UML 2 erfolgreich einsetzen, Addison-Wesley, 2007
- Rumpe, B.: Modellierung mit UML: Sprache, Konzepte und Methodik, Springer-Verlag, 2. Auflage, 2011
- Seidl, M., Brandsteidl, M., Huemer, C., Kappek, G.: UML@classroom - Eine Einführung in die objektorientierte Modellierung, dpunkt-Verlag, 2012
- Rupp, C.; Queins, S., et al. UML 2 glasklar: Praxiswissen für die UML-Modellierung, Carl Hanser Verlag, 2012

Die einschlägige Originalliteratur zur UML findet man auf der Website der Object Management Group ( <http://www.omg.org/spec/UML> ).

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

**[1] Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule)

Dieses Modul ist daneben auch in den Studienfächern "International Information Systems (IIS) (Master of Science)" verwendbar.

---

**Studien-/Prüfungsleistungen:**

Analyse und Design objektorientierter Softwaresysteme mit der Unified Modeling Language (UML)  
(Prüfungsnummer: 510375)

Prüfungsleistung, mündliche Prüfung, Dauer (in Minuten): 30

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Analyse und Design objektorientierter Softwaresysteme mit der Unified Modeling Language (UML)
- Übungen zu Analyse und Design objektorientierter Softwaresysteme mit der Unified Modeling Language (UML)

Erstablingung: SS 2019, 1. Wdh.: SS 2020

1. Prüfer: Detlef Kips (100178)

---

**Modulbezeichnung:** **Nailing your Thesis (PROJ 5-ECTS) (OSS-NYT-PROJ)** **5 ECTS**  
(Nailing your Thesis (PROJ 5-ECTS))

Modulverantwortliche/r: Dirk Riehle  
Lehrende: Dirk Riehle

Startsemester: SS 2019                      Dauer: 1 Semester                      Turnus: halbjährlich (WS+SS)  
Präsenzzeit: k.A. Std.                      Eigenstudium: 150 Std.                      Sprache: Englisch

**Lehrveranstaltungen:**

Nailing your Thesis (PROJ) (SS 2019, Sonstige Lehrveranstaltung, 2 SWS, Dirk Riehle)

**Inhalt:**

This course teaches students how to perform scientific research for their final thesis or a research paper. The goal is to prepare students for a Bachelor or Master research thesis.

The course covers the following topics:

- Science and society
- The research process
- Exploratory research
- Confirmatory research
- Writing a thesis/paper
- The scientific community

Students can choose one or both of two components:

- VUE (lecture + exercise), 4 SWS, 5 ECTS. VUE combines lectures with homework and exercises. VUE is run as a 3h block.
- PROJ (small research project), 2 SWS, 5 ECTS. In PROJ, students perform a small research project, either individually or in teams. The available projects will be presented at the beginning of the course. Students perform the research, write a paper, and hold a presentation about their work.

If you only want to take OSS-NYT-PROJ please still come to OSS-NYT-VUE class on the first day to learn about the projects.

The overall schedule can be found at <https://goo.gl/VqoFO> . Please sign up for the course on StudOn (link accessible through schedule spreadsheet) as soon as possible.

**Lernziele und Kompetenzen:**

- Understand how to perform research
- Understand how to write a research thesis

**Literatur:**

The syllabus, schedule, literature, and more can be found at <http://nythesis.com>

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

**[1] Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule)

Dieses Modul ist daneben auch in den Studienfächern "International Information Systems (IIS) (Master of Science)" verwendbar.

**Studien-/Prüfungsleistungen:**

Nailing your Thesis (PROJ 5-ECTS) (Prüfungsnummer: 580491)

(englische Bezeichnung: Nailing your Thesis (PROJ 5-ECTS))

Prüfungsleistung, Portfolio

Anteil an der Berechnung der Modulnote: 100%

weitere Erläuterungen:

- Exercise participation
- Homework assignments

- Project work

Erstablingung: SS 2019, 1. Wdh.: keine Angabe

1. Prüfer: Dirk Riehle (100100)

---

---

**Modulbezeichnung:** Free/Libre, and Open Source Software (5-ECTS) (OSS-FLOSS) 5 ECTS  
(Free/Libre, and Open Source Software (5-ECTS))

Modulverantwortliche/r: Dirk Riehle  
Lehrende: Dirk Riehle

---

Startsemester: SS 2019	Dauer: 1 Semester	Turnus: jährlich (SS)
Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Englisch

---

**Lehrveranstaltungen:**

Free/Libre, and Open Source Software (VUE) (SS 2019, Vorlesung mit Übung, 4 SWS, Dirk Riehle)

---

**Inhalt:**

This course teaches students engineering, product management, and business practices of open source. It takes the classic form of a lecture with associated weekly exercises.

The topics of the lectures are:

- Introduction to open source
- Open source and intellectual property
- Open source project communities
- Open source software engineering
- Corporate open source governance
- Strategic corporate open sourcing
- Open source developer foundations
- Open source user foundations
- Open source distributors
- Single-vendor open source firms
- Open source labor economics

More details can be found in the course syllabus at <http://goo.gl/gEjk2>.

Students of computer science (Informatik) should note that this course is less about technology and more about economics and the software industry.

The course combines weekly lectures with homework. Homework typically involves reading materials and summarizing these readings in short one-page documents. Some homework consists of installing a tool (FOSSology) and using it to create a bill-of-material for open source projects, including checking for license compliance. Students should be able to install and use the tool.

The overall schedule can be found at <http://goo.gl/V6gYj>. Please sign up for the course on StudOn (link accessible through schedule spreadsheet) as soon as possible.

**Lernziele und Kompetenzen:**

- Understand how open source projects work
- Understand key aspects of the software industry
- Understand how open source is changing the software industry

**Literatur:**

- <http://goo.gl/D8qnu>
- 

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

[1] **Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule)

Dieses Modul ist daneben auch in den Studienfächern "International Information Systems (IIS) (Master of Science)" verwendbar.

---

**Studien-/Prüfungsleistungen:**



Free/Libre, and Open Source Software (5-ECTS) (Prüfungsnummer: 532135)

(englische Bezeichnung: Free/Libre, and Open Source Software (5-ECTS))

Prüfungsleistung, Portfolio

Anteil an der Berechnung der Modulnote: 100%

weitere Erläuterungen:

- Exercise participation
- Homework assignments

Erstablingung: SS 2019, 1. Wdh.: keine Angabe

1. Prüfer: Dirk Riehle (100100)

---

---

**Modulbezeichnung:** **Product Management (VUE 5-ECTS) (OSS-PROD-VUE)** **5 ECTS**  
 (Product Management (VUE 5-ECTS))

Modulverantwortliche/r: Dirk Riehle  
 Lehrende: Dirk Riehle

---

Startsemester: WS 2019/2020	Dauer: 1 Semester	Turnus: jährlich (WS)
Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Englisch

---

**Lehrveranstaltungen:**

Product Management (VUE) (WS 2019/2020, Vorlesung mit Übung, 2 SWS, Dirk Riehle)

---

**Inhalt:**

This course teaches students the concepts, methods, and tools of software product management. Product management is an important function in software development organizations. A product manager conceives and defines new products. His or her task is to understand the market incl. customers, to develop a product vision from that understanding, to translate it into product requirements, define those requirements on a by-feature basis and work with engineering to ensure these features are properly realized in the product under development.

- Role, tasks, and responsibilities of a product manager
- Process, methods, techniques and tools of product management
- Managing incremental/sustaining as well as disruptive innovation
- Open source product management; new trends in product management

Students can choose one or both of two components:

- VUE (lecture + homework), 2 SWS, 5 ECTS. VUE uses teaching cases as commonly used in MBA programs. The teaching cases are available for free at <http://pmbycase.com>.
- PROJ (small project), 2 SWS, 5 ECTS. In PROJ, students perform a small product management project, either individually or in teams. In such a project, students write a business plan for a market opportunity or a startup. There are two sources of projects: Those supplied by the professorship and those supplied by students. If we supply any projects, we will tell you about them in the first lecture. Student-supplied projects can be suggested by students to Prof. Riehle at least one week before class starts and must meet the requirements described here: <https://goo.gl/sNPi1i>

PROD projects are run as shared projects, in which all participants contribute and get to participate in the project results. Read more at <https://wp.me/pDU66-2p4>.

Class is run as two 90min blocks. The first block discusses the teaching cases. The second block is a coaching session for the projects (10 ECTS only).

The overall schedule can be found at <http://goo.gl/tTA10> . Please sign up for the course on StudOn (link accessible through schedule spreadsheet) as soon as possible.

**Lernziele und Kompetenzen:**

- Understand the role, function, and responsibilities of a product manager
- Understand key concepts, methods, and tools of software product management
- Understand different business situations, incl. Incremental vs. disruptive innovation

**Literatur:**

- <http://goo.gl/41Dgsr>
- 

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

**[1] Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule)

Dieses Modul ist daneben auch in den Studienfächern "International Information Systems (IIS) (Master of Science)", "Maschinenbau (Master of Science)", "Mechatronik (Bachelor of Science)", "Mechatronik (Master of Science)" verwendbar.

---

**Studien-/Prüfungsleistungen:**

Product Management (VUE 5-ECTS) (Prüfungsnummer: 663131)

(englische Bezeichnung: Product Management (VUE 5-ECTS))

Prüfungsleistung, Portfolio

Anteil an der Berechnung der Modulnote: 100%

weitere Erläuterungen:

- In-class participation
- Homework assignments
- Project work

Erstablingung: WS 2019/2020, 1. Wdh.: keine Angabe

1. Prüfer: Dirk Riehle (100100)

---

**Modulbezeichnung:** **Nailing your Thesis (VUE 5-ECTS) (OSS-NYT-VUE)** **5 ECTS**  
(Nailing your Thesis (VUE 5-ECTS))

Modulverantwortliche/r: Dirk Riehle  
Lehrende: Dirk Riehle

Startsemester: SS 2019	Dauer: 1 Semester	Turnus: halbjährlich (WS+SS)
Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Englisch

**Lehrveranstaltungen:**

Nailing your Thesis (VUE) (SS 2019, Vorlesung mit Übung, 4 SWS, Dirk Riehle)

**Inhalt:**

This course teaches students how to perform scientific research for their final thesis or a research paper. The goal is to prepare students for a Bachelor or Master research thesis.

The course covers the following topics:

- Science and society
- The research process
- Exploratory research
- Confirmatory research
- Writing a thesis/paper
- The scientific community

Students can choose one or both of two components:

- VUE (lecture + exercise), 4 SWS, 5 ECTS. VUE combines lectures with homework and exercises. VUE is run as a 3h block.
- PROJ (small research project), 2 SWS, 5 ECTS. In PROJ, students perform a small research project, either individually or in teams. The available projects will be presented at the beginning of the course. Students perform the research, write a paper, and hold a presentation about their work.

If you only want to take OSS-NYT-PROJ please still come to OSS-NYT-VUE class on the first day to learn about the projects.

The overall schedule can be found at <https://goo.gl/VqoFO> . Please sign up for the course on StudOn (link accessible through schedule spreadsheet) as soon as possible.

**Lernziele und Kompetenzen:**

- Understand how to perform research
- Understand how to write a research thesis

**Literatur:**

The syllabus, schedule, literature, and more can be found at <http://nythesis.com>

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

**[1] Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule)

Dieses Modul ist daneben auch in den Studienfächern "International Information Systems (IIS) (Master of Science)", "Maschinenbau (Bachelor of Science)", "Maschinenbau (Master of Science)", "Mechatronik (Bachelor of Science)", "Mechatronik (Master of Science)", "Wirtschaftsingenieurwesen (Bachelor of Science)" verwendbar.

**Studien-/Prüfungsleistungen:**

Nailing your Thesis (VUE 5-ECTS) (Prüfungsnummer: 480491)

(englische Bezeichnung: Nailing your Thesis (VUE 5-ECTS))

Prüfungsleistung, Portfolio

Anteil an der Berechnung der Modulnote: 100%

weitere Erläuterungen:

- Exercise participation
- Homework assignments
- Project work

Erstablingung: SS 2019, 1. Wdh.: keine Angabe  
1. Prüfer: Dirk Riehle (100100)

---

---

**Modulbezeichnung:** Testen von Softwaresystemen (PSWT-TSWS) 5 ECTS  
 (Testing of Software Systems)

Modulverantwortliche/r: Norbert Oster

Lehrende: Norbert Oster, Klaudia Dussa-Zieger

---

Startsemester: SS 2019

Dauer: 1 Semester

Turnus: jährlich (SS)

Präsenzzeit: 60 Std.

Eigenstudium: 90 Std.

Sprache: Deutsch

---

**Lehrveranstaltungen:**

Testen von Softwaresystemen (SS 2019, Vorlesung, 4 SWS, Klaudia Dussa-Zieger et al.)

---

**Inhalt:**

- Test-Terminologie und Software-Qualität nach ISO/IEC 9126
- Fundamentaler Testprozess
- Teststufen im Softwarelebenszyklus
- Statischer Test: Reviews
- Erfahrungsbasiertes Testen
- Black-Box-Testverfahren: Äquivalenzklassen-/Grenzwerttest, Zustandsbezogener Test, Entscheidungstabellentest
- Statische Analyse: Daten- und Kontrollflussanomalien
- White-Box-Testverfahren: Kontrollflussbasiert, Datenflussbasiert, Bedingungsüberdeckung
- Mutationstest
- Testmanagement: Planung, Kostenschätzung, Überwachung, Risikobewertung, Priorisierung, Fehlermanagement
- Formale Verifikation: Theorem Proving und Model Checking
- Quantitative Zuverlässigkeitsbewertung

**Lernziele und Kompetenzen:**

Die Studierenden

- unterscheiden die verschiedenen Fachbegriffe aus der Test-Domäne
- erläutern die unterschiedlichen Aspekte der Software-Qualität
- beschreiben den fundamentalen Testprozess und seine einzelnen Teilaufgaben
- erläutern die wichtigsten Aspekte der Testpsychologie und entscheiden auf dieser Basis z.B. über das Testteam
- beschreiben die typischen Teststufen und ordnen diese den Phasen im Softwarelebenszyklus zu
- stellen die Unterschiede zwischen dynamischem Test, Review und statischer Analyse heraus
- unterscheiden verschiedene Review-Arten und veranschaulichen deren typische Arbeitsschritte und Rollen
- differenzieren unterschiedliche Formen erfahrungsbasierten Testens
- wenden das Verfahren der Äquivalenzklassenbildung an und ermitteln entsprechende Testfälle für den Grenzwerttest
- entwickeln Entscheidungstabellen für beliebige Testaufgaben und bestimmen die entsprechenden Testfälle
- erläutern typische Daten-/Kontrollflussanomalien an selbst-gewählten Beispielen
- unterscheiden verschiedene kontrollfluss-, datenfluss und bedingungsorientierte Testkriterien
- wenden die grundlegenden White-Box-Testkriterien an und leiten entsprechende Testfälle ab
- begründen die Ordnung der White-Box-Überdeckungskriterien in ihrer Subsumptionshierarchie
- erläutern das Konzept des Mutationstests zur quantitativen Bewertung der Testgüte
- gestalten einen konkreten Testprozess aus der Sicht eines Testmanagers
- beschreiben die wichtigsten Facetten des Fehlermanagements
- erläutern den Unterschied zwischen Theorem Proving und Model Checking und skizzieren das jeweilige Vorgehen
- wenden Theorem Proving auf sequentiellen Code an und skizzieren den Beweis der Interferenzfreiheit bei Nebenläufigkeit
- beschreiben Voraussetzungen, Annahmen und Vorgehen bei verschiedenen Arten der quantitativen

#### Zuverlässigkeitsbewertung

- entscheiden je nach Art des Softwareprodukts welche Art der quantitativen Zuverlässigkeitsbewertung zulässig ist
- nennen die wichtigsten Normen und Standards sowie deren typische Anforderungen
- bewerten und benutzen Werkzeuge für verschiedene Testaufgaben

#### Literatur:

- Spillner, Andreas; Linz, Tilo: Basiswissen Softwaretest, dpunkt-Verlag
- Liggesmeyer, Peter: Software-Qualität, Spektrum Verlag
- Spillner, Andreas; Roßner, Thomas; Winter, Mario; Linz, Tilo: Praxiswissen Softwaretest - Testmanagement, dpunkt-Verlag
- Lyu, Michael R.: Handbook of Software Reliability Engineering, McGraw-Hill

---

#### Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

##### [1] Informatik (Bachelor of Science)

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule)

Dieses Modul ist daneben auch in den Studienfächern "International Information Systems (IIS) (Master of Science)" verwendbar.

---

#### Studien-/Prüfungsleistungen:

Testen von Softwaresystemen (Prüfungsnummer: 189989)

Prüfungsleistung, mündliche Prüfung, Dauer (in Minuten): 30

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Testen von Softwaresystemen

Erstablingung: SS 2019, 1. Wdh.: WS 2019/2020

1. Prüfer: Norbert Oster (100120)

---

---

**Modulbezeichnung:** **Organisation und Qualitätskontrolle im Software Engineering 5 ECTS**  
**- Option A (OrgA-SWE)**  
 (Management and Quality Assurance in Software Engineering -  
 Option A)

Modulverantwortliche/r: Francesca Saglietti  
 Lehrende: Francesca Saglietti

---

Startsemester: SS 2019	Dauer: 2 Semester	Turnus: jährlich (SS)
Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Deutsch

---

**Lehrveranstaltungen:**

Das Modul besteht aus dem Teil Organisation und Qualitätskontrolle der Vorlesung mit Übung Grundlagen des Software Engineering (d.h. nur aus dem letzten Drittel der Vorlesung mit Übung) sowie den praktischen Übungen Software Engineering in der Praxis.

Grundlagen des Software Engineering (SS 2019, Vorlesung, 4 SWS, Francesca Saglietti)  
 Übungen zu Grundlagen des Software Engineering (SS 2019, Übung, 2 SWS, Marc Spisländer)

**Praktische Übung**

Software Engineering in der Praxis (WS 2019/2020, Übung, 3 SWS, Xiaochen Wu et al.)

---

**Inhalt:**

Das Modul bietet eine umfassende Übersicht analytischer und organisatorischer Verfahren des modernen Software Engineering. In praktischen Übungen werden Werkzeuge zur Entwicklung und Analyse komplexer Software vorgestellt, deren industrielle Einsetzbarkeit anschließend von den Teilnehmern anhand in der Praxis repräsentativer Aufgabenstellungen erprobt wird.

**Lernziele und Kompetenzen:**

Die Studenten

- benutzen Software-Metriken zur Bestimmung der Software-Komplexität
- wenden unterschiedliche Nachweisverfahren, wie z.B. statische Analyse, funktionales und strukturelles Testen, oder Korrektheitsbeweise an
- stellen die Bedeutung der Wartung im Software-Entwicklungsprozess heraus
- erläutern verschiedene Aspekte des Software-Projektmanagements
- erläutern Potenzial und Grenzen unterschiedlicher Werkzeuge zur Unterstützung softwaretechnischer Tätigkeiten
- wenden unterschiedliche Werkzeuge an, um sowohl selbständig als auch in Teams Beispielaufgaben aus dem Bereich der objektorientierten Analyse, des objektorientierten Entwurfs, des Testens, des Beweisens und des Projektmanagements zu lösen.

**Literatur:**

Lehrbuch der Softwaretechnik (Band 1), Helmut Balzert, 2000

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

**[1] Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule)

---

**Studien-/Prüfungsleistungen:**

Organisation und Qualitätskontrolle im modernen Software Engineering (mit prakt. Übungen Software Engineering in der Praxis) (Prüfungsnummer: 659041)

Prüfungsleistung, Portfolio

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Grundlagen des Software Engineering
- Übungen zu Grundlagen des Software Engineering
- Software Engineering in der Praxis



weitere Erläuterungen:

Die Bewertung der Prüfungsleistung setzt sich zusammen aus einer 30-minütigen schriftlichen Prüfung (Gewichtung 1/2) über die Grundlagen des Software Engineering (Teil Organisation und Qualitätskontrolle) und einer 60-minütigen Prüfung am Rechner (Gewichtung 1/2) über Software Engineering in der Praxis. Beide Einzelleistungen müssen mit mindestens 4,0 bestanden werden.

Erstablingung: SS 2019, 1. Wdh.: WS 2019/2020

1. Prüfer: Francesca Saglietti (100112)

---

---

**Modulbezeichnung:** Product Management (PROJ 5-ECTS) (OSS-PROD-PROJ) 5 ECTS  
(Product Management (PROJ 5-ECTS))

Modulverantwortliche/r: Dirk Riehle

Lehrende: Dirk Riehle

---

Startsemester: WS 2019/2020

Dauer: 1 Semester

Turnus: jährlich (WS)

Präsenzzeit: k.A. Std.

Eigenstudium: 150 Std.

Sprache: Englisch

---

**Lehrveranstaltungen:**

Product Management (PROJ) (WS 2019/2020, Sonstige Lehrveranstaltung, 2 SWS, Dirk Riehle)

---

**Inhalt:**

This course teaches students the concepts, methods, and tools of software product management. Product management is an important function in software development organizations. A product manager conceives and defines new products. His or her task is to understand the market incl. customers, to develop a product vision from that understanding, to translate it into product requirements, define those requirements on a by-feature basis and work with engineering to ensure these features are properly realized in the product under development.

- Role, tasks, and responsibilities of a product manager
- Process, methods, techniques and tools of product management
- Managing incremental/sustaining as well as disruptive innovation
- Open source product management; new trends in product management

Students can choose one or both of two components:

- VUE (lecture + homework), 2 SWS, 5 ECTS. VUE uses teaching cases as commonly used in MBA programs. The teaching cases are available for free at <http://pmbycase.com>.
- PROJ (small project), 2 SWS, 5 ECTS. In PROJ, students perform a small product management project, either individually or in teams. In such a project, students write a business plan for a market opportunity or a startup. There are two sources of projects: Those supplied by the professorship and those supplied by students. If we supply any projects, we will tell you about them in the first lecture. Student-supplied projects can be suggested by students to Prof. Riehle at least one week before class starts and must meet the requirements described here: <https://goo.gl/sNPi1i>

PROD projects are run as shared projects, in which all participants contribute and get to participate in the project results. Read more at <https://wp.me/pDU66-2p4>.

Class is run as two 90min blocks. The first block discusses the teaching cases. The second block is a coaching session for the projects (10 ECTS only).

The overall schedule can be found at <http://goo.gl/tTAI0> . Please sign up for the course on StudOn (link accessible through schedule spreadsheet) as soon as possible.

**Lernziele und Kompetenzen:**

- Understand the role, function, and responsibilities of a product manager
- Understand key concepts, methods, and tools of software product management
- Understand different business situations, incl. Incremental vs. disruptive innovation

**Literatur:**

- <http://goo.gl/41Dgsr>
- 

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

**[1] Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule)

Dieses Modul ist daneben auch in den Studienfächern "International Information Systems (IIS) (Master of Science)", "Maschinenbau (Master of Science)", "Mechatronik (Bachelor of Science)" verwendbar.

---

**Studien-/Prüfungsleistungen:**

Product Management (PROJ 5-ECTS) (Prüfungsnummer: 981857)

(englische Bezeichnung: Product Management (PROJ 5-ECTS))

Prüfungsleistung, Portfolio

Anteil an der Berechnung der Modulnote: 100%

weitere Erläuterungen:

- In-class participation
- Homework assignments
- Project work

Erstablingung: WS 2019/2020, 1. Wdh.: keine Angabe

1. Prüfer: Dirk Riehle (100100)

---

---

**Modulbezeichnung: Software-Projektmanagement (PSWT-SPM)** **5 ECTS**  
(Software Project Management)

Modulverantwortliche/r: Bernd Hindel  
Lehrende: Bernd Hindel

---

Startsemester: WS 2019/2020	Dauer: 1 Semester	Turnus: jährlich (WS)
Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Deutsch

---

**Lehrveranstaltungen:**

Software-Projektmanagement (WS 2019/2020, Vorlesung, 4 SWS, Bernd Hindel)

---

**Inhalt:**

Zahlreiche Statistiken zeigen: Nur wenige Software-Projekte werden erfolgreich (hinsichtlich Zeit-, Budget- und Funktionsvorgaben) abgeschlossen. Sehr viele Projekte werden nur mit erheblichen Defiziten zu Ende gebracht, noch viel zu viele scheitern gänzlich. Oft liegen die Gründe im ungenügenden Projektmanagement.

Die Vorlesung gibt einen Überblick zu grundlegenden Disziplinen des Projektmanagements und zeigt deren Wirkungsweisen an Hand von Praxisbeispielen.

Gliederung:

1. Einführung Grundbegriffe des Projektmanagements, unterschiedliche Projektgrößen, unterschiedliche Projektarten, Erfolg und Misserfolg in Projekten
2. Projektstart und Planung, Kickoff-Meeting, Anforderungssammlung, Projektstrukturplan, Aufwandschätzung, Aktivitäten-, Ressourcen- und Kostenplan
3. Projektkontrolle und Steuerung, Fortschrittsüberwachung, Besprechungen, Berichte, Änderungsmanagement
4. Personalmanagement, Der Faktor Mensch, Teamwork, Führungsgrundsätze, Gesprächsstrategien, Konflikte lösen
5. Änderungsmanagement Konfigurationen, Änderungswünsche, Change Control Board, Built- und Release-Mechanismen
6. Qualitäts- und Risikomanagement Qualitätsplan, Audits und Reviews, Risikoermittlung, Risikobewertung und Verfolgung, Gegenmaßnahmen
7. Reifegrad Modelle und Standards CMMI, SPiCE, ISO9001, ISO/IEC12207

**Lernziele und Kompetenzen:**

Die Studierenden

- kennen die Grundbegriffe des Projektmanagements
  - unterscheiden unterschiedliche Projektgrößen, unterschiedliche Projektarten
  - verstehen die Ursachen für Erfolg und Misserfolg in Projekten
  - planen selbständig Projekte und organisieren das Kickoff-Meeting
  - erstellen Anforderungen, Projektstrukturplan, Aufwandsschätzung, Aktivitäten-, Ressourcen- und Kostenplan
  - verstehen Projektkontrolle und Steuerung, Fortschrittsüberwachung, Besprechungen, Berichte, Änderungsmanagement
  - kennen die Grundzüge des Personalmanagements (Der Faktor Mensch, Teamwork, Führungsgrundsätze, Gesprächsstrategien, Konflikte lösen)
  - planen und steuern Änderungsmanagement (Konfigurationen, Änderungswünsche, Change Control Board, Built- und Release-Mechanismen)
  - setzen Qualitäts- und Risikomanagement ein (Qualitätsplan, Audits und Reviews, Risikoermittlung, Risikobewertung und Verfolgung, Gegenmaßnahmen)
  - kennen die wichtigsten Reifegrad Modelle und Standards (CMMI, SPiCE, ISO9001, ISO/IEC12207)
- 

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

**[1] Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule)

Dieses Modul ist daneben auch in den Studienfächern "International Information Systems (IIS) (Master of Science)" verwendbar.

---

**Studien-/Prüfungsleistungen:**

Software Projektmanagement (Prüfungsnummer: 312443)

(englische Bezeichnung: Software Project Management)

Prüfungsleistung, Portfolio

Anteil an der Berechnung der Modulnote: 100%

weitere Erläuterungen:

Aktive Mitarbeit in der Übung über Hausaufgaben und Kolloquium.

Erstablingung: WS 2019/2020, 1. Wdh.: WS 2020/2021

1. Prüfer: Oster/ Riehle/ Hindel/ Kips /Jung (ps0593)

---

**Modulbezeichnung:** **Organisation und Qualitätskontrolle im Software Engineering 5 ECTS**  
**- Option B (OrgB-SWE)**  
 (Management and Quality Assurance in Software Engineering -  
 Option B)

Modulverantwortliche/r: Francesca Saglietti

Lehrende: Francesca Saglietti, Marc Spisländer, Andreas Neubaum, Xiaochen Wu, Xiaochen Wu, Andreas Neubaum

Startsemester: SS 2019

Dauer: 2 Semester

Turnus: jährlich (SS)

Präsenzzeit: 60 Std.

Eigenstudium: 90 Std.

Sprache: Deutsch

### Lehrveranstaltungen:

#### Organisation und Qualitätskontrolle

Im Rahmen des Moduls ist das letztes Drittel der folgenden Vorlesung mit Übung zu belegen:

Grundlagen des Software Engineering (SS 2019, Vorlesung, 4 SWS, Francesca Saglietti)

Übungen zu Grundlagen des Software Engineering (SS 2019, Übung, 2 SWS, Marc Spisländer)

#### Seminar

Im Rahmen des Moduls ist eines der folgenden Seminare zu belegen:

Einführung in die Kryptografie (WS 2019/2020, Seminar, 2 SWS, Anwesenheitspflicht, Andreas Neubaum et al.)

Design Patterns und Anti-Patterns (SS 2019, Seminar, 2 SWS, Anwesenheitspflicht, Xiaochen Wu et al.)

### Inhalt:

Über analytischen Verfahren hinaus werden auch wesentliche Aspekte des Projektmanagements (Teamkoordination, Kostenschätzung, menschliche Faktoren) sowie ein breites Instrumentarium zur Messung quantitativer Indikatoren vorgestellt. Im Seminar Design Patterns und Anti-Patterns werden zahlreiche in der Praxis bewährte Entwurfsmuster präsentiert, sowie typische Fehler, die während des gesamten Lebenszyklus' eines Software-Systems auftreten können, zusammen mit ihrer Erkennung und Behebung vorgestellt. Im Seminar Einführung in die Kryptografie werden symmetrische und asymmetrische Verfahren zur Identifikation, Verschlüsselung und Signierung behandelt. Dazu werden sowohl die mathematischen Grundlagen vermittelt als auch die entsprechenden Algorithmen vorgestellt.

### Lernziele und Kompetenzen:

Die Studenten

- benutzen Software-Metriken zur Bestimmung der Software-Komplexität
- wenden unterschiedliche Nachweisverfahren, wie z.B. statische Analyse, funktionales und strukturelles Testen, oder Korrektheitsbeweise an
- stellen die Bedeutung der Wartung im Software-Entwicklungsprozess heraus
- erläutern verschiedene Aspekte des Software-Projektmanagements

Je nach belegtem Seminar:

- erläutern die Studenten etablierte Design Patterns sowie bekannte Anti-Patterns;
- klassifizieren sie und charakterisieren deren Stärken und Schwächen bzw. Negativfolgen oder:
- erläutern sie wesentliche Konzepte der modernen Kryptografie;
- klassifizieren sie Kryptoverfahren und charakterisieren deren Stärken und Schwächen;

### Literatur:

- Lehrbuch der Softwaretechnik (Band 1), Helmut Balzert, 2000
- Entwurfsmuster: Elemente wiederverwendbarer objektorientierter Software, Erich Gamma und Richard Helm und Ralph Johnson und John Vlissides, Addison-Wesley-Verlag, 2010
- Einführung in die Kryptographie, Johannes Buchmann, Springer Verlag, 2010

### Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

[1] Informatik (Bachelor of Science)

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule)

---

**Studien-/Prüfungsleistungen:**

Organisation und Qualitätskontrolle im modernen Software Engineering (mit Seminar Design Patterns und Anti-Patterns) (Prüfungsnummer: 152768)

Prüfungsleistung, Portfolio

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Grundlagen des Software Engineering
- Übungen zu Grundlagen des Software Engineering
- Design Patterns und Anti-Patterns

weitere Erläuterungen:

Die Bewertung der Prüfungsleistung setzt sich zusammen aus einer 30-minütigen schriftlichen Prüfung (Gewichtung 1/2) über die Grundlagen des Software Engineering (Teil Organisation und Qualitätskontrolle), einem Seminarvortrag (Gewichtung 1/4), und einer 15-minütigen mündlichen Prüfung (Gewichtung 1/4) über Seminarthemen. Alle drei Einzelleistungen müssen mit mindestens 4,0 bestanden werden.

Erstablingung: SS 2019, 1. Wdh.: WS 2019/2020

1. Prüfer: Francesca Saglietti (100112)

Organisation und Qualitätskontrolle im modernen Software Engineering (mit Seminar Einführung in die Kryptografie) (Prüfungsnummer: 755863)

Prüfungsleistung, Portfolio

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Grundlagen des Software Engineering
- Übungen zu Grundlagen des Software Engineering
- Einführung in die Kryptografie

weitere Erläuterungen:

Die Bewertung der Prüfungsleistung setzt sich zusammen aus einer 30-minütigen schriftlichen Prüfung (Gewichtung 1/2) über die Grundlagen des Software Engineering (Teil Organisation und Qualitätskontrolle), einem Seminarvortrag (Gewichtung 1/4), und einer 15-minütigen mündlichen Prüfung (Gewichtung 1/4) über Seminarthemen. Alle drei Einzelleistungen müssen mit mindestens 4,0 bestanden werden.

Erstablingung: SS 2019, 1. Wdh.: WS 2019/2020

1. Prüfer: Francesca Saglietti (100112)

---

---

**Modulbezeichnung:** **Erweiterte Grundlagen des Software Engineering - Option A (SWE-GL-Ext-A)** **10 ECTS**

(Extended Foundations of Software Engineering - Option A)

Modulverantwortliche/r: Francesca Saglietti

Lehrende: Francesca Saglietti

---

Startsemester: SS 2019

Dauer: 2 Semester

Turnus: jährlich (SS)

Präsenzzeit: 135 Std.

Eigenstudium: 165 Std.

Sprache: Deutsch

---

**Lehrveranstaltungen:**

Grundlagen des Software Engineering (SS 2019, Vorlesung, 4 SWS, Francesca Saglietti)

Übungen zu Grundlagen des Software Engineering (SS 2019, Übung, 2 SWS, Marc Spisländer)

**Praktische Übung**

Software Engineering in der Praxis (WS 2019/2020, Übung, 3 SWS, Xiaochen Wu et al.)

---

**Inhalt:**

Das Modul befasst sich mit einem breiten Spektrum an ingenieurwissenschaftlichen Prinzipien und alternativen Vorgehensweisen bei der Konzeption, Entwicklung, Analyse, Organisation und Pflege großer, komplexer Softwaresysteme. In praktischen Übungen werden Werkzeuge zur Entwicklung und Analyse komplexer Systeme vorgestellt, deren industrielle Einsetzbarkeit anschließend von den Teilnehmern anhand in der Praxis repräsentativer Aufgabenstellungen erprobt wird.

**Lernziele und Kompetenzen:**

Die Studierenden:

- erklären systematische und strukturierte Prozessmodelle (wie das Wasserfall- und V-Modell) zur Entwicklung komplexer Software-Systeme
- erfassen wesentliche Prinzipien der Software-Entwicklung (wie Kohäsion und Kopplung)
- erläutern systematische Methoden zur Anforderungsermittlung an und wesentliche Ziele der Anforderungsanalyse
- benutzen ausgewählte Spezifikations Sprachen (wie Endliche Automaten, Petri-Netze und Z), um komplexe Problemstellungen eindeutig zu formulieren
- wenden UML-Diagramme (wie Use Case-, Klassen-, Sequenz- und Kommunikationsdiagramme) zum Zweck objektorientierter Analyse- und Design-Aktivitäten an;
- beschreiben unterschiedliche Arten der Wiederverwendung von Entwurfselementen und lösen typische Entwurfsprobleme durch Anwendung etablierter Entwurfsmuster
- benutzen Software-Metriken zur Bestimmung der Software-Komplexität
- wenden unterschiedliche Nachweisverfahren, wie z.B. statische Analyse, funktionales und strukturelles Testen, oder Korrektheitsbeweise an
- stellen die Bedeutung der Wartung im Software-Entwicklungsprozess heraus
- erläutern verschiedene Aspekte des Software-Projektmanagements
- erläutern Potenzial und Grenzen unterschiedlicher Werkzeuge zur Unterstützung softwaretechnischer Tätigkeiten;
- wenden unterschiedliche Werkzeuge an, um sowohl selbständig als auch in Teams Beispielaufgaben aus dem Bereich der objektorientierten Analyse, des objektorientierten Entwurfs, des Testens, des Beweisens und des Projektmanagements zu lösen;

**Literatur:**

Lehrbuch der Softwaretechnik, Helmut Balzert, Spektrum Akad. Verlag, 2000

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

**[1] Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule)

Dieses Modul ist daneben auch in den Studienfächern "International Information Systems (IIS) (Master of Science)" verwendbar.



**Studien-/Prüfungsleistungen:**

Grundlagen des Software Engineering mit prakt. Übungen Software Engineering in der Praxis (Prüfungsnummer: 234129)

Prüfungsleistung, Portfolio

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Grundlagen des Software Engineering
- Übungen zu Grundlagen des Software Engineering
- Software Engineering in der Praxis

weitere Erläuterungen:

Die Bewertung der Prüfungsleistung setzt sich zusammen aus einer 90-minütigen schriftlichen Prüfung (Gewichtung 3/4) über die Grundlagen des Software Engineering und einer 60-minütigen Prüfung am Rechner (Gewichtung 1/4) über Software Engineering in der Praxis. Beide Einzelleistungen müssen mit mindestens 4,0 bestanden werden.

Erstablingung: SS 2019, 1. Wdh.: WS 2019/2020

1. Prüfer: Francesca Saglietti (100112)

---

---

**Modulbezeichnung:** **Konstruktives Software Engineering (Konstr-SWE)** **5 ECTS**  
(Constructive Phases of Software Engineering)

Modulverantwortliche/r: Francesca Saglietti  
Lehrende: Francesca Saglietti

---

Startsemester: SS 2019	Dauer: 1 Semester	Turnus: jährlich (SS)
Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Deutsch oder Englisch

---

**Lehrveranstaltungen:**

Das Modul besteht aus den ersten zwei Monaten der Vorlesung mit Übung Grundlagen des Software Engineering.  
Grundlagen des Software Engineering (SS 2019, Vorlesung, 4 SWS, Francesca Saglietti)  
Übungen zu Grundlagen des Software Engineering (SS 2019, Übung, 2 SWS, Marc Spisländer)

---

**Inhalt:**

Das Modul befasst sich mit einem breiten Spektrum an ingenieurwissenschaftlichen Prinzipien und alternativen Vorgehensweisen bei Konzeption, Entwicklung und Pflege großer, komplexer Softwaresysteme. Sie bietet eine umfassende Übersicht konstruktiver Verfahren des modernen Software Engineering an.

**Lernziele und Kompetenzen:**

Die Studierenden:

- erklären systematische und strukturierte Prozessmodelle (wie das Wasserfall- und V-Modell) zur Entwicklung komplexer Software-Systeme
- erfassen wesentliche Prinzipien der Software-Entwicklung (wie Kohäsion und Kopplung)
- erläutern systematische Methoden zur Anforderungsermittlung an und wesentliche Ziele der Anforderungsanalyse
- benutzen ausgewählte Spezifikationsprachen (wie Endliche Automaten, Petri-Netze und Z), um komplexe Problemstellungen eindeutig zu formulieren
- wenden UML-Diagramme (wie Use Case-, Klassen-, Sequenz- und Kommunikationsdiagramme) zum Zweck objektorientierter Analyse- und Design-Aktivitäten an;
- beschreiben unterschiedliche Arten der Wiederverwendung von Entwurfselementen und lösen typische Entwurfsprobleme durch Anwendung etablierter Entwurfsmuster

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

**[1] Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule)

Dieses Modul ist daneben auch in den Studienfächern "International Information Systems (IIS) (Master of Science)", "Maschinenbau (Master of Science)", "Wirtschaftsingenieurwesen (Master of Science)" verwendbar.

---

**Studien-/Prüfungsleistungen:**

Konstruktives Software Engineering (Prüfungsnummer: 471229)

Prüfungsleistung, Klausur, Dauer (in Minuten): 60

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Grundlagen des Software Engineering
- Übungen zu Grundlagen des Software Engineering

weitere Erläuterungen:

Die Prüfung erstreckt sich nur über den Teil Konstruktive Phasen der Vorlesung mit Übung Grundlagen des Software Engineering.

Erstablingung: SS 2019, 1. Wdh.: WS 2019/2020

1. Prüfer: Francesca Saglietti (100112)

## Konstruktives Software Engineering

(englische Bezeichnung: Constructive Phases of Software Engineering)

Prüfungsleistung, Klausur, Dauer (in Minuten): 60

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Grundlagen des Software Engineering
- Übungen zu Grundlagen des Software Engineering

weitere Erläuterungen:

Die Prüfung erstreckt sich nur über den Teil Konstruktive Phasen der Vorlesung mit Übung Grundlagen des Software Engineering.

Erstabelleung: SS 2019, 1. Wdh.: WS 2019/2020

1. Prüfer: Francesca Saglietti (100112)

---

### Organisatorisches:

Das Modul umfasst die ersten zwei Monate der Lehrveranstaltung »Grundlagen des Software Engineering« (Vorlesung und Übung) im Umfang von 5 ECTS.

**Modulbezeichnung:** **Software-Anwendungen mit KI (VUE 10-ECTS) (OSS-SAKI) 10 ECTS**  
 (Software-Applications with AI (VUE 10-ECTS))

Modulverantwortliche/r: Dirk Riehle

Lehrende: Dirk Riehle

Startsemester: SS 2019

Dauer: 1 Semester

Turnus: jährlich (SS)

Präsenzzeit: 60 Std.

Eigenstudium: 240 Std.

Sprache: Deutsch

### Lehrveranstaltungen:

Software-Anwendungen mit KI (VUE) (SS 2019, Vorlesung mit Übung, 2 SWS, Dirk Riehle)

Introduction to Complex Data Analysis in Python (KU) (SS 2019, Kurs, Michael Dorner et al.)

### Empfohlene Voraussetzungen:

Grundlagenveranstaltungen in maschinellem Lernen und künstlicher Intelligenz

### Inhalt:

Dieser Kurs lehrt fortgeschrittene Methoden des maschinellen Lernens resp. der künstlichen Intelligenz anhand von vier nicht-trivialen Anwendungsbeispielen mit realen Daten aus der Industrie.

Jedes der vier Beispiele stellt eine umfangreiche Hausaufgabe für Studierende dar, in der unterschiedliche Problemarten (Korrelation, Klassifikation, etc.) mit unterschiedlichen Methoden (Clustering, Bayesian Networks, etc.) in unterschiedlichen Fachgebieten (Automobilindustrie, Finanzindustrie, etc.) kombiniert werden. Jede Aufgabe wird von einem dazugehörigen Industriepartner mitbetreut.

Die vier Beispiele werden nacheinander abgearbeitet und strukturieren die Kurszeit in vier gleich große Abschnitte von jeweils drei Wochen, von denen jeder Abschnitt dieselbe Struktur hat:

- 1) Vorbereitung auf den anstehenden Abschnitt durch Wiederholung relevanter Literatur
- 2) Einführung in das Problem; Diskussion von Bibliotheken und Vorgehen zur Problemlösung
- 3) Wiederholte Diskussion (zwei weitere Sitzungen) des Problems und der Herangehensweise
- 4) Abgabe der Problemlösung, bestehend aus Erläuterung sowie Quelltext und Ergebnissen

Die Programmierung findet in Python statt. Eine Woche vor Kursbeginn können Studierende, welche Python noch nicht kennen oder ihre Kenntnisse auffrischen wollen, an unserem Einführungskurs zu Python und ML teilnehmen, siehe <https://goo.gl/X6j4nZ>.

Es wird erwartet, dass Studierende aktiv mitarbeiten, sich etwaige fehlende Grundlagen selbst aneignen, und die technischen Aufgaben eigenständig lösen werden. Der Zeitplan ist unter <https://goo.gl/5ynxcQ> einsehbar. Bitte registrieren Sie sich für den Kurs auf StudOn (Verweis über Zeitplan verfügbar).

### Lernziele und Kompetenzen:

Erwerb praktischer Kompetenz in der Anwendung von ML und KI-Methoden auf reale industrielle Daten

### Literatur:

(Literaturliste wird zeitnah bereitgestellt)

### Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

#### [1] Informatik (Bachelor of Science)

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule)

Dieses Modul ist daneben auch in den Studienfächern "International Information Systems (IIS) (Master of Science)" verwendbar.

### Studien-/Prüfungsleistungen:

Software-Anwendungen mit KI (VUE 10-ECTS) (Prüfungsnummer: 992696)

(englische Bezeichnung: Software-Applications with AI (VUE 10-ECTS))

Prüfungsleistung, Portfolio

Anteil an der Berechnung der Modulnote: 100%

weitere Erläuterungen:

Gleichgewichtete Summe der Noten der Hausaufgaben

Erstablingung: SS 2019, 1. Wdh.: keine Angabe  
1. Prüfer: Dirk Riehle (100100)

---

---

**Modulbezeichnung:** **Praktische Softwaretechnik (PSWT-PSWT)** **5 ECTS**  
(Applied Software Engineering)

Modulverantwortliche/r: Dirk Riehle, Bernd Hindel, Detlef Kips

Lehrende: Detlef Kips, Martin Jung, Bernd Hindel, Dirk Riehle, Norbert Oster

---

Startsemester: WS 2019/2020	Dauer: 1 Semester	Turnus: jährlich (WS)
Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Deutsch oder Englisch

---

**Lehrveranstaltungen:**

Praktische Softwaretechnik (WS 2019/2020, Vorlesung, 4 SWS, Detlef Kips et al.)

---

**Inhalt:**

Software ist überall und Software ist komplex. Nicht triviale Software wird von Teams entwickelt. Oft müssen bei der Entwicklung von Softwaresystemen eine Vielzahl von funktionalen und nicht funktionalen Anforderungen berücksichtigt werden. Hierfür ist eine disziplinierte und ingenieurmäßige Vorgehensweise notwendig.

Die Vorlesung "Praktische Softwaretechnik" soll ...

- ein Bewusstsein für die typischen Problemstellungen schaffen, die bei der Durchführung umfangreicher Softwareentwicklungsprojekte auftreten,
- ein breites Basiswissen über die Konzepte, Methoden, Notationen und Werkzeuge der modernen Softwaretechnik vermitteln und
- die Möglichkeiten und Grenzen ihres Einsatzes im Kontext realistischer Projektumgebungen anhand praktischer Beispiele demonstrieren und bewerten.

Die Vorlesung adressiert inhaltlich alle wesentlichen Bereiche der Softwaretechnik. Vorgestellt werden unter anderem

- traditionelle sowie agile Methoden der Softwareentwicklung,
- Methoden der Anforderungsanalyse und des Systementwurfs,
- Konzepte der Softwarearchitektur, -implementierung und Dokumentation und
- Testen und Qualitätssicherung sowie Prozessverbesserung.

Weitere Materialien und Informationen sind hier zu finden:

- Zeitplan: <http://goo.gl/0fy1T>
- Materialien: Auf StudOn über den Zeitplan

Die Teilnahme ist begrenzt. Bitte registrieren Sie sich zeitig für den Kurs auf StudOn, um sicherzustellen, dass Sie einen Platz erhalten.

**Lernziele und Kompetenzen:**

Die Studierenden

- verstehen den Unterschied zwischen "Programmieren im Kleinen" und "Programmieren im Großen" (Softwaretechnik)
- wenden grundlegende Methoden der Softwaretechnik über den gesamten Projekt- und Produktlebenszyklus an
- kennen die Rolle und Zuständigkeiten der Berufsbilder "Projektleiter", "Anforderungsermittler", "Softwareentwickler" und "Qualitätssicherer"

**Literatur:**

siehe <http://goo.gl/JSoUbV>

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

**[1] Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule)

Dieses Modul ist daneben auch in den Studienfächern "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Bachelor of Science)", "International Information Systems (IIS) (Master of Science)" verwendbar.

---

**Studien-/Prüfungsleistungen:**

Praktische Softwaretechnik (Prüfungsnummer: 816763)

Prüfungsleistung, Klausur, Dauer (in Minuten): 90

Anteil an der Berechnung der Modulnote: 100%

Erstablingung: WS 2019/2020, 1. Wdh.: WS 2020/2021

1. Prüfer: Oster/ Riehle/ Hindel/ Kips /Jung (ps0593)

---

---

**Modulbezeichnung:** **Advanced Design and Programming (5-ECTS) (OSS-ADAP) 5 ECTS**  
 (Advanced Design and Programming (5-ECTS))

Modulverantwortliche/r: Dirk Riehle

Lehrende: Dirk Riehle

---

Startsemester: WS 2019/2020

Dauer: 1 Semester

Turnus: jährlich (WS)

Präsenzzeit: 60 Std.

Eigenstudium: 90 Std.

Sprache: Deutsch

---

**Lehrveranstaltungen:**

Advanced Design and Programming (VUE) (WS 2019/2020, Vorlesung mit Übung, 4 SWS, Dirk Riehle)

---

**Vorhergehende Module:**

Algorithmen und Datenstrukturen

---

**Inhalt:**

This course teaches principles and practices of advanced object-oriented design and programming.

Dieser Kurs wird auf Deutsch gehalten.

It consists of a weekly lecture with exercises, homework and self-study, totaling 4 SWS, 5 ECTS. This is a hands-on course and students should be familiar with their Java IDE. Students learn the following concepts:

*Class-Level*

- Method design
- Class design
- Classes and interfaces
- Subtyping and inheritance
- Implementing inheritance
- Design by contract

*Collaboration-Level*

- Values vs. objects
- Role objects
- Type objects
- Object creation
- Collaboration-based design
- Design patterns

*Component-Level*

- Error handling
- Meta-object protocols
- Frameworks and components
- Domain-driven design
- API evolution

The running example is the photo sharing and rating software Wahlzeit, see <https://github.com/dirkriehle/wahlzeit>.

Class is held as a three hour session with a short break in between. The class iterates over short lectures, discussion, and exercise chunks of 10-30min each. Students should bring a laptop with a well-working Java programming setup.

The overall schedule can be found at <http://goo.gl/bePPn>. Please sign up for the course on StudOn (link accessible through schedule spreadsheet) as soon as possible.

**Lernziele und Kompetenzen:**

- Learn to recognize, analyse, and apply advanced concepts of object-oriented design and programming
- Learn to work effectively with a realistic tool set-up, involving an IDE, configuration management, and a service hoster

**Literatur:**



- See <http://goo.gl/BZpU8>

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

**[1] Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule)

Dieses Modul ist daneben auch in den Studienfächern "International Information Systems (IIS) (Master of Science)", "Maschinenbau (Master of Science)" verwendbar.

---

**Studien-/Prüfungsleistungen:**

Advanced Design and Programming (5-ECTS) (Prüfungsnummer: 881857)

(englische Bezeichnung: Advanced Design and Programming (5-ECTS))

Prüfungsleistung, Portfolio

Anteil an der Berechnung der Modulnote: 100%

weitere Erläuterungen:

- Exercise participation
- Homework assignments

Erstablingung: WS 2019/2020, 1. Wdh.: keine Angabe

1. Prüfer: Dirk Riehle (100100)

---

---

**Modulbezeichnung:** Product Management (VUE+PROJ 10-ECTS) (OSS-PROD-VUE+PROJ) 10 ECTS  
 (Product Management (VUE+PROJ 10-ECTS))

Modulverantwortliche/r: Dirk Riehle  
 Lehrende: Dirk Riehle

---

Startsemester: WS 2019/2020	Dauer: 1 Semester	Turnus: jährlich (WS)
Präsenzzeit: 60 Std.	Eigenstudium: 240 Std.	Sprache: Englisch

---

**Lehrveranstaltungen:**

Product Management (VUE) (WS 2019/2020, Vorlesung mit Übung, 2 SWS, Dirk Riehle)  
 Product Management (PROJ) (WS 2019/2020, Sonstige Lehrveranstaltung, 2 SWS, Dirk Riehle)

---

**Inhalt:**

This course teaches students the concepts, methods, and tools of software product management. Product management is an important function in software development organizations. A product manager conceives and defines new products. His or her task is to understand the market incl. customers, to develop a product vision from that understanding, to translate it into product requirements, define those requirements on a by-feature basis and work with engineering to ensure these features are properly realized in the product under development.

- Role, tasks, and responsibilities of a product manager
- Process, methods, techniques and tools of product management
- Managing incremental/sustaining as well as disruptive innovation
- Open source product management; new trends in product management

Students can choose one or both of two components:

- VUE (lecture + homework), 2 SWS, 5 ECTS. VUE uses teaching cases as commonly used in MBA programs. The teaching cases are available for free at <http://pmbycase.com>.
- PROJ (small project), 2 SWS, 5 ECTS. In PROJ, students perform a small product management project, either individually or in teams. In such a project, students write a business plan for a market opportunity or a startup. There are two sources of projects: Those supplied by the professorship and those supplied by students. If we supply any projects, we will tell you about them in the first lecture. Student-supplied projects can be suggested by students to Prof. Riehle at least one week before class starts and must meet the requirements described here: <https://goo.gl/sNPi1i>

PROD projects are run as shared projects, in which all participants contribute and get to participate in the project results. Read more at <https://wp.me/pDU66-2p4>.

Class is run as two 90min blocks. The first block discusses the teaching cases. The second block is a coaching session for the projects (10 ECTS only).

The overall schedule can be found at <http://goo.gl/tTA10> . Please sign up for the course on StudOn (link accessible through schedule spreadsheet) as soon as possible.

**Lernziele und Kompetenzen:**

- Understand the role, function, and responsibilities of a product manager
- Understand key concepts, methods, and tools of software product management
- Understand different business situations, incl. Incremental vs. disruptive innovation

**Literatur:**

- <http://goo.gl/41DgSr>

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

[1] **Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule)

Dieses Modul ist daneben auch in den Studienfächern "International Information Systems (IIS) (Master of Science)" verwendbar.

**Studien-/Prüfungsleistungen:**

Product Management (VUE+PROJ 10-ECTS) (Prüfungsnummer: 863131)

(englische Bezeichnung: Product Management (VUE+PROJ 10-ECTS))

Prüfungsleistung, Portfolio

Anteil an der Berechnung der Modulnote: 100%

weitere Erläuterungen:

- In-class participation
- Homework assignments
- Project work

Erstablingung: WS 2019/2020, 1. Wdh.: keine Angabe

1. Prüfer: Dirk Riehle (100100)

---

---

**Modulbezeichnung:** Grundlagen des Software Engineering (SWEGL) 7.5 ECTS  
 (Foundations of Software Engineering)

Modulverantwortliche/r: Francesca Saglietti  
 Lehrende: Francesca Saglietti

---

Startsemester: SS 2019	Dauer: 1 Semester	Turnus: jährlich (SS)
Präsenzzeit: 90 Std.	Eigenstudium: 135 Std.	Sprache: Deutsch

---

**Lehrveranstaltungen:**

Grundlagen des Software Engineering (SS 2019, Vorlesung, 4 SWS, Francesca Saglietti)  
 Übungen zu Grundlagen des Software Engineering (SS 2019, Übung, 2 SWS, Marc Spisländer)

---

**Inhalt:**

Das Modul befasst sich mit einem breiten Spektrum an ingenieurwissenschaftlichen Prinzipien und alternativen Vorgehensweisen bei der Konzeption, Entwicklung, Analyse, Organisation und Pflege großer, komplexer Softwaresysteme.

**Lernziele und Kompetenzen:**

Die Studierenden:

- erklären systematische und strukturierte Prozessmodelle (wie das Wasserfall- und V-Modell) zur Entwicklung komplexer Software-Systeme
- erfassen wesentliche Prinzipien der Software-Entwicklung (wie Kohäsion und Kopplung)
- erläutern systematische Methoden zur Anforderungsermittlung an und wesentliche Ziele der Anforderungsanalyse
- benutzen ausgewählte Spezifikationssprachen (wie Endliche Automaten, Petri-Netze und Z), um komplexe Problemstellungen eindeutig zu formulieren
- wenden UML-Diagramme (wie Use Case-, Klassen-, Sequenz- und Kommunikationsdiagramme) zum Zweck objektorientierter Analyse- und Design-Aktivitäten an;
- beschreiben unterschiedliche Arten der Wiederverwendung von Entwurfselementen und lösen typische Entwurfsprobleme durch Anwendung etablierter Entwurfsmuster
- benutzen Software-Metriken zur Bestimmung der Software-Komplexität
- wenden unterschiedliche Nachweisverfahren, wie z.B. statische Analyse, funktionales und strukturelles Testen, oder Korrektheitsbeweise an
- stellen die Bedeutung der Wartung im Software-Entwicklungsprozess heraus
- erläutern verschiedene Aspekte des Software-Projektmanagements

**Literatur:**

Lehrbuch der Softwaretechnik, Helmut Balzert, Spektrum Akad. Verlag, 2000

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

[1] **Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule)

Dieses Modul ist daneben auch in den Studienfächern "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Bachelor of Science)", "Informatik (Master of Science)", "Informations- und Kommunikationstechnik (Bachelor of Science)", "Mathematik (Bachelor of Science)", "Mechatronik (Bachelor of Science)", "Mechatronik (Master of Science)" verwendbar.

---

**Studien-/Prüfungsleistungen:**

Grundlagen des Software Engineering (Vorlesung mit Übungen) (Prüfungsnummer: 35501)

Prüfungsleistung, Klausur, Dauer (in Minuten): 90

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Grundlagen des Software Engineering
- Übungen zu Grundlagen des Software Engineering

Erstablingung: SS 2019, 1. Wdh.: WS 2019/2020  
1. Prüfer: Francesca Saglietti (100112)

---

---

**Modulbezeichnung:** **Erweiterte Grundlagen des Software Engineering - Option B (SWE-GL-Ext-B)** **10 ECTS**

(Extended Foundations of Software Engineering - Option B)

Modulverantwortliche/r: Francesca Saglietti

Lehrende: Francesca Saglietti

---

Startsemester: SS 2019

Dauer: 2 Semester

Turnus: halbjährlich (WS+SS)

Präsenzzeit: 120 Std.

Eigenstudium: 180 Std.

Sprache: Deutsch

---

### Lehrveranstaltungen:

Es sind die Vorlesung mit Übung sowie ein Seminar zu belegen

#### **Vorlesung mit Übung**

Grundlagen des Software Engineering (SS 2019, Vorlesung, 4 SWS, Francesca Saglietti)

Übungen zu Grundlagen des Software Engineering (SS 2019, Übung, 2 SWS, Marc Spisländer)

#### **Seminar**

Design Patterns und Anti-Patterns (SS 2019, Seminar, 2 SWS, Anwesenheitspflicht, Xiaochen Wu et al.)

Einführung in die Kryptografie (WS 2019/2020, Seminar, 2 SWS, Anwesenheitspflicht, Andreas Neubaum et al.)

---

### Inhalt:

Das Modul befasst sich mit einem breiten Spektrum an ingenieurwissenschaftlichen Prinzipien und alternativen Vorgehensweisen bei der Konzeption, Entwicklung, Analyse, Organisation und Pflege großer, komplexer Softwaresysteme unter besonderer Berücksichtigung neuartiger, musterbasierter Entwurfsmethoden.

### Lernziele und Kompetenzen:

Die Studierenden:

- erklären systematische und strukturierte Prozessmodelle (wie das Wasserfall- und V-Modell) zur Entwicklung komplexer Software-Systeme
- erfassen wesentliche Prinzipien der Software-Entwicklung (wie Kohäsion und Kopplung)
- erläutern systematische Methoden zur Anforderungsermittlung an und wesentliche Ziele der Anforderungsanalyse
- benutzen ausgewählte Spezifikationssprachen (wie Endliche Automaten, Petri-Netze und Z), um komplexe Problemstellungen eindeutig zu formulieren
- wenden UML-Diagramme (wie Use Case-, Klassen-, Sequenz- und Kommunikationsdiagramme) zum Zweck objektorientierter Analyse- und Design-Aktivitäten an;
- beschreiben unterschiedliche Arten der Wiederverwendung von Entwurfselementen und lösen typische Entwurfsprobleme durch Anwendung etablierter Entwurfsmuster
- benutzen Software-Metriken zur Bestimmung der Software-Komplexität
- wenden unterschiedliche Nachweisverfahren, wie z.B. statische Analyse, funktionales und strukturelles Testen, oder Korrektheitsbeweise an
- stellen die Bedeutung der Wartung im Software-Entwicklungsprozess heraus
- erläutern verschiedene Aspekte des Software-Projektmanagements

Je nach belegtem Seminar:

- erläutern die Studenten etablierte Design Patterns sowie bekannte Anti-Patterns;
  - klassifizieren sie und charakterisieren deren Stärken und Schwächen bzw. Negativfolgen
- oder:

- erläutern sie wesentliche Konzepte der modernen Kryptografie;
- klassifizieren sie Kryptoverfahren und charakterisieren deren Stärken und Schwächen;

### Literatur:

- Lehrbuch der Softwaretechnik, Helmut Balzert, Spektrum Akad. Verlag, 2000
- Entwurfsmuster: Elemente wiederverwendbarer objektorientierter Software, Erich Gamma und Richard Helm und Ralph Johnson und John Vlissides, Addison-Wesley-Verlag, 2010

---

### Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

#### [1] Informatik (Bachelor of Science)

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule)

Dieses Modul ist daneben auch in den Studienfächern "International Information Systems (IIS) (Master of Science)" verwendbar.

---

### Studien-/Prüfungsleistungen:

Erweiterte Grundlagen des Software Engineering - Option B (Prüfungsnummer: 113545)

(englische Bezeichnung: Extended Foundations of Software Engineering - Option B)

Prüfungsleistung, Portfolio

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Grundlagen des Software Engineering
- Übungen zu Grundlagen des Software Engineering
- Design Patterns und Anti-Patterns

weitere Erläuterungen:

Die Bewertung der Prüfungsleistung setzt sich zusammen aus einer 90-minütigen schriftlichen Prüfung (Gewichtung 3/4) über die Grundlagen des Software Engineering, einem Seminarvortrag (Gewichtung 1/8) und einer 15-minütigen mündlichen Prüfung (Gewichtung 1/8) über die Seminarthemen. Alle drei Einzelleistungen müssen mit mindestens 4,0 bestanden werden. Nach etwa der Hälfte der Vorlesungszeit wird die Teilnahme an einer freiwilligen Klausur (45 Minuten) über die Vorlesung mit Übungen Grundlagen des Software Engineering angeboten. Die dort erzielten Punkte können die Note der schriftlichen Prüfung um bis zu 0,7 verbessern.

Erstablingung: SS 2019, 1. Wdh.: WS 2019/2020

1. Prüfer: Francesca Saglietti (100112)

Grundlagen des Software Engineering mit Seminar Einführung in die Kryptografie (Prüfungsnummer: 140760)

Prüfungsleistung, Portfolio

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Grundlagen des Software Engineering
- Übungen zu Grundlagen des Software Engineering
- Einführung in die Kryptografie

weitere Erläuterungen:

Die Bewertung der Prüfungsleistung setzt sich zusammen aus einer 90-minütigen schriftlichen Prüfung (Gewichtung 3/4) über die Grundlagen des Software Engineering, einem Seminarvortrag (Gewichtung 1/8) und einer 15-minütigen mündlichen Prüfung (Gewichtung 1/8) über die Seminarthemen. Alle drei Einzelleistungen müssen mit mindestens 4,0 bestanden werden.

Erstablingung: WS 2019/2020, 1. Wdh.: SS 2020

1. Prüfer: Francesca Saglietti (100112)

---

---

**Modulbezeichnung:** Bewertung der Softwarezuverlässigkeit (SWE-ZUV) 5 ECTS  
(Software Reliability)

Modulverantwortliche/r: Francesca Saglietti  
Lehrende: Francesca Saglietti, Maria Bonner

---

Startsemester: SS 2019	Dauer: 1 Semester	Turnus: jährlich (SS)
Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Deutsch

---

**Lehrveranstaltungen:**

Softwarezuverlässigkeit (SS 2019, Vorlesung, 2 SWS, Francesca Saglietti)  
Übungen zu Softwarezuverlässigkeit (SS 2019, Übung, 2 SWS, Marc Spisländer)

---

**Inhalt:**

Das Modul befasst sich schwerpunktmäßig mit quantitativen Verfahren zur Bestimmung der erzielten Zuverlässigkeit eines Softwaresystems. Dies wird durch Berichte aktueller Erfahrungen aus der industriellen Entwicklungs- und Genehmigungspraxis abgerundet und ergänzt.

**Lernziele und Kompetenzen:**

Die Studierenden:

- wenden Konzepte der Zuverlässigkeitstheorie an
- unterscheiden zwischen verschiedenen Teststrategien (wie z.B. Zufallstest, struktureller und funktionaler Test)
- unterscheiden zwischen verschiedenen Zuverlässigkeitswachstumsmodellen und erläutern Möglichkeiten zur Verbesserung deren Vorhersagegenauigkeit
- wenden Hypothesentests (statistische Stichprobentheorie und sequentieller Wahrscheinlichkeitsverhältnis-Test) zur Ableitung quantitativer Aussagen zur Softwarezuverlässigkeit an.

**Literatur:**

Handbook of Software Reliability Engineering, Michael R. Lyu, 1995

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

**[1] Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule)

Dieses Modul ist daneben auch in den Studienfächern "International Information Systems (IIS) (Master of Science)" verwendbar.

---

**Studien-/Prüfungsleistungen:**

Softwarezuverlässigkeit (Vorlesung mit Übung) (Prüfungsnummer: 357823)

Prüfungsleistung, mündliche Prüfung, Dauer (in Minuten): 30

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Softwarezuverlässigkeit
- Übungen zu Softwarezuverlässigkeit

Prüfungssprache: Deutsch oder Englisch

Erstabledung: SS 2019, 1. Wdh.: WS 2019/2020

1. Prüfer: Francesca Saglietti (100112)

---



---

**Modulbezeichnung:** **Software Architecture (VUE+PROJ 10-ECTS) (OSS-ARCH-VUE+PROJ)** **10 ECTS**  
 (Software Architecture (VUE+PROJ 10-ECTS))

Modulverantwortliche/r: Martin Jung, Dirk Riehle  
 Lehrende: Martin Jung, Dirk Riehle

---

Startsemester: SS 2019	Dauer: 1 Semester	Turnus: jährlich (SS)
Präsenzzeit: 60 Std.	Eigenstudium: 240 Std.	Sprache: Deutsch

---

**Lehrveranstaltungen:**

Software Architecture (PROJ) (SS 2019, Sonstige Lehrveranstaltung, 2 SWS, Martin Jung et al.)  
 Softwarearchitektur (SS 2019, Vorlesung mit Übung, 4 SWS, Martin Jung et al.)

---

**Inhalt:**

This course teaches students concepts, methods, and tools of software architecture.

Die Architektur eines Softwaresystems beschreibt die wesentlichen Komponenten des Systems, ihre Beziehungen und Struktur, sowie das Verhalten und die Dynamik der Beziehungen und Struktur dieser Komponenten. Dieser Kurs vermittelt in einer Vorlesung zunächst die folgenden Aspekte von Softwarearchitektur:

- Grundlegende Bausteine und ihre Beziehungen
- Softwarearchitekturbeschreibungssprachen
- Softwarearchitekturstile und -muster
- Bibliotheken, Rahmenwerke und Plattformen
- Formale sowie de-facto Industriestandards
- Die Softwarearchitekturen von Beispielsystemen
- Nicht technische Kriterien in der Architektur
- Werkzeuge für Softwarearchitekten
- Vorgehensmodelle der Softwarearchitektur
- Architekturgetriebene Entwicklung
- Die Rolle und Funktion der Softwarearchitektin

Studierende können eine oder beide von zwei Komponenten wählen:

- VUE (Vorlesung + Übungen), 4 SWS, 5 ECTS
- PROJ (kleines Projekt), 2 SWS, 5 ECTS. Die Projekte werden von unseren Industriepartnern bereitgestellt. Hier dokumentieren, analysieren und bewerten Studierende die Softwarearchitektur eines realen Softwaresystems. Diese verschiedenen Aspekte werden im Laufe des Semesters inkrementell abgearbeitet und am Ende dem Industriepartner in einer Präsentation vorgestellt.

Der Unterricht findet als 3h-Block während der Vorlesungszeit statt. Der Zeitplan befindet sich hier: <http://goo.gl/ZXJjg> . Der Zeitplan enthält auch einen Link auf den zur Veranstaltung gehörigen StudOn Kurs. Bitte registrieren Sie sich auf StudOn sobald wie möglich.

**Lernziele und Kompetenzen:**

- Ganzheitliches Verständnis des Konzepts "Softwarearchitektur"
- Befähigung zur Bewertung, Auswahl und Konstruktion problemangemessener Architekturen
- Kenntnis architekturgetriebener Entwicklungsmethodik und entsprechender Werkzeuge
- Kenntnis der typischen Verantwortlichkeiten und der Methodik eines Softwarearchitekten

**Literatur:**

- <http://goo.gl/ou7mja>

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

**[1] Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule)

Dieses Modul ist daneben auch in den Studienfächern "International Information Systems (IIS) (Master of Science)" verwendbar.

**Studien-/Prüfungsleistungen:**

Software Architecture (VUE+PROJ 10-ECTS) (Prüfungsnummer: 733977)

(englische Bezeichnung: Software Architecture (VUE+PROJ 10-ECTS))

Prüfungsleistung, Portfolio

Anteil an der Berechnung der Modulnote: 100%

weitere Erläuterungen:

- Unterricht
- Hausaufgaben
- Mündliche Prüfung
- Projektarbeit

Erstablingung: SS 2019, 1. Wdh.: keine Angabe

1. Prüfer: Martin Jung (100803), 2. Prüfer: Dirk Riehle (100100)

---

---

**Modulbezeichnung:** Grundlagen des Software Engineering (SWEGL) 7.5 ECTS  
 (Foundations of Software Engineering)

Modulverantwortliche/r: Francesca Saglietti  
 Lehrende: Francesca Saglietti

---

Startsemester: SS 2019	Dauer: 1 Semester	Turnus: jährlich (SS)
Präsenzzeit: 90 Std.	Eigenstudium: 135 Std.	Sprache: Deutsch

---

**Lehrveranstaltungen:**

Grundlagen des Software Engineering (SS 2019, Vorlesung, 4 SWS, Francesca Saglietti)  
 Übungen zu Grundlagen des Software Engineering (SS 2019, Übung, 2 SWS, Marc Spisländer)

---

**Inhalt:**

Das Modul befasst sich mit einem breiten Spektrum an ingenieurwissenschaftlichen Prinzipien und alternativen Vorgehensweisen bei der Konzeption, Entwicklung, Analyse, Organisation und Pflege großer, komplexer Softwaresysteme.

**Lernziele und Kompetenzen:**

Die Studierenden:

- erklären systematische und strukturierte Prozessmodelle (wie das Wasserfall- und V-Modell) zur Entwicklung komplexer Software-Systeme
- erfassen wesentliche Prinzipien der Software-Entwicklung (wie Kohäsion und Kopplung)
- erläutern systematische Methoden zur Anforderungsermittlung an und wesentliche Ziele der Anforderungsanalyse
- benutzen ausgewählte Spezifikationssprachen (wie Endliche Automaten, Petri-Netze und Z), um komplexe Problemstellungen eindeutig zu formulieren
- wenden UML-Diagramme (wie Use Case-, Klassen-, Sequenz- und Kommunikationsdiagramme) zum Zweck objektorientierter Analyse- und Design-Aktivitäten an;
- beschreiben unterschiedliche Arten der Wiederverwendung von Entwurfselementen und lösen typische Entwurfsprobleme durch Anwendung etablierter Entwurfsmuster
- benutzen Software-Metriken zur Bestimmung der Software-Komplexität
- wenden unterschiedliche Nachweisverfahren, wie z.B. statische Analyse, funktionales und strukturelles Testen, oder Korrektheitsbeweise an
- stellen die Bedeutung der Wartung im Software-Entwicklungsprozess heraus
- erläutern verschiedene Aspekte des Software-Projektmanagements

**Literatur:**

Lehrbuch der Softwaretechnik, Helmut Balzert, Spektrum Akad. Verlag, 2000

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

[1] **Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule | Vertiefungsrichtung Software Engineering)

Dieses Modul ist daneben auch in den Studienfächern "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Bachelor of Science)", "Informatik (Master of Science)", "Informations- und Kommunikationstechnik (Bachelor of Science)", "Mathematik (Bachelor of Science)", "Mechatronik (Bachelor of Science)", "Mechatronik (Master of Science)" verwendbar.

---

**Studien-/Prüfungsleistungen:**

Grundlagen des Software Engineering (Vorlesung mit Übungen) (Prüfungsnummer: 35501)

Prüfungsleistung, Klausur, Dauer (in Minuten): 90

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Grundlagen des Software Engineering
- Übungen zu Grundlagen des Software Engineering

Erstablingung: SS 2019, 1. Wdh.: WS 2019/2020  
1. Prüfer: Francesca Saglietti (100112)

---

---

**Modulbezeichnung:** High End Simulation in Practice (HESP) 7.5 ECTS  
 (High End Simulation in Practice)

Modulverantwortliche/r: Harald Köstler  
 Lehrende: Harald Köstler

---

Startsemester: SS 2019	Dauer: 1 Semester	Turnus: jährlich (SS)
Präsenzzeit: 90 Std.	Eigenstudium: 135 Std.	Sprache: Englisch

---

**Lehrveranstaltungen:**

High End Simulation in Practice (SS 2019, Vorlesung, 2 SWS, Harald Köstler)  
 Exercises to High End Simulation in Practice (SS 2019, Übung, 4 SWS, Sebastian Eibl)

---

**Inhalt:**

Algorithmen und Datenstrukturen für parallele Strömungssimulation mittels der Lattice Boltzmann Methode in C++. Kopplung von Strömungssimulation und Starrkörpersimulation. Simulation von Partikeln in Strömung. Grundlagen der GPU Programmierung.

**Lernziele und Kompetenzen:**

*Fachkompetenz*

*Wissen*

Grundlegende Gesetze, die zur Simulation von physikalischen Effekten notwendig sind, darlegen.

*Verstehen*

Physikalische Grundgleichungen beschreiben.

*Anwenden*

Lösungsmethoden für physikalische Gleichungen implementieren. Numerische Algorithmen modifizieren um sie effizient zu parallelisieren. Numerische Algorithmen auf GPUs portieren.

*Analysieren*

Parallele Implementierungen auf GPU und CPU strukturieren.

*Evaluiieren (Beurteilen)*

Simulationsergebnisse validieren.

*Erschaffen*

Neue Simulationssoftware für GPU und CPU entwickeln.

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

[1] **Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule)

---

**Studien-/Prüfungsleistungen:**

High End Simulation in Practice (Vorlesung mit Übung) (Prüfungsnummer: 399607)

Prüfungsleistung, Portfolio, Dauer (in Minuten): 30

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- High End Simulation in Practice
- Exercises to High End Simulation in Practice

weitere Erläuterungen:

Die Prüfung wird bei maximal 20 Teilnehmern als 30 minütige mündliche Prüfung durchgeführt, ab 25 Teilnehmern wird die Prüfung als 60 minütige Klausur durchgeführt. Der Inhalt der Übungen ist Bestandteil der Prüfung.

Prüfungssprache: Deutsch oder Englisch

Erstablingung: SS 2019, 1. Wdh.: WS 2019/2020

1. Prüfer: Harald Köstler (100124)

---

---

**Modulbezeichnung:** **Programming Techniques for Supercomputers (Lecture and Tutorial) (PTfS)** 7.5 ECTS

(Programming Techniques for Supercomputers (Lecture and Tutorial))

Modulverantwortliche/r: Gerhard Wellein

Lehrende: Gerhard Wellein

---

Startsemester: SS 2019

Dauer: 1 Semester

Turnus: jährlich (SS)

Präsenzzeit: 90 Std.

Eigenstudium: 135 Std.

Sprache: Englisch

---

**Lehrveranstaltungen:**

Programming Techniques for Supercomputers (SS 2019, Vorlesung, 4 SWS, Gerhard Wellein)

Programming Techniques for Supercomputers (SS 2019, Übung, 2 SWS, Gerhard Wellein)

---

**Empfohlene Voraussetzungen:**

Experience in C/C++ or Fortran programming required; basic knowledge of MPI and OpenMP programming

---

**Inhalt:**

- Introduction to the architecture of modern supercomputers
- Single core architecture and optimisation strategies
- Memory hierarchy and data access optimization
- Concepts of parallel computers and parallel computing
- Efficient "shared memory" parallelisation (OpenMP)
- Parallelisation approaches for multi-core processors including GPUs
- Efficient "distributed memory" parallelisation (MPI)
- Roofline performance model
- Serial and parallel performance modelling
- Energy efficient implementation and execution of parallel programs

**Lernziele und Kompetenzen:**

Students

- acquire a comprehensive overview of programming modern supercomputers efficiently for numerical simulations
- learn modern optimisation and parallelisation strategies, guided by structured performance modelling
- acquire an insight into innovative programming techniques and alternative supercomputer architectures
- are able to implement basic numerical methods with high hardware efficiency on parallel computers
- are able to program and use modern supercomputer with high (energy) efficiency

**Literatur:**

G. Hager and G. Wellein: Introduction to High Performance Computing for Scientists and Engineers. CRC Computational Science Series, 2010. ISBN 978-1439811924

J. Hennessy and D. Patterson: Computer Architecture. A Quantitative Approach. Morgan Kaufmann Publishers, Elsevier, 2003. ISBN 1-55860-724-2

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

[1] **Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule)

---

**Studien-/Prüfungsleistungen:**

Programming Techniques for Supercomputers (Lecture and Tutorial)\_ (Prüfungsnummer: 278169)

Prüfungsleistung, Klausur, Dauer (in Minuten): 90

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Programming Techniques for Supercomputers
- Programming Techniques for Supercomputers

weitere Erläuterungen:

Written exam will also cover material presented and exercised in the tutorials. In der Klausur wird auch Stoff der Übungen abgeprüft.

Prüfungssprache: Englisch

Erstablingung: SS 2019, 1. Wdh.: WS 2019/2020

1. Prüfer: Gerhard Wellein (100166)

---

---

**Modulbezeichnung:** Programming Techniques for Supercomputers (Lecture) (PTfS) 5 ECTS  
 (Programming Techniques for Supercomputers (Lecture))

Modulverantwortliche/r: Gerhard Wellein  
 Lehrende: Gerhard Wellein

---

Startsemester: SS 2019	Dauer: 1 Semester	Turnus: jährlich (SS)
Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Deutsch

---

**Lehrveranstaltungen:**

Programming Techniques for Supercomputers (SS 2019, Vorlesung, 4 SWS, Gerhard Wellein)

---

**Empfohlene Voraussetzungen:**

Programmierkenntnisse in C/C++ oder Fortran

---

**Inhalt:**

- Introduction to the architecture of modern supercomputers
- Single core architecture and optimisation strategies
- Memory hierarchy and data access optimization
- Concepts of parallel computers and parallel computing
- Efficient "shared memory" parallelisation (OpenMP)
- Parallelisation approaches for multi-core processors including GPUs
- Efficient "distributed memory" parallelisation (MPI)
- Roofline performance model
- Serial and parallel performance modelling

**Lernziele und Kompetenzen:**

Students

- acquire a comprehensive overview of programming modern supercomputers efficiently for numerical simulations
- learn modern optimisation and parallelisation strategies, guided by structured performance modelling
- acquire an insight into innovative programming techniques and alternative supercomputer architectures
- are able to implement basic numerical methods with high hardware efficiency on parallel computers

**Literatur:**

- G. Hager and G. Wellein: Introduction to High Performance Computing for Scientists and Engineers. CRC Computational Science Series, 2010. ISBN 978-1439811924
  - J. Hennessy and D. Patterson: Computer Architecture. A Quantitative Approach. Morgan Kaufmann Publishers, Elsevier, 2003. ISBN 1-55860-724-2
- 

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

[1] **Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule)

Dieses Modul ist daneben auch in den Studienfächern "Computational Engineering (Master of Science with Honours)", "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Master of Science)", "Informatik (Master of Science)", "Mathematik (Bachelor of Science)" verwendbar.

---

**Studien-/Prüfungsleistungen:**



Mündliche Prüfung Programmiertechniken für Supercomputer / Programming Techniques for Supercomputers (Prüfungsnummer: 37403)

(englische Bezeichnung: Oral Examination on Programming Techniques for Supercomputers)

Prüfungsleistung, Klausur, Dauer (in Minuten): 60

Anteil an der Berechnung der Modulnote: 100% Prüfungssprache: Englisch

Erstablingung: SS 2019, 1. Wdh.: WS 2019/2020

1. Prüfer: Gerhard Wellein (100166)

---

---

**Modulbezeichnung:** **Advanced Programming Techniques (AdvPT)** **7.5 ECTS**  
 (Advanced Programming Techniques)

Modulverantwortliche/r: Harald Köstler  
 Lehrende: Harald Köstler

---

Startsemester: WS 2019/2020	Dauer: 1 Semester	Turnus: jährlich (WS)
Präsenzzeit: 60 Std.	Eigenstudium: 165 Std.	Sprache: Englisch

---

**Lehrveranstaltungen:**

Advanced Programming Techniques (WS 2019/2020, Vorlesung mit Übung, 4 SWS, Harald Köstler)  
 Exercises for Advanced Programming Techniques (WS 2019/2020, Übung, 2 SWS, Dominik Thönnies et al.)

---

**Inhalt:**

Der Inhalt der Vorlesung besteht aus zahlreichen fortgeschrittenen C++-Themen, die ausgerichtet sind auf die richtige und effiziente Nutzung von C++ für eine professionelle Softwareentwicklung.

The content of the lecture will consist of various topics of advanced C++ programming, aimed at teaching the proper and efficient usage of C++ for professional software development.

These are basic language concepts, the C++11/C++14/C++17 standards, object oriented programming in C++, static and dynamic polymorphism, template metaprogramming, and C++ idioms and design patterns.

A good preparation for the lecture is the C++ primer book from S. Lippman et al. One should at least have several hundred hours of programming experience in C/C++ or any related object oriented programming language. Knowledge of basic concepts like pointers, references, inheritance and polymorphism is required.

**Lernziele und Kompetenzen:**

*Fachkompetenz*

*Wissen*

Lernende können die grundlegenden Sprachkonstrukte in den verschiedenen C++ Standards wiedergeben.

Students know the basic language constructs from different C++ standards.

*Verstehen*

Lernende verstehen das C++ Objektmodell und können es mit anderen Programmiersprachen vergleichen.

Students understand the C++ object model and are able to compare it to other programming languages.

*Anwenden*

Lernenden können Standardalgorithmen in einer objektorientierten Programmiersprache implementieren.

Students can implement standard algorithms in an object oriented programming language.

*Analysieren*

Lernende können gängige Design Patterns klassifizieren und deren Anwendbarkeit für bestimmte Probleme diskutieren.

Students are able to classify common design patterns and to discuss their usability for certain problems.

*Evaluieren (Beurteilen)*

Lernende können entscheiden, welches Software Design passend für eine bestimmte Aufgabe ist. Sie können auch den Implementierungsaufwand dafür abschätzen.

Students can decide, which software design fits for a certain task. They are also able to estimate the programming effort for it.

*Erschaffen*

Lernende entwickeln selbständig in einer Gruppe ein größeres Softwarepaket im Bereich Simulation und Optimierung.

Students develop together in a group a larger software project in the area of simulation and optimization on their own.

**Literatur:**

- S. Lippman: C++ Primer, Addison-Wesley
- S. Meyers: Effective C++ Third Edition, Addison-Wesley
- H. Sutter: Exceptional C++, Addison-Wesley

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

**[1] Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule | Vertiefungsrichtung Systemsimulation)

Dieses Modul ist daneben auch in den Studienfächern "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Bachelor of Science)", "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Master of Science)", "Informatik (Master of Science)", "Materials Physics (Master of Science)", "Mathematik (Bachelor of Science)", "Medizintechnik (Bachelor of Science)", "Medizintechnik (Master of Science)" verwendbar.

---

**Studien-/Prüfungsleistungen:**

Advanced Programming Techniques (Vorlesung mit Übung) (Prüfungsnummer: 465562)

Prüfungsleistung, Portfolio

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Advanced Programming Techniques
- Exercises for Advanced Programming Techniques

weitere Erläuterungen:

Die Note ergibt sich aus einer 60minütigen Klausur. Voraussetzung zur Teilnahme an der Klausur ist die erfolgreiche Bearbeitung der Übungsaufgaben. 60 minute written examination. Successful submission of exercises is the prerequisite for the written exam.

Erstablingung: WS 2019/2020, 1. Wdh.: SS 2020

1. Prüfer: Harald Köstler (100124)

---

---

**Modulbezeichnung:** **Programming Techniques for Supercomputers (Lecture) (PTfS)** **5 ECTS**  
 (Programming Techniques for Supercomputers (Lecture))

Modulverantwortliche/r: Gerhard Wellein  
 Lehrende: Gerhard Wellein

---

Startsemester: SS 2019	Dauer: 1 Semester	Turnus: jährlich (SS)
Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Deutsch

---

**Lehrveranstaltungen:**

Programming Techniques for Supercomputers (SS 2019, Vorlesung, 4 SWS, Gerhard Wellein)

---

**Empfohlene Voraussetzungen:**

Programmierkenntnisse in C/C++ oder Fortran

---

**Inhalt:**

- Introduction to the architecture of modern supercomputers
- Single core architecture and optimisation strategies
- Memory hierarchy and data access optimization
- Concepts of parallel computers and parallel computing
- Efficient "shared memory" parallelisation (OpenMP)
- Parallelisation approaches for multi-core processors including GPUs
- Efficient "distributed memory" parallelisation (MPI)
- Roofline performance model
- Serial and parallel performance modelling

**Lernziele und Kompetenzen:**

Students

- acquire a comprehensive overview of programming modern supercomputers efficiently for numerical simulations
- learn modern optimisation and parallelisation strategies, guided by structured performance modelling
- acquire an insight into innovative programming techniques and alternative supercomputer architectures
- are able to implement basic numerical methods with high hardware efficiency on parallel computers

**Literatur:**

- G. Hager and G. Wellein: Introduction to High Performance Computing for Scientists and Engineers. CRC Computational Science Series, 2010. ISBN 978-1439811924
  - J. Hennessy and D. Patterson: Computer Architecture. A Quantitative Approach. Morgan Kaufmann Publishers, Elsevier, 2003. ISBN 1-55860-724-2
- 

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

**[1] Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule | Vertiefungsrichtung Systemsimulation)

Dieses Modul ist daneben auch in den Studienfächern "Computational Engineering (Master of Science with Honours)", "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Master of Science)", "Informatik (Master of Science)", "Mathematik (Bachelor of Science)" verwendbar.

---

**Studien-/Prüfungsleistungen:**

Mündliche Prüfung Programmiertechniken für Supercomputer / Programming Techniques for Supercomputers (Prüfungsnummer: 37403)

(englische Bezeichnung: Oral Examination on Programming Techniques for Supercomputers)

Prüfungsleistung, Klausur, Dauer (in Minuten): 60

Anteil an der Berechnung der Modulnote: 100% Prüfungssprache: Englisch

Erstablingung: SS 2019, 1. Wdh.: WS 2019/2020

1. Prüfer: Gerhard Wellein (100166)

---

---

**Modulbezeichnung:** **Simulation und Wissenschaftliches Rechnen 1 (SiWiR1)** **7.5 ECTS**  
 (Simulation and Scientific Computing 1)

Modulverantwortliche/r: Ulrich Rüde, Christoph Pflaum  
 Lehrende: Ulrich Rüde, Christoph Pflaum

---

Startsemester: WS 2019/2020	Dauer: 1 Semester	Turnus: halbjährlich (WS+SS)
Präsenzzeit: 90 Std.	Eigenstudium: 135 Std.	Sprache: Deutsch oder Englisch

---

**Lehrveranstaltungen:**

Simulation und Wissenschaftliches Rechnen 1 (WS 2019/2020, Vorlesung mit Übung, 2 SWS, Christoph Pflaum)  
 Übungen zu Simulation und Wissenschaftliches Rechnen (WS 2019/2020, Übung, 2 SWS, Christoph Rettinger)  
 Tutorium zu Simulation und Wissenschaftliches Rechnen (WS 2019/2020, Tutorium, 2 SWS, N.N.)

---

**Empfohlene Voraussetzungen:**

Voraussetzung ist ein Modul im Bereich Numerik

---

**Inhalt:**

- Performance Optimierung für numerische Algorithmen
- OpenMP Parallelisierung
- Finite Differenzen Diskretisierung im Ort
- Praktische Abschätzung des Diskretisierungsfehlers und der Konvergenzgeschwindigkeit numerischer Verfahren
- Software Entwicklung im Bereich des wissenschaftlichen Rechnens
- MPI Parallelisierung
- Finite Differenzen Diskretisierung für zeitabhängige Probleme

**Lernziele und Kompetenzen:**

Die Studierenden

- lernen Techniken zur Optimierung von Algorithmen im Bereich des wissenschaftlichen Rechnens
- lernen selbständig Algorithmen auf Parallelrechnern zu implementieren und zu optimieren
- lernen theoretisch die Stabilität von numerischen Algorithmen zu untersuchen

**Literatur:**

- Lehrbuch: G. Hager und G. Wellein, Introduction to High Performance Computing for Scientists and Engineers, CRC Press, 2010.
- Lehrbuch: Goedecker und Adofly Hoisie. Performance Optimization of Numerically Intensive Codes, SIAM, 2001.
- Lehrbuch: Gropp, Lusk, Skjellum, Using MPI. The MIT Press, 1999.
- Lehrbuch: Alexandrescu, Modern C++ Design, Generic Programming and Design Patterns. Addison-Wesley, 2001.
- Lehrbuch: Burden, Faires, Numerical Analysis, Brooks, 2001.
- Lehrbuch: Chandra at. al., Programming in OpenMP, Academic Press, 2001.

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

**[1] Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule | Vertiefungsrichtung Systemsimulation)

Dieses Modul ist daneben auch in den Studienfächern "Computational Engineering (Master of Science with Honours)", "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Bachelor of Science)", "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Master of Science)", "Informatik (Master of Science)", "Mathematik (Bachelor of Science)", "Mechatronik (Bachelor of Science)", "Mechatronik (Master of Science)", "Medizintechnik (Bachelor of Science)" verwendbar.

**Studien-/Prüfungsleistungen:**

Simulation und Wissenschaftliches Rechnen (Prüfungsnummer: 981660)

Prüfungsleistung, Klausur, Dauer (in Minuten): 90

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Simulation und Wissenschaftliches Rechnen 1
- Übungen zu Simulation und Wissenschaftliches Rechnen
- Tutorium zu Simulation und Wissenschaftliches Rechnen

weitere Erläuterungen:

Schriftliche Prüfung im Umfang von 90 Minuten und 50% der Punkte der Übungsaufgaben und Vortrag im Tutorium.

Prüfungssprache: Englisch

Erstablingung: WS 2019/2020, 1. Wdh.: SS 2020

1. Prüfer: Christoph Pflaum (100175)

---

---

**Modulbezeichnung:** **Organic Computing (OC)** **5 ECTS**  
 (Organic Computing)

Modulverantwortliche/r: Rolf Wanka  
 Lehrende: Rolf Wanka

---

Startsemester: SS 2019	Dauer: 1 Semester	Turnus: jährlich (SS)
Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Deutsch oder Englisch

---

**Lehrveranstaltungen:**

Organic Computing (SS 2019, Vorlesung, 2 SWS, Rolf Wanka)  
 Übungen zu Organic Computing (SS 2019, Übung, 2 SWS, Alexander Raß)

---

**Inhalt:**

Unter Organic Computing (OC) versteht man den Entwurf und den Einsatz von selbst-organisierenden Systemen, die sich den jeweiligen Umgebungsbedürfnissen dynamisch anpassen. Diese Systeme zeichnen sich dadurch aus, dass sie die sog. Self-\*-Eigenschaft besitzen, d.h. sie sind selbst-konfigurierend, selbst-optimierend, selbst-heilend, selbst-schützend, selbst-erklärend, ...

Als Vorbild für solche technischen Systeme werden Strukturen und Methoden biologischer und anderer natürlicher Systeme gewählt.

**Lernziele und Kompetenzen:**

*Fachkompetenz*

*Wissen*

Lernende können Wissen abrufen und wiedergeben. Sie kennen konkrete Einzelheiten wie Begriffe, Definitionen, Fakten, Regeln, Theorien, Merkmale, Kriterien, Abläufe. Sie lernen den Begriff des Organic Computings von anderen Paradigmen zu unterscheiden.

*Verstehen*

Lernende können Beispiele anführen und Aufgabenstellungen interpretieren.

*Anwenden*

Lernende können ein neues Problem wie z.B. Ranking-Erstellung durch Transfer des Wissens lösen.

*Analysieren*

Lernende können ein Problem in einzelne Teile zerlegen und so die Struktur des Problems verstehen.

*Sozialkompetenz*

Fähigkeit und Bereitschaft, zielorientiert mit anderen zusammenzuarbeiten.

**Literatur:**

- Ch. Müller-Schloer, Ch. von der Malsburg, R. P. Würt. Organic Computing. Informatik-Spektrum, Band 27, Nummer 4, S. 332-336. (LINK)
  - I. C. Trelea. The particle swarm optimization algorithm: convergence analysis and parameter selection. Information Processing Letters 85 (2003) 317-325. (LINK)
  - J. M. Kleinberg. Authoritative sources in a hyperlinked environment. Journal of the ACM 46 (1999) 604-632. (LINK)
  - M. Dorigo. V. Maniezzo. A Colorni. Ant system: an autocatalytic optimizing process. Technical Report 91-016, Politecnico di Milano, 1991. (LINK)
  - A. Badr. A. Fahmy. A proof of convergence for Ant algorithms. Information Sciences 160 (2004) 267-279.
  - M. Clerc. J. Kennedy. The particle swarm - Explosion, stability, and convergence in a multidimensional complex space. IEEE Transactions on Evolutionary Computation 8 (2002) 58-73.
- 

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

**[1] Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule)

Dieses Modul ist daneben auch in den Studienfächern "Computational Engineering (Master of Science with Honours)", "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Master of Science)", "Informatik



(Master of Science)", "Information and Communication Technology (Master of Science)", "Informations- und Kommunikationstechnik (Master of Science)", "Mathematik (Bachelor of Science)", "Medizintechnik (Master of Science)" verwendbar.

---

**Studien-/Prüfungsleistungen:**

Organic Computing (Prüfungsnummer: 39701)

(englische Bezeichnung: Organic Computing)

Prüfungsleistung, Klausur, Dauer (in Minuten): 90

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Organic Computing
- Übungen zu Organic Computing

Erstablingung: SS 2019, 1. Wdh.: WS 2019/2020

1. Prüfer: Rolf Wanka (100367)

---

**Bemerkungen:**

Auch für CE

---

**Modulbezeichnung:** Introduction to Finite Model Theory (FMT) 7.5 ECTS  
 (Introduction to Finite Model Theory)

Modulverantwortliche/r: Tadeusz Litak  
 Lehrende: Tadeusz Litak

---

Startsemester: WS 2019/2020	Dauer: 1 Semester	Turnus: jährlich (WS)
Präsenzzeit: 56 Std.	Eigenstudium: 169 Std.	Sprache: Englisch

---

**Lehrveranstaltungen:**

Introduction to Finite Model Theory (WS 2019/2020, Vorlesung mit Übung, 4 SWS, Tadeusz Litak)

---

**Inhalt:**

We will discuss basic results of finite model theory. This module can be seen as a natural continuation of GLoln.

**Lernziele und Kompetenzen:**

*Fachkompetenz*

*Wissen*

The students will learn basic notions and results of finite model theory.

*Verstehen*

The students will be able to summarize and prove major results of finite model theory and explain the deep difference between finite and infinite model theory, by using 0-1 laws or by showing the failure of most preservation theorems.

*Anwenden*

The students will be able to use basic tools of finite model theory such as Ehrenfeucht-Fraïssé games or Herbrand structures.

The students will use preservation theorems of Rosen or Rossman to characterize expressive power of logics over finite structures.

*Analysieren*

The students will be able to determine, e.g., whether a given query is expressible over all finite models (or restricted subclasses of structures) in a chosen language.

*Evaluiieren (Beurteilen)*

The best students, having gained a deep understanding of finite model theory, will be able to examine and evaluate choices involved in the design of query languages, with particular attention to the criterion of expressive completeness and contrasting expressive power of a given language with its computational complexity.

*Erschaffen*

Making the right design choices is an important skill whenever the need arises to develop a new domain-specific language.

**Literatur:**

- Leonid Libkin, Elements of Finite Model Theory, Springer 2012
- Heinz-Dieter Ebbinghaus, Jörg Flum, Finite Model Theory, Springer 1995
- Phokion Kolaitis. The Expressive Power of Logics on Finite Models, in Finite Model Theory and its Applications, EATCS Series: Texts in Theoretical Computer Science, Springer 2007

Supplementary reading on logic and classical (unrestricted) model theory

- Herbert B. Enderton. A Mathematical Introduction to Logic (1 ed. 1972). Academic Press Second edition, 2001. ISBN 978-0-12-238452-3
  - Wilfrid Hodges. Model theory, Cambridge University Press, Cambridge 1993, 772pp.
- 

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

[1] **Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule)

---

**Studien-/Prüfungsleistungen:**

Introduction to Finite Model Theory (Prüfungsnummer: 468977)

(englische Bezeichnung: Introduction to Finite Model Theory)

Prüfungsleistung, mündliche Prüfung, Dauer (in Minuten): 30

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Introduction to Finite Model Theory

weitere Erläuterungen:

Die Übungsleistungen (Bearbeitung von Übungsaufgaben) gehen zu 50 % in die Note der mündlichen Prüfung ein.

Prüfungssprache: Englisch

Erstabledung: WS 2019/2020, 1. Wdh.: SS 2020

1. Prüfer: Tadeusz Litak (100372)

---

---

**Modulbezeichnung:** Introduction to Dependently Typed Programming (IDenT) 7.5 ECTS  
 (Introduction to Dependently Typed Programming)

Modulverantwortliche/r: Sergey Goncharov  
 Lehrende: Sergey Goncharov

---

Startsemester: WS 2019/2020	Dauer: 1 Semester	Turnus: jährlich (WS)
Präsenzzeit: 60 Std.	Eigenstudium: 165 Std.	Sprache: Englisch

---

**Lehrveranstaltungen:**

Introduction to Dependently Typed Programming (WS 2019/2020, Vorlesung mit Übung, 4 SWS, Sergey Goncharov)

---

**Vorhergehende Module:**

Theorie der Programmierung

---

**Inhalt:**

The course provides an introduction to dependently typed programming using Agda as the running programming language for practical exercises. The idea of the course is to present theoretical and practical aspects of dependent type theory through the lens of logic, programming, their connection via the propositions-as-types paradigm and explore the applications of these concepts to functional programming and verification.

**Lernziele und Kompetenzen:**

*Fachkompetenz*

*Wissen*

Students demonstrate an understanding of the role of dependent types and the propositions-as-types paradigm in the context of functional programming and as a semantic mechanism for programming and system specification; Students reproduce the main definitions and results on dependent type semantics and higher order logic, and explain them from a programming perspective.

*Anwenden*

Students use dependent types to formalise examples, formulate and prove the corresponding verification goals. Students use their theoretic knowledge in dependent type theory for implementing practical problems in concrete programming languages, specifically in Agda.

*Analysieren*

Students analyse programming challenges, reduce larger tasks to subtasks and correspondingly organise their implementations in a modular fashion, by using the available abstraction mechanisms of dependently typed programming. Students formulate and prove verification goals in a modular way and use interactive mechanisms of the Agda programming environment both to conduct the proofs and to facilitate the development process.

*Selbstkompetenz*

Students will be regularly provided with small challenges in form of exercises to be able to have a gradual progress with the lecture material.

**Literatur:**

- Ulf Norell and James Chapman. Dependently Typed Programming in Agda.
- Ana Bove and Peter Dybjer. Dependent Types at Work.
- Ana Bove, Peter Dybjer, and Ulf Norell. A Brief Overview of Agda - A Functional Language with Dependent Types
- Yoshiki Kinoshita. On the Agda Language
- Anton Setzer. Lecture notes on Interactive Theorem Proving.
- Daniel Peebles. Introduction to Agda. Video of talk from the January 2011 Boston Haskell session at MIT.
- Conor McBride. Introduction to Dependently Typed Programming using Agda.
- Andreas Abel. Agda lecture notes.
- Jan Malakhovski. Brutal [Meta]Introduction to Dependent Types in Agda

- Thorsten Altenkirch. Computer Aided Formal Reasoning
- Daniel Licata. Dependently Typed Programming in Agda
- Tesla Ice Zhang. Some books about Formal Verification in Agda (in Chinese)
- Phil Wadler. Programming Languages Foundations in Agda
- Aaron Stump. Verified Functional Programming in Agda
- Diviánszky Péter. Agda Tutorial
- Musa Al-hassy. A slow-paced introduction to reflection in Agda

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

**[1] Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule)

---

**Studien-/Prüfungsleistungen:**

Introduction to Dependently Typed Programming (Prüfungsnummer: 184367)

(englische Bezeichnung: Introduction to Dependently Typed Programming)

Prüfungsleistung, mündliche Prüfung, Dauer (in Minuten): 25

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Introduction to Dependently Typed Programming

weitere Erläuterungen:

Die Modulnote setzt sich zu 50% aus dem Ergebnis einer 30-minütigen mündlichen Prüfung am Semesterende und zu 50% aus der Bewertung der Leistungen aus dem Übungsbetrieb zusammen.

Prüfungssprache: Englisch

Erstabledung: WS 2019/2020, 1. Wdh.: SS 2020

1. Prüfer: Sergey Goncharov (100373)

---

---

**Modulbezeichnung:** **Monad-Based Programming (MBProg)** **7.5 ECTS**  
(Monad-Based Programming)

Modulverantwortliche/r: Sergey Goncharov  
Lehrende: Sergey Goncharov

---

Startsemester: SS 2019	Dauer: 1 Semester	Turnus: jährlich (SS)
Präsenzzeit: 60 Std.	Eigenstudium: 165 Std.	Sprache: Englisch

---

**Lehrveranstaltungen:**

Monad-Based Programming (SS 2019, Vorlesung mit Übung, 4 SWS, Sergey Goncharov)

---

**Inhalt:**

The course provides a background to various topics of the theory of programming. As a guiding paradigm monad-based functional programming is chosen. The idea of the course is to provide clear computational insights to various concepts of computer science and to practice these by concrete implementations in suitable programming languages such as Haskell.

**Lernziele und Kompetenzen:**

*Fachkompetenz*

*Wissen*

Students demonstrate an understanding of the role of computational monads in the context of functional programming and as a semantic tool for programming and system specification; Students reproduce the main definitions and results on monads, monad combination, and further categorical constructions and explain them from a programming perspective.

*Anwenden*

Students use the monad-based approach to formalise examples involving various kinds of computational effects as monads. Students use monads for practical programming in programming languages, such as Haskell.

*Analysieren*

Students identify various computational effects as monads and provide an appropriate treatment of problems from various semantic domains (probabilistic, nondeterministic, concurrent), possibly providing a monad-based software implementation.

*Selbstkompetenz*

Students will be regularly provided with small challenges in form of exercises to be able to have a gradual progress with the lecture material.

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

**[1] Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule)

---

**Studien-/Prüfungsleistungen:**

Monad-Based Programming (Prüfungsnummer: 845618)

(englische Bezeichnung: Monad-Based Programming)

Prüfungsleistung, mündliche Prüfung, Dauer (in Minuten): 30

Anteil an der Berechnung der Modulnote: 100%

weitere Erläuterungen:

Die Modulnote setzt sich zu 50% aus dem Ergebnis einer 30-minütigen mündlichen Prüfung am Semesterende und zu 50% aus der Bewertung der Leistungen aus dem Übungsbetrieb zusammen.

Prüfungssprache: Englisch

Erstablesung: SS 2019, 1. Wdh.: WS 2019/2020 (nur für Wiederholer)

1. Prüfer: Sergey Goncharov (100373)

---

**Modulbezeichnung:** **Praktische Semantik von Programmiersprachen (SemProg)** 7.5 ECTS  
(Practical Semantics of Programming Languages)

Modulverantwortliche/r: Tadeusz Litak  
Lehrende: Tadeusz Litak

Startsemester: SS 2019	Dauer: 1 Semester	Turnus: jährlich (SS)
Präsenzzeit: 56 Std.	Eigenstudium: 169 Std.	Sprache: Deutsch und Englisch

**Lehrveranstaltungen:**

Praktische Semantik von Programmiersprachen (SS 2019, Vorlesung mit Übung, 4 SWS, Tadeusz Litak)

**Inhalt:**

We study the foundations of the imperative and functional languages, including semantics and type systems. The special feature of this course is that theory is done in a very practical and hands-on way: we not just prove, but program all the results from first-principles. The basic tool used in the course is Coq proof assistant, which can be regarded as a functional programming language in its own right. It has been used, for example, to verify correctness of Java Card technology, C compilers or, more recently, fragments of x86 architecture.

**Lernziele und Kompetenzen:**

*Fachkompetenz*

*Wissen*

The students explain the basics of both programming semantics and proof assistants, in particular Coq.

*Verstehen*

The students prove theorems using a proof assistant.

*Anwenden*

The students transfer proofs into programs and programs into proofs.

*Analysieren*

The students examine behaviour of simple programs using formal semantics

*Evaluiieren (Beurteilen)*

The students evaluate the role played by logic and type theory in scientific approach to programming.

*Erschaffen*

The students provide formal semantics to a simple programming language.

**Literatur:**

Online book "Software Foundations" <http://www.cis.upenn.edu/~bcpierce/sf/>

Online books by Adam Chlipala: "Certified Programming with Dependent Types" <http://adam.chlipala.net/cpdt/> and "Formal Reasoning About Programs" <http://adam.chlipala.net/frap/>

Supplementary reading on the theory of programming: Types and Programming Languages Benjamin C. Pierce, The MIT Press

Supplementary reading on Coq: Interactive Theorem Proving and Program Development Coq'Art: The Calculus of Inductive Constructions Series: Texts in Theoretical Computer Science. An EATCS Series Bertot, Yves, Casteran, Pierre

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

[1] **Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule)

**Studien-/Prüfungsleistungen:**

Mündliche Prüfung zu Praktische Semantik von Programmiersprachen (Prüfungsnummer: 599478)

(englische Bezeichnung: Practical Semantics of Programming Languages)

Prüfungsleistung, mündliche Prüfung, Dauer (in Minuten): 30

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Praktische Semantik von Programmiersprachen

weitere Erläuterungen:

In die Note der mündliche Prüfung gehen auch die Übungsleistungen ein.

Erstablingung: SS 2019, 1. Wdh.: WS 2019/2020 (nur für Wiederholer)

1. Prüfer: Tadeusz Litak (100372), 2. Prüfer: Lutz Schröder (100190)

---



---

**Modulbezeichnung:** Randomisierte Algorithmen (RAND) 7.5 ECTS  
 (Randomized Algorithms)

Modulverantwortliche/r: Rolf Wanka  
 Lehrende: Rolf Wanka

---

Startsemester: SS 2019	Dauer: 1 Semester	Turnus: jährlich (SS)
Präsenzzeit: 60 Std.	Eigenstudium: 165 Std.	Sprache: Deutsch oder Englisch

---

**Lehrveranstaltungen:**

Randomisierte Algorithmen (SS 2019, Vorlesung, 2 SWS, Rolf Wanka)  
 Übungen zu Randomisierte Algorithmen (SS 2019, Übung, 2 SWS, Alexander Raß)

---

**Inhalt:**

Bei der Lösung kombinatorischer oder zahlentheoretischer Probleme ist es oft möglich, durch Würfeln schnell und einfach mit hoher Wahrscheinlichkeit oder im Durchschnitt zu hervorragenden Lösungen zu kommen. In diesem Modul lernen wir Konzepte wie die Probabilistische Methode, Irrläufe (Random Walks) und Varianzanalysen von Zufallsprozessen kennen und wenden sie auf graphentheoretische Probleme und effiziente Datenstrukturen an. Zu den vorgestellten Inhalten gehören:

- Schnelle Wiederholung wahrscheinlichkeitstheoretischer Begriffe und Resultate
- Die Probabilistische Methode und ihre Anwendung auf die Berechnung maximaler Schnitte und unabhängiger Mengen
- Random Walks und ihre Anwendung auf das Erfüllbarkeitsproblem

Neueste Ergebnisse dieses Forschungsgebietes werden inhaltlich in das Modul eingebunden.

**Lernziele und Kompetenzen:**

*Fachkompetenz*

*Wissen*

Lernende können Wissen abrufen und wiedergeben. Sie kennen konkrete Einzelheiten wie Begriffe, Definitionen, Regeln, Gesetzmäßigkeiten, Theorien.

*Verstehen*

Lernende können Beispiele anführen, Aufgabenstellungen interpretieren oder ein Problem in eigenen Worten wiedergeben.

*Anwenden*

Lernende können ein neues Problem durch Transfer des Wissens lösen.

*Analysieren*

Lernende können ein Problem in einzelne Teile zerlegen und so die Struktur des Problems verstehen; sie können Zusammenhänge erkennen und Folgerungen ableiten

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

[1] **Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule)

---

**Studien-/Prüfungsleistungen:**

Randomisierte Algorithmen (Vorlesung mit Übung) (Prüfungsnummer: 164985)

Prüfungsleistung, mündliche Prüfung, Dauer (in Minuten): 30

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Randomisierte Algorithmen
- Übungen zu Randomisierte Algorithmen

Erstabelleung: SS 2019, 1. Wdh.: WS 2019/2020

1. Prüfer: Rolf Wanka (100367)

---

---

**Modulbezeichnung:** **Approximationsalgorithmen (APPROXA)** **7.5 ECTS**  
 (Approximation Algorithms)

Modulverantwortliche/r: Rolf Wanka  
 Lehrende: Rolf Wanka

---

Startsemester: SS 2019	Dauer: 1 Semester	Turnus: jährlich (SS)
Präsenzzeit: 60 Std.	Eigenstudium: 165 Std.	Sprache: Deutsch oder Englisch

---

**Lehrveranstaltungen:**

Approximationsalgorithmen (SS 2019, Vorlesung, 2 SWS, Rolf Wanka)  
 Übungen zu Approximationsalgorithmen (SS 2019, Übung, 2 SWS, Rolf Wanka)

---

**Inhalt:**

Für viele kombinatorische Optimierungsprobleme hat sich herausgestellt, daß sie vermutlich nicht durch schnelle exakte Algorithmen gelöst werden können, weshalb man sich mit Näherungslösungen zufrieden geben muß. In dieser Vorlesung werden Approximationsalgorithmen vorgestellt, die für eine Reihe populärer Optimierungsprobleme beweisbar gute Lösungen in vertretbarer Zeit berechnen.

Im ersten Teil der Veranstaltung werden die grundlegenden Begriffe vorgestellt, mit Beispielalgorithmen ausgeführt und jeweils die Grenzen aufgezeigt.

Im zweiten Teil werden allgemeine Techniken eingeführt und anhand instruktiver Beispiele mit Leben erfüllt.

**Lernziele und Kompetenzen:**

*Fachkompetenz*

*Wissen*

Lernende können Wissen abrufen und wiedergeben. Sie kennen konkrete Einzelheiten wie Begriffe, Definitionen, Fakten, Gesetzmäßigkeiten und Theorien.

*Verstehen*

Lernende können Beispiele anführen und Aufgabenstellungen interpretieren ,

*Anwenden*

Lernende können neue Probleme durch Transfer des Wissens lösen.

*Analysieren*

Lernende können ein Problem in einzelne Teile zerlegen und so die Struktur des Problems verstehen und der Anwendung der erlernten Methoden zugänglich machen.

**Literatur:**

- R. Wanka. Approximationsalgorithmen - Eine Einführung. Teubner, 2007.
  - K. Jansen, M. Margraf. Approximative Algorithmen und Nichtapproximierbarkeit. de Gruyter, 2008.
  - G. Ausiello, P. Crescenzi, G. Gambosi, V. Kann, A. Marchetti-Spaccamela, M. Protasi. Complexity and Approximation – Combinatorial Optimization Problems and Their Approximability Properties. Springer, 1999.
  - E. W. Mayr, H. J. Prömel, and A. Steger (Hrsg.). Lectures on Proof Verification and Approximation Algorithms. Springer, 1998.
  - V. V. Vazirani. Approximation Algorithms. Springer, 2001.
- 

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

**[1] Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule)

---

**Studien-/Prüfungsleistungen:**

Approximationsalgorithmen (Vorlesung mit Übung) (Prüfungsnummer: 247639)

Prüfungsleistung, mündliche Prüfung, Dauer (in Minuten): 30

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Approximationsalgorithmen

- Übungen zu Approximationsalgorithmen

Erstablingung: SS 2019, 1. Wdh.: WS 2019/2020

1. Prüfer: Rolf Wanka (100367)

---

---

**Modulbezeichnung:** Algebra des Programmierens (AlgProg) 7.5 ECTS  
 (Algebra of Programming)

Modulverantwortliche/r: Stefan Milius  
 Lehrende: Henning Urvat, Stefan Milius

---

Startsemester: WS 2019/2020	Dauer: 1 Semester	Turnus: jährlich (WS)
Präsenzzeit: 42 Std.	Eigenstudium: 183 Std.	Sprache: Deutsch und Englisch

---

**Lehrveranstaltungen:**

Algebra des Programmierens (WS 2019/2020, Vorlesung mit Übung, 4 SWS, Henning Urvat et al.)

---

**Inhalt:**

Induktive Datentypen wie z.B. Listen, Stacks und Bäume werden abstrakt beschrieben. Strukturelle Induktion und Rekursion für solche Datentypen (z.B. die fold-Operation auf Listen) werden auf Grundlage der Initiale-Algebra-Semantik entwickelt. Dadurch werden verschiedene effektive Programmiertricks auf eine solide mathematische Grundlage gestellt. Grundlagen und Methoden der Kategorientheorie werden eingeführt und erklärt, insbesondere initiale Algebren und ihre Konstruktion. Evtl. werden Koalgebren behandelt, die es ermöglichen, verschiedene zustandsbasierte Systeme und ihre Semantik in einer einheitlichen Theorie zu studieren.

**Lernziele und Kompetenzen:**

Die Studierenden erklären induktive Datentypen und die entsprechenden strukturellen Induktionsprinzipien (Defitions- und Beweisprinzip) und wenden diese in Programmierbeispielen an. Sie leiten ferner solche Prinzipien für neue induktive Datentypen her und stellen sie als Spezialfall der Initiale-Algebra-Semantik dar. Die Studierenden erläutern grundlegende Begriffe der Kategorientheorie und wenden die behandelten Begriffe und Methoden auf verschiedene Spezialfälle in der Informatik an (abstrakte Datentypen, zustandsbasierte Systeme). Sie formulieren die Semantik von induktiven Programmen und zustandsbasierten Systemen mit Hilfe von (Ko)Algebren formulieren. Die Studierenden erlangen die Fähigkeit kategorientheoretische Beweise zu führen und die entsprechenden Argumente klar schriftlich niederzulegen.

**Literatur:**

- R. Bird and O. de Moor: Algebra of Programming, Prentice Hall, 1996.
- J. Adamek, H. Herrlich and G.E. Strecker: Abstract and Concrete Categories: The Joy of Cats, 2nd edition, Dover Publishers, 2009.

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

[1] **Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule | Vertiefungsrichtung Theoretische Informatik)

Dieses Modul ist daneben auch in den Studienfächern "Informatik (Master of Science)", "Mathematik (Bachelor of Science)" verwendbar.

---

**Studien-/Prüfungsleistungen:**

Algebra des Programmierens (Vorlesung mit Übung) (Prüfungsnummer: 861501)

Prüfungsleistung, mündliche Prüfung, Dauer (in Minuten): 30

Anteil an der Berechnung der Modulnote: 100%

weitere Erläuterungen:

In die Note der mündliche Prüfung gehen auch die Übungsleistungen ein.

Erstablingung: WS 2019/2020, 1. Wdh.: SS 2020 (nur für Wiederholer)

1. Prüfer: Lutz Schröder (100190)



---

**Modulbezeichnung:** Algebraische und Logische Aspekte der Automaten-  
 Automatentheorie (ALoA) 7.5 ECTS  
 (Algebraic and Logical Aspects of Automata Theory)

Modulverantwortliche/r: Stefan Milius  
 Lehrende: Stefan Milius

---

Startsemester: WS 2019/2020	Dauer: 1 Semester	Turnus: unregelmäßig
Präsenzzeit: 45 Std.	Eigenstudium: 180 Std.	Sprache: Deutsch

---

**Lehrveranstaltungen:**

Algebraische und Logische Aspekte der Automaten-  
 Automatentheorie (WS 2019/2020, Vorlesung mit Übung,  
 4 SWS, Stefan Milius)

---

**Vorhergehende Module:**

Grundlagen der Logik in der Informatik  
 Berechenbarkeit und Formale Sprachen

---

**Inhalt:**

Automaten als mathematische Formalisierung zustandsbasierter Systeme gehören zu den wichtigsten Werkzeugen der Theoretischen Informatik und besitzen zahlreiche Anwendungen, von der Compilerentwicklung bis zur Verifikation reaktiver Systeme. In dieser Veranstaltung, die an die Anfängervorlesungen des Informatikstudiums anknüpft, werden wichtige Querverbindungen zwischen der Automaten-  
 Automatentheorie und Gebieten der Mathematik (Algebra, Topologie und Logik) hergestellt. Folgende Themen werden behandelt:

- Erkennung von regulären Sprachen durch Monoide und Halbgruppen
- Proendliche Gleichungen und Varietäten von Sprachen
- Logische Beschreibung regulärer Sprachen, Ehrenfeucht-Fraïssé-Spiele
- Automaten, Algebra und Logik auf unendlichen Wörtern und Bäumen

**Lernziele und Kompetenzen:**

Die Studierenden lernen die algebraische und logische Sichtweise der Automaten-  
 Automatentheorie kennen und können diese auf verschiedene Automatenmodelle und Typen von formalen Sprachen anwenden.

Die Studierenden machen sich mit mathematischen Grundlagen aus der Halbgruppentheorie und der Topologie vertraut.

Die Studierenden erlangen die Fähigkeit, auf Basis dieser Grundlagen mathematische Beweise zu führen und die entsprechenden Argumente klar schriftlich niederzulegen.

**Literatur:**

- J.-E. Pin: Mathematical Foundations of Automata Theory
- D. Perrin, J.-E. Pin: Infinite Words, Academic Press, 2004
- H. Straubing: Finite Automata, Formal Logic, and Circuit Complexity, Birkhäuser, 1994
- E. Grädel, W. Thomas, T. Wilke: Automata, Logic, and Infinite Games, Springer, 2002

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

[1] **Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule | Vertiefungsrichtung Theoretische Informatik)

Dieses Modul ist daneben auch in den Studienfächern "Informatik (Master of Science)", "Mathematik (Bachelor of Science)" verwendbar.

---

**Studien-/Prüfungsleistungen:**

Algebraische und Logische Aspekte der Automatentheorie (Prüfungsnummer: 787141)

(englische Bezeichnung: Algebraic and Logical Aspects of Automata Theory)

Prüfungsleistung, mündliche Prüfung, Dauer (in Minuten): 30

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Algebraische und Logische Aspekte der Automatentheorie

Prüfungssprache: Deutsch oder Englisch

Erstablesung: WS 2019/2020, 1. Wdh.: SS 2020 (nur für Wiederholer)

1. Prüfer: Stefan Milius (100294)

---

**Bemerkungen:**

Webseite der Lehrveranstaltung: <https://www8.cs.fau.de/course:aloa>

---

**Modulbezeichnung:** Effiziente kombinatorische Algorithmen (EffAlg) 7.5 ECTS  
 (Efficient Combinatorial Algorithms)

Modulverantwortliche/r: Rolf Wanka  
 Lehrende: Rolf Wanka

---

Startsemester: WS 2019/2020	Dauer: 1 Semester	Turnus: jährlich (WS)
Präsenzzeit: 60 Std.	Eigenstudium: 165 Std.	Sprache: Deutsch oder Englisch

---

**Lehrveranstaltungen:**

Effiziente kombinatorische Algorithmen (WS 2019/2020, Vorlesung, 2 SWS, Rolf Wanka)  
 Übung Effiziente kombinatorische Algorithmen (WS 2019/2020, Übung, 2 SWS, Rolf Wanka)

---

**Vorhergehende Module:**

Berechenbarkeit und Formale Sprachen

---

**Inhalt:**

In diesem Modul werden effiziente exakte Algorithmen für diskrete Probleme vorgestellt. Zuerst werden nichttriviale tiefensuchbasierte Linearzeitverfahren für die Berechnung zweifacher Zusammenhangskomponenten auf ungerichteten Graphen und starker Zusammenhangskomponenten auf gerichteten Graphen untersucht. Danach werden Polynomialzeit-Verfahren zur Berechnung maximaler Flüsse präsentiert. Eine Einführung in den Entwurf und die Analyse parametrisierter Algorithmen an Hand des Vertex-Cover-Problems und eine Einführung in den Bereich der sog. mild-exponentiellen Algorithmen für das Erfüllbarkeitsproblem und weiterer NP-vollständiger Probleme runden das Modul ab.

**Literatur:**

- A. V. Aho, J. E. Hopcroft, J. D. Ullman. The Design and Analysis of Computer Algorithms. Addison-Wesley, 1975.
- Venkatesh Raman, Saket Saurabh, Somnath Sikdar. Efficient Exact Algorithms through Enumerating Maximal Independent Sets and Other Techniques. Theory of Computing Systems 41 (2007) 563-587.
- Frank Gurski, Irene Rothe, Jörg Rothe, Egon Wanke. Exakte Algorithmen für schwere Graphenprobleme. Springer 2010.
- Sven Oliver Krumke, Hartmut Noltemeier. Graphentheoretische Konzepte und Algorithmen. Vieweg+Teubner, 2. Auflage 2009.
- Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein. Introduction to Algorithms (2nd Edition). MIT Press, 2001.
- Fedor V. Fomin, Dieter Kratsch. Exact Exponential Algorithms. Springer, 2010.
- Volker Heun. Grundlegende Algorithmen. Vieweg, 2. Auflage 2003.
- Juraj Hromkovic. Algorithmics for Hard Problems. Springer, 2001.
- Stephan Hußmann, Brigitte Lutz-Westphal (Hrsg.). Kombinatorische Optimierung erleben. Vieweg, 2007.
- Jon Kleinberg, Eva Tardos. Algorithm Design. Pearson / Addison Wesley, 2006.
- Sven Oliver Krumke, Hartmut Noltemeier. Graphentheoretische Konzepte und Algorithmen. Vieweg+Teubner, 2. Auflage 2009.
- Christos H. Papadimitriou, Kenneth Steiglitz. Combinatorial Optimization: Algorithms and Complexity. Dover Publications, 1998.
- Volker Turau. Algorithmische Graphentheorie. Oldenbourg, 3. Auflage 2009.
- Vöcking et al. (Hrsg.) Taschenbuch der Algorithmen. Springer 2008.

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

**[1] Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule | Vertiefungsrichtung Theoretische Informatik)



Dieses Modul ist daneben auch in den Studienfächern "Informatik (Master of Science)", "Mathematik (Bachelor of Science)" verwendbar.

---

**Studien-/Prüfungsleistungen:**

Effiziente kombinatorische Algorithmen (Vorlesung mit Übung) (Prüfungsnummer: 843472)

Prüfungsleistung, mündliche Prüfung, Dauer (in Minuten): 30

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Effiziente kombinatorische Algorithmen
- Übung Effiziente kombinatorische Algorithmen

Erstablesung: WS 2019/2020, 1. Wdh.: SS 2020

1. Prüfer: Rolf Wanka (100367)

---

**Modulbezeichnung:** Einführung in die moderne Kryptographie (ModCrypt) 7.5 ECTS  
(Introduction to Modern Cryptography)

Modulverantwortliche/r: Dominique Schröder  
Lehrende: Dominique Schröder

Startsemester: WS 2019/2020      Dauer: 1 Semester      Turnus: jährlich (WS)  
Präsenzzeit: 60 Std.      Eigenstudium: 165 Std.      Sprache: Englisch

**Lehrveranstaltungen:**

Einführung in die moderne Kryptographie (WS 2019/2020, Vorlesung, 4 SWS, Dominique Schröder)  
Einführung in die moderne Kryptographie (Übung) (WS 2019/2020, Übung, 2 SWS, Dominique Schröder)

**Empfohlene Voraussetzungen:**

No previous knowledge in Cryptography or computer Security is required.

**Inhalt:**

This course gives a comprehensive introduction to modern cryptography. The course also serves as a base for other courses on cryptography that are offered by the chair. The topics covered are the following:

- Information theoretic security
- Computational security
- Private key Encryption
- Message Authentication Codes
- Hash functions
- Public key Encryption
- Digital Signatures

More advanced topics may be covered if time permits.

**Lernziele und Kompetenzen:**

On successfully passing the course, the student is guaranteed to be knowledgeable on the basic concepts of provable security.

**Literatur:**

Introduction to Modern Cryptography Jonathan Katz and Yehuda Lindell 2nd Edition (2014) (Chapman & Hall/CRC Cryptography and Network Security Series) ISBN-13: 978-1466570269

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

[1] **Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule | Vertiefungsrichtung Theoretische Informatik)

Dieses Modul ist daneben auch in den Studienfächern "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Bachelor of Science)", "Informatik (Master of Science)", "Mathematik (Bachelor of Science)" verwendbar.

**Studien-/Prüfungsleistungen:**

Einführung in die algorithmische Kryptographie (Prüfungsnummer: 997109)

(englische Bezeichnung: Introduction to Algorithmic Cryptography)

Prüfungsleistung, Portfolio

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Einführung in die moderne Kryptographie
- Einführung in die moderne Kryptographie (Übung)

weitere Erläuterungen:

Um das Modul zu bestehen müssen sowohl 50 % der Übungspunkte erzielt werden als auch die Klausur bestanden werden. Die Note setzt sich zu 100 % aus der Note der Klausur zusammen.

Prüfungssprache: Englisch

Erstablingung: WS 2019/2020, 1. Wdh.: SS 2020

1. Prüfer: Dominique Schröder (100450)

---

**Modulbezeichnung:** Formale Methoden der Softwareentwicklung (FMSoft) 7.5 ECTS  
(Formal Methods of Software Development)

Modulverantwortliche/r: Tadeusz Litak  
Lehrende: Tadeusz Litak

Startsemester: WS 2019/2020 Dauer: 1 Semester Turnus: jährlich (SS)  
Präsenzzeit: 56 Std. Eigenstudium: 169 Std. Sprache: Deutsch und Englisch

### Lehrveranstaltungen:

Formale Methoden der Softwareentwicklung (WS 2019/2020, Vorlesung mit Übung, 4 SWS, Tadeusz Litak)

### Inhalt:

In the first part of the course, we will engage in the formal verification of reactive systems. Students learn the syntax and semantics of the temporal logics LTL, CTL, and CTL\* and their application in the specification of e.g. safety and liveness properties of systems. Simple models of systems are designed and verified using model checkers and dedicated frameworks for asynchronous and synchronous reactive systems, and the algorithms working in the background are explained.

The second part of the course focuses on functional correctness of programs; more precisely, we discuss the theory of pre- and postconditions, Hoare triples, loop invariants, and weakest (liberal) preconditions, in order to introduce automatised correctness proofs using the Hoare calculus.

### Lernziele und Kompetenzen:

Students are going to acquire the following competences:

#### *Fachkompetenz*

##### *Wissen*

- Reproduce the definition of syntax and semantics of temporal logics LTL, CTL, and CTL\*.
- Reproduce the definition of semantics of a simple programming languages like IMP, with special focus on axiomatic semantics (Hoare rules).
- Explain how CTL can be characterised in terms of fixpoints.

##### *Verstehen*

The students understand the workings of state of the art automatic frameworks, clarifying the role of model checking algorithms, semantics and Hoare calculi in formal verification.

##### *Anwenden*

In a series of exercises, the students use state of the art tools for

- model checking
- specification and verification of reactive systems,
- verification of functional correctness or memory safety of simple programs.

##### *Analysieren*

- Choose the optimal tool for a given verification or specification problem.
- Differentiate between safety and liveness properties.
- Contrast several related temporal logics (LTL, CTL, CTL\*) and properties expressible/inexpressible in each of them.

### Literatur:

- G. Winskel: The Formal Semantics of Programming Languages: An Introduction, The MIT Press, 1993.
- M. Huth, M. Ryan: Logic in Computer Science, Cambridge University Press, 2. Aufl., 2004.

### Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

#### [1] Informatik (Bachelor of Science)

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule | Vertiefungsrichtung Theoretische Informatik)

Dieses Modul ist daneben auch in den Studienfächern "Informatik (Master of Science)", "Mathematik (Bachelor of Science)" verwendbar.

---

**Studien-/Prüfungsleistungen:**

Formale Methoden der Softwareentwicklung (Prüfungsnummer: 151316)

Prüfungsleistung, mündliche Prüfung, Dauer (in Minuten): 30

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Formale Methoden der Softwareentwicklung

weitere Erläuterungen:

Die Modulnote setzt sich zu 50% aus dem Ergebnis einer 30-minütigen mündlichen Prüfung am Semesterende und zu 50% aus der Bewertung der Leistungen aus dem Übungsbetrieb zusammen.

Erstabledung: WS 2019/2020, 1. Wdh.: SS 2020 (nur für Wiederholer)

1. Prüfer: Tadeusz Litak (100372)

---

<b>Modulbezeichnung:</b> Betriebssystemtechnik (BST) (Operating Systems Engineering)	<b>5 ECTS</b>	
Modulverantwortliche/r:	Wolfgang Schröder-Preikschat	
Lehrende:	Wolfgang Schröder-Preikschat	
Startsemester: SS 2019	Dauer: 1 Semester	Turnus: jährlich (SS)
Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Deutsch

### Lehrveranstaltungen:

- Betriebssystemtechnik (SS 2019, Vorlesung, 2 SWS, Wolfgang Schröder-Preikschat)
- Übungen zu Betriebssystemtechnik (SS 2019, Übung, 2 SWS, Bernhard Heinloth et al.)
- Rechnerübungen zu Betriebssystemtechnik (SS 2019, Übung, Bernhard Heinloth et al.)

### Inhalt:

Schwerpunktthema der Veranstaltung ist die Verwaltung von Prozessadressräumen. Untersucht werden Verfahren und Techniken zur Trennung logischer Adressräume, zum Adressraumgrenzen überschreitenden Zugriff und zum Schutz von Prozessen. Vorgestellt wird die Implementierung von Systemaufrufen und seiten- wie auch segmentbasierte Techniken zur Abbildung logischer/virtueller Adressräume auf reale. Vor diesem Hintergrund werden verschiedene Betriebssystemarchitekturen verglichen und gängige Adressraummodelle von Betriebssystemen erläutert. Weitere Themen bildet die Interprozesskommunikation durch Nachrichtenversenden bei getrennten Adressräumen, aber auch die Nachbildung virtuell gemeinsamen Speichers auf Basis solcher Ansätze.

### Lernziele und Kompetenzen:

Studierende, die das Modul erfolgreich abgeschlossen haben:

- erläutern und implementieren Entwurfsprinzipien für Systemaufrufe und diskutieren deren spezifischen Vor-/Nachteile.
- vergleichen grundlegende BS-Architekturen (Monolith, Mikrokern, Makrokern, Exokern) anhand fundamentaler Charakteristika (Robustheit, Performanz, Portierbarkeit) und deren Einfluss auf die Implementierung von Mechanismen (Systemaufrufe, Adressraumschutz).
- unterscheiden Hierarchiekonzepte des Softwareentwurfs (Benutzthierarchie, funktionale Hierarchie) und erläutern deren Implikationen beim Betriebssystementwurf.
- klassifizieren Schutz-, Verwaltungs-, und Virtualisierungstechniken für Programmzustände (Seitennummerierung, Segmentierung, Sprachbasierung, Capabilities) und implementieren diese auf der IA-32-Architektur
- diskutieren Adressraummodelle (Mehradressraummodell, Einadressraummodell, mehrstufige und inverse Seitenabbildungen, Mitbenutzung) und deren Implementierbarkeit auf gängigen Hardwarearchitekturen.
- implementieren Mechanismen und Abstraktionen zur Interprozesskommunikation.
- erläutern das Zusammenspiel zwischen Kommunikation und Synchronisation im Bezug auf die besonderen Herausforderungen bei der Implementierung von Betriebssystemabstraktionen (lost wakeup, lost update, gepufferte/ungepufferte Zugriffe)
- diskutieren Prinzipien der Mitbenutzung von Code und Daten unter Berücksichtigung der Betriebssystem- und Adressraumarchitektur.
- erläutern die Funktionsweise eines Bindeladers und skizzieren effiziente Implementierungstechniken für positionsunabhängige Strukturen.
- erschließen sich typische Probleme (Nebenläufigkeit, Compilerverhalten, Debuggen ohne dedizierte Hilfsmittel) und Fehlerquellen bei der hardwarenahen Softwareentwicklung.
- können in Kleingruppen kooperativ arbeiten.
- können die ihre Entwurfs- und Implementierungsentscheidungen kompakt präsentieren und argumentativ vertreten.
- reflektieren ihre Entscheidungen kritisch und leiten Alternativen ab.
- können offen und konstruktiv mit Schwachpunkten und Irrwegen umgehen.

### Literatur:

- Wolfgang Schröder-Preikschat. The Logical Design of Parallel Operating Systems. Prentice Hall.

- Brian W. Kernighan, Dennis M. Ritchie. The C Programming Language. Prentice Hall.
- 
- David M. Weiss, Chi Tau Robert Lai. Software Product-Line Engineering: A Family-Based Software Development Process. Addison-Wesley.
- Krzysztof Czarnecki, Ulrich W. Eisenecker. Generative Programming. Methods, Tools and Applications. Addison-Wesley.

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

**[1] Informatik (Bachelor of Science): 5-6. Semester**

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule)

---

**Studien-/Prüfungsleistungen:**

Betriebssystemtechnik (Prüfungsnummer: 406841)

(englische Bezeichnung: Operating Systems Engineering)

Prüfungsleistung, Portfolio

Anteil an der Berechnung der Modulnote: 100%

weitere Erläuterungen:

30-minütige mündliche Prüfung + erfolgreiche Teilnahme an den Übungen

Erstablingung: SS 2019, 1. Wdh.: WS 2019/2020

1. Prüfer: Wolfgang Schröder-Preikschat (100114)

---

**Modulbezeichnung:** Betriebssystemtechnik (V + EÜ) (BST-VEU) 7.5 ECTS  
 (Operating Systems Engineering)

Modulverantwortliche/r: Wolfgang Schröder-Preikschat

Lehrende: Wolfgang Schröder-Preikschat

---

Startsemester: SS 2019

Dauer: 1 Semester

Turnus: jährlich (SS)

Präsenzzeit: 90 Std.

Eigenstudium: 135 Std.

Sprache: Deutsch

---

### Lehrveranstaltungen:

Betriebssystemtechnik (SS 2019, Vorlesung, 2 SWS, Wolfgang Schröder-Preikschat)

Übungen zu Betriebssystemtechnik (SS 2019, Übung, 2 SWS, Bernhard Heinloth et al.)

Rechnerübungen zu Betriebssystemtechnik (SS 2019, Übung, Bernhard Heinloth et al.)

---

### Inhalt:

Schwerpunktthema der Veranstaltung ist die Verwaltung von Prozessadressräumen. Untersucht werden Verfahren und Techniken zur Trennung logischer Adressräume, zum Adressraumgrenzen überschreitenden Zugriff und zum Schutz von Prozessen. Vorgestellt wird die Implementierung von Systemaufrufen und seiten- wie auch segmentbasierte Techniken zur Abbildung logischer/virtueller Adressräume auf reale. Vor diesem Hintergrund werden verschiedene Betriebssystemarchitekturen verglichen und gängige Adressraummodelle von Betriebssystemen erläutert. Weitere Themen bildet die Interprozesskommunikation durch Nachrichtenversenden bei getrennten Adressräumen, aber auch die Nachbildung virtuell gemeinsamen Speichers auf Basis solcher Ansätze.

### Lernziele und Kompetenzen:

Studierende, die das Modul erfolgreich abgeschlossen haben:

- erläutern und implementieren Entwurfsprinzipien für Systemaufrufe und diskutieren deren spezifischen Vor-/Nachteile.
- vergleichen grundlegende BS-Architekturen (Monolith, Mikrokern, Makrokern, Exokern) anhand fundamentaler Charakteristika (Robustheit, Performanz, Portierbarkeit) und deren Einfluss auf die Implementierung von Mechanismen (Systemaufrufe, Adressraumschutz).
- unterscheiden Hierarchiekonzepte des Softwareentwurfs (Benutzthierarchie, funktionale Hierarchie) und erläutern deren Implikationen beim Betriebssystementwurf.
- klassifizieren Schutz-, Verwaltungs-, und Virtualisierungstechniken für Programmzustände (Seitennummerierung, Segmentierung, Sprachbasierung, Capabilities) und implementieren diese auf der IA-32-Architektur
- diskutieren Adressraummodelle (Mehradressraummodell, Einadressraummodell, mehrstufige und inverse Seitenabbildungen, Mitbenutzung) und deren Implementierbarkeit auf gängigen Hardwarearchitekturen.
- implementieren Mechanismen und Abstraktionen zur Interprozesskommunikation.
- erläutern das Zusammenspiel zwischen Kommunikation und Synchronisation im Bezug auf die besonderen Herausforderungen bei der Implementierung von Betriebssystemabstraktionen (lost wakeup, lost update, gepufferte/ungepufferte Zugriffe)
- diskutieren Prinzipien der Mitbenutzung von Code und Daten unter Berücksichtigung der Betriebssystem- und Adressraumarchitektur.
- erläutern die Funktionsweise eines Bindeladers und skizzieren effiziente Implementierungstechniken für positionsunabhängige Strukturen.
- erschließen sich typische Probleme (Nebenläufigkeit, Compilerverhalten, Debuggen ohne dedizierte Hilfsmittel) und Fehlerquellen bei der hardwarenahen Softwareentwicklung.
- können in Kleingruppen kooperativ arbeiten.
- können die ihre Entwurfs- und Implementierungsentscheidungen kompakt präsentieren und argumentativ vertreten.
- reflektieren ihre Entscheidungen kritisch und leiten Alternativen ab.
- können offen und konstruktiv mit Schwachpunkten und Irrwegen umgehen.

### Literatur:

- Wolfgang Schröder-Preikschat. The Logical Design of Parallel Operating Systems. Prentice Hall.



- Brian W. Kernighan, Dennis M. Ritchie. The C Programming Language. Prentice Hall.
- 
- David M. Weiss, Chi Tau Robert Lai. Software Product-Line Engineering: A Family-Based Software Development Process. Addison-Wesley.
- Krzysztof Czarnecki, Ulrich W. Eisenecker. Generative Programming. Methods, Tools and Applications. Addison-Wesley.

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

**[1] Informatik (Bachelor of Science): 5-6. Semester**

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule)

---

**Studien-/Prüfungsleistungen:**

Betriebssystemtechnik (V + EÜ) (Prüfungsnummer: 350752)

(englische Bezeichnung: Operating Systems Engineering)

Prüfungsleistung, Portfolio

Anteil an der Berechnung der Modulnote: 100%

weitere Erläuterungen:

30-minütige mündliche Prüfung + erfolgreiche Teilnahme an den Übungen

Erstablingung: SS 2019, 1. Wdh.: WS 2019/2020

1. Prüfer: Wolfgang Schröder-Preikschat (100114)

---

**Modulbezeichnung:** Verteilte Systeme - V+Ü (VS) 5 ECTS  
 (Distributed Systems - L+E)

Modulverantwortliche/r: Tobias Distler

Lehrende: Tobias Distler, Jürgen Kleinöder

Startsemester: SS 2019

Dauer: 1 Semester

Turnus: jährlich (SS)

Präsenzzeit: 60 Std.

Eigenstudium: 90 Std.

Sprache: Deutsch

---

**Lehrveranstaltungen:**

Verteilte Systeme (SS 2019, Vorlesung, 2 SWS, Tobias Distler et al.)

Übungen zu Verteilte Systeme (SS 2019, Übung, 2 SWS, Michael Eischer et al.)

---

**Empfohlene Voraussetzungen:**

Gute Programmierkenntnisse in Java

---

**Inhalt:**

Verteilte Systeme bestehen aus mehreren Rechnern, die über ein Netzwerk miteinander verbunden sind und einen gemeinsamen Dienst erbringen. Obwohl die beteiligten Rechner hierfür in weiten Teilen unabhängig voneinander agieren, erscheinen sie ihren Nutzern gegenüber in der Gesamtheit dabei trotzdem als ein einheitliches System. Die Einsatzmöglichkeiten für verteilte Systeme erstrecken sich über ein weites Spektrum an Szenarien: Von der Zusammenschaltung kleinster Rechenknoten zur Sammlung von Daten im Rahmen von Sensornetzwerken über Steuerungssysteme für Kraftfahrzeuge und Industrieanlagen bis hin zu weltumspannenden, Internet-gestützten Infrastrukturen mit Komponenten in Datenzentren auf verschiedenen Kontinenten.

Ziel dieses Moduls ist es, die sich durch die speziellen Eigenschaften verteilter Systeme ergebenden Problemstellungen zu verdeutlichen und Ansätze zu vermitteln, mit deren Hilfe sie gelöst werden können; Beispiele hierfür sind etwa die Interaktion zwischen heterogenen Systemkomponenten, der Umgang mit erhöhten Netzwerklatenzen sowie die Wahrung konsistenter Zustände über Rechnergrenzen hinweg. Gleichzeitig zeigt das Modul auf, dass die Verteiltheit eines Systems nicht nur Herausforderungen mit sich bringt, sondern auf der anderen Seite auch Chancen eröffnet. Dies gilt insbesondere in Bezug auf die im Vergleich zu nicht verteilten Systemen erzielbare höhere Widerstandsfähigkeit eines Gesamtsystems gegenüber Fehlern wie den Ausfällen ganzer Rechner oder sogar kompletter Datenzentren.

Ausgehend von den einfachsten, aus nur einem Client und einem Server bestehenden verteilten Systemen, beschäftigt sich die Vorlesung danach mit der deutlich komplexeren Replikation der Server-Seite und behandelt anschließend die Verteilung eines Systems über mehrere, mitunter weit voneinander entfernte geografische Standorte. In allen Abschnitten umfasst die Betrachtung des jeweiligen Themas eine Auswahl aus Grundlagen, im Praxiseinsatz befindlicher Ansätze und Techniken sowie für den aktuellen Stand der Forschung repräsentativer Konzepte.

Im Rahmen der Übungen wird zunächst ein plattformunabhängiges Fernaufrufsystem schrittweise entwickelt und parallel dazu getestet. Als Vorlage und Orientierungshilfe dient dabei das in der Praxis weit verbreitete Java RMI. In den weiteren Übungsaufgaben stehen anschließend klassische Problemstellungen von verteilten Systemen wie fehlertolerante Replikation und verteilte Synchronisation im Mittelpunkt.

**Lernziele und Kompetenzen:**

Studierende, die das Modul erfolgreich abgeschlossen haben:

- beschreiben charakteristische Merkmale und Eigenschaften verteilter Systeme sowie grundlegende Probleme im Zusammenhang mit ihrer Realisierung.
- untersuchen die Unterschiede zwischen lokalen Methodenaufrufen und Fernmethodenaufrufen.
- vergleichen Ansätze zur Konvertierung von Nachrichten zwischen verschiedenen Datenrepräsentationen.
- konzipieren eine eigene auf Java RMI basierende Anwendung.
- entwickeln ein eigenes Fernaufrufsystem nach dem Vorbild von Java RMI.
- gestalten ein Modul zur Unterstützung verschiedener Fernaufrufsemantiken (Maybe, Last-of-Many) für das eigene Fernaufrufsystem.

- beurteilen auf Basis eigener Experimente mit Fehlerinjektionen die Auswirkungen von Störeinflüssen auf verschiedene Fernaufrufsemantiken.
- klassifizieren Mechanismen zur Bereitstellung von Fehlertoleranz, insbesondere verschiedene Arten der Replikation (aktiv vs. passiv).
- entwickeln ein eigene aktiv replizierte Anwendung zur Erforschung der mit dieser Replikationsart verbundenen Problemstellungen (z.B. Determinismus).
- vergleichen verschiedene Konsistenzgarantien georeplizierter Systeme.
- illustrieren das Problem einer fehlenden gemeinsamen Zeitbasis in verteilten Systemen.
- erforschen logische Uhren als Mittel zur Reihenfolgebestimmung und Methoden zur Synchronisation physikalischer Uhren.
- unterscheiden grundlegende Zustellungs- und Ordnungsgarantien beim Multicast von Nachrichten.
- gestalten ein Protokoll für den zuverlässigen und totalgeordneten Versand von Nachrichten in einer Gruppe von Knoten.
- entwickeln einen Dienst zur Verwaltung verteilter Sperrobjekte auf Basis von Lamport-Locks.
- bewerten die Qualität einer aktuellen Publikation aus der Fachliteratur.
- erschließen sich typische Probleme (Nebenläufigkeit, Konsistenz) und Fehlerquellen bei der Programmierung verteilter Anwendungen.
- können in Kleingruppen kooperativ arbeiten.
- können ihre Entwurfs- und Implementierungsentscheidungen kompakt präsentieren und argumentativ vertreten.
- reflektieren ihre Entscheidungen kritisch und leiten Alternativen ab.
- können offen und konstruktiv mit Schwachpunkten und Irrwegen umgehen.

---

### Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

#### [1] Informatik (Bachelor of Science): 5-6. Semester

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule)

Dieses Modul ist daneben auch in den Studienfächern "Informations- und Kommunikationstechnik (Bachelor of Science)", "Informations- und Kommunikationstechnik (Master of Science)", "International Information Systems (IIS) (Master of Science)", "Mathematik (Bachelor of Science)", "Mechatronik (Bachelor of Science)", "Mechatronik (Master of Science)" verwendbar.

---

### Studien-/Prüfungsleistungen:

Verteilte Systeme (Prüfungsnummer: 52801)

(englische Bezeichnung: Distributed Systems)

Prüfungsleistung, mündliche Prüfung, Dauer (in Minuten): 30

Anteil an der Berechnung der Modulnote: 100%

Erstablingung: SS 2019, 1. Wdh.: WS 2019/2020

1. Prüfer: Tobias Distler (100377)

Verteilte Systeme - V+Ü (Prüfungsnummer: 649073)

(englische Bezeichnung: Distributed Systems - L+E)

Prüfungsleistung, Portfolio, Dauer (in Minuten): 30

Anteil an der Berechnung der Modulnote: 100%

weitere Erläuterungen:

Erfolgreiche Bearbeitung aller Übungsaufgaben (Bewertung jeweils mit "ausreichend") und mündliche Prüfung.

Die Modulnote ergibt sich zu 100% aus der Bewertung der mündlichen Prüfung.

Erstablingung: SS 2019, 1. Wdh.: WS 2019/2020

1. Prüfer: Tobias Distler (100377)

---

**Modulbezeichnung:** **Automotive Systems & Software Engineering (ASSE)** **5 ECTS**  
 (Automotive Systems & Software Engineering)

Modulverantwortliche/r: Reinhard German  
 Lehrende: Christian Allmann

---

Startsemester: SS 2019	Dauer: 1 Semester	Turnus: jährlich (SS)
Präsenzzeit: 40 Std.	Eigenstudium: 110 Std.	Sprache: Deutsch

---

**Lehrveranstaltungen:**

Automotive Systems & Software Engineering (SS 2019, Vorlesung, 4 SWS, Christian Allmann et al.)

---

**Inhalt:**

Allgemein:

- Überblick Systeme und Anwendungen in der automotive Welt
- Embedded Automotive Systems: Abgrenzung Echtzeitsysteme; Definition Steuergeräte, Sensoren, Aktuatoren; Definition verteilte, vernetzte Funktionen; Betriebssysteme, Kommunikationsschnittstellen
- Entwicklungsprozesse für automotive Anwendungen: Allgemeine Vorgehensmodelle; Vorgehensmodelle in der Elektronikentwicklung; Kooperationsmodelle OEM-Zulieferer; Lieferantenmanagement

System und Software:

- Softwareentwicklungsprozess
- Anforderungsmanagement: Lastenhefte - Aufbau/Handhabung/Tools; Erhebung und Analyse von Anforderungen; Systembeschreibungen (u.a. UML, SysML); (Semi-)Formale Beschreibungen
- Modellbasierte Entwicklung: Rapid Prototyping; Modellierungstechniken; Automatische Codegenerierung; Tools (Simulink, Target Link); Autosar
- Test und Diagnose: SW-Test; Integrationsstufen; SIL - PIL - HIL-Test; On- Offboard Diagnose
- Virtuelle Entwicklung von Elektronik
- Begleitende Prozesse: Projektmanagement; Systemsicherheit (ISO WD 26262); Konfigurationsmanagement; Qualitätsmanagement; Variantenmanagement
- Architektur: Architekturmodelle (u.a. EAST-ADL); Bussysteme - Typen & Eigenschaften; SW- & HW-Architektur; Auslegung und Bewertung

Anwendung:

- Überblick der Anwendungsdomänen: Infotainment/ Fahrerassistenz/ Karosserie
- Systemauslegung von Fahrerassistenzsystemen: Überblick Sensoren; Umwelt und Umfeldmodelle; Bildverarbeitung; Konzeption, Erprobung, Umsetzung; Tools (ADTF)
- Alternative Antriebskonzepte: Elektronikumfänge; Betriebsstrategie
- Fahrdynamische Systeme: x-by-wire Technologien

**Lernziele und Kompetenzen:**

- Vertiefung der Themen aus „Entwicklung und Test von verteilten, eingebetteten Systemen im Bereich Automotive“WS, Dr. Hehn
- Herausforderungen in der System- und Softwareentwicklung aus Sicht eines OEMs kennenlernen
- Bedeutung des Requirements Engineering in Vor- & Serienentwicklung verstehen und die Methoden anwenden können
- Herausforderungen in der Entwicklung von Fahrerassistenzsystemen erkennen und die Konzepte nachvollziehen können
- Entwicklung eines eigenen FAS durchführen und prüfen
- Erfahrung im Umgang mit typischen Entwicklungstools
- Absicherung der Systementwicklung am HIL
- Neue Ideen und Themen im Bereich automotive Entwicklung verstehen

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

[1] **Informatik (Bachelor of Science): 5-6. Semester**

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule)

**Studien-/Prüfungsleistungen:**

Automotive Systems & Software Engineering (Prüfungsnummer: 313638)

(englische Bezeichnung: Automotive Systems & Software Engineering)

Prüfungsleistung, Klausur, Dauer (in Minuten): 90

Anteil an der Berechnung der Modulnote: 100%

Erstabelleung: SS 2019, 1. Wdh.: WS 2019/2020

1. Prüfer: Anatoli Djanatliev (100361)

---

---

**Modulbezeichnung:** Echtzeitsysteme 2 - Verlässliche Echtzeitsysteme **5 ECTS**  
 (V+Ü) (EZS2)  
 (Real-Time Systems 2 - Dependable Real-Time Systems (V+Ü))

Modulverantwortliche/r: Peter Ulbrich  
 Lehrende: Peter Ulbrich, Peter Wägemann

---

Startsemester: SS 2019	Dauer: 1 Semester	Turnus: jährlich (SS)
Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Deutsch

---

**Lehrveranstaltungen:**

Echtzeitsysteme 2 - Verlässliche Echtzeitsysteme (SS 2019, Vorlesung, 2 SWS, Peter Ulbrich)  
 Übungen zu Echtzeitsysteme 2 - Verlässliche Echtzeitsysteme (SS 2019, Übung, Simon Schuster et al.)

---

**Inhalt:**

Viele Echtzeitsysteme sind in Bereiche des täglichen Lebens eingebettete, die hohe Anforderungen an die funktionale Sicherheit dieser Systeme stellen. Beispiele hierfür sind Fahrerassistenzsysteme in modernen Automobilen, medizinische Geräte, Prozessanlagen in Kernkraftwerken oder Chemiefabriken oder Flugzeuge. Fehlfunktionen in diesen Anwendungen ziehen mitunter katastrophale Konsequenzen nach sich - Menschen können ernsthaft verletzt oder sogar getötet werden, Landstriche können unbewohnbar gemacht oder zumindest großer finanzieller Schaden verursacht werden.

Dieses Modul betrachtet Methoden und Werkzeuge, die uns helfen können, einerseits **zuverlässig Software zu entwickeln** (also Fehler im Programm zu entdecken und zu vermeiden), und andererseits **zuverlässige Software zu entwickeln** (also Abstraktionen, die auch im Fehlerfall ihre Gültigkeit behalten). Hierbei steht weniger die Vermittlung theoretischer Grundkenntnisse auf diesen Gebieten im Vordergrund, also vielmehr

- die praktische Anwendung existierende Werkzeuge und Methoden
- sowie die Erfahrung und das Verständnis ihrer Grenzen.

Auf diese Weise soll ein Fundament für die konstruktive Umsetzung verlässlicher Echtzeitsysteme gelegt werden. Dieses Modul soll daher fundierte Anknüpfungspunkte für die Entwicklung verlässlicher Echtzeitsysteme vermitteln, die Ad-hoc-Techniken möglichst ersetzen sollen.

**Lernziele und Kompetenzen:**

Studierende, die das Modul erfolgreich abgeschlossen haben:

- nennen die Konzepte und die Taxonomie verlässlicher Systeme, unterscheiden Software- und Hardwarefehler und klassifizieren Fehler (Defekt, Fehler, Fehlverhalten).
- stellen Fehlerbäume auf.
- organisieren Softwareentwicklungsprojekte mittels der Versionsverwaltung git.
- vergleichen die verschiedenen Arten der Redundanz als Grundvoraussetzung für Fehlererkennung und -toleranz.
- entwickeln fehlertolerante Systeme mittels Replikation.
- diskutieren die Fehlerhypothese und die Sicherstellung von Replikdeterminismus.
- erläutern die Vor- und Nachteile softwarebasierter Replikation und den Einsatz von Diversität.
- wenden Informationsredundanz zur Härtung von Daten- und Kontrollflüssen an.
- bewerten die Effektivität der arithmetischer Codierung von Programmen und verallgemeinern diesen Ansatz auf die verschiedenen Implementierungsebenen (Maschinenprogramm zu Prozessinkarnation).
- interpretieren den Einfluss der Ausführungsplattform (Hardware, Betriebssystem) auf die Leistungsfähigkeit der Fehlererkennung.
- konzipieren eine fehlertolerante Ausführungsumgebung für ein softwarebasiertes TMR-System basierend auf ANBD-Codierung.
- nennen die Grundlagen der systematischen Fehlerinjektion.
- überprüfen die Wirksamkeit von Fehlertoleranzmechanismen mittels Fehlerinjektion auf der Befehlsatzebene.
- entwickeln Testfälle für die Fehlerinjektion mittels des fail\* Werkzeugs.
- setzen Messergebnisse in Relation zu dem tatsächlichen Fehlerraum.

- beschreiben die Grundlagen der Fehlererholung (Vorwärts- bzw. Rückwärtskorrektur) und Reintegration fehlgeschlagener Knoten.
- vergleichen den Zustandstransfer am Beispiel der Running bzw. Recursive State Restoration.
- benennen Konzepte der Rückwärtskorrektur durch Entwurfalternativen (Recovery Blocks).
- fassen die Grundlagen des dynamischen Testens zusammen.
- unterscheiden Black-Box und White-Box Testverfahren.
- konzipieren und implementieren Testfälle.
- überprüfen die Testüberdeckung anhand grundlegender Überdeckungskriterien (Anweisungs- bis Bedingungsüberdeckung).
- geben die Grundlagen der statischen Programmanalyse wieder.
- nennen die Funktionsweise von Hoare- WP-Kalkül.
- verifizieren eine Ampelsteuerung mittels des FramaC Werkzeugs zur statischen Analyse von C Programmen.
- beschreiben den Korrektheitsnachweis mittels abstrakter Interpretation und unterscheiden die konkrete von der abstrakten Programmsemantik.
- erläutern die Funktionsweise von Sammel- und Präfixsemantiken.
- erstellen einen Korrektheitsbeweis für einen a-b-Filter mittels des Astrée Werkzeugs zur abstrakten Interpretation von C Programmen.
- bewerten die Verlässlichkeit kommerzieller, sicherheitskritischer Systeme anhand von Fallstudien (Sizewell B, Airbus A320).
- erschließen sich typische Probleme und Fehlerquellen bei der Programmierung von eingebetteten Systemen im Allgemeinen.
- klassifizieren Fallstricke und Mehrdeutigkeiten in der Programmiersprache C99 im Besonderen.
- können in Gruppen kooperativ und effektiv arbeiten.
- können ihre Entwurfs- und Implementierungsentscheidungen kompakt präsentieren und argumentativ vertreten.
- reflektieren ihre Entscheidungen kritisch und leiten Alternativen ab.
- können offen und konstruktiv mit Schwachpunkten in der Konzeption wie Umsetzung umgehen.

#### **Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

**[1] Informatik (Bachelor of Science): 5-6. Semester**

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule)

Dieses Modul ist daneben auch in den Studienfächern "Maschinenbau (Master of Science)", "Medizintechnik (Master of Science)", "Wirtschaftsingenieurwesen (Master of Science)" verwendbar.

#### **Studien-/Prüfungsleistungen:**

Verlässliche Echtzeitsysteme (Prüfungsnummer: 876012)

(englische Bezeichnung: Dependable Real-Time Systems)

Prüfungsleistung, Portfolio

Anteil an der Berechnung der Modulnote: 100%

weitere Erläuterungen:

mündliche Prüfung (30 Minuten) + erfolgreiche Bearbeitung aller Übungsaufgaben

Erstablægung: SS 2019, 1. Wdh.: WS 2019/2020

1. Prüfer: Peter Ulbrich (100360)

---

**Modulbezeichnung:** Echtzeitsysteme 2 - Verlässliche Echtzeitsysteme 7.5 ECTS  
 (V+EÜ) (EZS2)  
 (Real-Time Systems 2 - Dependable Real-Time Systems)

Modulverantwortliche/r: Peter Ulbrich  
 Lehrende: Peter Ulbrich, Peter Wägemann

---

Startsemester: SS 2019	Dauer: 1 Semester	Turnus: jährlich (SS)
Präsenzzeit: 60 Std.	Eigenstudium: 165 Std.	Sprache: Deutsch

---

**Lehrveranstaltungen:**

Echtzeitsysteme 2 - Verlässliche Echtzeitsysteme (SS 2019, Vorlesung, 2 SWS, Peter Ulbrich)  
 Übungen zu Echtzeitsysteme 2 - Verlässliche Echtzeitsysteme (SS 2019, Übung, Simon Schuster et al.)

---

**Inhalt:**

Viele Echtzeitsysteme sind in Bereiche des täglichen Lebens eingebettete, die hohe Anforderungen an die funktionale Sicherheit dieser Systeme stellen. Beispiele hierfür sind Fahrerassistenzsysteme in modernen Automobilen, medizinische Geräte, Prozessanlagen in Kernkraftwerken oder Chemiefabriken oder Flugzeuge. Fehlfunktionen in diesen Anwendungen ziehen mitunter katastrophale Konsequenzen nach sich - Menschen können ernsthaft verletzt oder sogar getötet werden, Landstriche können unbewohnbar gemacht oder zumindest großer finanzieller Schaden verursacht werden.

Dieses Modul betrachtet Methoden und Werkzeuge, die uns helfen können, einerseits **zuverlässig Software zu entwickeln** (also Fehler im Programm zu entdecken und zu vermeiden), und andererseits **zuverlässige Software zu entwickeln** (also Abstraktionen, die auch im Fehlerfall ihre Gültigkeit behalten). Hierbei steht weniger die Vermittlung theoretischer Grundkenntnisse auf diesen Gebieten im Vordergrund, also vielmehr

- die praktische Anwendung existierende Werkzeuge und Methoden
- sowie die Erfahrung und das Verständnis ihrer Grenzen.

Auf diese Weise soll ein Fundament für die konstruktive Umsetzung verlässlicher Echtzeitsysteme gelegt werden. Dieses Modul soll daher fundierte Anknüpfungspunkte für die Entwicklung verlässlicher Echtzeitsysteme vermitteln, die Ad-hoc-Techniken möglichst ersetzen sollen.

**Lernziele und Kompetenzen:**

Studierende, die das Modul erfolgreich abgeschlossen haben:

- nennen die Konzepte und die Taxonomie verlässlicher Systeme, unterscheiden Software- und Hardwarefehler und klassifizieren Fehler (Defekt, Fehler, Fehlverhalten).
- stellen Fehlerbäume auf.
- organisieren Softwareentwicklungsprojekte mittels der Versionsverwaltung git.
- vergleichen die verschiedenen Arten der Redundanz als Grundvoraussetzung für Fehlererkennung und -toleranz.
- entwickeln fehlertolerante Systeme mittels Replikation.
- diskutieren die Fehlerhypothese und die Sicherstellung von Replikdeterminismus.
- erläutern die Vor- und Nachteile softwarebasierter Replikation und den Einsatz von Diversität.
- wenden Informationsredundanz zur Härtung von Daten- und Kontrollflüssen an.
- bewerten die Effektivität der arithmetischer Codierung von Programmen und verallgemeinern diesen Ansatz auf die verschiedenen Implementierungsebenen (Maschinenprogramm zu Prozessinkarnation).
- interpretieren den Einfluss der Ausführungsplattform (Hardware, Betriebssystem) auf die Leistungsfähigkeit der Fehlererkennung.
- konzipieren eine fehlertolerante Ausführungsumgebung für ein softwarebasiertes TMR-System basierend auf ANBD-Codierung.
- nennen die Grundlagen der systematischen Fehlerinjektion.
- überprüfen die Wirksamkeit von Fehlertoleranzmechanismen mittels Fehlerinjektion auf der Befehlsatzebene.
- entwickeln Testfälle für die Fehlerinjektion mittels des fail\* Werkzeugs.
- setzen Messergebnisse in Relation zu dem tatsächlichen Fehlerraum.



- beschreiben die Grundlagen der Fehlererholung (Vorwärts- bzw. Rückwärtskorrektur) und Reintegration fehlgeschlagener Knoten.
- vergleichen den Zustandstransfer am Beispiel der Running bzw. Recursive State Restoration.
- benennen Konzepte der Rückwärtskorrektur durch Entwurfalternativen (Recovery Blocks).
- fassen die Grundlagen des dynamischen Testens zusammen.
- unterscheiden Black-Box und White-Box Testverfahren.
- konzipieren und implementieren Testfälle.
- überprüfen die Testüberdeckung anhand grundlegender Überdeckungskriterien (Anweisungs- bis Bedingungsüberdeckung).
- geben die Grundlagen der statischen Programmanalyse wieder.
- nennen die Funktionsweise von Hoare- WP-Kalkül.
- verifizieren eine Ampelsteuerung mittels des FramaC Werkzeugs zur statischen Analyse von C Programmen.
- beschreiben den Korrektheitsnachweis mittels abstrakter Interpretation und unterscheiden die konkrete von der abstrakten Programmsemantik.
- erläutern die Funktionsweise von Sammel- und Präfixsemantiken.
- erstellen einen Korrektheitsbeweis für einen a-b-Filter mittels des Astrée Werkzeugs zur abstrakten Interpretation von C Programmen.
- bewerten die Verlässlichkeit kommerzieller, sicherheitskritischer Systeme anhand von Fallstudien (Sizewell B, Airbus A320).
- erschließen sich typische Probleme und Fehlerquellen bei der Programmierung von eingebetteten Systemen im Allgemeinen.
- klassifizieren Fallstricke und Mehrdeutigkeiten in der Programmiersprache C99 im Besonderen.
- können in Gruppen kooperativ und effektiv arbeiten.
- können ihre Entwurfs- und Implementierungsentscheidungen kompakt präsentieren und argumentativ vertreten.
- reflektieren ihre Entscheidungen kritisch und leiten Alternativen ab.
- können offen und konstruktiv mit Schwachpunkten in der Konzeption wie Umsetzung umgehen.

---

#### Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

[1] **Informatik (Bachelor of Science): 5-6. Semester**

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule)

---

#### Studien-/Prüfungsleistungen:

Echtzeitsysteme 2 - Verlässliche Echtzeitsysteme (Prüfungsnummer: 939179)

(englische Bezeichnung: Real-Time Systems 2 - Dependable Real-Time Systems)

Prüfungsleistung, Portfolio

Anteil an der Berechnung der Modulnote: 100%

weitere Erläuterungen:

mündliche Prüfung (30 Minuten) + erfolgreiche Bearbeitung aller Übungsaufgaben

Erstablingung: SS 2019, 1. Wdh.: WS 2019/2020

1. Prüfer: Peter Ulbrich (100360)

---

---

**Modulbezeichnung:** **Verteilte Systeme - V+EÜ (VS)** **7.5 ECTS**  
 (Distributed Systems - L+EE)

Modulverantwortliche/r: Tobias Distler  
 Lehrende: Tobias Distler, Jürgen Kleinöder

---

Startsemester: SS 2019	Dauer: 1 Semester	Turnus: jährlich (SS)
Präsenzzeit: 90 Std.	Eigenstudium: 135 Std.	Sprache: Deutsch

---

**Lehrveranstaltungen:**

Verteilte Systeme (SS 2019, Vorlesung, 2 SWS, Tobias Distler et al.)  
 Erweiterte Übungen zu Verteilte Systeme (SS 2019, Übung, 4 SWS, Michael Eischer et al.)

---

**Empfohlene Voraussetzungen:**

Gute Programmierkenntnisse in Java

---

**Inhalt:**

Verteilte Systeme bestehen aus mehreren Rechnern, die über ein Netzwerk miteinander verbunden sind und einen gemeinsamen Dienst erbringen. Obwohl die beteiligten Rechner hierfür in weiten Teilen unabhängig voneinander agieren, erscheinen sie ihren Nutzern gegenüber in der Gesamtheit dabei trotzdem als ein einheitliches System. Die Einsatzmöglichkeiten für verteilte Systeme erstrecken sich über ein weites Spektrum an Szenarien: Von der Zusammenschaltung kleinster Rechenknoten zur Sammlung von Daten im Rahmen von Sensornetzwerken über Steuerungssysteme für Kraftfahrzeuge und Industrieanlagen bis hin zu weltumspannenden, Internet-gestützten Infrastrukturen mit Komponenten in Datenzentren auf verschiedenen Kontinenten.

Ziel dieses Moduls ist es, die sich durch die speziellen Eigenschaften verteilter Systeme ergebenden Problemstellungen zu verdeutlichen und Ansätze zu vermitteln, mit deren Hilfe sie gelöst werden können; Beispiele hierfür sind etwa die Interaktion zwischen heterogenen Systemkomponenten, der Umgang mit erhöhten Netzwerklatenzen sowie die Wahrung konsistenter Zustände über Rechnergrenzen hinweg. Gleichzeitig zeigt das Modul auf, dass die Verteiltheit eines Systems nicht nur Herausforderungen mit sich bringt, sondern auf der anderen Seite auch Chancen eröffnet. Dies gilt insbesondere in Bezug auf die im Vergleich zu nicht verteilten Systemen erzielbare höhere Widerstandsfähigkeit eines Gesamtsystems gegenüber Fehlern wie den Ausfällen ganzer Rechner oder sogar kompletter Datenzentren.

Ausgehend von den einfachsten, aus nur einem Client und einem Server bestehenden verteilten Systemen, beschäftigt sich die Vorlesung danach mit der deutlich komplexeren Replikation der Server-Seite und behandelt anschließend die Verteilung eines Systems über mehrere, mitunter weit voneinander entfernte geografische Standorte. In allen Abschnitten umfasst die Betrachtung des jeweiligen Themas eine Auswahl aus Grundlagen, im Praxiseinsatz befindlicher Ansätze und Techniken sowie für den aktuellen Stand der Forschung repräsentativer Konzepte.

Im Rahmen der Übungen wird zunächst ein plattformunabhängiges Fernaufrufsystem schrittweise entwickelt und parallel dazu getestet. Als Vorlage und Orientierungshilfe dient dabei das in der Praxis weit verbreitete Java RMI. In den weiteren Übungsaufgaben stehen anschließend klassische Problemstellungen von verteilten Systemen wie fehlertolerante Replikation und verteilte Synchronisation im Mittelpunkt.

**Lernziele und Kompetenzen:**

Studierende, die das Modul erfolgreich abgeschlossen haben:

- beschreiben charakteristische Merkmale und Eigenschaften verteilter Systeme sowie grundlegende Probleme im Zusammenhang mit ihrer Realisierung.
- untersuchen die Unterschiede zwischen lokalen Methodenaufrufen und Fernmethodenaufrufen.
- vergleichen Ansätze zur Konvertierung von Nachrichten zwischen verschiedenen Datenrepräsentationen.
- konzipieren eine eigene auf Java RMI basierende Anwendung.
- entwickeln ein eigenes Fernaufrufsystem nach dem Vorbild von Java RMI.
- bewerten die Serialisierungsroutinen von Java RMI.
- erproben die manuelle Serialisierung von Nachrichten.

- bewerten die Performanz des eigenen Fernaufrufsystems.
- gestalten ein Modul zur Unterstützung verschiedener Fernaufrufsemantiken (Maybe, Last-of-Many, At-Most-Once) für das eigene Fernaufrufsystem.
- beurteilen auf Basis eigener Experimente mit Fehlerinjektionen die Auswirkungen von Störeinflüssen auf verschiedene Fernaufrufsemantiken.
- klassifizieren Mechanismen zur Bereitstellung von Fehlertoleranz, insbesondere verschiedene Arten der Replikation (aktiv vs. passiv).
- entwickeln ein eigene aktiv replizierte Anwendung zur Erforschung der mit dieser Replikationsart verbundenen Problemstellungen (z.B. Determinismus).
- konzipieren einen Zustandstransfermechanismus für die eigene replizierte Anwendung, mit dessen Hilfe der Neustart von Replikaten toleriert werden kann.
- vergleichen verschiedene Konsistenzgarantien georeplizierter Systeme.
- illustrieren das Problem einer fehlenden gemeinsamen Zeitbasis in verteilten Systemen.
- erforschen logische Uhren als Mittel zur Reihenfolgebestimmung und Methoden zur Synchronisation physikalischer Uhren.
- unterscheiden grundlegende Zustellungs- und Ordnungsgarantien beim Multicast von Nachrichten.
- gestalten ein Protokoll für den zuverlässigen und totalgeordneten Versand von Nachrichten in einer Gruppe von Knoten.
- entwickeln einen Dienst zur Verwaltung verteilter Sperrobjekte auf Basis von Lamport-Locks.
- bewerten die Qualität einer aktuellen Publikation aus der Fachliteratur.
- erschließen sich typische Probleme (Nebenläufigkeit, Konsistenz) und Fehlerquellen bei der Programmierung verteilter Anwendungen.
- können in Kleingruppen kooperativ arbeiten.
- können ihre Entwurfs- und Implementierungsentscheidungen kompakt präsentieren und argumentativ vertreten.
- reflektieren ihre Entscheidungen kritisch und leiten Alternativen ab.
- können offen und konstruktiv mit Schwachpunkten und Irrwegen umgehen.

---

### Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

[1] **Informatik (Bachelor of Science): 5-6. Semester**

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule)

---

### Studien-/Prüfungsleistungen:

Verteilte Systeme (Vorlesung mit erweiterten Übungen) (Prüfungsnummer: 743260)

(englische Bezeichnung: Distributed Systems (Lecture with Extended Exercises))

Prüfungsleistung, Portfolio, Dauer (in Minuten): 30

Anteil an der Berechnung der Modulnote: 100%

weitere Erläuterungen:

Erfolgreiche Bearbeitung aller Übungsaufgaben zu den erweiterten Übungen (Bewertung jeweils mit "ausreichend") und mündliche Prüfung.

Die Modulnote ergibt sich zu 100% aus der Bewertung der mündlichen Prüfung.

Erstablægung: SS 2019, 1. Wdh.: WS 2019/2020

1. Prüfer: Tobias Distler (100377)

---

**Modulbezeichnung:** Betriebssysteme-V+EÜ (BS-VEU) 7.5 ECTS  
 (Operating Systems L+EE)

Modulverantwortliche/r: Volkmar Sieh

Lehrende: Volkmar Sieh, Wolfgang Schröder-Preikschat

---

Startsemester: WS 2019/2020

Dauer: 1 Semester

Turnus: jährlich (WS)

Präsenzzeit: 90 Std.

Eigenstudium: 135 Std.

Sprache: Deutsch

---

**Lehrveranstaltungen:**

Betriebssysteme (WS 2019/2020, Vorlesung, 2 SWS, Volkmar Sieh)

Erweiterte Übungen zu Betriebssysteme (WS 2019/2020, Übung, 4 SWS, Bernhard Heinloth et al.)

Übungen zu Betriebssysteme (Rechnerübungen) (WS 2019/2020, Übung, Bernhard Heinloth et al.)

---

**Empfohlene Voraussetzungen:**

- C/C++ (Übungsaufgaben werden in C++ implementiert)
- Assembler (Grundkenntnisse)

**Vorhergehende Module:**

Systemprogrammierung

---

**Inhalt:**

Inhalt des Moduls ist Vermittlung grundlegender Konzepte, Methoden und Techniken von Betriebssystemen aus dem Blickwinkel einer Betriebssystementwicklerin. Im Rahmen der Übungen entwickeln die Studierenden in einem "bottom-up" Entwurf- und Entwicklungsprozess ihr eigenes Mehrkernbetriebssystem für die IA-32 Plattform, ausgehend von der "nackten Hardware" über grundlegende Ein-Ausgabemöglichkeiten, Unterbrechungsbearbeitung bis hin zu quasi- und echtparalleler Programmausführung.

**Lernziele und Kompetenzen:**

Studierende, die das Modul erfolgreich abgeschlossen haben:

- erläutern den Startvorgang eines Rechensystems am Beispiel eines IA32 PCs.
- beschreiben die spezifischen Herausforderungen bei der Softwareentwicklung für "bare metal".
- beschreiben den Ablauf einer Unterbrechungsbehandlung von der Hardware bis zur (System-)software.
- skizzieren Besonderheiten und Strategien der Unterbrechungsbehandlung in Hardware für Mehrkernsystemen am Beispiel des IA32-APICs.
- diskutieren die Aufgabenteilung zwischen Hardware und Systemsoftware bei der Unterbrechungsbearbeitung.
- unterscheiden die verschiedenen Typen von Kontrollflüssen in einem Betriebssystem anhand des Ebenenmodells.
- unterscheiden harte, mehrstufige, und weiche Verfahren zur Unterbrechungssynchronisation in Betriebssystemen und können diese für ein Mehrkernsystem implementieren.
- klassifizieren konkrete Konkurrenzsituationen anhand des Ebenenmodells und leiten daraus geeignete Synchronisationsmaßnahmen ab.
- schildern die IA32-Architektur und gängige PC-Technologie und deren Schnittstellen zur Systemsoftware.
- erläutern grundlegende Bausteine für die Implementierung von Quasi-Parallelität (Fortsetzungen, Koroutinen, Fäden) und grenzen diese gegeneinander ab.
- erläutern die Interaktionen zwischen Hardware, Übersetzer und Systemsoftware, die dabei zu beachten sind.
- entwickeln den Koroutinenwechsel für einen gegebene Architektur.
- erläutern die Implikationen von Quasi-Parallelität auf das Ebenenmodell und die daraus abgeleiteten Synchronisationsmaßnahmen.
- beschreiben die Implementierung von (verdrängendem) Scheduling in einem Betriebssystem.
- analysieren das Zusammenspiel von Scheduling und Unterbrechungssynchronisation.

- nennen Kriterien und Dimensionen des Scheduling von Betriebsmitteln, insbesondere der CPU.
- erläutern die konkrete Umsetzung am Beispiel der Scheduler in Linux und Windows.
- unterscheiden grundlegende Möglichkeiten der Koordinierung und Synchronisation von Fäden (aktives/passives Warten, nichtverdrängbare kritische Abschnitte).
- entwickeln Mechanismen für die Synchronisation auf Fadenebene für Mehrkernsystemen.
- erklären die dabei zu beachtenden Synchronisationsprobleme (lost update, lost wakeup) und geeignete Gegenmaßnahmen.
- interpretieren die Bedeutung von Gerätetreibern in der Betriebssystempraxis.
- erläutern die Anforderungen an ein Treibermodell.
- vergleichen die Umsetzung von Treibermodellen in Windows und Linux.
- vergleichen grundlegende BS-Architekturen (Bibliothek, Monolith, Mikrokern, Exokern, Hypervisor) anhand fundamentaler Charakteristika (Robustheit, Performanz, Portierbarkeit) und Mechanismen.
- schildern die grundlegenden Paradigmen zur Interprozesskommunikation in Betriebssystemen (speicherbasiert vs. nachrichtenbasiert).
- erläutern die grundlegenden Primitiven dieser Verfahren.
- skizzieren, wie unter Anwendung dieser Primitiven höhere Synchronisationskonstrukte implementiert werden (Monitore, Leser-/Schreiber-Sperre).
- illustrieren die Dualität der Paradigmen.
- erschließen sich typische Probleme (Nebenläufigkeit, Compilerverhalten, Debuggen ohne dedizierte Hilfsmittel) und Fehlerquellen bei der hardwarenahen Softwareentwicklung für Mehrkernsysteme.
- können in Kleingruppen kooperativ arbeiten.
- können die ihre Entwurfs- und Implementierungsentscheidungen kompakt präsentieren und argumentativ vertreten.
- reflektieren ihre Entscheidungen kritisch und leiten Alternativen ab.
- können offen und konstruktiv mit Schwachpunkten und Irrwegen umgehen.

#### Literatur:

Hinweise zu geeigneter Literatur erfolgen in der Vorlesung.

#### Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

##### [1] Informatik (Bachelor of Science)

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule | Vertiefungsrichtung Verteilte Systeme und Betriebssysteme)

Dieses Modul ist daneben auch in den Studienfächern "Informatik (Master of Science)", "Informations- und Kommunikationstechnik (Master of Science)", "Mathematik (Bachelor of Science)" verwendbar.

#### Studien-/Prüfungsleistungen:

Betriebssysteme (Vorlesung mit erweiterten Übungen) (Prüfungsnummer: 820947)

Prüfungsleistung, Portfolio

Anteil an der Berechnung der Modulnote: 100%

weitere Erläuterungen:

30-minütige mündliche Prüfung

+ aktive, erfolgreiche Teilnahme an den erweiterten Übungen

Erstablingung: WS 2019/2020, 1. Wdh.: SS 2020

1. Prüfer: Volkmar Sieh (100548)

---

**Modulbezeichnung:** Betriebssysteme-V+Ü (BS-VU) 5 ECTS  
 (Operating Systems L+E)

Modulverantwortliche/r: Volkmar Sieh

Lehrende: Volkmar Sieh, Wolfgang Schröder-Preikschat

---

Startsemester: WS 2019/2020

Dauer: 1 Semester

Turnus: jährlich (WS)

Präsenzzeit: 60 Std.

Eigenstudium: 90 Std.

Sprache: Deutsch

---

**Lehrveranstaltungen:**

Betriebssysteme (WS 2019/2020, Vorlesung, 2 SWS, Volkmar Sieh)

Übungen zu Betriebssysteme (WS 2019/2020, Übung, 2 SWS, Bernhard Heinloth et al.)

Übungen zu Betriebssysteme (Rechnerübungen) (WS 2019/2020, Übung, Bernhard Heinloth et al.)

---

**Empfohlene Voraussetzungen:**

- C/C++ (Übungsaufgaben werden in C++ implementiert)
- Assembler (Grundkenntnisse)

**Vorhergehende Module:**

Systemprogrammierung

---

**Inhalt:**

Inhalt des Moduls ist Vermittlung grundlegender Konzepte, Methoden und Techniken von Betriebssystemen aus dem Blickwinkel einer Betriebssystementwicklerin. Im Rahmen der Übungen entwickeln die Studierenden in einem "bottom-up" Entwurf- und Entwicklungsprozess ihr eigenes (Mehrkern-)betriebssystem für die IA-32 Plattform, ausgehend von der "nackten Hardware" über grundlegende Ein-Ausgabemöglichkeiten, Unterbrechungsbearbeitung bis hin zu quasi- und echtparalleler Programmausführung.

**Lernziele und Kompetenzen:**

Studierende, die das Modul erfolgreich abgeschlossen haben:

- erläutern den Startvorgang eines Rechensystems am Beispiel eines IA32 PCs.
- beschreiben die spezifischen Herausforderungen bei der Softwareentwicklung für "bare metal".
- beschreiben den Ablauf einer Unterbrechungsbehandlung von der Hardware bis zur (System-)software.
- skizzieren Besonderheiten und Strategien der Unterbrechungsbehandlung in Hardware für Mehrkernsystemen am Beispiel des IA32-APICs.
- diskutieren die Aufgabenteilung zwischen Hardware und Systemsoftware bei der Unterbrechungsbearbeitung.
- unterscheiden die verschiedenen Typen von Kontrollflüssen in einem Betriebssystem anhand des Ebenenmodells.
- unterscheiden harte, mehrstufige, und weiche Verfahren zur Unterbrechungssynchronisation in Betriebssystemen und können diese implementieren.
- klassifizieren konkrete Konkurrenzsituationen anhand des Ebenenmodells und leiten daraus geeignete Synchronisationsmaßnahmen ab.
- schildern die IA32-Architektur und gängige PC-Technologie und deren Schnittstellen zur Systemsoftware.
- erläutern grundlegende Bausteine für die Implementierung von Quasi-Parallelität (Fortsetzungen, Koroutinen, Fäden) und grenzen diese gegeneinander ab.
- erläutern die Interaktionen zwischen Hardware, Übersetzer und Systemsoftware, die dabei zu beachten sind.
- entwickeln den Koroutinenwechsel für einen gegebene Architektur.
- erläutern die Implikationen von Quasi-Parallelität auf das Ebenenmodell und die daraus abgeleiteten Synchronisationsmaßnahmen.
- beschreiben die Implementierung von (verdrängendem) Scheduling in einem Betriebssystem.
- analysieren das Zusammenspiel von Scheduling und Unterbrechungssynchronisation.

- nennen Kriterien und Dimensionen des Scheduling von Betriebsmitteln, insbesondere der CPU.
- erläutern die konkrete Umsetzung am Beispiel der Scheduler in Linux und Windows.
- unterscheiden grundlegende Möglichkeiten der Koordinierung und Synchronisation von Fäden (aktives/passives Warten, nichtverdrängbare kritische Abschnitte).
- entwickeln Mechanismen für die Synchronisation auf Fadenebene.
- erklären die dabei zu beachtenden Synchronisationsprobleme (lost update, lost wakeup) und geeignete Gegenmaßnahmen.
- interpretieren die Bedeutung von Gerätetreibern in der Betriebssystempraxis.
- erläutern die Anforderungen an ein Treibermodell.
- vergleichen die Umsetzung von Treibermodellen in Windows und Linux.
- vergleichen grundlegende BS-Architekturen (Bibliothek, Monolith, Mikrokern, Exokern, Hypervisor) anhand fundamentaler Charakteristika (Robustheit, Performanz, Portierbarkeit) und Mechanismen.
- schildern die grundlegenden Paradigmen zur Interprozesskommunikation in Betriebssystemen (speicherbasiert vs. nachrichtenbasiert).
- erläutern die grundlegenden Primitiven dieser Verfahren.
- skizzieren, wie unter Anwendung dieser Primitiven höhere Synchronisationskonstrukte implementiert werden (Monitore, Leser-/Schreiber-Sperre).
- illustrieren die Dualität der Paradigmen.
- erschließen sich typische Probleme (Nebenläufigkeit, Compilerverhalten, Debuggen ohne dedizierte Hilfsmittel) und Fehlerquellen bei der hardwarenahen Softwareentwicklung.
- können in Kleingruppen kooperativ arbeiten.
- können die ihre Entwurfs- und Implementierungsentscheidungen kompakt präsentieren und argumentativ vertreten.
- reflektieren ihre Entscheidungen kritisch und leiten Alternativen ab.
- können offen und konstruktiv mit Schwachpunkten und Irrwegen umgehen.

#### Literatur:

Hinweise zu geeigneter Literatur erfolgen in der Vorlesung.

#### Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

##### [1] Informatik (Bachelor of Science)

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule | Vertiefungsrichtung Verteilte Systeme und Betriebssysteme)

Dieses Modul ist daneben auch in den Studienfächern "Informatik (Master of Science)", "Informations- und Kommunikationstechnik (Master of Science)", "Mathematik (Bachelor of Science)" verwendbar.

#### Studien-/Prüfungsleistungen:

Betriebssysteme (Vorlesung mit Übungen) (Prüfungsnummer: 150033)

Prüfungsleistung, Portfolio

Anteil an der Berechnung der Modulnote: 100%

weitere Erläuterungen:

20 bis 30 minütige mündliche Prüfung

Erstablesung: WS 2019/2020, 1. Wdh.: SS 2020

1. Prüfer: Volkmar Sieh (100548)

**Modulbezeichnung:** Betriebssysteme und Echtzeitsysteme - V+Ü (BS+EZS-VU) 10 ECTS  
(Operating Systems and Real-Time Systems - V+Ü)

Modulverantwortliche/r: Peter Ulbrich

Lehrende: Volkmar Sieh, Peter Ulbrich

Startsemester: WS 2019/2020

Dauer: 1 Semester

Turnus: jährlich (WS)

Präsenzzeit: 120 Std.

Eigenstudium: 180 Std.

Sprache: Deutsch

### Lehrveranstaltungen:

Betriebssysteme (WS 2019/2020, Vorlesung, 2 SWS, Volkmar Sieh)

Übungen zu Betriebssysteme (WS 2019/2020, Übung, 2 SWS, Bernhard Heinloth et al.)

Übungen zu Betriebssysteme (Rechnerübungen) (WS 2019/2020, Übung, Bernhard Heinloth et al.)

Echtzeitsysteme (WS 2019/2020, Vorlesung, 2 SWS, Peter Ulbrich)

Übungen zu Echtzeitsysteme (WS 2019/2020, Übung, 2 SWS, Simon Schuster et al.)

Rechnerübungen zu Echtzeitsysteme (WS 2019/2020, Übung, Simon Schuster)

### Vorhergehende Module:

Systemprogrammierung

### Inhalt:

Inhalt des Moduls ist Vermittlung grundlegender Konzepte, Methoden und Techniken von Betriebssystemen aus dem Blickwinkel einer (Echtzeit-)Betriebssystementwicklerin. Im Rahmen der Übungen entwickeln die Studierenden in einem "bottom-up" Entwurf- und Entwicklungsprozess ihr eigenes (Mehrkern-)betriebssystem für die IA-32 Plattform, ausgehend von der "nackten Hardware" über grundlegende Ein-Ausgabemöglichkeiten, Unterbrechungsbearbeitung bis hin zu quasi- und echtparalleler Programmausführung.

Ferner werden grundlegenden Techniken und Mechanismen vermittelt werden, die man im Betriebssystemumfeld verwendet, um Echtzeitsysteme und Echtzeitbetriebssysteme zu realisieren:

- zeitgesteuerte und ereignisgesteuerte Systeme
- statische und dynamische Ablaufplanungsverfahren
- Fadensynchronisation in Echtzeitbetriebssystemen
- Behandlung von periodischen und nicht-periodischen Ereignissen

In den begleitenden Übungen werden die vorgestellten Techniken bei der Entwicklung eines kleinen Echtzeitsystems praktisch umgesetzt.

### Lernziele und Kompetenzen:

Studierende, die das Modul erfolgreich abgeschlossen haben:

- erläutern den Startvorgang eines Rechensystems am Beispiel eines IA32 PCs.
- beschreiben die spezifischen Herausforderungen bei der Softwareentwicklung für "bare metal".
- beschreiben den Ablauf einer Unterbrechungsbehandlung von der Hardware bis zur (System-)software.
- skizzieren Besonderheiten und Strategien der Unterbrechungsbehandlung in Hardware für Mehrkernsystemen am Beispiel des IA32-APICs.
- diskutieren die Aufgabenteilung zwischen Hardware und Systemsoftware bei der Unterbrechungsbearbeitung.
- unterscheiden die verschiedenen Typen von Kontrollflüssen in einem Betriebssystem anhand des Ebenenmodells.
- unterscheiden harte, mehrstufige, und weiche Verfahren zur Unterbrechungssynchronisation in Betriebssystemen und können diese implementieren.
- klassifizieren konkrete Konkurrenzsituationen anhand des Ebenenmodells und leiten daraus geeignete Synchronisationsmaßnahmen ab.
- schildern die IA32-Architektur und gängige PC-Technologie und deren Schnittstellen zur Systemsoftware.



- erläutern grundlegende Bausteine für die Implementierung von Quasi-Parallelität (Fortsetzungen, Koroutinen, Fäden) und grenzen diese gegeneinander ab.
- erläutern die Interaktionen zwischen Hardware, Übersetzer und Systemsoftware, die dabei zu beachten sind.
- entwickeln den Koroutinenwechsel für einen gegebene Architektur.
- erläutern die Implikationen von Quasi-Parallität auf das Ebenenmodell und die daraus abgeleiteten Synchronisationsmaßnahmen.
- beschreiben die Implementierung von (verdrängendem) Scheduling in einem Betriebssystem.
- analysieren das Zusammenspiel von Scheduling und Unterbrechungssynchronisation.
- nennen Kriterien und Dimensionen des Scheduling von Betriebsmitteln, insbesondere der CPU.
- erläutern die konkrete Umsetzung am Beispiel der Scheduler in Linux und Windows.
- unterscheiden grundlegende Möglichkeiten der Koordinierung und Synchronisation von Fäden (aktives/passives Warten, nichtverdrängbare kritische Abschnitte).
- entwickeln Mechanismen für die Synchronisation auf Fadenebene.
- erklären die dabei zu beachtenden Synchronisationsprobleme (lost update, lost wakeup) und geeignete Gegenmaßnahmen.
- interpretieren die Bedeutung von Gerätetreibern in der Betriebssystempraxis.
- erläutern die Anforderungen an ein Treibermodell.
- vergleichen die Umsetzung von Treibermodellen in Windows und Linux.
- vergleichen grundlegende BS-Architekturen (Bibliothek, Monolith, Mikrokern, Exokern, Hypervisor) anhand fundamentaler Charakteristika (Robustheit, Performanz, Portierbarkeit) und Mechanismen.
- schildern die grundlegenden Paradigmen zur Interprozesskommunikation in Betriebssystemen (speicherbasiert vs. nachrichtenbasiert).
- erläutern die grundlegenden Primitiven dieser Verfahren.
- skizzieren, wie unter Anwendung dieser Primitiven höhere Synchronisationskonstrukte implementiert werden (Monitore, Leser-/Schreiber-Sperre).
- illustrieren die Dualität der Paradigmen.
- erschließen sich typische Probleme (Nebenläufigkeit, Compilerverhalten, Debuggen ohne dedizierte Hilfsmittel) und Fehlerquellen bei der hardwarenahen Softwareentwicklung.
- können in Kleingruppen kooperativ arbeiten.
- können die ihre Entwurfs- und Implementierungsentscheidungen kompakt präsentieren und argumentativ vertreten.
- reflektieren ihre Entscheidungen kritisch und leiten Alternativen ab.
- können offen und konstruktiv mit Schwachpunkten und Irrwegen umgehen.
- unterscheiden die verschiedenen Komponenten eines Echtzeitsystems.
- bewerten die Verbindlichkeiten von Terminvorgaben (weich, fest, hart).
- erläutern die Zusammensetzung des Laufzeitverhaltes einer Echtzeitanwendung.
- klassifizieren die Berührungspunkte zwischen physikalischem Objekt und kontrollierendem Echtzeitsystem.
- interpretieren die Zeitparameter des durch das Echtzeitrechensystem zu kontrollierenden Objekts.
- nennen die Zeitparameter des zugrundeliegenden Rechensystems (Unterbrechungslatenz, Ausführungszeit, ...).
- unterscheiden synchrone und asynchrone Programmunterbrechung (insbesondere Trap/Interrupt, Ausnahmebehandlung und Zustandssicherung).
- skizzieren die Verwaltungsgemeinkosten des schlimmsten Falls.
- entwickeln in der Programmiersprache C und wenden die GNU Werkzeugkette für den Infineon TriCore Microcontroller sowie den Lauterbach Debugger an.
- erstellen Echtzeitanwendungen auf Basis der eCos OS-Schnittstelle
- ordnen die Strukturelemente von Echtzeitanwendungen zu: Aufgabe, Arbeitsauftrag und Faden.
- erläutern die Implikationen von zeitlichem Mehrfachbetrieb auf die Verwaltungsgemeinkosten.
- unterscheiden die Umsetzungsalternativen zur Ablaufsteuerung und die Trennung der Belange in Einplanung (Strategie) und Einlastung (Mechanismus).

- benennen die grundsätzliche Verfahren der Ablaufsteuerung (taktgesteuert, reihum, vorranggesteuert).
- erklären die grundlegenden Zeitparameter einer Aufgabe (Auslösezeitpunkt, Termin, Antwortzeit, Latenz, Ausführungszeit, Schlupfzeit).
- unterscheiden die Grundlagen der Planbarkeit (gültig vs. zulässig, Optimalität von Einplanungsalgorithmen).
- beschreiben den Unterschied zwischen konstruktiver und analytischer Einhaltung von Terminen.
- vergleiche die Möglichkeiten (statisch, dynamisch) der zeitliche Analyse von Echtzeitanwendungen.
- erklären die Grundlagen und Beschränkungen von dynamischer (worst-case?) und statischer WCET-Analyse (makroskopisch und mikroskopisch).
- illustrieren Lösungsverfahren zur Bestimmung des längsten Ausführungspfads (Timing Schema, IPET).
- erstellen Zeitmessung mittels Zeitgeber / Oszilloskop und bestimmen den längsten Pfad durch Code-Review.
- erproben werkzeuggestützte WCET-Analyse mittels des absint aiT Analysewerkzeugs.
- beschreiben die Grundlagen der Abfertigung periodischer Echtzeitsysteme (Periode, Phase, Hyperperiode).
- skizzieren das periodische Modell und dessen Folgen (Entwicklungskomfort vs. Analysierbarkeit).
- erklären die ereignisgesteuerte Ausführung (feste und dynamische Priorität, Verdrängbarkeit) mittels ereignisorientierter Planer (Berechnungskomplexität, MLQ-Scheduler, O(1)-Scheduler).
- unterscheiden die zeitgesteuerte Ausführung (Busy Loop, Ablaufplan) und die Abfertigung von Arbeitsaufträgen im Abfrage- bzw. Unterbrecherbetrieb.
- wenden die Grundlagen der ereignisgesteuerten Ablaufplanung periodischer Echtzeitsysteme an.
- unterscheiden Verfahren zur statischen (RM, DM) und dynamischen Prioritätsvergabe (EDF, LRT, LST).
- nennen den Unterschied zwischen Anwendungs- und Systemebene (Mehrdeutigkeit von Prioritäten).
- erläutern den Optimalitätsnachweis des RM-, DM- und EDF-Algorithmus und dessen Ausnahmen.
- beschreiben grundlegende Verfahren zur Planbarkeitsanalyse (CPU-Auslastung, Antwortzeitanalyse).
- implementieren komplexe Aufgabensysteme in eCos.
- unterscheiden die Grundlagen der zeitgesteuerten Ablaufplanung periodischer Echtzeitsysteme.
- erstellen regelmäßige, zyklische Ablaufpläne (cyclic executive model, Rahmen).
- vergleiche Methoden der manuellen und algorithmischen Ablaufplanung.
- unterscheiden optimale von heuristischen Verfahren (List Scheduling, Branch & Bound).
- diskutieren die Konsequenzen eines Betriebswechsels in Echtzeitsystemen.
- erstellen takt- beziehungsweise ereignisgesteuerte Abläufe in eCos beziehungsweise tt-eCos.
- klassifizieren die Grundlagen der Abfertigung nicht-periodischer Echtzeitsysteme (minimale Zwischenankunftszeit).
- definieren die Verbindlichkeiten von nicht-periodischen Aufgaben (aperiodisch, sporadisch)
- zeigen die sich ergebenden Restriktionen des periodischen Modells (Mischbetrieb, Prioritätswarteschlangen, Übernahmepfprüfung) auf.
- beschreiben die Basistechniken des Laufzeitsystems (Zusteller, Unterbrecherbetrieb, Hintergrundbetrieb).
- quantifizieren die Eigenschaften und Auswirkungen auf den periodischen Teil des Echtzeitsystems.
- formulieren die Grundlagen des Slack-Stealing.
- beschreiben den Einsatz von bandweite-bewahrenden Zustellern.
- unterscheiden aufschiebbarer Zusteller und Sporadic Server (SpSL und POSIX).
- wenden eine Übernahmepfprüfung bei sporadischen Aufgaben mittels dichte- oder schlupfbasierten Akzeptanztests an.
- arbeiten einen strukturierter Ablaufplan (Rahmen) aus und untersuchen den Einsatz von Slack-Stealing.
- ermitteln gerichtete Abhängigkeiten und Rangfolgen in Echtzeitanwendungen (Abhängigkeits- und Aufgabengraph).

- stellen Umsetzungsalternativen für Abhängigkeiten einander gegenüber (naiv, implizit, explizit).
- beschreiben das Konzept der zeitlichen Domänen und physikalischer bzw. logischer Ereignisse.
- übertragen Abhängigkeiten auf das Problem der Ablaufplanung (modifizierte Auslösezeitpunkt/Termin, Phasenversatz).
- konzipieren Rangfolge und aperiodische Steuerung in eCos.
- implementieren einen aperiodischer Moduswechsel mit Zustandsüberführung in eCos.
- wenden die Grundlagen von Wettstreit um Betriebsmitteln, Konkurrenz und Konfliktsituationen (kritische Abschnitte, (un)kontrollierte Prioritätsumkehr) an.
- beschreiben echtzeitfähige Synchronisationsprotokolle (NPCS, PI, PCP).
- nennen die Vor- und Nachteile der Techniken (transitive Blockung, Verklemmungen).
- hinterfragen die Vereinfachung des PCP durch stapelbezogene Grenzprioritäten.
- bestimmen die Ablaufplanung unter Berücksichtigung von Blockierungszeiten und Selbstsuspendierung.
- implementieren Zugriffskontrolle (NPCS, PI, PCP) in Echtzeitanwendungen mit eCos.
- erläutern die Anforderungen an verteilte Echtzeitsysteme (Komposition, Erweiterbarkeit, Komplexität, Ereignis- vs. Zustandsnachricht).
- fassen die Grundlagen von Knoten, Netzwerkschnittstellen und Netzübergängen sowie die Konzepte der expliziten und impliziten Flusskontrolle zusammen.
- erschließen sich typische Probleme (zeitliche Analyse, Beobachtbarkeit, Synchronisation, Rangfolge) und Fehlerquellen bei der Programmierung von Echtzeitanwendungen.

---

#### **Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

##### **[1] Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule | Vertiefungsrichtung Verteilte Systeme und Betriebssysteme)

Dieses Modul ist daneben auch in den Studienfächern "Informatik (Master of Science)", "Mathematik (Bachelor of Science)" verwendbar.

---

#### **Studien-/Prüfungsleistungen:**

Betriebssysteme und Echtzeitsysteme (Vorlesungen mit Übungen) (Prüfungsnummer: 341400)

Prüfungsleistung, Portfolio

Anteil an der Berechnung der Modulnote: 100%

weitere Erläuterungen:

30-minütige mündliche Prüfung

+ erfolgreiche Bearbeitung aller Übungsaufgaben

Erstablingung: WS 2019/2020, 1. Wdh.: SS 2020

1. Prüfer: Peter Ulbrich (100360)

---

---

**Modulbezeichnung: Echtzeitsysteme-V+Ü (EZS-VU)**  
 (Real-Time Systems L+E)

**5 ECTS**

 Modulverantwortliche/r: Peter Ulbrich  
 Lehrende: Peter Ulbrich

Startsemester: WS 2019/2020

Dauer: 1 Semester

Turnus: jährlich (WS)

Präsenzzeit: 60 Std.

Eigenstudium: 90 Std.

Sprache: Deutsch

---

**Lehrveranstaltungen:**

Echtzeitsysteme (WS 2019/2020, Vorlesung, 2 SWS, Peter Ulbrich)

Übungen zu Echtzeitsysteme (WS 2019/2020, Übung, 2 SWS, Simon Schuster et al.)

Rechnerübungen zu Echtzeitsysteme (WS 2019/2020, Übung, Simon Schuster)

---

**Vorhergehende Module:**

Systemprogrammierung

Systemnahe Programmierung in C

---

**Inhalt:**

Videobearbeitung in Echtzeit, Echtzeitstrategiespiel, echtzeitfähig - der Begriff Echtzeit ist wohl einer der am meisten strapazierten Begriffe der Informatik und wird in den verschiedensten Zusammenhängen benutzt. Diese Vorlesung beschäftigt sich mit dem Begriff Echtzeit aus der Sicht von Betriebssystemen - was versteht man eigentlich unter dem Begriff Echtzeit im Betriebssystemumfeld, wo und warum setzt man sog. Echtzeitbetriebssysteme ein und was zeichnet solche Echtzeitbetriebssysteme aus?

In dieser Vorlesung geht es darum, die oben genannten Fragen zu beantworten, indem die grundlegenden Techniken und Mechanismen vermittelt werden, die man im Betriebssystemumfeld verwendet, um Echtzeitsysteme und Echtzeitbetriebssysteme zu realisieren. Im Rahmen dieser Vorlesung werden unter anderem folgende Themen behandelt:

- zeitgesteuerte und ereignisgesteuerte Systeme
- statische und dynamische Ablaufplanungsverfahren
- Fadensynchronisation in Echtzeitbetriebssystemen
- Behandlung von periodischen und nicht-periodischen Ereignissen

In den begleitenden Übungen werden die in der Vorlesung vorgestellten Techniken bei der Entwicklung eines kleinen Echtzeitsystems praktisch umgesetzt.

**Lernziele und Kompetenzen:**

Studierende, die das Modul erfolgreich abgeschlossen haben:

- unterscheiden die verschiedenen Komponenten eines Echtzeitsystems.
- bewerten die Verbindlichkeiten von Terminvorgaben (weich, fest, hart).
- erläutern die Zusammensetzung des Laufzeitverhaltes einer Echtzeitanwendung.
- klassifizieren die Berührungspunkte zwischen physikalischem Objekt und kontrollierendem Echtzeitsystem.
- interpretieren die Zeitparameter des durch das Echtzeitrechensystem zu kontrollierenden Objekts.
- nennen die Zeitparameter des zugrundeliegenden Rechensystems (Unterbrechungslatenz, Ausführungszeit, ...).
- unterscheiden synchrone und asynchrone Programmunterbrechung (insbesondere Trap/Interrupt, Ausnahmebehandlung und Zustandssicherung).
- skizzieren die Verwaltungsgemeinkosten des schlimmsten Falls.
- entwickeln in der Programmiersprache C und wenden die GNU Werkzeugkette für den ARM Cortex M4 Microcontroller an.
- erstellen Echtzeitanwendungen auf Basis der eCos OS-Schnittstelle
- ordnen die Strukturelemente von Echtzeitanwendungen zu: Aufgabe, Arbeitsauftrag und Faden.
- erläutern die Implikationen von zeitlichem Mehrfachbetrieb auf die Verwaltungsgemeinkosten.
- unterscheiden die Umsetzungsalternativen zur Ablaufsteuerung und die Trennung der Belange in Einplanung (Strategie) und Einlastung (Mechanismus).

- benennen die grundsätzliche Verfahren der Ablaufsteuerung (taktgesteuert, reihum, vorranggesteuert).
- erklären die grundlegenden Zeitparameter einer Aufgabe (Auslösezeitpunkt, Termin, Antwortzeit, Latenz, Ausführungszeit, Schlupfzeit).
- unterscheiden die Grundlagen der Planbarkeit (gültig vs. zulässig, Optimalität von Einplanungsalgorithmen).
- beschreiben den Unterschied zwischen konstruktiver und analytischer Einhaltung von Terminen.
- vergleiche die Möglichkeiten (statisch, dynamisch) der zeitliche Analyse von Echtzeitanwendungen.
- erklären die Grundlagen und Beschränkungen von dynamischer (worst-case?) und statischer WCET-Analyse (makroskopisch und mikroskopisch).
- illustrieren Lösungsverfahren zur Bestimmung des längsten Ausführungspfads (Timing Schema, IPET).
- erstellen Zeitmessung mittels Zeitgeber / Oszilloskop und bestimmen den längsten Pfad durch Code-Review.
- erproben werkzeuggestützte WCET-Analyse mittels des absint aiT Analysewerkzeugs.
- beschreiben die Grundlagen der Abfertigung periodischer Echtzeitsysteme (Periode, Phase, Hyperperiode).
- skizzieren das periodische Modell und dessen Folgen (Entwicklungskomfort vs. Analysierbarkeit).
- erklären die ereignisgesteuerte Ausführung (feste und dynamische Priorität, Verdrängbarkeit) mittels ereignisorientierter Planer (Berechnungskomplexität, MLQ-Scheduler, O(1)-Scheduler).
- unterscheiden die zeitgesteuerte Ausführung (Busy Loop, Ablaufplan) und die Abfertigung von Arbeitsaufträgen im Abfrage- bzw. Unterbrecherbetrieb.
- wenden die Grundlagen der ereignisgesteuerten Ablaufplanung periodischer Echtzeitsysteme an.
- unterscheiden Verfahren zur statischen (RM, DM) und dynamischen Prioritätsvergabe (EDF, LRT, LST).
- nennen den Unterschied zwischen Anwendungs- und Systemebene (Mehrdeutigkeit von Prioritäten).
- erläutern den Optimalitätsnachweis des RM-, DM- und EDF-Algorithmus und dessen Ausnahmen.
- beschreiben grundlegende Verfahren zur Planbarkeitsanalyse (CPU-Auslastung, Antwortzeitanalyse).
- implementieren komplexe Aufgabensysteme in eCos.
- unterscheiden die Grundlagen der zeitgesteuerten Ablaufplanung periodischer Echtzeitsysteme.
- erstellen regelmäßige, zyklische Ablaufpläne (cyclic executive model, Rahmen).
- vergleiche Methoden der manuellen und algorithmischen Ablaufplanung.
- unterscheiden optimale von heuristischen Verfahren (List Scheduling, Branch & Bound).
- diskutieren die Konsequenzen eines Betriebswechsels in Echtzeitsystemen.
- erstellen takt- beziehungsweise ereignisgesteuerte Abläufe in eCos beziehungsweise tt-eCos.
- klassifizieren die Grundlagen der Abfertigung nicht-periodischer Echtzeitsysteme (minimale Zwischenankunftszeit).
- definieren die Verbindlichkeiten von nicht-periodischen Aufgaben (aperiodisch, sporadisch)
- zeigen die sich ergebenden Restriktionen des periodischen Modells (Mischbetrieb, Prioritätswarteschlangen, Übernahmepfprüfung) auf.
- beschreiben die Basistechniken des Laufzeitsystems (Zusteller, Unterbrecherbetrieb, Hintergrundbetrieb).
- quantifizieren die Eigenschaften und Auswirkungen auf den periodischen Teil des Echtzeitsystems.
- formulieren die Grundlagen des Slack-Stealing.
- beschreiben den Einsatz von bandweite-bewahrenden Zustellern.
- unterscheiden aufschiebbarer Zusteller und Sporadic Server (SpSL und POSIX).
- wenden eine Übernahmepfprüfung bei sporadischen Aufgaben mittels dichte- oder schlupfbasierten Akzeptanztests an.
- arbeiten einen strukturierter Ablaufplan (Rahmen) aus und untersuchen den Einsatz von Slack-Stealing.
- ermitteln gerichtete Abhängigkeiten und Rangfolgen in Echtzeitanwendungen (Abhängigkeits- und Aufgabengraph).

- stellen Umsetzungsalternativen für Abhängigkeiten einander gegenüber (naiv, implizit, explizit).
- beschreiben das Konzept der zeitlichen Domänen und physikalischer bzw. logischer Ereignisse.
- übertragen Abhängigkeiten auf das Problem der Ablaufplanung (modifizierte Auslösezeitpunkt/Termin, Phasenversatz).
- konzipieren Rangfolge und aperiodische Steuerung in eCos.
- implementieren einen aperiodischer Moduswechsel mit Zustandsüberführung in eCos.
- wenden die Grundlagen von Wettstreit um Betriebsmitteln, Konkurrenz und Konfliktsituationen (kritische Abschnitte, (un)kontrollierte Prioritätsumkehr) an.
- beschreiben echtzeitfähige Synchronisationsprotokolle (NPCS, PI, PCP).
- nennen die Vor- und Nachteile der Techniken (transitive Blockung, Verklemmungen).
- hinterfragen die Vereinfachung des PCP durch stapelbezogene Grenzprioritäten.
- bestimmen die Ablaufplanung unter Berücksichtigung von Blockierungszeiten und Selbstsuspendierung.
- implementieren Zugriffskontrolle (NPCS, PI, PCP) in Echtzeitanwendungen mit eCos.
- erläutern die Anforderungen an verteilte Echtzeitsysteme (Komposition, Erweiterbarkeit, Komplexität, Ereignis- vs. Zustandsnachricht).
- fassen die Grundlagen von Knoten, Netzwerkschnittstellen und Netzübergängen sowie die Konzepte der expliziten und impliziten Flusskontrolle zusammen.
- erschließen sich typische Probleme (zeitliche Analyse, Beobachtbarkeit, Synchronisation, Rangfolge) und Fehlerquellen bei der Programmierung von Echtzeitanwendungen.
- können in Gruppen kooperativ und effektiv arbeiten.
- können ihre Entwurfs- und Implementierungsentscheidungen kompakt präsentieren und argumentativ vertreten.
- reflektieren ihre Entscheidungen kritisch und leiten Alternativen ab.
- können offen und konstruktiv mit Schwachpunkten in der Konzeption wie Umsetzung umgehen.

#### Literatur:

- Hermann Kopetz. Real-Time Systems: Design Principles for Distributed Embedded Applications. Kluwer Academic Publishers, 1997.
- Jane W. S. Liu. Real-Time Systems. Prentice-Hall, Inc., 2000.
- Wolfgang Schröder-Preikschat. Softwaresysteme 1. Vorlesungsfolien. 2006.

---

#### Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

##### [1] Informatik (Bachelor of Science)

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule | Vertiefungsrichtung Verteilte Systeme und Betriebssysteme)

Dieses Modul ist daneben auch in den Studienfächern "Computational Engineering (Master of Science with Honours)", "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Master of Science)", "Informatik (Master of Science)", "Information and Communication Technology (Master of Science)", "Informations- und Kommunikationstechnik (Bachelor of Science)", "Informations- und Kommunikationstechnik (Master of Science)", "Maschinenbau (Bachelor of Science)", "Maschinenbau (Master of Science)", "Mathematik (Bachelor of Science)", "Mechatronik (Bachelor of Science)", "Mechatronik (Master of Science)", "Medizintechnik (Bachelor of Science)", "Medizintechnik (Master of Science)", "Wirtschaftsingenieurwesen (Bachelor of Science)", "Wirtschaftsingenieurwesen (Master of Science)" verwendbar.

---

#### Studien-/Prüfungsleistungen:

Echtzeitsysteme (Vorlesung mit Übungen) (Prüfungsnummer: 707303)

(englische Bezeichnung: Real Time Systems (Lecture with Exercises))

Prüfungsleistung, Portfolio

Anteil an der Berechnung der Modulnote: 100%

weitere Erläuterungen:

30-minütige mündliche Prüfung über den Stoff der Vorlesung und die Übungen.

Teilnahme an den Übungen und die erfolgreiche Bearbeitung aller Übungsaufgaben wird hierzu dringend empfohlen!

Erstablingung: WS 2019/2020, 1. Wdh.: SS 2020

1. Prüfer: Peter Ulbrich (100360)

---

---

**Modulbezeichnung:** Echtzeitsysteme-V+EÜ (EZS-VEU) 7.5 ECTS  
 (Real-Time Systems L+EE)

Modulverantwortliche/r: Peter Ulbrich  
 Lehrende: Peter Ulbrich

---

Startsemester: WS 2019/2020	Dauer: 1 Semester	Turnus: jährlich (WS)
Präsenzzeit: 90 Std.	Eigenstudium: 135 Std.	Sprache: Deutsch

---

**Lehrveranstaltungen:**

Echtzeitsysteme (WS 2019/2020, Vorlesung, 2 SWS, Peter Ulbrich)  
 Erweiterte Übungen zu Echtzeitsysteme (WS 2019/2020, Übung, 4 SWS, Simon Schuster et al.)  
 Rechnerübungen zu Echtzeitsysteme (WS 2019/2020, Übung, Simon Schuster)

---

**Inhalt:**

Videobearbeitung in Echtzeit, Echtzeitstrategiespiel, echtzeitfähig - der Begriff Echtzeit ist wohl einer der am meisten strapazierten Begriffe der Informatik und wird in den verschiedensten Zusammenhängen benutzt. Diese Vorlesung beschäftigt sich mit dem Begriff Echtzeit aus der Sicht von Betriebssystemen - was versteht man eigentlich unter dem Begriff Echtzeit im Betriebssystemumfeld, wo und warum setzt man sog. Echtzeitbetriebssysteme ein und was zeichnet solche Echtzeitbetriebssysteme aus? In dieser Vorlesung geht es darum, die oben genannten Fragen zu beantworten, indem die grundlegenden Techniken und Mechanismen vermittelt werden, die man im Betriebssystemumfeld verwendet, um Echtzeitsysteme und Echtzeitbetriebssysteme zu realisieren. Im Rahmen dieser Vorlesung werden unter anderem folgende Themen behandelt:

- zeitgesteuerte und ereignisgesteuerte Systeme
- statische und dynamische Ablaufplanungsverfahren
- Fadensynchronisation in Echtzeitbetriebssystemen
- Behandlung von periodischen und nicht-periodischen Ereignissen

In den begleitenden Übungen werden die in der Vorlesung vorgestellten Techniken bei der Entwicklung eines kleinen Echtzeitsystems praktisch umgesetzt.

**Lernziele und Kompetenzen:**

Studierende, die das Modul erfolgreich abgeschlossen haben:

- unterscheiden die verschiedenen Komponenten eines Echtzeitsystems.
- bewerten die Verbindlichkeiten von Terminvorgaben (weich, fest, hart).
- erläutern die Zusammensetzung des Laufzeitverhaltes einer Echtzeitanwendung.
- klassifizieren die Berührungspunkte zwischen physikalischem Objekt und kontrollierendem Echtzeitsystem.
- interpretieren die Zeitparameter des durch das Echtzeitrechensystem zu kontrollierenden Objekts.
- nennen die Zeitparameter des zugrundeliegenden Rechensystems (Unterbrechungslatenz, Ausführungszeit, ...).
- unterscheiden synchrone und asynchrone Programmunterbrechung (insbesondere Trap/Interrupt, Ausnahmebehandlung und Zustandssicherung).
- skizzieren die Verwaltungsgemeinkosten des schlimmsten Falls.
- entwickeln in der Programmiersprache C und wenden die GNU Werkzeugkette für den ARM Cortex M4 an.
- erstellen Echtzeitanwendungen auf Basis der eCos OS-Schnittstelle
- ordnen die Strukturelemente von Echtzeitanwendungen zu: Aufgabe, Arbeitsauftrag und Faden.
- erläutern die Implikationen von zeitlichem Mehrfachbetrieb auf die Verwaltungsgemeinkosten.
- unterscheiden die Umsetzungsalternativen zur Ablaufsteuerung und die Trennung der Belange in Einplanung (Strategie) und Einlastung (Mechanismus).
- benennen die grundsätzliche Verfahren der Ablaufsteuerung (taktgesteuert, reihum, vorranggesteuert).
- erklären die grundlegenden Zeitparameter einer Aufgabe (Auslösezeitpunkt, Termin, Antwortzeit, Latenz, Ausführungszeit, Schlupfzeit).



- unterscheiden die Grundlagen der Planbarkeit (gültig vs. zulässig, Optimalität von Einplanungsalgorithmen).
- beschreiben den Unterschied zwischen konstruktiver und analytischer Einhaltung von Terminen.
- vergleiche die Möglichkeiten (statisch, dynamisch) der zeitliche Analyse von Echtzeitanwendungen.
- erkläre die Grundlagen und Beschränkungen von dynamischer (worst-case?) und statischer WCET-Analyse (makroskopisch und mikroskopisch).
- illustrieren Lösungsverfahren zur Bestimmung des längsten Ausführungspfads (Timing Schema, IPET).
- erstellen Zeitmessung mittels Zeitgeber / Oszilloskop und bestimmen den längsten Pfad durch Code-Review.
- erproben und konzipieren werkzeuggestützte WCET-Analyse mittels des absint aiT Analysewerkzeugs.
- Entwickeln und annotieren Flußrestriktionen für die statische WCET-Analyse.
- beschreiben die Grundlagen der Abfertigung periodischer Echtzeitsysteme (Periode, Phase, Hyperperiode).
- skizzieren das periodische Modell und dessen Folgen (Entwicklungskomfort vs. Analysierbarkeit).
- erkläre die ereignisgesteuerte Ausführung (feste und dynamische Priorität, Verdrängbarkeit) mittels ereignisorientierter Planer (Berechnungskomplexität, MLQ-Scheduler, O(1)-Scheduler).
- unterscheiden die zeitgesteuerte Ausführung (Busy Loop, Ablaufplan) und die Abfertigung von Arbeitsaufträgen im Abfrage- bzw. Unterbrecherbetrieb.
- wenden die Grundlagen der ereignisgesteuerten Ablaufplanung periodischer Echtzeitsysteme an.
- unterscheiden Verfahren zur statischen (RM, DM) und dynamischen Prioritätsvergabe (EDF, LRT, LST).
- nennen den Unterschied zwischen Anwendungs- und Systemebene (Mehrdeutigkeit von Prioritäten).
- erläutere den Optimalitätsnachweis des RM-, DM- und EDF-Algorithmus und dessen Ausnahmen.
- beschreiben grundlegende Verfahren zur Planbarkeitsanalyse (CPU-Auslastung, Antwortzeitanalyse).
- implementieren komplexe Aufgabensysteme in eCos.
- unterscheiden die Grundlagen der zeitgesteuerten Ablaufplanung periodischer Echtzeitsysteme.
- erstellen regelmäßige, zyklische Ablaufpläne (cyclic executive model, Rahmen).
- vergleiche Methoden der manuellen und algorithmischen Ablaufplanung.
- unterscheiden optimale von heuristischen Verfahren (List Scheduling, Branch & Bound).
- diskutieren die Konsequenzen eines Betriebswechsels in Echtzeitsystemen.
- erstellen takt- beziehungsweise ereignisgesteuerte Abläufe in eCos beziehungsweise tt-eCos.
- entwickeln ein softwarebasiertes Oszilloskop und erstellen dessen zeitliche Analyse und Ablaufplanung.
- klassifizieren die Grundlagen der Abfertigung nicht-periodischer Echtzeitsysteme (minimale Zwischenankunftszeit).
- definieren die Verbindlichkeiten von nicht-periodischen Aufgaben (aperiodisch, sporadisch)
- zeigen die sich ergebenden Restriktionen des periodischen Modells (Mischbetrieb, Prioritätswarteschlangen, Übernahmeprüfung) auf.
- beschreiben die Basistechniken des Laufzeitsystems (Zusteller, Unterbrecherbetrieb, Hintergrundbetrieb).
- quantifizieren die Eigenschaften und Auswirkungen auf den periodischen Teil des Echtzeitsystems.
- formulieren die Grundlagen des Slack-Stealing.
- beschreiben den Einsatz von bandweite-bewahrenden Zustellern.
- unterscheiden aufschiebbarer Zusteller und Sporadic Server (SpSL und POSIX).
- wenden eine Übernahmeprüfung bei sporadischen Aufgaben mittels dichte- oder schlupfbasierten Akzeptanztests an.
- arbeiten einen strukturierter Ablaufplan (Rahmen) aus und untersuchen den Einsatz von Slack-Stealing.
- ermitteln gerichtete Abhängigkeiten und Rangfolgen in Echtzeitanwendungen (Abhängigkeits- und Aufgabengraph).

- stellen Umsetzungsalternativen für Abhängigkeiten einander gegenüber (naiv, implizit, explizit).
- beschreiben das Konzept der zeitlichen Domänen und physikalischer bzw. logischer Ereignisse.
- übertragen Abhängigkeiten auf das Problem der Ablaufplanung (modifizierte Auslösezeitpunkt/Termin, Phasenversatz).
- konzipieren Rangfolge und aperiodische Steuerung in eCos.
- implementieren einen aperiodischer Moduswechsel mit Zustandsüberführung in eCos.
- gestalten einen Signal-Trigger für das entwickelte softwarebasierten Oszilloskops.
- konzipieren explizite Synchronisation mittels Nachrichten in eCos.
- wenden die Grundlagen von Wettstreit um Betriebsmitteln, Konkurrenz und Konfliktsituationen (kritische Abschnitte, (un)kontrollierte Prioritätsumkehr) an.
- beschreiben echtzeitfähige Synchronisationsprotokolle (NPCS, PI, PCP).
- nennen die Vor- und Nachteile der Techniken (transitive Blockung, Verklemmungen).
- hinterfragen die Vereinfachung des PCP durch stapelbezogene Grenzprioritäten.
- bestimmen die Ablaufplanung unter Berücksichtigung von Blockierungszeiten und Selbstsuspendierung.
- implementieren Zugriffskontrolle (NPCS, PI, PCP) in Echtzeitanwendungen mit eCos.
- analysieren Blockade für die Zugriffskontrolle in eCos.
- erläutern die Anforderungen an verteilte Echtzeitsysteme (Komposition, Erweiterbarkeit, Komplexität, Ereignis- vs. Zustandsnachricht).
- fassen die Grundlagen von Knoten, Netzwerkschnittstellen und Netzübergängen sowie die Konzepte der expliziten und impliziten Flusskontrolle zusammen.
- erschließen sich typische Probleme (zeitliche Analyse, Beobachtbarkeit, Synchronisation, Rangfolge) und Fehlerquellen bei der Programmierung von Echtzeitanwendungen.
- können in Gruppen kooperativ und effektiv arbeiten.
- können ihre Entwurfs- und Implementierungsentscheidungen kompakt präsentieren und argumentativ vertreten.
- reflektieren ihre Entscheidungen kritisch und leiten Alternativen ab.
- können offen und konstruktiv mit Schwachpunkten in der Konzeption wie Umsetzung umgehen.

#### Literatur:

- Hermann Kopetz. Real-Time Systems: Design Principles for Distributed Embedded Applications. Kluwer Academic Publishers, 1997.
- Jane W. S. Liu. Real-Time Systems. Prentice-Hall, Inc., 2000.
- Wolfgang Schröder-Preikschat. System-programmierung. Vorlesungsfolien. 2006.

---

#### Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

##### [1] Informatik (Bachelor of Science)

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule | Vertiefungsrichtung Verteilte Systeme und Betriebssysteme)

Dieses Modul ist daneben auch in den Studienfächern "Informatik (Master of Science)", "Information and Communication Technology (Master of Science)", "Informations- und Kommunikationstechnik (Master of Science)", "Mathematik (Bachelor of Science)", "Mechatronik (Bachelor of Science)", "Mechatronik (Master of Science)" verwendbar.

---

#### Studien-/Prüfungsleistungen:

Echtzeitsysteme mit erweiterten Übungen (Prüfungsnummer: 179490)

(englische Bezeichnung: Real-Time Systems with Extended Exercises)

Prüfungsleistung, Portfolio

Anteil an der Berechnung der Modulnote: 100%

weitere Erläuterungen:

30-minütige mündliche Prüfung über den Stoff der Vorlesung und die erweiterten Übungen.

Teilnahme an den erweiterten Übungen und die erfolgreiche Bearbeitung aller Übungsaufgaben wird hierzu dringend empfohlen!

Erstablingung: WS 2019/2020, 1. Wdh.: SS 2020

1. Prüfer: Peter Ulbrich (100360)

---

---

**Modulbezeichnung:** **Middleware - Cloud Computing V+EÜ (MW-VEU)** **7.5 ECTS**  
 (Middleware - Cloud Computing L+EE)

Modulverantwortliche/r: Tobias Distler  
 Lehrende: Tobias Distler

---

Startsemester: WS 2019/2020	Dauer: 1 Semester	Turnus: jährlich (WS)
Präsenzzeit: 90 Std.	Eigenstudium: 135 Std.	Sprache: Deutsch

---

**Lehrveranstaltungen:**

Middleware - Cloud Computing (WS 2019/2020, Vorlesung mit Übung, 4 SWS, Tobias Distler)  
 Übungen zu Middleware - Cloud Computing (WS 2019/2020, Übung, Michael Eischer et al.)  
 Übungen zu Middleware - Cloud Computing (Rechnerübungen) (WS 2019/2020, Übung, Michael Eischer et al.)

---

**Inhalt:**

- Überblick Cloud Computing
- Grundlagen verteilter Programmierung (Web Services/SOAP/REST)
- Virtualisierung als Basis für Cloud Computing
- Infrastructure as a Service (IaaS) am Beispiel von Eucalyptus und Amazon EC2
- Verteilte Programmierung für datenlastige Cloud-Anwendungen
- Skalierbare Verarbeitung von großen Datenmengen
- Interoperabilität und Multi-Cloud Computing
- Fehlertoleranz und Sicherheit im Kontext von Cloud Computing
- Aktuelle Forschungstrends (z.B. einbruchstolerante Systeme)

**Lernziele und Kompetenzen:**

Studierende, die das Modul erfolgreich abgeschlossen haben:

- nennen unterschiedliche Ausprägungen von Cloud-Computing.
- erläutern verschiedene Cloud-Architekturen.
- stellen Vor- und Nachteile von Cloud-Computing gegenüber.
- beschreiben die Bestandteile des SOAP-Nachrichten-Frameworks und ihre jeweiligen Funktionen.
- unterscheiden die Herangehensweisen bei der Entwicklung von SOAP- im Vergleich zu REST-Anwendungen.
- organisieren den Austausch von Informationen in einer verteilten Anwendung unter Verwendung eines Verzeichnisdienstes.
- wenden WSDL zur Beschreibung von Web-Services an.
- erproben die automatische Erzeugung von Proxy-Objekten auf Basis von WSDL-Beschreibungen.
- entwickeln eigene auf Web-Services basierende Anwendungen.
- bewerten die Vor- und Nachteile der Bündelung von Nachrichten beim Aufruf von Web-Services.
- erläutern die Anforderungen an ein virtualisiertes System.
- beschreiben die für die Virtualisierung eines Systems erforderlichen Kriterien.
- vergleichen zwischen unterschiedlichen Virtualisierungstechniken und -ebenen.
- schildern den Aufbau und die Funktionsweise von Xen und Linux-VServer.
- erproben das Einrichten eines Abbilds für eine virtuelle Maschine.
- entwickeln einen Dienst zur hierarchischen Speicherung von Schlüssel-Wert-Paaren.
- gestalten einen Mechanismus zur effizienten persistenten Speicherung von Schlüssel-Wert-Paaren.
- skizzieren die Architektur einer Infrastruktur-Cloud sowie die Aufgabenbereiche hierfür zentraler Komponenten am Beispiel von Eucalyptus.
- erproben das Bereitstellen von Anwendungen in einer Infrastruktur-Cloud.
- zeigen die Grundlagen Software-definierter Netzwerke am Beispiel von Onix und B4 auf.
- bewerten verschiedene im Bereich Cloud-Computing zum Einsatz kommende Datenspeichersysteme (Google File System, Bigtable, Windows Azure Storage, Amazon Dynamo) hinsichtlich der Kriterien Verfügbarkeit, Konsistenz und Partitionstoleranz.

- erläutern eine auf Vektoruhren basierende Methode zur Auflösung im Zusammenhang mit letztendlicher Konsistenz auftretender Konflikte.
- entwickeln ein verteiltes, repliziertes Dateisystem nach dem Vorbild von HDFS, das auf die hierarchische Speicherung großer Datenmengen ausgelegt ist.
- erkunden das Bereitstellen selbst entwickelter Dienste mittels Docker.
- erstellen ein Framework zur parallelen Bearbeitung von Daten nach dem Vorbild von MapReduce.
- konzipieren eigene MapReduce-Anwendungen zur Verarbeitung strukturierter bzw. unstrukturierter Rohdaten.
- diskutieren die Fehlertoleranzmechanismen in Google MapReduce.
- skizzieren die Abbildung von Pig-Latin-Programmen auf MapReduce.
- schildern die grundsätzliche Funktionsweise von Systemen zur Kühlung von Datenzentren mittels Umgebungsluft.
- beschreiben das Grundkonzept einer temperaturabhängigen Lastverteilung von Prozessen in einem Datenzentrum.
- stellen diverse Ansätze zur Erhöhung der Energieeffizienz von MapReduce-Clustern gegenüber.
- unterscheiden die Architekturen und Funktionsweisen der Koordinierungsdienste Chubby und ZooKeeper.
- entwickeln einen eigenen Koordinierungsdienst nach dem Vorbild von ZooKeeper.
- ermitteln die Konsistenzeigenschaften der eigenen Koordinierungsdienstimplementierung.
- erschließen sich die mit der Implementierung passiv replizierter Systeme verbundenen Problematiken.
- erläutern unterschiedliche Ansätze zur Reduzierung bzw. Tolerierung von Tail-Latenz.
- skizzieren das Grundkonzept von Erasure-Codes.
- beschreiben den Aufbau eines auf die Clouds mehrerer Anbieter gestützten Datenspeichersystems.
- entwickeln einen auf Hot-Standy-Replikate basierenden Mechanismus zur Erkennung und Beseitigung von Überlastsituationen für virtuelle Maschinen in einer Infrastruktur-Cloud am Beispiel von Amazon EC2.
- erläutern den Einsatz passiver Replikation zur Bereitstellung von Fehlertoleranzmechanismen für virtuelle Maschinen am Beispiel von Remus.
- schildern die Grundlagen der Migration von virtuellen Maschinen.
- unterscheiden zwischen Ausfalltoleranz und byzantinischer Fehlertoleranz.
- bewerten die Qualität einer aktuellen Publikation aus der Fachliteratur.
- erschließen sich typische Probleme (Nebenläufigkeit, Konsistenz, Skalierbarkeit) und Fehlerquellen bei der Programmierung verteilter Anwendungen.
- können in Kleingruppen kooperativ arbeiten.
- können ihre Entwurfs- und Implementierungsentscheidungen kompakt präsentieren und argumentativ vertreten.
- reflektieren ihre Entscheidungen kritisch und leiten Alternativen ab.
- können offen und konstruktiv mit Schwachpunkten und Irrwegen umgehen.

#### Literatur:

- Mache Creeger. *Cloud computing: An overview*. Queue Distributed Computing, 7(5), 2009.
- Michael Armbrust, Armando Fox, Rean Griffith, Anthony D. Joseph, Randy Katz, Andy Konwinski, Gunho Lee, David Patterson, Ariel Rabkin, Ion Stoica, and Matei Zaharia. *Above the clouds: A Berkeley view of cloud computing*. Technical Report UCB/EECS-2009-28, EECS Department, University of California, Berkeley, 2009.
- Michael Armbrust, Armando Fox, Rean Griffith, Anthony D. Joseph, Randy Katz, Andy Konwinski, Gunho Lee, David Patterson, Ariel Rabkin, Ion Stoica, and Matei Zaharia. A view of cloud computing. *Communications of the ACM*, 53(4):5058, 2010.

weite Literaturangaben auf der Vorlesungs-Webseite

---

#### Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

**[1] Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule | Vertiefungsrichtung Verteilte Systeme und Betriebssysteme)

Dieses Modul ist daneben auch in den Studienfächern "Informatik (Master of Science)", "Informations- und Kommunikationstechnik (Master of Science)", "Mathematik (Bachelor of Science)" verwendbar.

---

**Studien-/Prüfungsleistungen:**

Middleware - Cloud Computing (Vorlesung mit erweiterten Übungen) (Prüfungsnummer: 722831)

Prüfungsleistung, Portfolio

Anteil an der Berechnung der Modulnote: 100%

weitere Erläuterungen:

erfolgreiche Bearbeitung aller während des Semesters gestellten Übungsaufgaben + mündliche Prüfung am Ende des Semesters

Erstablingung: WS 2019/2020, 1. Wdh.: SS 2020

1. Prüfer: Tobias Distler (100377)

---

---

**Modulbezeichnung:** **Middleware - Cloud Computing V+Ü (MW-VU)** **5 ECTS**  
 (Middleware - Cloud Computing L+E)

Modulverantwortliche/r: Tobias Distler  
 Lehrende: Tobias Distler

---

Startsemester: WS 2019/2020	Dauer: 1 Semester	Turnus: jährlich (WS)
Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Deutsch

---

**Lehrveranstaltungen:**

Middleware - Cloud Computing (WS 2019/2020, Vorlesung mit Übung, 4 SWS, Tobias Distler)  
 Übungen zu Middleware - Cloud Computing (WS 2019/2020, Übung, Michael Eischer et al.)  
 Übungen zu Middleware - Cloud Computing (Rechnerübungen) (WS 2019/2020, Übung, Michael Eischer et al.)

---

**Inhalt:**

- Überblick Cloud Computing
- Grundlagen verteilter Programmierung (Web Services/SOAP/REST)
- Virtualisierung als Basis für Cloud Computing
- Infrastructure as a Service (IaaS) am Beispiel von Eucalyptus und Amazon EC2
- Verteilte Programmierung für datenlastige Cloud-Anwendungen
- Skalierbare Verarbeitung von großen Datenmengen
- Interoperabilität und Multi-Cloud Computing
- Fehlertoleranz und Sicherheit im Kontext von Cloud Computing
- Aktuelle Forschungstrends (z.B. einbruchstolerante Systeme)

**Lernziele und Kompetenzen:**

Studierende, die das Modul erfolgreich abgeschlossen haben:

- nennen unterschiedliche Ausprägungen von Cloud-Computing.
- erläutern verschiedene Cloud-Architekturen.
- stellen Vor- und Nachteile von Cloud-Computing gegenüber.
- beschreiben die Bestandteile des SOAP-Nachrichten-Frameworks und ihre jeweiligen Funktionen.
- unterscheiden die Herangehensweisen bei der Entwicklung von SOAP- im Vergleich zu REST-Anwendungen.
- organisieren den Austausch von Informationen in einer verteilten Anwendung unter Verwendung eines Verzeichnisdienstes.
- wenden WSDL zur Beschreibung von Web-Services an.
- erproben die automatische Erzeugung von Proxy-Objekten auf Basis von WSDL-Beschreibungen.
- entwickeln eigene auf Web-Services basierende Anwendungen.
- erläutern die Anforderungen an ein virtualisiertes System.
- beschreiben die für die Virtualisierung eines Systems erforderlichen Kriterien.
- vergleichen zwischen unterschiedlichen Virtualisierungstechniken und -ebenen.
- schildern den Aufbau und die Funktionsweise von Xen und Linux-VServer.
- erproben das Einrichten eines Abbilds für eine virtuelle Maschine.
- entwickeln einen Dienst zur Speicherung von Schlüssel-Wert-Paaren.
- skizzieren die Architektur einer Infrastruktur-Cloud sowie die Aufgabenbereiche hierfür zentraler Komponenten am Beispiel von Eucalyptus.
- erproben das Bereitstellen von Anwendungen in einer Infrastruktur-Cloud.
- zeigen die Grundlagen Software-definierter Netzwerke am Beispiel von Onix und B4 auf.
- bewerten verschiedene im Bereich Cloud-Computing zum Einsatz kommende Datenspeichersysteme (Google File System, Bigtable, Windows Azure Storage, Amazon Dynamo) hinsichtlich der Kriterien Verfügbarkeit, Konsistenz und Partitionstoleranz.
- erläutern eine auf Vektoruhren basierende Methode zur Auflösung im Zusammenhang mit letztendlicher Konsistenz auftretender Konflikte.

- entwickeln ein verteiltes Dateisystem nach dem Vorbild von HDFS, das auf die Speicherung großer Datenmengen ausgelegt ist.
- erkunden das Bereitstellen selbst entwickelter Dienste mittels Docker.
- erstellen ein Framework zur parallelen Bearbeitung von Daten nach dem Vorbild von MapReduce.
- konzipieren eigene MapReduce-Anwendungen zur Verarbeitung strukturierter Rohdaten.
- diskutieren die Fehlertoleranzmechanismen in Google MapReduce.
- skizzieren die Abbildung von Pig-Latin-Programmen auf MapReduce.
- schildern die grundsätzliche Funktionsweise von Systemen zur Kühlung von Datenzentren mittels Umgebungsluft.
- beschreiben das Grundkonzept einer temperaturabhängigen Lastverteilung von Prozessen in einem Datenzentrum.
- stellen diverse Ansätze zur Erhöhung der Energieeffizienz von MapReduce-Clustern gegenüber.
- unterscheiden die Architekturen und Funktionsweisen der Koordinierungsdienste Chubby und ZooKeeper.
- entwickeln einen eigenen Koordinierungsdienst nach dem Vorbild von ZooKeeper.
- ermitteln die Konsistenzigenschaften der eigenen Koordinierungsdienstimplementierung.
- erschließen sich die mit der Implementierung passiv replizierter Systeme verbundenen Problematiken.
- erläutern unterschiedliche Ansätze zur Reduzierung bzw. Tolerierung von Tail-Latenz.
- skizzieren das Grundkonzept von Erasure-Codes.
- beschreiben den Aufbau eines auf die Clouds mehrerer Anbieter gestützten Datenspeichersystems.
- entwickeln einen Mechanismus zur Erkennung und Beseitigung von Überlastsituationen für virtuelle Maschinen in einer Infrastruktur-Cloud am Beispiel von Amazon EC2.
- erläutern den Einsatz passiver Replikation zur Bereitstellung von Fehlertoleranzmechanismen für virtuelle Maschinen am Beispiel von Remus.
- schildern die Grundlagen der Migration von virtuellen Maschinen.
- unterscheiden zwischen Ausfalltoleranz und byzantinischer Fehlertoleranz.
- bewerten die Qualität einer aktuellen Publikation aus der Fachliteratur.
- erschließen sich typische Probleme (Nebenläufigkeit, Konsistenz, Skalierbarkeit) und Fehlerquellen bei der Programmierung verteilter Anwendungen.
- können in Kleingruppen kooperativ arbeiten.
- können ihre Entwurfs- und Implementierungsentscheidungen kompakt präsentieren und argumentativ vertreten.
- reflektieren ihre Entscheidungen kritisch und leiten Alternativen ab.
- können offen und konstruktiv mit Schwachpunkten und Irrwegen umgehen.

#### Literatur:

- Mache Creeger. *Cloud computing: An overview*. Queue Distributed Computing, 7(5), 2009.
- Michael Armbrust, Armando Fox, Rean Griffith, Anthony D. Joseph, Randy Katz, Andy Konwinski, Gunho Lee, David Patterson, Ariel Rabkin, Ion Stoica, and Matei Zaharia. *Above the clouds: A Berkeley view of cloud computing*. Technical Report UCB/EECS-2009-28, EECS Department, University of California, Berkeley, 2009.
- Michael Armbrust, Armando Fox, Rean Griffith, Anthony D. Joseph, Randy Katz, Andy Konwinski, Gunho Lee, David Patterson, Ariel Rabkin, Ion Stoica, and Matei Zaharia. A view of cloud computing. *Communications of the ACM*, 53(4):5058, 2010.

weitere Literaturangaben auf der Vorlesungs-Webseite

---

#### Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

##### [1] Informatik (Bachelor of Science)

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule | Vertiefungsrichtung Verteilte Systeme und Betriebssysteme)

Dieses Modul ist daneben auch in den Studienfächern "Informatik (Master of Science)", "Informations- und



Kommunikationstechnik (Master of Science)", "International Information Systems (IIS) (Master of Science)",  
"Mathematik (Bachelor of Science)", "Wirtschaftsinformatik (Bachelor of Science)" verwendbar.

---

**Studien-/Prüfungsleistungen:**

Middleware - Cloud Computing (Vorlesung mit Übungen) (Prüfungsnummer: 557235)

(englische Bezeichnung: Middleware - Cloud Computing (Lecture with Exercises))

Prüfungsleistung, Portfolio

Anteil an der Berechnung der Modulnote: 100%

weitere Erläuterungen:

erfolgreiche Bearbeitung aller während des Semesters gestellten Übungsaufgaben + mündliche Prüfung am Ende des Semesters

Erstablingung: WS 2019/2020, 1. Wdh.: SS 2020

1. Prüfer: Tobias Distler (100377)

---

---

**Modulbezeichnung: Nebenläufige Systeme (CS)** **5 ECTS**  
 (Concurrent Systems)

Modulverantwortliche/r: Wolfgang Schröder-Preikschat  
 Lehrende: Wolfgang Schröder-Preikschat

---

Startsemester: WS 2019/2020	Dauer: 1 Semester	Turnus: jährlich (WS)
Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Deutsch oder Englisch

---

**Lehrveranstaltungen:**

Concurrent Systems (WS 2019/2020, Vorlesung, Wolfgang Schröder-Preikschat et al.)  
 Concurrent Systems - Exercises (WS 2019/2020, Übung, Stefan Reif et al.)

---

**Empfohlene Voraussetzungen:**

Prerequisite are hard skills (i.e., computer-science expertise) on structured computer organisation, algorithm design and development, and principles of programming in C or C++. These skills are mandatory, knowledge gaps will not be closed in the class. Further, but optional, hard skills are assembly language programming, system programming, and operating systems. As appropriate, knowledge gaps will be closed on demand by the instructors.

---

**Inhalt:**

The module imparts knowledge on concurrent systems from a systems programming point of view, especially operating systems. In a more general term, subject matter are software systems that allow for several simultaneous computations, potentially interacting with each other. Commonalities and differences of concurrence in a program flow due to multiplication (real parallelism, multi/many-core processors) or multiplexing (pseudo parallelism, partial virtualisation) of hardware processing units are covered. Thematic priority is the coordination of cooperation and concurrency between interacting (i.e., control- or data-flow dependent) processes with emphasis on explicit synchronisation. Objective is to establish understanding of different synchronisation paradigms, namely blocking and non-blocking synchronisation, and sense and sensibility of the pros and cons of the various approaches according to these paradigms.

**Lernziele und Kompetenzen:**

Students who successfully concluded the module:

- understand fundamentals and dimensions of concurrency (causality, interdependencies),
- classify characteristics of processes (sequential vs. non-sequential, concurrent, simultaneous, interacting) against the background of resource classes and sharing,
- evaluate situations of contention, race conditions and how to manage them in systems software,
- interpret data races (non-repeatable read, inconsistent read, dirty read, lost update) and control-flow races (lost wake-up, time-of-check to time-of-use),
- implement elementary methods of multilateral synchronisation (blocking synchronisation/mutual exclusion, non-blocking synchronisation) and unilateral synchronisation (condition variable),
- describe pessimistic methods for the implementation of critical sections (blocking synchronisation: lock, semaphore, mutex),
- develop optimistic methods for safeguarding interacting processes in systems software (non-blocking synchronisation: lock-free, wait-free),
- distinguish the use of atomic read-modify-write instructions such as TAS, FAS, FAA, CAS, and LL/SC, and
- discuss hardware-supported concepts, such as transactional memory.

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

**[1] Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule | Vertiefungsrichtung Verteilte Systeme und Betriebssysteme)

Dieses Modul ist daneben auch in den Studienfächern "Advanced Signal Processing & Communications Engineering (Master of Science)", "Informatik (Master of Science)", "Mathematik (Bachelor of Science)" verwendbar.

---

**Studien-/Prüfungsleistungen:**

Nebenläufige Systeme (Prüfungsnummer: 554695)

(englische Bezeichnung: Concurrent Systems)

Prüfungsleistung, Portfolio

Anteil an der Berechnung der Modulnote: 100%

weitere Erläuterungen:

20-minütige mündliche Prüfung + erfolgreiche Bearbeitung aller Übungsaufgaben

Erstablingung: WS 2019/2020, 1. Wdh.: SS 2020

1. Prüfer: Wolfgang Schröder-Preikschat (100114)

---

**Modulbezeichnung:** Informationssysteme im Gesundheitswesen (MEDINFSYS) 5 ECTS  
(Information Systems in Healthcare)

Modulverantwortliche/r: Hans-Ulrich Prokosch  
Lehrende: Hans-Ulrich Prokosch

Startsemester: WS 2019/2020      Dauer: 1 Semester      Turnus: jährlich (WS)  
Präsenzzeit: 45 Std.      Eigenstudium: 105 Std.      Sprache: Deutsch

**Lehrveranstaltungen:**

Informationssysteme im Gesundheitswesen (WS 2019/2020, Vorlesung mit Übung, 4 SWS, Hans-Ulrich Prokosch)

**Inhalt:**

In dieser Vorlesung wird ein Überblick über die Architektur und Einsatzgebiete von Informationssystemen im Gesundheitswesen gegeben. Es wird hierbei sowohl auf Informationssysteme im Krankenhaus als auch auf Systeme im niedergelassenen Bereich eingegangen. Aufgabengebiete und Funktionalitäten beispielhafter medizinischer Informationssysteme werden vorgestellt. Gleichzeitig wird auch auf die EDV-gestützte Vernetzung der verschiedenen Institutionen im Gesundheitswesen (Gesundheitstelematik, Telemedizin) sowie die Entstehung elektronischer Patientenakten eingegangen.

Die Veranstaltung besteht aus einem Vorlesungsteil (2x 90 Minuten) und einem Teil für eigene praktische Übungen, in denen die jeweiligen Themen u.a. durch Informationsrecherche im Internet vertieft werden.

**Lernziele und Kompetenzen:**

Die Studenten

*Fachkompetenz*

*Verstehen*

- erläutern unterschiedliche Prozesse des Gesundheitswesens die durch IT-Systeme unterstützt werden.
- unterscheiden zwischen der Vielzahl verschiedener IT-Systeme im Gesundheitswesen (z.B. klinische Arbeitsplatzsysteme, Elektronische Krankenakte, Abteilungssysteme in diagnostischen und therapeutischen Funktionsbereichen, Arztpraxissysteme, Apothekeninformationssysteme, Anwendungen zur Forschungs-IT) und deren Funktionalität
- verstehen die Heterogenität der IT-Landschaften im Gesundheitswesen und die Herausforderungen in Bezug auf die Interoperabilität zwischen den Systemen

*Anwenden*

- identifizieren Kommunikationsprozesse im Krankenhaus
- erfassen der passenden Kommunikations- und Nachrichtenstandards des Gesundheitswesens zur Etablierung von Schnittstellen
- analysieren IT-Landschaften in Einrichtungen des Gesundheitswesens
- konzipieren eine in sich konsistente Gesamt-IT-Architektur

*Analysieren*

- erschließen sich Prozessabläufe im Gesundheitswesen und können daraus notwendige IT-Systeme und deren Schnittstellen ableiten

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

**[1] Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Wahlpflichtbereich (5. und 6. Semester) | Wahlpflichtmodule | Vertiefungsrichtung Medizinische Informatik)

**[2] Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Nebenfach | Nebenfach Medizin (Medizinische Informatik) | Schwerpunkt Informationssysteme im Gesundheitswesen)

Dieses Modul ist daneben auch in den Studienfächern "Informatik (Master of Science)", "Mathematik (Bachelor of Science)" verwendbar.

---

**Studien-/Prüfungsleistungen:**

Informationssysteme im Gesundheitswesen (Prüfungsnummer: 29101)

(englische Bezeichnung: Health Care Information Systems)

Prüfungsleistung, Klausur, Dauer (in Minuten): 60

Anteil an der Berechnung der Modulnote: 100% Prüfungssprache: Deutsch

Erstablingung: WS 2019/2020, 1. Wdh.: SS 2020

1. Prüfer: Hans-Ulrich Prokosch (031003)

---

**Organisatorisches:**

Studierende melden sich bitte mit Angabe ihrer Matrikelnummer, ihrer StudOn-Kennung und ihres Abschlusses (Bachelor/Master) per Mail <mailto:martin.ross@fau.de> an.

---

**Modulbezeichnung:** **Marketing Fallstudien (MaFa)** **5 ECTS**  
(Marketing Case Studies)

Modulverantwortliche/r: Christian Götz  
Lehrende: Christian Götz

---

Startsemester: WS 2019/2020	Dauer: 1 Semester	Turnus: jährlich (WS)
Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Deutsch

---

**Lehrveranstaltungen:**

Marketing Fallstudien (WS 2019/2020, Vorlesung mit Übung, 4 SWS, Christian Götz)

---

**Inhalt:**

- Kommunikation und Kommunikationsmodelle
- Einführung in das moderne Marketingmanagement
- Grundlagen der Marketinginstrumente
- Aspekte der Marktsegmentierung und Produktplatzierung
- Marktforschung und ihre Methoden
- Kundenprädispositionen und Kaufbereitschaft
- Kommunikationsstrategie und Werbemanagement
- Grundlegende Instrumente im Kommunikationsmix
- Grundzüge des Neuromarketings und der Werbepsychologie

**Lernziele und Kompetenzen:**

Die Studierenden

- erläutern die wesentlichen Inhalte und Methoden der modernen Marketinginstrumente im Marketing-Mix (Produkt-, Kontrahierungs-, Distributions- und Kommunikationspolitik)
- beschreiben die wesentlichen Designs und Methoden (Beobachtung, Befragung, Experiment) der modernen Marktforschung
- erklären den Prozess der Entwicklung von zielgruppenspezifischen Werbebotschaften und Aussagen
- kennen wesentliche Komponenten der modernen Werbepsychologie
- können diese in bestimmten Fallbeispielen (z.B. bei der Anordnung und dem Aufbau des Point of Sale) verifizieren
- wissen über die Wirkung von Werbebotschaften und können diese an Beispielen analysieren
- können Erkenntnisse aus der Vorlesung auf ihr eigenes On- und Offline-Einkaufsverhalten anwenden

**Literatur:**

- Homburg, Christian: Marketingmanagement
  - Meffert, Heribert: Marketing
  - Kotler, Philip: Marketing Management
  - Moser, Klaus: Markt- und Werbepsychologie
  - Scheier, Christian, Held, Dirk: Wie Werbung wirkt. Erkenntnisse des Neuromarketings
- 

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

**[1] Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Nebenfach | Nebenfach Betriebswirtschaftslehre | Wahlbereich)

Dieses Modul ist daneben auch in den Studienfächern "Informatik (Master of Science)" verwendbar.

---

**Studien-/Prüfungsleistungen:**

Marketing-Fallstudien (Prüfungsnummer: 24911)

Prüfungsleistung, Präsentation/Hausarbeit

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Marketing Fallstudien

weitere Erläuterungen:  
Prüfungsleistung = 100 % Hausarbeitsnote

Erstablingung: WS 2019/2020, 1. Wdh.: SS 2020  
1. Prüfer: Christian Götz (100187)

---

**Organisatorisches:**

StudOn:

**Bemerkungen:**

Übung nach Vereinbarung

---

**Modulbezeichnung:** **BWL für Ingenieure (BWL-ING)** **5 ECTS**  
(Business Administration for Engineers)

Modulverantwortliche/r: Kai-Ingo Voigt  
Lehrende: Kai-Ingo Voigt

---

Startsemester: WS 2019/2020	Dauer: 2 Semester	Turnus: jährlich (SS)
Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Deutsch

---

**Lehrveranstaltungen:**

BWL für Ingenieure I (WS 2019/2020, Vorlesung, Kai-Ingo Voigt et al.)  
BWL für Ingenieure II (SS 2020, Vorlesung mit Übung, 4 SWS, Kai-Ingo Voigt et al.)

---

**Inhalt:**

BW 1 (konstitutive Grundlagen):  
Grundlagen und Vertiefung spezifischer Aspekte der Rechtsform-, Standort-, Organisations- und Strategiewahl  
BW 2 (operative Leistungsprozesse):  
Betrachtung der unternehmerischen Kernprozesse Forschung und Entwicklung mit Fokus auf das Technologie- und Innovationsmanagement, Beschaffung und Produktion sowie Marketing und Vertrieb  
BW 3 (Unternehmensgründung):  
Grundlagen der Gründungsplanung und des Gründungsmanagements  
BW 3 Übung (Vertiefung und Businessplanerstellung):  
Vertiefung einzelner Schwerpunkte aus den Bereichen BW 1, 2 und 3 sowie ausgewählte Fallstudien zu wichtigen Elementen eines Businessplans

**Lernziele und Kompetenzen:**

Die Studierenden

- erwerben Kenntnisse über Grundfragen der allgemeinen Betriebswirtschaftslehre
- verstehen die Kernprozesse der Unternehmung und die damit verbundenen zentralen Fragestellungen
- erwerben ein Verständnis für den Entwicklungsprozess der Unternehmung sowie deren Kernprozesse, insbesondere verfügen sie über breites und integriertes Wissen einschließlich der wissenschaftlichen Grundlagen in den Bereichen Forschung und Entwicklung, Beschaffung, Produktion, Marketing und Vertrieb.
- können Fragen des Technologie- und Innovationsmanagements anhand der Anwendung ausgewählter Methoden und Instrumente erschließen
- wissen um die Bestandteile eines Businessplans, deren Bedeutung und sind in der Lage, diese zu verfassen und zu beurteilen

**Literatur:**

Voigt, Industrielles Management, 2008

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

**[1] Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Nebenfach | Nebenfach Betriebswirtschaftslehre)

Dieses Modul ist daneben auch in den Studienfächern "Informatik (Master of Science)", "International Production Engineering and Management (Bachelor of Science)", "Life Science Engineering (Master of Science)", "Maschinenbau (Bachelor of Science)", "Maschinenbau (Master of Science)", "Materialwissenschaft und Werkstofftechnik (Bachelor of Science)", "Mechatronik (Master of Science)", "Medizintechnik (Master of Science)", "Wirtschaftsingenieurwesen (Bachelor of Science)" verwendbar.

---

**Studien-/Prüfungsleistungen:**



Betriebswirtschaftslehre für Ingenieure (Vorlesung) (Prüfungsnummer: 25701)

(englische Bezeichnung: Lecture: Business Studies for Engineers)

Prüfungsleistung, Klausur, Dauer (in Minuten): 60

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- BWL für Ingenieure I
- BWL für Ingenieure II

Erstablesung: SS 2020, 1. Wdh.: WS 2020/2021

1. Prüfer: Kai-Ingo Voigt (090120)

---

**Modulbezeichnung:** Allgemeine Biologie I: Biologie für Nebenfächler (NF\_Bio I) 5 ECTS  
(Biology for minor subject students)

Modulverantwortliche/r: Michael Lebert

Lehrende: Johann Helmut Brandstätter, Ingrid Brehm, Renato Frischknecht, Michael Lebert,  
Gerald Seidel

Startsemester: WS 2019/2020

Dauer: 1 Semester

Turnus: jährlich (WS)

Präsenzzeit: 60 Std.

Eigenstudium: 90 Std.

Sprache: Deutsch

**Lehrveranstaltungen:**

Biologie für Nebenfächler (WS 2019/2020, Vorlesung, 4 SWS, Michael Lebert et al.)

**Inhalt:**

**Botanik:**

- Bau und Leistung der Pflanzenzelle
- Morphologie und Anatomie der Pflanzenorgane
- Systematik und Evolution von Pflanzen
- Vermehrung von Pflanzen
- Pflanzenphysiologie
- Pflanze und Umwelt

**Zoologie:**

- Stoffwechsel, Kreislauf und Atmung
- erregbare Zellen: Muskelzellen und Nervenzellen
- zelluläre Neurophysiologie (Ruhepotential, Aktionspotential, axonale Weiterleitung der Erregung, Synapse)

**Mikrobiologie:**

- Einführung in die Mikrobiologie
- Zellstruktur und Zellfunktion
- Grundlagen der Molekularbiologie und Bakteriengenetik
- Mikrobiologie der Prokaryoten (Physiologie, Taxonomie und Phylogenie)
- Grundlagen der Virologie

**Lernziele und Kompetenzen:**

Die Studierenden

- können die Struktur und Funktionen der Biomoleküle in Ihren Grundzügen beschreiben und erläutern;
- verstehen die Zelltypen verschiedener Organismen und können deren Zellbestandteile- und -bausteine darstellen und erklären;
- kennen die Grundbegriffe der Zytologie, Morphologie und Anatomie der Pflanzen und sind in der Lage diese Einordnungen anzuwenden;
- sind in der Lage, die Physiologie der Pflanzen darzustellen;
- können die Anpassungen von Pflanzen darlegen;
- sind befähigt, die Evolution der Pflanzen in den Grundzügen zu erklären;
- können zelluläre Unterschiede zwischen Pflanzen und Tieren erläutern;
- sind in der Lage, die fundamentalen Prozesse des Energiestoffwechsels der Tiere &#8208; und damit verbundene Anpassungen (Kreislauf und Atmung) in den Grundzügen darzustellen und zu beschreiben;
- verstehen die zellulären und molekularen Grundlagen der Muskelkontraktion und können diese darstellen und verdeutlichen;
- können zelluläre Grundlagen sowie grundlegende Funktionsmechanismen von Nervenzellen einordnen
- verstehen den Einfluss von Mikroorganismen auf Ökosysteme und deren Nutzung in Landwirtschaft, Biotechnik, Medizin und Lebensmittelproduktion;
- erwerben basale Kenntnisse der Bakteriengenetik, der Physiologie, der taxonomischer Einteilung und den Grundlagen der Virologie.

**Literatur:**

Nultsch, Allgemeine Botanik, Thieme Verlag Fuchs, Allgemeine Mikrobiologie Thieme&#8208;Verlag

Wehner, Gehring, Kühn, Zoologie, Thieme Brock: Mikrobiologie, Pearson Verlag Campbell, Biologie, Pearson

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

**[1] Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Nebenfach | Nebenfach Biologie)

Dieses Modul ist daneben auch in den Studienfächern "Informatik (Master of Science)", "Physische Geographie (Bachelor of Science)", "Psychologie (Bachelor of Science)", "Psychologie (Master of Science)" verwendbar.

---

**Studien-/Prüfungsleistungen:**

Biologie für Nebenfachstudierende (Prüfungsnummer: 29201)

(englische Bezeichnung: Biologie für Nebenfachstudierende)

Prüfungsleistung, Klausur, Dauer (in Minuten): 90

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Biologie für Nebenfächler

Erstablingung: WS 2019/2020, 1. Wdh.: SS 2020

1. Prüfer: Michael Lebert (070138)

---

**Modulbezeichnung:** Allgemeine Biologie II: Morphologie und Anatomie der Organismen (NF\_Bio II) 5 ECTS  
(Biology for minor subject students II: Morphology and anatomy of Organisms)

Modulverantwortliche/r: Ralph Rübsam  
Lehrende: Michael Lebert, Ralph Rübsam

Startsemester: WS 2019/2020      Dauer: 1 Semester      Turnus: jährlich (WS)  
Präsenzzeit: 75 Std.      Eigenstudium: 75 Std.      Sprache: Deutsch

**Lehrveranstaltungen:**

Morphologie und Anatomie der Organismen (WS 2019/2020, Übung, 5 SWS, Franz Klebl et al.)

**Inhalt:**

- Einführung in die Lichtmikroskopie (Hellfeld-, Dunkelfeld-, Phasenkontrast-Mikroskopie)
- Charakteristika eukaryontischer Zellen am Beispiel von Amöben und Ciliaten (u.a. Phagozytose, verschiedene Fortbewegungstypen)
- Entwicklung eines Tieres am Wirbeltierbeispiel (Huhn)
- Organisationsprinzipien vielzelliger Tiere am Beispiel repräsentativer Tiergruppen (Cnidaria, Plathelminthes, Annelida, Arthropoda, Vertebrata)
- Evolutive Abwandlung und ökologische Anpassungen dieser Baupläne
- Algen und Pflanzen: u.a. Cyanobakterien, Kieselalgen und Grünalgen (Organisationsstufen), Moose und Farne (Aufbau und Generationswechsel), Höhere Pflanzen (Wurzel und Physiologie der Wasser- und Nährstoffaufnahme, Spross mit Leitgeweben und sekundärem Dickenwachstum, Blatt und Photosynthese, Blüte, Fortpflanzung und Frucht)

**Lernziele und Kompetenzen:**

Die Studierenden

- lernen grundlegende Mikroskopier- und Präparationstechniken
- sind in der Lage mikroskopische und anatomische Präparate zeichnerisch zu protokollieren
- erkennen typische tierische Gewebe in histologischen Präparaten und - kennen die charakteristischen Phasen der Entwicklung eines Wirbeltieres und können die dabei ablaufenden Prozesse wiedergeben
- verstehen die basalen Funktionen wichtiger tierischer Organsysteme und können diese in den verschiedenen Bauplänen miteinander vergleichen
- kennen die grundsätzlichen Trends der Evolution pflanzlicher und tierischer Baupläne und können deren adaptive Bedeutung ermessen
- bekommen ein vertieftes Verständnis von anatomischen und zellulären Funktionsbeziehungen bei Pflanzen und Tieren
- verfügen über Grundlegende Kenntnisse der Formenkunde

**Literatur:**

Campbell: Biologie; Wehner/Gehring: Zoologie

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

**[1] Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Nebenfach | Nebenfach Biologie)

Dieses Modul ist daneben auch in den Studienfächern "Informatik (Master of Science)", "Physische Geographie (Bachelor of Science)", "Psychologie (Bachelor of Science)", "Psychologie (Master of Science)" verwendbar.

**Studien-/Prüfungsleistungen:**

Morphologie und Anatomie der Organismen (Prüfungsnummer: 29202)

(englische Bezeichnung: Morphology and Anatomy of Organisms)

Prüfungsleistung, Klausur, Dauer (in Minuten): 45

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Morphologie und Anatomie der Organismen  
Prüfungssprache: Deutsch

Erstablingung: WS 2019/2020, 1. Wdh.: SS 2020  
1. Prüfer: Ralph Rübsam (070223)

---

**Modulbezeichnung:** Ökologie und Diversität A (ÖkoDivA) **5 ECTS**  
(Ecology and Diversity A)

Modulverantwortliche/r: Jürgen Schmidl

Lehrende: Jürgen Schmidl, Andreas Feigenspan

Startsemester: WS 2019/2020

Dauer: 1 Semester

Turnus: jährlich (WS)

Präsenzzeit: 75 Std.

Eigenstudium: 75 Std.

Sprache: Deutsch

**Lehrveranstaltungen:**

Ökologische und systematische Diversität der Organismen A: Einführung in die Zoologie (WS 2019/2020, Vorlesung, 2 SWS, Andreas Feigenspan et al.)

Ökologische und systematische Diversität der Organismen A: Zoologische Bestimmungsübungen (WS 2019/2020, Übung, 3 SWS, Jürgen Schmidl)

**Empfohlene Voraussetzungen:**

keine

**Inhalt:**

**Vorlesung:** Grundkenntnisse zu folgenden Teilgebieten der Zoologie:

- Physiologie (Grundlagen der Sinnesphysiologie, Exkretion, Verdauung, Thermoregulation, Hormonsteuerung etc.)
- Morphologie (Systematik des Tierreiches, Kennenlernen ausgewählter Baupläne)
- Evolution (Mechanismen und Aspekte der Evolution)
- Phylogenie (Methoden der Systematik und Taxonomie, Artkonzepte)
- Ökologie (Großlebensräume der Erde und Einnischung von Tierarten- und Gruppen)
- Biogeographie (Konzepte und geologisch-historische Grundlagen der globalen Verbreitung der Tiergruppen)

**Übungen:**

- Morphologie, Systematik und Diversität der wichtigsten Tiergruppen und ihrer typischen Vertreter
- Praktische Übungen zum Bestimmen heimischer Tiergruppen mittels Bestimmungsschlüssel und Stereomikroskop
- Biologie und Ökologie der zuvor bestimmten Arten und Gruppen

**Lernziele und Kompetenzen:**

Die Studierenden

- können die wichtigsten Tiergruppen unterscheiden und ihre typischer Vertreter erkennen;
- verstehen die Diversität im Tierreich;
- können die Grundlagen der Physiologie, Morphologie, Evolution, Phylogenie, Ökologie und Biogeographie darstellen und erklären;
- sind befähigt, zum Erkennen und Lösen von relevanten Problemen aus ausgewählten Teilgebieten der Zoologie;
- sind in der Lage, mit dem Bestimmungsschlüssel umzugehen;
- sind fähig, die Vorlesungsinhalte in Übungen am Stereomikroskop praktisch umzusetzen;
- können aufgrund der regelmäßigen aktiven Teilnahme an den Laborübungen fachgerecht mit dem Stereomikroskop umgehen.

**Literatur:**

Brohmer: Fauna von Deutschland; Wehner/Gehring: Zoologie

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

**[1] Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Nebenfach | Nebenfach Biologie)

Dieses Modul ist daneben auch in den Studienfächern "Biologie (1. Staatsprüfung für das Lehramt an Gymnasien)", "Biologie (Bachelor of Science)", "Informatik (Master of Science)", "Physische Geographie (Bachelor of Science)", "Psychologie (Bachelor of Science)", "Psychologie (Master of Science)" verwendbar.

---

**Studien-/Prüfungsleistungen:**

Ökologische und Systematische Diversität der Organismen A (Prüfungsnummer: 29401)

(englische Bezeichnung: Introduction to Zoology)

Prüfungsleistung, Klausur, Dauer (in Minuten): 45

Anteil an der Berechnung der Modulnote: 100%

Erstabelleung: WS 2019/2020, 1. Wdh.: SS 2020

1. Prüfer: Jürgen Schmidl (111461)

---

**Modulbezeichnung:** **Mikrobiologische Übungen (NF\_Bio V)** **5 ECTS**  
(Laboratory Course in Microbiology)

Modulverantwortliche/r: Gerald Seidel  
Lehrende: Gerald Seidel

Startsemester: WS 2019/2020      Dauer: 1 Semester      Turnus: jährlich (WS)  
Präsenzzeit: 90 Std.      Eigenstudium: 60 Std.      Sprache: Deutsch

**Lehrveranstaltungen:**

Mikrobiologische Übungen für Naturwissenschaftler und Techniker (WS 2019/2020, Übung, 6 SWS, Andreas Burkovski et al.)

**Inhalt:**

Mikroskop, Färbetechniken, Kultur- und Sterilisationsverfahren, -Wachstum von Bakterien, Antibiotika-Transformation von *Acinetobacter spec.*, -Identifizierung/Diagnostik von Bakterien

- grundlegende Techniken der Molekularbiologie
- Experimente: Beobachtung von Bakterien im Mikroskop, verschiedene Darstellungsverfahren
- Nachweis von Keimen in der Luft
- Erlernen verschiedener Techniken, Herstellung von Nährmedien, Bestimmung Zellzahl in einer Kolonie, Bestimmung der Phagenzahl in einem Plaque, Sterilisationsversuche
- selektive Anreicherung von Bakterien, Bakterienwachstumskurve; Einfluss von Antibiotika auf das Wachstum von Bakterien
- Isolierung von Antibiotika-Produzenten
- Nachweis und Identifizierung von Bakterien, Resistenzbestimmung, Isolierung von Antibiotika-Produzenten,
- Plasmid-Isolierung und Spaltung mit Restriktionsenzymen
- Agarose-Gelelektrophorese, Protein-Isolierung und Polyacrylamid-Gelelektrophorese

**Lernziele und Kompetenzen:**

Aneignung der Grundkenntnisse der Mikrobiologie und molekularbiologischen Grundtechniken

**Literatur:**

- Lehrbuch: Brock Mikrobiologie, M. T. Madigan & J. M. Martinko, aktuelle Ausgabe (z.Zt. 2013)
- Lehrbuch: Allgemeine Mikrobiologie, G. Fuchs, aktuelle Ausgabe (z.Zt. 2007)
- Lehrbuch: Mikrobiologische Methoden, E. Bast

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

**[1] Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Nebenfach | Nebenfach Biologie)

Dieses Modul ist daneben auch in den Studienfächern "Informatik (Master of Science)", "Physische Geographie (Bachelor of Science)", "Psychologie (Bachelor of Science)", "Psychologie (Master of Science)" verwendbar.

**Studien-/Prüfungsleistungen:**

Praktikum Mikrobiologische Übungen für Naturwissenschaftler (Prüfungsnummer: 29802)  
(englische Bezeichnung: Medical Engineering (2,5 ECTS))

Studienleistung, Protokollheft

Zugeordnete Lehrveranstaltungen:

- Mikrobiologische Übungen für Naturwissenschaftler und Techniker

Erstablingung: WS 2019/2020, 1. Wdh.: keine Angabe

1. Prüfer: Gerald Seidel (111463)



**Modulbezeichnung:** Ökologie und Diversität B (ÖkoDivB) **5 ECTS**  
(Ecology and Diversity B)

Modulverantwortliche/r: Ruth Stadler

Lehrende: Ruth Stadler, Jürgen Schmidl, Regula Muheim-Lenz, Ulrike Daigl,

Startsemester: SS 2019

Dauer: 1 Semester

Turnus: jährlich (SS)

Präsenzzeit: 75 Std.

Eigenstudium: 75 Std.

Sprache: Deutsch

### Lehrveranstaltungen:

Übungen zur Biologie und Systematik einheimischer Pflanzen (SS 2019, Vorlesung mit Übung, 4 SWS, Anwesenheitspflicht, Ruth Stadler)

### Inhalt:

#### Zoologische Geländeübungen:

- Erkundung typischer Biotoptypen in der Umgebung Erlangens (Kiefernwald, Weiher incl. Plankton- und Saprobienthematik, Wiese, Fließwasser) und ihrer Tiergemeinschaften und Ökologie
- Einführung in Präparation von Wirbellosen und Anlegen einer wissenschaftlichen zoologischen Sammlung

#### Übungen:

Erkundung von Beispielarten in der Umgebung Erlangens an folgenden Standorten:

- Rathsberg: Laubmischwald: Caryophyllaceae: *Stellaria*, Ranunculaceae: *Anemone*, Violaceae: *Viola*, Liliaceae: *Polygonatum*
- Regnitztal: Auwald: Brassicaceae: *Alliaria*, Lamiaceae: *Lamium*, Salicaceae: *Salix*
- Schwabachtal: Sandmagerrasen: Fabaceae: *Cytisus*, Rosaceae: *Potentilla*, Euphorbiaceae: *Euphorbia*
- Regnitzwiesen: Kräuter der Fettwiese: Apiaceae: *Anthriscus*, Asteroideae: *Leucanthemum*, Cichorioideae: *Taraxacum*, Polygonaceae: *Rumex*
- Regnitzwiesen: Gräser der Fettwiese: Poaceae: *Arrhenatherum*, *Poa*, *Lolium*, *Festuca*
- Walberla: Kalkmagerrasen: Plantaginaceae: *Veronica*, Plantago Orobanchaceae: *Rhinanthus*
- Tennenlohe: Sandäcker: Chenopodiaceae: *Chenopodium*, Geraniaceae: *Erodium*
- Heusteg: Verlandungsreihe eutropher Gewässer: Cyperaceae: *Carex*, Solanaceae: *Solanum*, Juncaceae: *Juncus*, Primulaceae: *Lysimachia*
- An verschiedenen Standorten: sandige, nährstoffreiche Ruderalfluren: Hypericaceae: *Hypericum* Onagraceae: *Oenothera*
- Reichswald: Nadelforst auf Sandböden: Ericaceae: *Vaccinium*, Gymnospermae: *Pinus*, Pteridophyta: *Dryopteris*

### Lernziele und Kompetenzen:

Die Studierenden

- können aufgrund der regelmäßigen aktiven Teilnahme an den Bestimmungsübungen und Geländeübungen die wichtigsten einheimischen Tier- und Pflanzenfamilien und deren typischer Vertreter an ihrem Standort (Exkursionen) erkennen und unterscheiden (Formenkenntnis);
- sind in der Lage, fachgerecht mit dem Bestimmungsschlüssel umzugehen;
- sind fähig, ein wissenschaftliches Herbar und eine zoologische Sammlung anzulegen;
- sind zur Teamarbeit befähigt.

### Literatur:

Brohmer Fauna von Deutschland, Schmeil-Fitschen: Flora von Deutschland;

Rothmaler: Exkursionsflora, Oberdorfer: Pfl.-soziol. Exkursionsflora

### Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

#### [1] Informatik (Bachelor of Science)

(Po-Vers. 2009s | Nebenfach | Nebenfach Biologie)

Dieses Modul ist daneben auch in den Studienfächern "Biologie (Bachelor of Science)", "Informatik (Master of Science)", "Physische Geographie (Bachelor of Science)", "Psychologie (Bachelor of Science)", "Psychologie (Master of Science)" verwendbar.

**Studien-/Prüfungsleistungen:**

Klausur zu Ökologische und Systematische Diversität der Organismen B (Prüfungsnummer: 28201)  
(englische Bezeichnung: Examination (Klausur) on Ecological and Systematic Diversity of Organisms B)

Prüfungsleistung, Klausur, Dauer (in Minuten): 45

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Übungen zur Biologie und Systematik einheimischer Pflanzen

Erstablingung: SS 2019, 1. Wdh.: WS 2019/2020

1. Prüfer: Ruth Stadler (070141)

---

---

**Modulbezeichnung: Abfallaufbereitung (ABA)** **5 ECTS**  
(Waste Treatment)

Modulverantwortliche/r: Stefanos Georgiadis  
Lehrende: Stefanos Georgiadis

---

Startsemester: WS 2019/2020	Dauer: 1 Semester	Turnus: jährlich (WS)
Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Deutsch

---

**Lehrveranstaltungen:**

Abfallaufbereitung (WS 2019/2020, Vorlesung, 2 SWS, Stefanos Georgiadis)  
Übungen zu Abfallaufbereitung (WS 2019/2020, Übung, 1 SWS, Stefanos Georgiadis)

---

**Inhalt:**

Die Vorlesung vermittelt einen Einblick in die heutige Abfallwirtschaft und das Abfallmanagement. Ausgehend von der Abfallgesetzgebung (Kreislaufwirtschafts- und Abfallgesetz, Deponieverordnung 2009) werden anhand zweier typischer Großstädte in unserer Region die anfallenden Abfallarten und Mengen sowie deren zeitliche Entwicklung betrachtet. Diese Entwicklungen erlauben Rückschlüsse auf die Wirksamkeit der getroffenen legislativen Maßnahmen. Im Anschluß daran werden verschiedene Entsorgungskonzepte mit der dazugehörigen Logistik erläutert und miteinander verglichen. In einem gesonderten Kapitel wird auf die Deponietechnik eingegangen. Es werden Grundlagen vermittelt, die dem angehenden Ingenieur, der sich beruflich damit auseinandersetzen muß, für Planung, Aufbau und fachliche Beurteilung von Deponien von Nutzen sind. Im letzten Teil der Vorlesung werden die in der Abfallaufbereitung üblichen verfahrenstechnischen Prozesse und die hierfür notwendigen Maschinen und Apparate besprochen.

**Lernziele und Kompetenzen:**

Die Studierenden

- entdecken die z.T. sehr komplexen Zusammenhänge in der heutigen Abfallwirtschaft
- können mit Regelwerken aus der Abfallgesetzgebung effizient arbeiten
- eignen sich gebräuchlichen Arbeitsmethoden im Abfallmanagement an und können Sie anwenden
- können genehmigungsrechtlichen Fragen innerhalb der Abfallwirtschaft einschätzen und bearbeiten

**Literatur:**

Vorlesungsskript *Abfallaufbereitung*

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

**[1] Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Nebenfach | Nebenfach Chemie- und Bioingenieurwesen)

Dieses Modul ist daneben auch in den Studienfächern "Chemical Engineering - Nachhaltige Chemische Technologien (Master of Science)", "Chemie- und Bioingenieurwesen (Master of Science)", "Energietechnik (Master of Science)", "Informatik (Master of Science)" verwendbar.

---

**Studien-/Prüfungsleistungen:**

Abfallaufbereitung (Prüfungsnummer: 51401)

(englische Bezeichnung: Waste Treatment)

Prüfungsleistung, mündliche Prüfung, Dauer (in Minuten): 30

Anteil an der Berechnung der Modulnote: 100%

Erstablegung: WS 2019/2020, 1. Wdh.: SS 2020

1. Prüfer: Stefanos Georgiadis (100440)

---

**Organisatorisches:**

Diese Vorlesung richtet sich sowohl an technisch ausgerichtete Studiengänge als auch an "*Nebenfächler*" aus naturwissenschaftlichen Disziplinen. Die Vorlesung soll insbesondere Studierende ansprechen, die

ihre Kenntnisse auf dem Abfallsektor spezialisieren möchten. Die Vorlesung wird in Deutsch gehalten.  
Das Manuskript ist in digitaler Form (pdf) in deutscher Sprache verfügbar.

**Für diese Lehrveranstaltung ist eine Anmeldung erforderlich!**

Die Anmeldung kann über StudOn erfolgen.

Anmeldelink: [http://www.studon.uni-erlangen.de/crs178512\\_join.html](http://www.studon.uni-erlangen.de/crs178512_join.html)

**Modulbezeichnung:** Entsorgung medizinischer Abfälle ohne Praktikum (EMA) 5 ECTS  
(Medical Waste Treatment (without practical course))

Modulverantwortliche/r: Stefanos Georgiadis  
Lehrende: Stefanos Georgiadis

Startsemester: SS 2019	Dauer: 1 Semester	Turnus: jährlich (SS)
Präsenzzeit: 45 Std.	Eigenstudium: 105 Std.	Sprache: Deutsch

**Lehrveranstaltungen:**

Entsorgung medizinischer Abfälle (SS 2019, Vorlesung, 2 SWS, Stefanos Georgiadis)  
Übungen zu Entsorgung medizinischer Abfälle (SS 2019, Übung, 1 SWS, Stefanos Georgiadis)

**Empfohlene Voraussetzungen:**

Grundkenntnisse auf dem Gebiet der heutigen Abfallwirtschaft (Haus-, Gewerbe-, Sondermüll; vgl. Vorlesung Abfallaufbereitung im WS) vorteilhaft, aber nicht Voraussetzung

**Inhalt:**

Die Vorlesung versucht einen Einblick in die Problematik des Abfallhandlings, der Logistik sowie der Entsorgung von medizinischen Abfällen in und außerhalb von Einrichtungen des Gesundheitsdienstes aus ingenieurtechnischer Sicht zu vermitteln:

- Abfallgesetzgebung
- Abfallarten und Mengen
- Klassifizierung von Abfällen / Europäisches Abfallverzeichnis
- Entsorgungskonzepte, Sammlung / Transport /(Ab-)Lagerung
- Behandlungsverfahren für Krankenausabfälle
- Abwasserentsorgung in Krankenhäusern
- Arzneimittelrückstände in Abwässern und Oberflächengewässern

**Lernziele und Kompetenzen:**

Die Studierenden

- bestimmen aus ingenieurtechnischer Sicht die Problematik des Abfallhandlings, der Logistik sowie der Entsorgung von medizinischen Abfällen, in und außerhalb von Einrichtungen des Gesundheitsdienstes
- untersuchen die verfahrenstechnische Behandlung/Konditionierung derartiger Abfälle inklusive ihrer Entsorgungspfade
- können im späteren Berufsleben zwischen Behörden und Industrie, (z.B. im Rahmen von Genehmigungs-/Abwicklungsverfahren für abfalltechnische Anlagen) als Gesprächspartner agieren.

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

[1] **Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Nebenfach | Nebenfach Chemie- und Bioingenieurwesen)

Dieses Modul ist daneben auch in den Studienfächern "Chemie- und Bioingenieurwesen (Master of Science)", "Informatik (Master of Science)", "Life Science Engineering (Master of Science)" verwendbar.

**Studien-/Prüfungsleistungen:**

Entsorgung medizinischer Abfälle (Prüfungsnummer: 53201)  
Prüfungsleistung, mündliche Prüfung, Dauer (in Minuten): 30  
Anteil an der Berechnung der Modulnote: 100%  
Zugeordnete Lehrveranstaltungen:  
• Entsorgung medizinischer Abfälle

Erstablingung: SS 2019, 1. Wdh.: WS 2019/2020  
1. Prüfer: Stefanos Georgiadis (100440)

**Organisatorisches:**

Für diese Lehrveranstaltung ist eine Anmeldung über StudOn erforderlich!

**Modulbezeichnung:** Umweltbioverfahrenstechnik ohne Praktikum (UBVT) 5 ECTS  
(Environmental Biotechnology)

Modulverantwortliche/r: Roman Breiter  
Lehrende: Roman Breiter

Startsemester: SS 2019	Dauer: 1 Semester	Turnus: jährlich (SS)
Präsenzzeit: 45 Std.	Eigenstudium: 105 Std.	Sprache: Deutsch und Englisch

**Lehrveranstaltungen:**

Umweltbioverfahrenstechnik (SS 2019, Vorlesung, 2 SWS, Roman Breiter)  
Übungen zu Umweltbioverfahrenstechnik (SS 2019, Übung, 1 SWS, Roman Breiter)

**Empfohlene Voraussetzungen:**

- Grundkenntnisse der aquatischen Chemie
- Grundkenntnisse der Mikrobiologie

Vorlesung:

Grundlagen der Biologie und Chemie

Übung:

Die Übung setzt den Besuch der Vorlesung Mikrobielle Verfahrenstechnik voraus.

Seminar:

Das Seminar setzt den Besuch der Vorlesung Mikrobielle Verfahrenstechnik voraus.

**Inhalt:**

Vorlesung:

- Stoffströme in biotechnologischen Prozessen der Reinigung von Wasser, Boden und Luft.
- Substratabbau und Wachstum (Trophieebenen, Energieproduktion, Zellsynthese, Schlammalter, endogener Abbau, Massenbilanzen)
- Zusammensetzung von Abwasser (Chemische Zusammensetzung von Partikeln und gelösten Stoffen, Kenngrößen für die Abwasserreinigung)
- Legislativer Hintergrund (Wasserhaushaltsgesetz, Abwassersatzungen, Direkt- und Indirekteinleitung, Grenzwerte)
- Mechanische Vorbehandlung von Abwasser (Siebe, Sandfang, Klärer)
- Vorgänge in natürlichen und belüfteten Teichsystemen (physikalische und biologische Belüftung, natürliche biologische Prozesse in Wasser und Sediment)
- Land treatment und Land application (Rieselfelder, Infiltrationen, Melioration)
- Pflanzenkläranlagen, Free Wetland Systems FWS, Vertical Submerged Beds VSB (Design, Reinigungsprinzipien)
- Abwasserbehandlung mit suspendierter Biomasse (Turmbiologie, Biohochreaktor, Belebtschlammverfahren, Verweilzeiten)
- Abwasserbehandlung mit sessiler Biomasse (Rotating Biological Contactor RBC, Membranbiologische Verfahren, Tropfkörper)
- Stickstoffeliminierung, Nitrifikation, Denitrifikation, N-Spezies und Belüftung)
- Phosphateliminierung (Chemische Verfahren, enhanced biological phosphate removal processes EBPR, A/O-Verfahren und Phostrip-Prozess)
- Hygienisierung (Legislative Anforderungen, humanpathogene Organismen und Viren, CT-Konzept, Ozonierung und UV-Behandlung)
- Anaerobe Verfahren der Schlamm- und Abwasserbehandlung
- Boden- und Grundwassersanierung (Gesetzeslage, Natural Attenuation, pump-and-treat-Verfahren)

Übung:

- Alle Themen der Vorlesung mit erweiterten Grundlagen

Seminar:

- Vertiefung von Inhalten der Vorlesung durch Wiederholung und weiteren Aufbau von Grundlagen

## Lernziele und Kompetenzen:

Vorlesung:

Die Studierenden

- erlernen die Identifikation von Stoffströmen im Umweltschutz aus dem Grundprinzip der Dekontamination und Reinigung, nachdem neben den gereinigten Umweltmedien Boden, Wasser und Luft nur untoxische Produkte und inerte, untoxische Rückstände entstehen dürfen.
- lernen den Zusammenhang zwischen der Weiterentwicklung umwelttechnischer Anlagen und gesetzlichen Regelungen kennen.
- wenden Grundlagen des Substratabbaus, Biomassenwachstums und der Verfügbarkeit von terminalen Elektronenakzeptoren auf biologische Prozesse in natürlichen, aquatischen Systemen an und
- können diese natürlichen Prozesse ingenieurstechnisch für die Abwasserreinigung und Grundwasser-sanierung optimieren und intensivieren.
- erlernen Grundlagen der C-, N- und P-Eliminierung und wenden diese auf komplexere Systeme mit mineralischen und organischen Feststoffen sowie gelöste Substanzen in aquatischen Systemen an.
- Leiten Verfahrensvarianten bei geänderten Randbedingungen (Frachten, Konzentrationen, Zusammensetzung, Belüftung) ab.
- Übertragen Kenntnisse von Prozessen im Biofilm (Diffusion, Substratabbau, Limitierungen) auf Prozesse mit den für die Abwasserreinigung typischen, natürlichen Randbedingungen (Mischpopulationen, Zonen verschiedener Elektronenakzeptoren, Makrofauna).
- Lernen die Grundlagen anaeroben Schadstoffabbaus in Biozönosen kennen und verknüpfen diese mit dem Design von anaeroben Behandlungsanlagen für Schlamm und Abwasser.
- Lernen aktuelle Entwicklungen der Sanierung von Boden und Grundwasser anhand von am Lehrstuhl durchgeführten Projekten kennen.

Übung:

- Auffrischung von Grundlagen

Seminar:

- Vertiefung der Grundlagen

## Literatur:

Vorlesung:

- Die englischsprachigen, teilweise durch deutsche Texte ergänzten Unterlagen stehen auf der Studon-Plattform zur Verfügung.
- Umfangreiche englischsprachige Tafelanschrift

Übung:

- Ausführliche Tafelanschrift
- Unterlagen integriert in die Vorlesungsunterlagen

Seminar:

- Ausführliche Tafelanschrift
- Unterlagen integriert in die Vorlesungsunterlagen

## Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

### [1] Informatik (Bachelor of Science)

(Po-Vers. 2009s | Nebenfach | Nebenfach Chemie- und Bioingenieurwesen)

Dieses Modul ist daneben auch in den Studienfächern "Chemical Engineering - Nachhaltige Chemische Technologien (Master of Science)", "Chemie- und Bioingenieurwesen (Master of Science)", "Informatik (Master of Science)", "Life Science Engineering (Master of Science)" verwendbar.

## Studien-/Prüfungsleistungen:



Umweltbioverfahrenstechnik (Prüfungsnummer: 51601)

Untertitel: nur Erlangen

(englischer UntertitelErlangen only)

Prüfungsleistung, mündliche Prüfung, Dauer (in Minuten): 30

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Umweltbioverfahrenstechnik
- Übungen zu Umweltbioverfahrenstechnik

Erstablesung: SS 2019, 1. Wdh.: WS 2019/2020

1. Prüfer: Roman Breiter (100651)

---

**Organisatorisches:**

Die Vorlesung mit Übung wird in Deutsch und zusätzlich in Englischer Sprache gehalten. In beiden Fällen sind Begleitmaterial und die umfangreiche Tafelanschrift in Englischer Sprache.

**WICHTIG:** Der erste Termin für die Übungen in Deutsch und in Englisch wird in der ersten Vorlesungsstunde besprochen.

Für diese Veranstaltung ist zur besseren Erreichbarkeit der Studierenden eine Anmeldung via StudOn erwünscht.

**Bemerkungen:**

Das Modul "Environmental Biotechnology" am FAU campus Busan, Südkorea (Curriculum: [http://www.fau-busan.ac.kr/en/studies/studies\\_01.html](http://www.fau-busan.ac.kr/en/studies/studies_01.html)) ist ein Core/Specialisation Module (Vertiefungsmodul) und wird ausschließlich dort angeboten und geprüft!

---

**Modulbezeichnung:** Grundlagen der Elektrotechnik I (GETI) 7.5 ECTS  
 (Foundations of Electrical Engineering I)

Modulverantwortliche/r: Georg Fischer

Lehrende: Georg Fischer, Christopher Beck, Sarah Linz, N.N.

---

Startsemester: WS 2019/2020

Dauer: 1 Semester

Turnus: jährlich (WS)

Präsenzzeit: 90 Std.

Eigenstudium: 135 Std.

Sprache: Deutsch

---

**Lehrveranstaltungen:**

Grundlagen der Elektrotechnik I (WS 2019/2020, Vorlesung, 4 SWS, Georg Fischer)

Übungen zu Grundlagen der Elektrotechnik I (WS 2019/2020, Übung, 2 SWS, Christopher Beck et al.)

Tutorium zu Grundlagen der Elektrotechnik I (GET1) (WS 2019/2020, Tutorium, Christopher Beck)

---

**Inhalt:**

Diese Vorlesung bietet einen Einstieg in die physikalischen Grundlagen der Elektrotechnik und Elektronik. Ausgehend von beobachtbaren Kraftwirkungen zwischen Ladungen und zwischen Strömen wird der Begriff des elektrischen und magnetischen Feldes eingeführt. Mit den daraus abgeleiteten integralen Größen Spannung, Strom, Widerstand, Kapazität und Induktivität wird das Verhalten der passiven Bauelemente diskutiert. Am Beispiel der Gleichstromschaltungen werden die Methoden der Netzwerkanalyse eingeführt und Fragen nach Wirkungsgrad und Zusammenschaltung von Quellen untersucht. Einen Schwerpunkt bildet das Faraday'sche Induktionsgesetz und seine Anwendungen. Die Bewegungsinduktion wird im Zusammenhang mit den Drehstromgeneratoren betrachtet, die Ruheinduktion wird sehr ausführlich am Beispiel der Übertrager und Transformatoren diskutiert. Einen weiteren Schwerpunkt bildet die Behandlung zeitlich periodischer Vorgänge. Die komplexe Wechselstromrechnung bei sinusförmigen Strom- und Spannungsformen und die Fourieranalyse bei zeitlich periodischen nicht sinusförmigen Signalen werden ausführlich behandelt.

1. Physikalische Grundbegriffe
2. Das elektrostatische Feld
3. Das stationäre elektrische Strömungsfeld
4. Einfache elektrische Netzwerke
5. Stromleitungsmechanismen
6. Das stationäre Magnetfeld
7. Das zeitlich veränderliche elektromagnetische Feld
8. Wechselspannung und Wechselstrom
9. Zeitlich periodische Vorgänge beliebiger Kurvenform

**Lernziele und Kompetenzen:**

Nach der Teilnahme an den Modulveranstaltungen sind die Studierenden in der Lage:

1. den Begriff des Feldes zu verstehen, 2. Gleich- und Wechselstromschaltungen mit Widerständen, Kapazitäten, Induktivitäten und Transformatoren zu entwickeln, 3. Schwingkreise und Resonanzerscheinungen zu analysieren, 4. Energie- und Leistungsberechnungen durchzuführen, 5. Schaltungen zur Leistungsanpassung und zur Blindstromkompensation zu bewerten, 6. das Drehstromsystem zu verstehen.

**Literatur:**

Manfred Albach: Grundlagen der Elektrotechnik I und II, Pearson-Verlag

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

[1] **Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Nebenfach | Nebenfach Elektrotechnik, Elektronik und Informationstechnik | Allgemeine Elektrotechnik)

[2] **Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Nebenfach | Nebenfach Elektrotechnik, Elektronik und Informationstechnik | Elektrische Energie- und Antriebstechnik)

Dieses Modul ist daneben auch in den Studienfächern "Berufspädagogik Elektrotechnik und Informationstechnik (Bachelor of Education)", "Berufspädagogik Technik (Bachelor of Science)", "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Bachelor of Science)", "Elektrotechnik, Elektronik und Informationstechnik (Bachelor of Science)", "Energietechnik (Bachelor of Science)", "Informatik (Master of Science)", "Mathematik (Bachelor of Science)", "Mechatronik (Bachelor of Science)", "Medizintechnik (Bachelor of Science)", "Technomathematik (Bachelor of Science)" verwendbar.

---

### **Studien-/Prüfungsleistungen:**

Grundlagen der Elektrotechnik I (Prüfungsnummer: 25601)

(englische Bezeichnung: Lecture: Foundations of Electrical Engineering I)

Prüfungsleistung, Klausur, Dauer (in Minuten): 120

Anteil an der Berechnung der Modulnote: 100%

Erstablesung: WS 2019/2020, 1. Wdh.: SS 2020

1. Prüfer: Georg Fischer (100226)

---

---

**Modulbezeichnung:** Grundlagen der Elektrotechnik II (GET II) 5 ECTS  
 (Principles of Electrical Engineering II)

Modulverantwortliche/r: Klaus Helmreich  
 Lehrende: Klaus Helmreich

---

Startsemester: SS 2019	Dauer: 1 Semester	Turnus: jährlich (SS)
Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Deutsch

---

**Lehrveranstaltungen:**

Grundlagen der Elektrotechnik II (SS 2019, Vorlesung, 2 SWS, Klaus Helmreich)  
 Grundlagen der Elektrotechnik II Übung (SS 2019, Übung, 2 SWS, Gerald Gold)

---

**Empfohlene Voraussetzungen:**

- Grundlagen der Elektrotechnik 1
- Mathematik I
- Mathematik II (begleitend)

---

**Inhalt:**

Diese Vorlesung stellt den zweiten Teil einer 3-semesterigen Lehrveranstaltung über Grundlagen der Elektrotechnik für Studenten der Elektrotechnik, Elektronik und Informationstechnik im Grundstudium dar. Inhalt der Vorlesung ist die Analyse elektrischer Grundsaltungen und Netzwerke aus konzentrierten Bauelementen bei sinus- und nichtsinusförmiger harmonischer Erregung.

Nach kurzer Einführung in die komplexe Wechselstromrechnung und den Umgang mit elementaren elektrischen Bauelementen werden zunächst Spannungs- und Stromquellen und ihre Zusammenschaltung mit einer Last sowie die Leistungsübertragung von der Quelle zur Last betrachtet. Nach Herleitung und beispielhafter Anwendung von Methoden und Sätzen zur Berechnung und Vereinfachung elektrischer Schaltungen (Überlagerungssatz, Reziprozitätstheorem, äquivalente Schaltungen, Miller-Theorem etc.) werden zunächst 2-polige Netzwerke analysiert und in einem weiteren Kapitel dann allgemeine Verfahren zur Netzwerkanalyse wie das Maschenstromverfahren und das Knotenpotenzialverfahren behandelt. Die Berechnung der verallgemeinerten Eigenschaften von Zweipolfunktionen bei komplexen Frequenzen führt im verlustlosen Fall zur schnellen Vorhersagbarkeit des Frequenzverhaltens und zu elementaren Verfahren der Schaltungssynthese. Der nachfolgende Vorlesungsteil über mehrpolige Netzwerke konzentriert sich nach der Behandlung von allgemeinen Mehrtoren auf 2-Tore und ihr Verhalten, ihre verschiedenen Möglichkeiten der Zusammenschaltung und die zweckmäßige Beschreibung in verschiedenen Matrixdarstellungen (Impedanz-, Admittanz-, Ketten-, Hybridmatrix). Das Übertragungsverhalten von einfachen und verketteten Zweitoren wird am Beispiel gängiger Filterarten durchgesprochen und das Bode-Diagramm zur schnellen Übersichtsdarstellung eingeführt.

Nach allgemeiner Einführung der Fourierreihenentwicklung periodischer Signale wird die Darstellung von nicht sinusförmigen periodischen Erregungen von Netzwerken mittels reeller und komplexer Fourierreihen und die stationäre Reaktion der Netzwerke auf diese Erregung behandelt. Als mögliche Ursache für nichtsinusförmige Ströme und Spannungen in Netzwerken werden nichtlineare Zweipole mit ihren Kennlinienformen vorgestellt und auf die Berechnung des erzeugten Oberwellenspektrums eingegangen.

**Lernziele und Kompetenzen:**

Die Studierenden

- besitzen spezialisiertes und vertieftes Wissen über die Umformung, Analyse und Synthese von einfachen und umfangreicheren Netzwerken bei sinus- und nichtsinusförmiger Erregung in komplexer Darstellung.
- können die im Inhalt beschriebenen Verfahren und Methoden der Netzwerkanalyse erklären und auf Schaltungsbeispiele anwenden.
- können Verfahren der Netzwerkanalyse hinsichtlich des Rechenaufwandes beurteilen und vergleichen.

**Literatur:**

Elektrotechnik, Albach, M., 2011.

Grundlagen der Elektrotechnik - Netzwerke, Schmidt, L.-P., Schaller, G., Martius, S., 2013.

(bisher: Grundlagen der Elektrotechnik 3, Schmidt, L.-P., Schaller, G., Martius, S., 2006.

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

**[1] Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Nebenfach | Nebenfach Elektrotechnik, Elektronik und Informationstechnik | Allgemeine Elektrotechnik)

**[2] Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Nebenfach | Nebenfach Elektrotechnik, Elektronik und Informationstechnik | Elektrische Energie- und Antriebstechnik)

Dieses Modul ist daneben auch in den Studienfächern "Berufspädagogik Elektrotechnik und Informationstechnik (Bachelor of Education)", "Berufspädagogik Technik (Bachelor of Science)", "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Bachelor of Science)", "Elektrotechnik, Elektronik und Informationstechnik (Bachelor of Science)", "Energietechnik (Bachelor of Science)", "Informatik (Master of Science)", "Mechatronik (Bachelor of Science)", "Medizintechnik (Bachelor of Science)", "Technomathematik (Bachelor of Science)" verwendbar.

---

**Studien-/Prüfungsleistungen:**

Grundlagen der Elektrotechnik II (Prüfungsnummer: 25701)

(englische Bezeichnung: Foundations of Electrical Engineering II)

Prüfungsleistung, Klausur, Dauer (in Minuten): 90

Anteil an der Berechnung der Modulnote: 100%

Erstablingung: SS 2019, 1. Wdh.: WS 2019/2020

1. Prüfer: Klaus Helmreich (100222)

---

**Organisatorisches:**

Siehe UniVIS-Eintrag der zugeordneten Lehrveranstaltungen!

**Modulbezeichnung:** Praktikum Grundlagen der Elektrotechnik für EEI, ME, BP, 2.5 ECTS  
INF, MATH (PR GET EEI /ME/BP/INF/MATH)

(Laboratory: Fundamentals of Electrical Engineering for EEI, ME,  
BP, INF, MATH)

Modulverantwortliche/r: Christopher Beck

Lehrende: Georg Fischer, Lorenz-Peter Schmidt, Reinhard Lerch

Startsemester: SS 2019

Dauer: 3 Semester

Turnus: halbjährlich (WS+SS)

Präsenzzeit: 36 Std.

Eigenstudium: 39 Std.

Sprache: Deutsch

### Lehrveranstaltungen:

Praktikum Grundlagen der Elektrotechnik I (EEI, ME, ET, BPT) (SS 2019, Praktikum, 1 SWS, Torsten Reißland)

Praktikum Grundlagen der Elektrotechnik II (EEI) (WS 2019/2020, Praktikum, 1 SWS, Jan Schür)

Praktikum Grundlagen der Elektrotechnik III (SS 2019, Praktikum, 1 SWS, N.N.)

### Empfohlene Voraussetzungen:

Voraussetzung:

- *Grundlagen der Elektrotechnik I*
- *Grundlagen der Elektrotechnik II*
- *Grundlagen der Elektrotechnik III*

### Inhalt:

Im Rahmen des Praktikums GET I werden 4 Versuche zu den folgenden Themen durchgeführt:

1. Wickelkondensator
2. Magnetfeldmessung
3. Transformator
4. Schwingkreis

Im Rahmen des Praktikums GET II werden 4 Versuche zu den folgenden Themen durchgeführt:

1. Ohmsche Netze; Zweitore
2. Quelle und Last; reaktiver Zweipol; Bode-Diagramm
3. Schaltungssimulation
4. Nichtsinusförmige periodische Signale und Fourierreihen

Im Rahmen des Praktikums GET III werden 4 Versuche zu den folgenden Themen durchgeführt:

1. Einschwingvorgänge
2. nichtlineare Netzwerke
3. Messschaltungen
4. Brückenschaltung

Die Dauer der einzelnen Versuche entspricht etwa der Dauer von 3-4 Vorlesungsstunden. Nähere Informationen zur Anmeldung und zur Gruppeneinteilung sind im Sekretariat des Lehrstuhls erhältlich bzw. werden am Ende der VL Grundlagen I besprochen.

Für die erfolgreiche Teilnahme an den Versuchen wird ein Schein ausgestellt.

### Lernziele und Kompetenzen:

Nach der Teilnahme an den Modulveranstaltungen sind die Studierenden in der Lage:

- Messaufbauten mit den grundlegenden Messgeräten wie z.B. Multimeter, Sinusgenerator, Oszilloskop sowie deren Bedienung zu verstehen,
- den inneren Aufbau von Kondensatoren und Transformatoren zu analysieren, indem sie einen Kondensator und einen Transformator selber herstellen,
- einfache Schaltungen messtechnisch zu analysieren und deren Verhalten zu verstehen,
- durch einen Vergleich von gemessenen und berechneten Ergebnissen den Einfluss von parasitären Eigenschaften zu verstehen,
- den grundlegenden Umgang mit nichtsinusförmigen periodischen Signalen zu verstehen.

### Literatur:

- Unterlagen zur Vorlesung *Grundlagen der Elektrotechnik I*

- Unterlagen zur Vorlesung *Grundlagen der Elektrotechnik II*
- R. Lerch: Elektrische Messtechnik, Springer, 5. Auflage
- Versuchsbeschreibungen

---

### Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

#### [1] Informatik (Bachelor of Science)

(Po-Vers. 2009s | Nebenfach | Nebenfach Elektrotechnik, Elektronik und Informationstechnik | Allgemeine Elektrotechnik)

Dieses Modul ist daneben auch in den Studienfächern "Berufspädagogik Elektrotechnik und Informationstechnik (Bachelor of Education)", "Elektrotechnik, Elektronik und Informationstechnik (Bachelor of Science)", "Informatik (Master of Science)", "Mathematik (Bachelor of Science)", "Mechatronik (Bachelor of Science)" verwendbar.

---

### Studien-/Prüfungsleistungen:

Praktikum Grundlagen der Elektrotechnik (Prüfungsnummer: 26201)

Studienleistung, Praktikumsleistung

Zugeordnete Lehrveranstaltungen:

- Praktikum Grundlagen der Elektrotechnik I (EEI, ME, ET, BPT)
- Praktikum Grundlagen der Elektrotechnik II (EEI)

•

weitere Erläuterungen:

Für jeden Versuch im Rahmen der drei Teilpraktika wird ein Testat erteilt. Die Studienleistung ist bestanden, wenn alle Testate vollständig vorliegen. Der Schein wird von Frau Konhäuser am Lehrstuhl EMF erstellt. PR GET I,II und III als Praktikumsleistung für 2,5 ECTS

Erstablingung: SS 2019, 1. Wdh.: WS 2019/2020

1. Prüfer: Georg Fischer (100226)

---

### Organisatorisches:

Die Anmeldung zum Praktikum GET I erfolgt über StudOn.

**Bitte die "Hinweise zum Praktikum" auf der Homepage des Lehrstuhls (LEMF) herunterladen und zur Anmeldung mitbringen!**

Die Anmeldung zum Praktikum GET II erfolgt über StudOn.

**Bitte die "Hinweise zum Praktikum" auf der Homepage des Lehrstuhls (LHFT) beachten!**

---

**Modulbezeichnung:** Grundlagen der Elektrotechnik III (GET III) 5 ECTS  
 (Fundamentals of Electrical Engineering III)

Modulverantwortliche/r: Stefan J. Rupitsch  
 Lehrende: Stefan J. Rupitsch

---

Startsemester: WS 2019/2020	Dauer: 1 Semester	Turnus: jährlich (WS)
Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Deutsch

---

**Lehrveranstaltungen:**

Grundlagen der Elektrotechnik III (WS 2019/2020, Vorlesung, 2 SWS, Dominik Gedeon)  
 Übungen zu Grundlagen der Elektrotechnik III (WS 2019/2020, Übung, 2 SWS, Philipp Dorsch)

---

**Empfohlene Voraussetzungen:**

Grundlagen der Elektrotechnik I und II

---

**Inhalt:**

- Umfang und Bedeutung der elektrischen Messtechnik
- Die Grundlagen des Messens
- Fourier-Transformation
- Laplace-Transformation
- Netzwerkanalyse im Zeit- und Laplace-Bereich
- Übertragungsfunktion und Bode-Diagramm
- Nichtlineare Bauelemente, Schaltungen und Systeme
- Operationsverstärker
- Messverstärker
- Messfehler
- Messung von Gleichstrom und Gleichspannung
- Ausschlagbrücken
- Abgleichbrücken, Messung von elektrischen Impedanzen

**Lernziele und Kompetenzen:**

Die Studierenden

- ordnen die behandelten Verfahren gemäß ihrer Eignung für spezifische Probleme (Zeit-/Frequenzbereich, Linear/Nichtlinear) ein.
- wählen geeignete Verfahren zur Analyse elektrischer Netzwerke aus und wenden diese an.
- interpretieren die Ergebnisse und zeigen Zusammenhänge zwischen den Lösungsverfahren auf.
- kennen einfache Grundsaltungen mit Operationsverstärkern und sind in der Lage, diese zu analysieren.
- kennen die behandelten Messschaltungen und ihre Einsatzmöglichkeiten.
- analysieren Brückenschaltungen.
- wenden grundlegende Konzepte der Messfehlerrechnung auf Messschaltungen an.
- reflektieren selbstständig den eigenen Lernprozess und nutzen die Präsenzzeit zur Klärung der erkannten Defizite.

**Literatur:**

Lehrbuch: „Elektrische Messtechnik“, R. Lerch, 7. Aufl. 2016, Springer-Verlag  
 Übungsbuch: „Elektrische Messtechnik - Übungen“, R. Lerch, M. Kaltenbacher, F. Lindinger, A. Sutor, 2. Aufl. 2005, Springer-Verlag

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

[1] **Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Nebenfach | Nebenfach Elektrotechnik, Elektronik und Informationstechnik | Automatisierungstechnik)



Dieses Modul ist daneben auch in den Studienfächern "Berufspädagogik Elektrotechnik und Informationstechnik (Bachelor of Education)", "Berufspädagogik Technik (Bachelor of Science)", "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Bachelor of Science)", "Elektrotechnik, Elektronik und Informationstechnik (Bachelor of Science)", "Informatik (Master of Science)", "Mechatronik (Bachelor of Science)", "Medizintechnik (Bachelor of Science)", "Technomathematik (Bachelor of Science)" verwendbar.

---

**Studien-/Prüfungsleistungen:**

Grundlagen der Elektrotechnik III (Prüfungsnummer: 25801)

Prüfungsleistung, Klausur, Dauer (in Minuten): 90

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Grundlagen der Elektrotechnik III
- Übungen zu Grundlagen der Elektrotechnik III

Erstablingung: WS 2019/2020, 1. Wdh.: SS 2020

1. Prüfer: Reinhard Lerch (100217)

---

---

**Modulbezeichnung:** Einführung in die Systemtheorie (EST) 5 ECTS  
 (Introduction to System Theory)

Modulverantwortliche/r: Knut Graichen

Lehrende: Knut Graichen, wissenschaftliche Mitarbeiter/innen

---

Startsemester: SS 2019

Dauer: 1 Semester

Turnus: jährlich (SS)

Präsenzzeit: 60 Std.

Eigenstudium: 90 Std.

Sprache: Deutsch

---

**Lehrveranstaltungen:**

Einführung in die Systemtheorie (SS 2019, Vorlesung, 2 SWS, Knut Graichen)

Übungen zu Einführung in die Systemtheorie (SS 2019, Übung, 2 SWS, Alexander Lomakin et al.)

---

**Empfohlene Voraussetzungen:**

Kenntnis der Laplace-Transformation

---

**Inhalt:**

Die Systemtheorie stellt mathematische Methoden zur Beschreibung und Analyse von Systemen und Signalen bereit.

In der Vorlesung wird die große Klasse der linearen und zeitinvarianten Systeme betrachtet - und zwar sowohl mit zeitkontinuierlichen als auch mit zeitdiskreten Ein- und Ausgangssignalen. Behandelt werden sowohl die Ein-Ausgangsbeschreibung (mittels Ein-/ Ausgangsdifferentialgleichung, Übertragungsfunktion, Gewichtsfunktion oder Sprungantwort bzw. mittels Ein-/Ausgangsdifferenzgleichung, z-Übertragungsfunktion oder Gewichtsfolge) als auch die Zustandsbeschreibung (allgemein, in Diagonal- und in Regelungsnormalfunktion) sowie grundlegende Systemeigenschaften (Ein-/Ausgangsstabilität, Zustandsstabilität, Steuer- und Beobachtbarkeit, Lösung der Zustandsgleichungen mittels Transitionsmatrix).

**Lernziele und Kompetenzen:**

Die Studierenden können

- Gegenstand und Zielstellung der Systemtheorie erläutern.
- Systeme anhand der Signalarten und der Systemoperatoreigenschaften klassifizieren.
- die Systemoperatoreigenschaften Dynamik, Kausalität, Linearität und Zeitinvarianz definieren sowie gegebene Systeme auf diese Eigenschaften hin untersuchen.
- mit Laplace und z-Transformation umgehen sowie bei linearen, zeitinvarianten Systemen (LZI-Systeme) zur Anwendung bringen.
- das Ein-/Ausgangsverhalten von LZI-Systemen im zeitkontinuierlichen Fall mittels Ein-/Ausgangsdifferentialgleichung, Übertragungsfunktion, Gewichtsfunktion und Sprungantwort bzw. im zeitdiskreten Fall mittels Ein-/Ausgangsdifferenzgleichung, z-Übertragungsfunktion und Gewichtsfolge beschreiben.
- die Zusammenhänge zwischen den unterschiedlichen Ein-/Ausgangsbeschreibungen aufzeigen.
- die Ein-/Ausgangsstabilität definieren, für Gewichtsfunktion bzw. Gewichtsfolge, Übertragungsfunktion und Sprungantwort von LZI-Systemen gültige Kriterien hierfür (inkl. Hurwitz-Kriterium) angeben und damit konkrete Systeme auf Ein-/Ausgangsstabilität überprüfen.
- das Verhalten von zeitkontinuierlichen und zeitdiskreten LZI-Systemen durch Zustandsgleichungen beschreiben, diese in Regelungsnormalform und in Diagonalform aufstellen sowie die Transformation auf Diagonalform bestimmen und ausführen.
- die vollständige Steuerbarkeit und vollständige Beobachtbarkeit dynamischer Systeme definieren, die Kriterien nach Kalman und Gilbert hierfür formulieren und damit konkrete Systeme auf diese Eigenschaften hin untersuchen.
- die Auswirkungen von Steuerbarkeits- bzw. Beobachtbarkeitsdefekten auf das Systemein-/ausgangsverhalten beschreiben und die Folgerungen hieraus, insbesondere für das Stabilitätsverhalten, formulieren.
- die allgemeine Lösung der Zustandsgleichungen von LZI-Systemen angeben und diese für gegebene Systeme berechnen.
- den Zusammenhang zwischen zeitkontinuierlicher und zeitdiskreter Zustandsbeschreibung eines LZI-

Systems herstellen.

- die asymptotische Zustandsstabilität definieren, das für LZI-Systeme gültige Eigenwertkriterium hierfür im zeitkontinuierlichen sowie im zeitdiskreten Fall angeben und damit die Zustandsstabilität gegebener Systeme überprüfen.
- die Vorlesungsinhalte auf verwandte Problemstellungen übertragen sowie sich weiterführende Methoden der Systemtheorie linearer, zeitinvarianter Systeme selbständig erschließen.

**Literatur:**

Eine Literaturübersicht wird in der ersten Vorlesung gegeben.

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

**[1] Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Nebenfach | Nebenfach Elektrotechnik, Elektronik und Informationstechnik | Automatisierungstechnik)

Dieses Modul ist daneben auch in den Studienfächern "Berufspädagogik Elektrotechnik und Informationstechnik (Bachelor of Education)", "Berufspädagogik Technik (Bachelor of Science)", "Energietechnik (Master of Science)", "Informatik (Master of Science)", "Mechatronik (Bachelor of Science)" verwendbar.

---

**Studien-/Prüfungsleistungen:**

Einführung in die Systemtheorie (Prüfungsnummer: 50001)

Prüfungsleistung, Klausur, Dauer (in Minuten): 90

Anteil an der Berechnung der Modulnote: 100%

Erstablingung: SS 2019, 1. Wdh.: WS 2019/2020

1. Prüfer: Knut Graichen (100423)

---

---

**Modulbezeichnung:** Regelungstechnik A (Grundlagen) (RT A) 5 ECTS  
 (Control System Design A (Fundamentals))

Modulverantwortliche/r: Knut Graichen  
 Lehrende: Knut Graichen

---

Startsemester: WS 2019/2020	Dauer: 1 Semester	Turnus: jährlich (WS)
Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Deutsch

---

**Lehrveranstaltungen:**

Regelungstechnik A (Grundlagen) (WS 2019/2020, Vorlesung, 2 SWS, Knut Graichen)  
 Übungen zu Regelungstechnik A (Grundlagen) (WS 2019/2020, Übung, 2 SWS, Jakob Gabriel)

---

**Empfohlene Voraussetzungen:**

Systemtheorie linearer zeitkontinuierlicher Systeme (inkl. Laplace-Transformation)

---

**Inhalt:**

- Gegenstand und Zielstellung der Regelungstechnik
- Modellbildung der Strecke im Zeit und Frequenzbereich und Darstellung als Strukturbild
- Analyse des Streckenverhaltens linearer Eingrößensysteme anhand von Übertragungsfunktion und Frequenzgang
- Auslegung einschleifiger Regelkreise
- Erweiterte Regelkreisstrukturen

**Lernziele und Kompetenzen:**

Die Studierenden können

- Gegenstand und Zielstellung der Regelungstechnik erläutern.
- Problemstellungen als Steuerungs- und Regelungsaufgabe identifizieren.
- das Streckenverhalten durch ein mathematisches Modell in Form des Strukturbilds beschreiben.
- eine Modellvereinfachung durch Linearisierung und Strukturbildumformung durchführen.
- aus Übertragungsfunktion und Frequenzgang das qualitative Streckenverhalten ermitteln.
- zu einem Frequenzgang Ortskurve und Bode-Diagramm angeben.
- den Aufbau einer Zwei-Freiheitsgrade-Regelung angeben und die Zweckbestimmung von Vorsteuerung und Regelung erläutern.
- Sollverläufe auf Zulässigkeit überprüfen und realisierbare Vorsteuerungen entwerfen.
- die Regelkreis-Stabilität definieren und mit dem Nyquist-Kriterium untersuchen.
- entscheiden, wann welcher Reglertyp in Frage kommt und nach welchen Gesichtspunkten dessen Parameter zu wählen sind.
- für lineare Eingrößensysteme einen geeigneten Regler entwerfen.
- ergänzende Maßnahmen zur Störverhaltensverbesserung beschreiben und zur Anwendung bringen.
- die Vorlesungsinhalte auf verwandte Problemstellungen übertragen und sich weiterführende Frequenzbereichsmethoden der Regelungstechnik selbständig erschließen.

**Literatur:**

Eine Literaturübersicht wird in der Vorlesung gegeben.

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

[1] **Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Nebenfach | Nebenfach Elektrotechnik, Elektronik und Informationstechnik | Automatisierungstechnik)

Dieses Modul ist daneben auch in den Studienfächern "Berufspädagogik Elektrotechnik und Informationstechnik (Bachelor of Education)", "Berufspädagogik Technik (Bachelor of Science)", "Elektrotechnik, Elektronik und Informationstechnik (Bachelor of Science)", "Informatik (Master of Science)", "Mechatronik (Bachelor of Science)", "Medizintechnik (Master of Science)", "Technomathematik (Bachelor of Science)", "Wirtschaftsingenieurwesen (Bachelor of Science)", "Wirtschaftsingenieurwesen (Master of Science)" verwendbar.

**Studien-/Prüfungsleistungen:**

Vorlesung Regelungstechnik A (Grundlagen) (Prüfungsnummer: 26501)

(englische Bezeichnung: Lecture: Control Engineering A (Foundations))

Prüfungsleistung, Klausur, Dauer (in Minuten): 90

Anteil an der Berechnung der Modulnote: 100%

Erstablesung: WS 2019/2020, 1. Wdh.: SS 2020, 2. Wdh.: WS 2020/2021

1. Prüfer: Knut Graichen (100423)

---

**Bemerkungen:**

Für Studienbeginner im SS 2011 und SS 2012 des Studiengangs EEI findet Regelungstechnik A im 4. FS statt.

---

**Modulbezeichnung:** Signale und Systeme I (SISY I) 5 ECTS  
 (Signals and Systems I)

Modulverantwortliche/r: André Kaup  
 Lehrende: André Kaup, Jürgen Seiler

---

Startsemester: WS 2019/2020	Dauer: 1 Semester	Turnus: jährlich (WS)
Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Deutsch

---

**Lehrveranstaltungen:**

Signale und Systeme I (WS 2019/2020, Vorlesung, 2,5 SWS, André Kaup)  
 Übung zu Signale und Systeme I (WS 2019/2020, Übung, 1,5 SWS, Jürgen Seiler)  
 Tutorium zu Signale und Systeme I (WS 2019/2020, Tutorium, 1 SWS, Simon Grosche)

---

**Empfohlene Voraussetzungen:**

Modul „Grundlagen der Elektrotechnik I+II“ **oder** Module „Einführung in die luK“ sowie „Elektronik und Schaltungstechnik“

---

**Inhalt:**

**Kontinuierliche Signale**

Elementare Operationen, Delta-Impuls, Energie und Leistung, Skalarprodukt und Orthogonalität, Faltung und Korrelation

**Fourier-Transformation**

Definition, Symmetrien, inverse Transformation, Sätze und Korrespondenzen

**Laplace-Transformation**

Definition, Eigenschaften und Sätze, Inverse Transformation, Korrespondenzen

**Kontinuierliche LTI-Systeme im Zeitbereich**

Impulsantwort, Sprungantwort, Beschreibung durch Differentialgleichungen, Direktformen, Zustandsraumdarstellung, äquivalente Zustandsraumdarstellungen, Transformation auf Diagonalfom

**Kontinuierliche LTI-Systeme im Frequenzbereich**

Eigenfunktionen, Systemfunktion und Übertragungsfunktion, Verkettung von LTI-Systemen, Zustandsraumbeschreibung im Frequenzbereich

**Kontinuierliche LTI-Systeme mit Anfangsbedingungen**

Lösung mit der Laplace-Transformation, Lösung über die Zustandsraumbeschreibung, Zusammenhang zwischen Anfangswert und Anfangszustand

**Kontinuierliche LTI-Systeme mit speziellen Übertragungsfunktionen**

Reellwertige Systeme, verzerrungsfreie Systeme, linearphasige Systeme, minimalphasige Systeme und Allpässe, idealer Tiefpass und idealer Bandpass

**Kausalität und Hilbert-Transformation**

Kausale kontinuierliche LTI-Systeme, Hilbert-Transformation, analytisches Signal

**Stabilität und rückgekoppelte Systeme**

Übertragungsstabilität, kausale stabile kontinuierliche LTI-Systeme, Stabilitätskriterium von Hurwitz, rückgekoppelte Systeme

**Abtastung und periodische Signale**

Delta-Impulskamm und seine Fourier-Transformierte, Fourier-Transformierte periodischer Signale, Abtasttheorem, ideale und nichtideale Abtastung und Rekonstruktion, Abtastung im Frequenzbereich

**Lernziele und Kompetenzen:**

**Die Studierenden**

- analysieren kontinuierliche Signale mit Hilfe der Fourier- und Laplace-Transformation
- bestimmen die Impulsantwort, Direktformen und Zustandsraumdarstellung für kontinuierliche lineare zeitinvariante Systeme
- berechnen System- und Übertragungsfunktionen für kontinuierliche lineare zeitinvariante Systeme
- analysieren die Eigenschaften von kontinuierlichen linearen zeitinvarianten Systemen aufgrund der Zeit- und Frequenzbereichsbeschreibung

- stufen kontinuierliche lineare zeitinvariante Systeme an-hand ihrer Eigenschaften Verzerrungsfreiheit, Linearphasigkeit und Minimalphasigkeit ein
- bewerten Kausalität und Stabilität von kontinuierlichen linearen zeitinvarianten Systemen
- beurteilen die Effekte und Grenzen einer Abtastung von kontinuierlichen Signalen

**Literatur:**

B. Girod, R. Rabenstein, A. Stenger, „Einführung in die Systemtheorie“, Wiesbaden: Teubner-Verlag, 2005

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

**[1] Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Nebenfach | Nebenfach Elektrotechnik, Elektronik und Informationstechnik | Informationstechnik)

Dieses Modul ist daneben auch in den Studienfächern "Berufspädagogik Technik (Bachelor of Science)", "Berufspädagogik Technik (Master of Education)", "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Bachelor of Science)", "Elektrotechnik, Elektronik und Informationstechnik (Bachelor of Science)", "Energietechnik (Master of Science)", "Informatik (Master of Science)", "Informations- und Kommunikationstechnik (Bachelor of Science)", "Mathematik (Bachelor of Science)", "Medizintechnik (Bachelor of Science)", "Technomathematik (Bachelor of Science)", "Wirtschaftsingenieurwesen (Bachelor of Science)" verwendbar.

---

**Studien-/Prüfungsleistungen:**

Signale und Systeme I (Prüfungsnummer: 26801)

(englische Bezeichnung: Signals and Systems I)

Prüfungsleistung, Klausur, Dauer (in Minuten): 90

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Signale und Systeme I
- Übung zu Signale und Systeme I
- Tutorium zu Signale und Systeme I

Erstablesung: WS 2019/2020, 1. Wdh.: SS 2020

1. Prüfer: André Kaup (100219)

---

---

**Modulbezeichnung:** Signale und Systeme II (SISY II) 5 ECTS  
 (Signals and Systems II)

Modulverantwortliche/r: André Kaup

Lehrende: André Kaup, Christian Herglotz, Fabian Brand

---

Startsemester: SS 2019

Dauer: 1 Semester

Turnus: jährlich (SS)

Präsenzzeit: 60 Std.

Eigenstudium: 90 Std.

Sprache: Deutsch

---

**Lehrveranstaltungen:**

Signale und Systeme II (SS 2019, Vorlesung, 2,5 SWS, André Kaup)

Übung zu Signale und Systeme II (SS 2019, Übung, 1,5 SWS, Christian Herglotz)

Tutorium zu Signale und Systeme II (SS 2019, Tutorium, 1 SWS, Nils Genser)

---

**Inhalt:**

**Diskrete Signale**

Elementare Operationen und Eigenschaften, spezielle diskrete Signale, Energie und Leistung, Skalarprodukt und Orthogonalität, Faltung und Korrelation

**Zeitdiskrete Fourier-Transformation (DTFT)**

Definition, Beispiele, Korrespondenzen, inverse zeitdiskrete Fourier-Transformation, Eigenschaften und Sätze

**Diskrete Fourier-Transformation (DFT)**

Definition, Beispiele, Korrespondenzen, Eigenschaften und Sätze, Faltung mittels der diskreten Fourier-Transformation, Matrixschreibweise, schnelle Fourier-Transformation (FFT)

**z-Transformation**

Definition, Beispiele, Korrespondenzen, inverse z-Transformation, Eigenschaften und Sätze

**Diskrete LTI-Systeme im Zeitbereich**

Beschreibung durch Impulsantwort und Faltung, Beschreibung durch Differenzgleichungen, Beschreibung durch Zustandsraumdarstellung

**Diskrete LTI-Systeme im Frequenzbereich**

Eigenfolgen, Systemfunktion und Übertragungsfunktion, Verkettung von LTI-Systemen, Zustandsraumbeschreibung im Frequenzbereich

**Diskrete LTI-Systeme mit speziellen Übertragungsfunktionen**

Reellwertige Systeme, verzerrungsfreie Systeme, linearphasige Systeme, minimalphasige Systeme und Allpässe, idealer Tiefpass und ideale Bandpässe, idealer Differenzierer

**Kausale diskrete LTI-Systeme und Hilbert-Transformation**

Kausale diskrete LTI-Systeme, Hilbert-Transformation für periodische Spektren, analytisches Signal und diskreter Hilbert-Transformator

**Stabilität diskreter LTI-Systeme**

BIBO-Stabilität, kausale stabile diskrete Systeme, Stabilitätskriterium für Systeme N-ter Ordnung

**Beschreibung von Zufallssignalen**

Erwartungswerte, stationäre und ergodische Zufallsprozesse, Autokorrelations- und Korrelationsfunktion, Leistungsdichtespektrum, komplexwertige Zufallssignale

**Zufallssignale und LTI-Systeme**

Verknüpfung von Zufallssignalen, Reaktion von LTI-Systemen auf Zufallssignale, Wienerfilter

**Lernziele und Kompetenzen:**

Die Studierenden

- analysieren diskrete Signale mit Hilfe der zeitdiskreten Fourier-Transformation und berechnen deren diskrete Fourier-Transformation
- bestimmen die Impulsantwort, Direktformen und Zustandsraumdarstellung für diskrete lineare zeitinvariante Systeme
- berechnen System- und Übertragungsfunktionen für diskrete lineare zeitinvariante Systeme
- analysieren die Eigenschaften von diskreten linearen zeitinvarianten Systemen aufgrund der Zeit- und Frequenzbereichsbeschreibung



- stufen diskrete lineare zeitinvariante Systeme anhand ihrer Eigenschaften Verzerrungsfreiheit, Linearphasigkeit und Minimalphasigkeit ein
- bewerten Kausalität und Stabilität von diskreten linearen zeitinvarianten Systemen
- bewerten diskrete Zufallssignale durch Berechnung von Erwartungswerten und Korrelationsfunktionen
- beurteilen die wesentlichen Effekte einer Filterung von diskreten Zufallssignalen durch diskrete lineare zeitinvariante Systeme

---

### Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

#### [1] Informatik (Bachelor of Science)

(Po-Vers. 2009s | Nebenfach | Nebenfach Elektrotechnik, Elektronik und Informationstechnik | Informationstechnik)

Dieses Modul ist daneben auch in den Studienfächern "Berufspädagogik Technik (Bachelor of Science)", "Berufspädagogik Technik (Master of Education)", "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Bachelor of Science)", "Elektrotechnik, Elektronik und Informationstechnik (Bachelor of Science)", "Energietechnik (Master of Science)", "Informatik (Master of Science)", "Informations- und Kommunikationstechnik (Bachelor of Science)", "Mathematik (Bachelor of Science)", "Medizintechnik (Bachelor of Science)", "Medizintechnik (Master of Science)", "Technomathematik (Bachelor of Science)", "Wirtschaftsingenieurwesen (Bachelor of Science)" verwendbar.

---

### Studien-/Prüfungsleistungen:

Signale und Systeme II (Prüfungsnummer: 26802)

(englische Bezeichnung: Signals and Systems II)

Prüfungsleistung, Klausur, Dauer (in Minuten): 90

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Signale und Systeme II
- Übung zu Signale und Systeme II

Erstablingung: SS 2019, 1. Wdh.: WS 2019/2020

1. Prüfer: André Kaup (100219)

---

**Modulbezeichnung:** Information Theory and Coding (ITC) **5 ECTS**  
(Information Theory and Coding)

Modulverantwortliche/r: Ralf Müller  
Lehrende: Ralf Müller

Startsemester: WS 2019/2020      Dauer: 1 Semester      Turnus: halbjährlich (WS+SS)  
Präsenzzeit: 60 Std.      Eigenstudium: 90 Std.      Sprache: Englisch

**Lehrveranstaltungen:**

Information Theory and Coding (WS 2019/2020, Vorlesung, 3 SWS, Ralf Müller)  
Tutorial for Information Theory and Coding (WS 2019/2020, Übung, 1 SWS, Ali Bereyhi)

**Inhalt:**

1. Introduction: binomial distribution, (7,4)-Hamming code, parity-check matrix, generator matrix
2. Probability, entropy, and inference: entropy, conditional probability, Bayes' law, likelihood, Jensen's inequality
3. Inference: inverse probability, statistical inference
4. The source coding theorem: information content, typical sequences, Chebychev inequality, law of large numbers
5. Symbol codes: unique decidability, expected codeword length, prefix-free codes, Kraft inequality, Huffman coding
6. Stream codes: arithmetic coding, Lempel-Ziv coding, Burrows-Wheeler transform
7. Dependent random variables: mutual information, data processing lemma
8. Communication over a noisy channel: discrete memory-less channel, channel coding theorem, channel capacity
9. The noisy-channel coding theorem: jointly-typical sequences, proof of the channel coding theorem, proof of converse, symmetric channels
10. Error-correcting codes and real channels: AWGN channel, multivariate Gaussian pdf, capacity of AWGN channel
11. Binary codes: minimum distance, perfect codes, why perfect codes are bad, why distance isn't everything
12. Message passing: distributed counting, path counting, low-cost path, min-sum (=Viterbi) algorithm
13. Exact marginalization in graphs: factor graphs, sum-product algorithm
14. Low-density parity-check codes: density evolution, check node degree, regular vs. irregular codes, girth
15. Lossy source coding: transform coding and JPEG compression

- 
1. Einleitung: Binomialverteilung, (7,4)-Hamming-Code, Paritätsmatrix, Generatormatrix
  2. Wahrscheinlichkeit, Entropie und Inferenz: Entropie, bedingte Wahrscheinlichkeit, Bayes'sches Gesetz, Likelihood, Jensen'sche Ungleichung
  3. Inferenz: Inverse Wahrscheinlichkeit, statistische Inferenz
  4. Das Quellencodierungstheorem: Informationsgehalt, typische Folgen, Tschebyschev'sche Ungleichung, Gesetz der großen Zahlen
  5. Symbolcodes: eindeutige Dekodierbarkeit, mittlere Codewortlänge, präfixfreie Codes, Kraft'sche Ungleichung, Huffmancodierung
  6. Stromcodes: arithmetische Codierung, Lempel-Ziv-Codierung, Burrows-Wheeler-Transformation
  7. Abhängige Zufallsvariablen: Transinformation, Datenverarbeitungslemma
  8. Kommunikation over gestörte Kanäle: diskreter gedächtnisloser Kanal, Kanalcodierungstheorem, Kanalkapazität
  9. Das Kanalcodierungstheorem: verbundtypische Folgen, Beweis des Kanalcodierungstheorems, Beweis des Umkehrsatzes, symmetrische Kanaäle
  10. Fehlerkorrigierende Codes und reale Kanäle: AWGN-Kanal, mehrdimensionale Gauß'sche WDF, Kapazität des AWGN-Kanals

11. Binäre Codes: Minimaldistanz, perfekte Codes, Warum perfekte Codes schlecht sind, Warum Distanz nicht alles ist
12. Nachrichtenaustausch: verteiltes Zählen, Pfadzählen, günstigster Pfad, Minimumsummenalgorithmus
13. Exakte Marginalisierung in Graphen: Faktorgraph, Summenproduktalgorithmus
14. LDPC-Codes: Dichteevolution, Knotenordnung, reguläre und irreguläre Codes, Graphumfang
15. Verlustbehaftete Quellencodierung: Transformationscodierung und JPEG-Kompression

### Lernziele und Kompetenzen:

The students apply Bayesian inference to problems in both communications and everyday's life.  
 The students explain the concept of digital communications by means of source compression and forward-error correction coding.  
 For the design of communication systems, they use the concepts of entropy and channel capacity.  
 They calculate these quantities for memoryless sources and channels.  
 The students proof both the source coding and the channel coding theorem.  
 The students compare various methods of source coding with respect to compression rate and complexity.  
 The students apply source compression methods to measure mutual information.  
 The students factorize multivariate functions, represent them by graphs, and marginalize them with respect to various variables.  
 The students explain the design of error-correcting codes and the role of minimum distance.  
 They decode error-correcting codes by means of maximum-likelihood decoding and message passing.  
 The students apply distributed algorithms to problems in both communications and everyday's life.  
 The students improve the properties of low-density parity-check codes by widening the girth and/or irregularity in the degree distribution.  
 The students transform source images into the frequency domain to improve lossy compression.

–

Die Studierenden wenden Bayes'sche Inferenz auf Probleme in der Nachrichtentechnik und im Alltagsleben an.  
 Die Studierenden erklären die konzeptuelle Trennung von digitaler Übertragung in Quellen- und Kanalcodierung.  
 Kommunikationssysteme entwerfen sie unter Betrachtung von Entropie und Kanalkapazität.  
 Sie berechnen diese Größen für gedächtnislose Quellen und Kanäle.  
 Die Studierenden beweisen sowohl das Quellen- als auch das Kanalcodierungstheorem.  
 Die Studierenden vergleichen verschiedenartige Quellencodierungsverfahren hinsichtlich Komplexität und Kompressionsrate.  
 Die Studierenden verwenden Quellencodierverfahren zur Messung von Transinformation.  
 Die Studierenden faktorisieren Funktionen mehrerer Veränderlicher, stellen diese als Graph dar und marginalisieren sie bezüglich mehrerer Veränderlicher.  
 Die Studierenden erklären den Entwurf von Kanalcodes und den Einfluss der Minimaldistanz.  
 Sie decodieren Kanalcodes gemäß maximaler Likelihood und Nachrichtenaustausch.  
 Die Studierenden wenden verteilte Algorithmen auf Probleme der Nachrichtentechnik und des Alltagslebens an.  
 Die Studierenden verbessern die Eigenschaften von LDPC-Codes durch Erhöhung des Umfangs und/oder durch irreguläre Knotenordnungsverteilungen.  
 Die Studierenden transformieren Bildquellen zur Verbesserung verlustbehafteter Kompression in den Frequenzbereich.

### Literatur:

MacKay, D.: Information Theory, Inference, and Learning Algorithms, Cambridge University Press, Cambridge, 2003.

---

### Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

#### [1] Informatik (Bachelor of Science)

(Po-Vers. 2009s | Nebenfach | Nebenfach Elektrotechnik, Elektronik und Informationstechnik | Informationstechnik)

Dieses Modul ist daneben auch in den Studienfächern "Advanced Signal Processing & Communications Engineering (Master of Science)", "Berufspädagogik Technik (Bachelor of Science)", "Berufspädagogik Technik (Master of Education)", "Communications and Multimedia Engineering (Master of Science)", "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Bachelor of Science)", "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Master of Science)", "Elektrotechnik, Elektronik und Informationstechnik (Bachelor of Science)", "Elektrotechnik, Elektronik und Informationstechnik (Master of Science)", "Informatik (Master of Science)", "Information and Communication Technology (Master of Science)", "Informations- und Kommunikationstechnik (Bachelor of Science)", "Informations- und Kommunikationstechnik (Master of Science)", "Mathematik (Bachelor of Science)", "Medizintechnik (Master of Science)", "Wirtschaftsingenieurwesen (Bachelor of Science)", "Wirtschaftsingenieurwesen (Master of Science)" verwendbar.

---

### **Studien-/Prüfungsleistungen:**

Information Theory and Coding / Informationstheorie und Codierung (Prüfungsnummer: 36011)

(englische Bezeichnung: Information Theory and Coding)

Prüfungsleistung, Klausur, Dauer (in Minuten): 90

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Information Theory and Coding
- Tutorial for Information Theory and Coding

Prüfungssprache: Englisch

Erstablingung: WS 2019/2020, 1. Wdh.: SS 2020

1. Prüfer: Ralf Müller (100355)

---

### **Bemerkungen:**

Schlüsselwörter: ASC

---

**Modulbezeichnung: Halbleiterbauelemente (HBEL)** **5 ECTS**  
(Semiconductor Devices)

Modulverantwortliche/r: Tobias Dirnecker

Lehrende: Tobias Dirnecker, Tobias Stolzke

---

Startsemester: SS 2019	Dauer: 1 Semester	Turnus: halbjährlich (WS+SS)
Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Deutsch

---

**Lehrveranstaltungen:**

Das Tutorium Halbleiterbauelemente stellt ein zusätzliches Angebot an die Studierenden zur Prüfungsvorbereitung dar. Es handelt sich dabei um eine freiwillige Wahlveranstaltung.

Halbleiterbauelemente (SS 2019, Vorlesung, 2 SWS, Tobias Dirnecker)

Übungen zu Halbleiterbauelemente (SS 2019, Übung, 2 SWS, Christian Martens)

Tutorium Halbleiterbauelemente (SS 2019, Tutorium, 2 SWS, Assistenten)

---

**Empfohlene Voraussetzungen:**

Grundlagen der Elektrotechnik I

---

**Inhalt:**

Nach einer Einleitung werden Bewegungsgleichungen von Ladungsträgern im Vakuum sowie die Ladungsträgeremission im Vakuum und daraus abgeleitete Bauelemente besprochen. Anschließend werden Ladungsträger im Halbleiter behandelt: Hier werden die wesentlichen Aspekte der Festkörperphysik zusammengefasst, die zum Verständnis moderner Halbleiterbauelemente nötig sind. Darauf aufbauend werden im Hauptteil der Vorlesung die wichtigsten Halbleiterbauelemente, d.h. Dioden, Bipolartransistoren und Feldeffekttransistoren detailliert dargestellt. Einführungen in die wesentlichen Grundlagen von Leistungsbauelementen und optoelektronischen Bauelementen runden die Vorlesung ab.

**Lernziele und Kompetenzen:**

Die Studierenden

*Fachkompetenz*

*Verstehen*

verstehen grundlegende physikalische Vorgänge (u.a. Drift, Diffusion, Generation, Rekombination) im Halbleiter  
interpretieren Informationen aus Bänderdiagrammen

*Anwenden*

beschreiben die Funktionsweisen moderner Halbleiterbauelemente  
berechnen Kenngrößen der wichtigsten Bauelemente  
übertragen - ausgehend von den wichtigsten Bauelementen, wie Dioden, Bipolartransistoren und Feldeffekttransistoren - diese Funktionsprinzipien auf Weiterentwicklungen für spezielle Anwendungsgebiete wie Leistungselektronik oder Optoelektronik

*Analysieren*

diskutieren das Verhalten der Bauelemente z.B. bei hohen Spannungen oder erhöhter Temperatur

**Literatur:**

- Vorlesungsskript, am LEB erhältlich
- Neamen, D.A.: Semiconductor Physics and Devices: Basic Principles, 2nd ed., McGraw-Hill (Richard D. Irwin, Inc., Burr Ridge), USA, 1997
- Müller, R.: Grundlagen der Halbleiter-Elektronik: Band 1 der Reihe Halbleiter-Elektronik, 7. Auflage, Springer Verlag, Berlin, 1995
- Th. Tille, D. Schmitt-Landsiedel: Mikroelektronik, Springer-Verlag, Berlin, 2004
- S.K. Banerjee, B.G. Streetman: Solid State Electronic Devices, Prentice Hall, 2005

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

[1] **Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Nebenfach | Nebenfach Elektrotechnik, Elektronik und Informationstechnik | Mikroelektronik)

Dieses Modul ist daneben auch in den Studienfächern "Berufspädagogik Elektrotechnik und Informationstechnik (Bachelor of Education)", "Berufspädagogik Technik (Bachelor of Science)", "Berufspädagogik Technik (Master of Education)", "Elektrotechnik, Elektronik und Informationstechnik (Bachelor of Science)", "Informatik (Master of Science)", "Mathematik (Bachelor of Science)", "Mechatronik (Bachelor of Science)", "Medizintechnik (Bachelor of Science)", "Medizintechnik (Master of Science)", "Wirtschaftsingenieurwesen (Bachelor of Science)" verwendbar.

---

### **Studien-/Prüfungsleistungen:**

Vorlesung Halbleiterbauelemente (Prüfungsnummer: 25901)

(englische Bezeichnung: Lecture: Semiconductor Devices)

Prüfungsleistung, Klausur, Dauer (in Minuten): 90

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Halbleiterbauelemente
- Übungen zu Halbleiterbauelemente

Erstablægung: SS 2019, 1. Wdh.: WS 2019/2020

1. Prüfer: Tobias Dirnecker (100278)

---

### **Organisatorisches:**

Unterlagen zur Vorlesung über StudOn

---

**Modulbezeichnung:** Halbleiterbauelemente (HBEL) 5 ECTS  
 (Semiconductor Devices)

Modulverantwortliche/r: Tobias Dirnecker

Lehrende: Tobias Dirnecker, Christian Martens

---

Startsemester: WS 2019/2020	Dauer: 1 Semester	Turnus: halbjährlich (WS+SS)
Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Deutsch

---

**Lehrveranstaltungen:**

Das Tutorium Halbleiterbauelemente stellt ein zusätzliches Angebot an die Studierenden zur Prüfungsvorbereitung dar.

Es handelt sich dabei um eine freiwillige Wahlveranstaltung.

Halbleiterbauelemente (WS 2019/2020, Vorlesung, 2 SWS, Tobias Dirnecker)

Übungen zu Halbleiterbauelemente (WS 2019/2020, Übung, 2 SWS, Christian Martens)

Tutorium Halbleiterbauelemente (WS 2019/2020, Tutorium, 2 SWS, Christian Martens)

---

**Empfohlene Voraussetzungen:**

Grundlagen der Elektrotechnik I

---

**Inhalt:**

Die Vorlesung Halbleiterbauelemente vermittelt den Studenten der Elektrotechnik die physikalischen Grundlagen moderner Halbleiterbauelemente. Der erste Teil der Vorlesung befasst sich nach einer Einleitung mit Bewegungsgleichungen von Ladungsträgern im Vakuum sowie der Ladungsträgeremission im Vakuum und daraus abgeleiteten Bauelementen. In der anschließenden Behandlung von Ladungsträgern im Halbleiter werden die wesentlichen Aspekte der Festkörperphysik zusammengefasst, die zum Verständnis moderner Halbleiterbauelemente nötig sind. Darauf aufbauend werden im Hauptteil der Vorlesung die wichtigsten Halbleiterbauelemente, d.h. Dioden, Bipolartransistoren und Feldeffekttransistoren detailliert dargestellt. Einführungen in die wesentlichen Grundlagen von Leistungsbaulementen und optoelektronischen Bauelementen runden die Vorlesung ab.

**Lernziele und Kompetenzen:**

Die Studierenden

*Fachkompetenz*

*Verstehen*

- verstehen grundlegende physikalische Vorgänge (u.a. Drift, Diffusion, Generation, Rekombination) im Halbleiter
- interpretieren Informationen aus Bänderdiagrammen

*Anwenden*

- beschreiben die Funktionsweisen moderner Halbleiterbauelemente
- berechnen Kenngrößen der wichtigsten Bauelemente
- übertragen - ausgehend von den wichtigsten Bauelementen, wie Dioden, Bipolartransistoren und Feldeffekttransistoren - diese Funktionsprinzipien auf Weiterentwicklungen für spezielle Anwendungsgebiete wie Leistungselektronik oder Optoelektronik

*Analysieren*

- diskutieren das Verhalten der Bauelemente z.B. bei hohen Spannungen oder erhöhter Temperatur

**Literatur:**

- Vorlesungsskript, am LEB erhältlich
- R. Müller: Grundlagen der Halbleiter-Elektronik, Band 1 der Reihe Halbleiter-Elektronik, Springer-Verlag, Berlin, 2002
- D.A. Neamen: Semiconductor Physics and Devices: Basic Principles, McGraw-Hill (Richard D. Irwin Inc.), 2002
- Th. Tille, D. Schmitt-Landsiedel: Mikroelektronik, Springer-Verlag, Berlin, 2004
- S.K. Banerjee, B.G. Streetman: Solid State Electronic Devices, Prentice Hall, 2005

---

### Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

#### [1] Informatik (Bachelor of Science)

(Po-Vers. 2009s | Nebenfach | Nebenfach Elektrotechnik, Elektronik und Informationstechnik | Mikroelektronik)

Dieses Modul ist daneben auch in den Studienfächern "Berufspädagogik Elektrotechnik und Informationstechnik (Bachelor of Education)", "Berufspädagogik Technik (Bachelor of Science)", "Berufspädagogik Technik (Master of Education)", "Elektrotechnik, Elektronik und Informationstechnik (Bachelor of Science)", "Informatik (Master of Science)", "Mathematik (Bachelor of Science)", "Mechatronik (Bachelor of Science)", "Medizintechnik (Bachelor of Science)", "Medizintechnik (Master of Science)", "Wirtschaftsingenieurwesen (Bachelor of Science)" verwendbar.

---

### Studien-/Prüfungsleistungen:

Halbleiterbauelemente (Prüfungsnummer: 25901)

(englische Bezeichnung: Semiconductor Devices)

Prüfungsleistung, Klausur, Dauer (in Minuten): 90

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Halbleiterbauelemente
- Übungen zu Halbleiterbauelemente

Erstablingung: WS 2019/2020, 1. Wdh.: SS 2020

1. Prüfer: Tobias Dirnecker (100278)

---

### Organisatorisches:

Unterlagen zur Vorlesung über StudOn



---

**Modulbezeichnung:** Entwurf Integrierter Schaltungen I (EIS I) **5 ECTS**  
(Design on Integrated Circuits I)

Modulverantwortliche/r: Sebastian M. Sattler  
Lehrende: Sebastian M. Sattler

---

Startsemester: WS 2019/2020	Dauer: 1 Semester	Turnus: jährlich (WS)
Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Deutsch

---

**Lehrveranstaltungen:**

Entwurf Integrierter Schaltungen I (WS 2019/2020, Vorlesung, 2 SWS, Sebastian M. Sattler)  
Übungen zu Entwurf Integrierter Schaltungen I (WS 2019/2020, Übung, 2 SWS, Florian Deeg et al.)

---

**Inhalt:**

Die Vorlesung führt in die Grundlagen des integrierten digitalen Schaltungsentwurfes auf Basis von CMOS ein. Ausgehend vom MOS Transistor wird die Complementary Logik erklärt und auf gängige statische und dynamische Schaltelemente und Ihre Erweiterungen auf hochintegrierte Schaltungen bis  $0.13\mu\text{m}$  eingegangen.

- Digitaler IC Entwurf für Deep Submicron
- MOS Transistor
- Herstellung, Layout und Simulation
- MOS Inverterschaltung
- Statische CMOS Gatter-Schaltungen
- Entwurf von Logik mit hoher Schaltrate
- Transfer-Gatter und dynamische Logik
- Entwurf von Speichern
- Zusätzliche Themen des Speicherentwurfs

Content

The course introduces students to the basics of digital integrated circuit design in CMOS. Starting from the MOS transistor, complementary logic is explained. Common static and dynamic switching elements are discussed as well as their extensions to large scale integrated circuits ( $0.18\mu\text{m}$  -  $0.13\mu\text{m}$ ).

- Deep Submicron Digital IC Design
- MOS Transistor
- Fabrication, Layout and Simulation
- MOS Inverter Circuits
- Static CMOS Gate-Circuits
- Design of Logic with High Switching Rate
- Transfer-Gates and Dynamic Logic
- Design of Memory
- Additional Topics of Memory Design

**Lernziele und Kompetenzen:**

Verstehen

- Überblick über existierende Integrationstechnologien und Entwurfsmethodiken für Integrierte Schaltungen in  $0.18\mu\text{m}$  und  $0.13\mu\text{m}$  CMOS gewinnen und dabei die Zusammenhänge zwischen technischen und wirtschaftlichen Gesichtspunkten der Halbleiterfertigung verstehen.

Evaluiieren (Beurteilen)

- Verhalten von MOS/CMOS-Transistoren analysieren und verschiedene statische und dynamische digitale Schaltungsstrukturen auf Transistorebene bewerten können.

Learning objectives and competencies:

Understand

- gain an overview of existing integration technologies and integrated circuit design techniques in CMOS ( $0.18\mu\text{m}$  -  $0.13\mu\text{m}$ ), understanding technical and economic aspects of semiconductor manufacturing.

Evaluate (Assess)

- Analyze the behavior of MOS / CMOS transistors and evaluate various static and dynamic digital circuit structures at transistor level.

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

**[1] Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Nebenfach | Nebenfach Elektrotechnik, Elektronik und Informationstechnik | Mikroelektronik)

Dieses Modul ist daneben auch in den Studienfächern "Berufspädagogik Technik (Bachelor of Science)", "Berufspädagogik Technik (Master of Education)", "Elektrotechnik, Elektronik und Informationstechnik (Bachelor of Science)", "Elektrotechnik, Elektronik und Informationstechnik (Master of Science)", "Informatik (Master of Science)", "Information and Communication Technology (Master of Science)", "Informations- und Kommunikationstechnik (Bachelor of Science)", "Informations- und Kommunikationstechnik (Master of Science)", "Mathematik (Bachelor of Science)", "Mechatronik (Bachelor of Science)", "Mechatronik (Master of Science)", "Wirtschaftsingenieurwesen (Master of Science)" verwendbar.

---

**Studien-/Prüfungsleistungen:**

Entwurf integrierter Schaltungen I / Design of Integrated Circuits I (Prüfungsnummer: 65901)

(englische Bezeichnung: Design of Integrated Circuits I)

Prüfungsleistung, Klausur, Dauer (in Minuten): 90

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Entwurf Integrierter Schaltungen I
- Übungen zu Entwurf Integrierter Schaltungen I

Erstablingung: WS 2019/2020, 1. Wdh.: SS 2020

1. Prüfer: Sebastian M. Sattler (100270)

---

---

**Modulbezeichnung: Schaltungstechnik (ST)** **5 ECTS**  
(Electronic Circuits)

Modulverantwortliche/r: Robert Weigel  
Lehrende: Robert Weigel

---

Startsemester: SS 2019	Dauer: 1 Semester	Turnus: jährlich (SS)
Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Deutsch

---

**Lehrveranstaltungen:**

Schaltungstechnik (SS 2019, Vorlesung, 2 SWS, Robert Weigel)  
Übungen zu Schaltungstechnik (SS 2019, Übung, 2 SWS, Marco Dietz et al.)

---

**Inhalt:**

- Halbleiterbauelemente: Diode, Bipolartransistor, MOSFET
- Transistor-Grundsaltungen: Arbeitspunkte, Großsignal-, Kleinsignalverhalten
- Verstärker: Stromquellen, Differenzverstärker, Impedanzwandler
- Operationsverstärker, innerer Aufbau, Modelle, Anwendungen
- Digital-Analog-/Analog-Digital-Umsetzer: Grundsaltungen, Modelle, Anwendungen

**Lernziele und Kompetenzen:**

- Die Studierenden verstehen den Aufbau und die Funktionsweisen von Halbleiterschaltungen wie Dioden- und Transistorgrundsaltungen, Verstärkern, Operationsverstärkern und Analog-Digital-/Digital-Analog-Umsetzern und können diese erläutern.
  - Die Studierenden können komplexe Schaltungen durch eine Zerlegung in grundlegende Funktionsblöcke analysieren und diese in ihrer Funktion beurteilen.
  - Die Studierenden verstehen die Entwicklungsmethodik beim Entwurf von grundlegenden Halbleiterschaltungen und können diese dimensionieren.
  - Die Studierenden können eine einfache, abstrakte Funktionsbeschreibung in grundlegende Halbleiterschaltungen abbilden und diese zur Erfüllung der abstrakten Funktion auslegen.
- 

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

**[1] Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Nebenfach | Nebenfach Elektrotechnik, Elektronik und Informationstechnik | Mikroelektronik)

Dieses Modul ist daneben auch in den Studienfächern "Berufspädagogik Elektrotechnik und Informationstechnik (Bachelor of Education)", "Berufspädagogik Technik (Bachelor of Science)", "Berufspädagogik Technik (Master of Education)", "Elektrotechnik, Elektronik und Informationstechnik (Bachelor of Science)", "Informatik (Master of Science)", "Mathematik (Bachelor of Science)", "Mechatronik (Bachelor of Science)", "Medizintechnik (Bachelor of Science)", "Medizintechnik (Master of Science)" verwendbar.

---

**Studien-/Prüfungsleistungen:**

Vorlesung Schaltungstechnik\_ (Prüfungsnummer: 26601)

(englische Bezeichnung: Electronic Circuits)

Prüfungsleistung, Klausur, Dauer (in Minuten): 90

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Schaltungstechnik
- Übungen zu Schaltungstechnik

Erstabelleung: SS 2019, 1. Wdh.: WS 2019/2020, 2. Wdh.: SS 2020

1. Prüfer: Robert Weigel (100221)

---

---

**Modulbezeichnung:** Das System Erde (Z-Edu-Geo1) 5 ECTS  
(System Earth for scientists)

Modulverantwortliche/r: Anette Regelous  
Lehrende: Anette Regelous

---

Startsemester: WS 2019/2020	Dauer: 1 Semester	Turnus: jährlich (WS)
Präsenzzeit: 75 Std.	Eigenstudium: 75 Std.	Sprache: Deutsch

---

**Lehrveranstaltungen:**

[Z-Edu-Geo1] + [Nebenfach] Das System Erde (WS 2019/2020, Seminar, 4 SWS, Anette Regelous)

---

**Inhalt:**

Die Inhalte des Seminars sind:

- Entstehung der Elemente
- Entstehung des Sonnensystems
- Aufbau der Erde
- Grundlagen des Vulkanismus und der Plattentektonik
- Einführung in die Seismik
- Entstehung und Entwicklung der kontinentalen und ozeanischen Kruste
- Übersicht über die Entwicklung des Lebens und der Atmosphäre
- Grundlagen über die wichtigsten Minerale und Gesteine
- Einführung in die Bildung von Rohstoffen

Die Inhalte der Übung sind die Bestimmung von

- ausgewählten Mineralen
- die wichtigsten metamorphen, magmatischen und Sedimentgesteine

**Lernziele und Kompetenzen:**

Die Studierenden können

- die Entstehung der Elemente erläutern
  - fachwissenschaftliche Grundlagen und die Zusammenhänge des System Erde erklären
  - die Entstehung und Entwicklung der ozeanischen und kontinentalen Kruste wiedergeben
  - Grundlagen der Forschungsmethodik wie z.B. Seismik erklären
  - die Entwicklung des Lebens und der Atmosphäre erläutern
  - in Gruppen Minerale und Gesteine kooperativ und verantwortlich bestimmen
  - komplexe fachbezogene Inhalte über Minerale und Gesteine in der Übung klar und zielgruppengerecht präsentieren und argumentativ vertreten
  - Minerale und Gesteine erkennen und nach vorgegebenen Kriterien bestimmen
- 

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

**[1] Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Nebenfach | Nebenfach Geowissenschaften)

Dieses Modul ist daneben auch in den Studienfächern "Energietechnik (Master of Science)", "Geowissenschaften im Lehramt (Zusatzstudien)", "Informatik (Master of Science)", "Mathematik (Bachelor of Science)", "Modulstudien Naturale (keine Abschlussprüfung angestrebt bzw. möglich)" verwendbar.

---

**Studien-/Prüfungsleistungen:**

Das System Erde (Prüfungsnummer: 36351)

Prüfungsleistung, Klausur, Dauer (in Minuten): 60

Anteil an der Berechnung der Modulnote: 100%

Erstablingung: WS 2019/2020, 1. Wdh.: SS 2020

1. Prüfer: Anette Regelous (080211)

---

---

**Modulbezeichnung:** Der Mensch im geowissenschaftlichen Umfeld II (NFGGeoUmf2) 5 ECTS  
 (Humans in the geological environment II)

Modulverantwortliche/r: Matthias Göbbels  
 Lehrende: Matthias Göbbels

---

Startsemester: SS 2019	Dauer: 1 Semester	Turnus: jährlich (SS)
Präsenzzeit: 75 Std.	Eigenstudium: 75 Std.	Sprache: Deutsch

---

**Lehrveranstaltungen:**

Der Mensch im geologischen Umfeld II (SS 2019, Übung, N.N.)

---

**Empfohlene Voraussetzungen:**

keine

---

**Inhalt:**

**Der Mensch im geologischen Umfeld II:** Den Studierenden erarbeiten sich an verschiedenen Beispielen, auf welche Weise und mit welchen Konsequenzen neben endogenen und exogenen Kräften auch der Mensch in das System Erde eingreifen kann. Zudem ist er ihr Nutzer, kann aber auch als Zerstörer in Erscheinung treten. Durch differenzierte Betrachtung ausgewählter Nutzungsmöglichkeiten.

- Lagerstätten (z. B. Ton/Sand/Kies, Natursteine, Erdöl/Erdgas, Kohle, Salze, Er-ze, Kaolin) und Aspekte ihrer wirtschaftlichen Nutzung (Abbauwürdigkeit, Preisabhängigkeit, Prospektion); Abbau und Rekultivierung
- Geothermie: oberflächennahe und tiefe Form, Grundlagen und Nutzungs-möglichkeiten
- Hydrogeologie: Grundwasserproblematik bezüglich Menge und Qualität, Nachhaltigkeit der Nutzung
- Baugrund: Eigenschaften des Untergrunds, Gefährdung von Bauwerken

**Exkursion** Ziel der Exkursion ist es, dass die Studierenden lernen, ihre geologische Umgebung bewusst wahrzunehmen, zu verstehen und zu interpretieren, indem sie die Fähigkeit zur gezielten Beobachtung und Beschreibung landschaftlicher Phänomene im Mikro- wie auch im Makrobereich entwickeln. Durch die originale Begegnung vor Ort lernen sie, die Landschaft als schützenswerte Ressource zu schätzen, für deren Erhalt auch persönliches Engagement erforderlich ist.

- Bestimmung von Mineralen und Gesteinen im Gelände
- Arbeit an einem Aufschluss
- Besuch eines Museums oder einer Forschungsstätte (z. B. Kontinentale Tief-bohrung)

**Lernziele und Kompetenzen:**

Die Studierenden können

- Lagerstätten und Aspekte ihrer wirtschaftlichen Nutzung wiedergeben, darstellen und klassifizieren
- den Abbau und die Rekultivierung von Lagerstätten hinterfragen und diskutieren
- die Prinzipien der oberflächennahen und tiefen Geothermie erläutern und deren Nutzungsmöglichkeiten beschreiben
- hydrogeologische Grundlagen wiedergeben, Grundwasserproblematiken herausstellen und die Nachhaltigkeit der Nutzung erläutern
- die Grundlagen der ingenieurgeologischen Baugrunduntersuchungen aufzählen, die Eigenschaften des Untergrunds nennen und die Gefährdung von Bauwerken erklären
- geowissenschaftliche Arbeitsmethoden im Gelände anwenden
- eine Landschaft in ihren Grundstrukturen visuell erfassen und geologisch beschreiben
- Gesteine und Strukturen im Gelände bestimmen
- Gesteinsproben nehmen
- einen Bericht verfassen

**Literatur:**

Wird von den Dozenten in den jeweiligen Veranstaltungen vorgestellt.

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

**[1] Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Nebenfach | Nebenfach Geowissenschaften)

Dieses Modul ist daneben auch in den Studienfächern "Energietechnik (Master of Science)", "Informatik (Master of Science)" verwendbar.

---

**Studien-/Prüfungsleistungen:**

Der Mensch im geologischen Umfeld II (Prüfungsnummer: 36353)

Prüfungsleistung, mündliche Prüfung, Dauer (in Minuten): 30

Anteil an der Berechnung der Modulnote: 40%

Erstablingung: SS 2019, 1. Wdh.: WS 2019/2020

1. Prüfer: Anette Regelous (080211)

---

---

**Modulbezeichnung:** Das System Erde (Z-Edu-Geo1) **5 ECTS**  
(System Earth for scientists)

Modulverantwortliche/r: Anette Regelous  
Lehrende: Anette Regelous

---

Startsemester: WS 2019/2020	Dauer: 1 Semester	Turnus: jährlich (WS)
Präsenzzeit: 75 Std.	Eigenstudium: 75 Std.	Sprache: Deutsch

---

**Lehrveranstaltungen:**

[Z-Edu-Geo1] + [Nebenfach] Das System Erde (WS 2019/2020, Seminar, 4 SWS, Anette Regelous)

---

**Inhalt:**

Die Inhalte des Seminars sind:

- Entstehung der Elemente
- Entstehung des Sonnensystems
- Aufbau der Erde
- Grundlagen des Vulkanismus und der Plattentektonik
- Einführung in die Seismik
- Entstehung und Entwicklung der kontinentalen und ozeanischen Kruste
- Übersicht über die Entwicklung des Lebens und der Atmosphäre
- Grundlagen über die wichtigsten Minerale und Gesteine
- Einführung in die Bildung von Rohstoffen

Die Inhalte der Übung sind die Bestimmung von

- ausgewählten Mineralen
- die wichtigsten metamorphen, magmatischen und Sedimentgesteine

**Lernziele und Kompetenzen:**

Die Studierenden können

- die Entstehung der Elemente erläutern
  - fachwissenschaftliche Grundlagen und die Zusammenhänge des System Erde erklären
  - die Entstehung und Entwicklung der ozeanischen und kontinentalen Kruste wiedergeben
  - Grundlagen der Forschungsmethodik wie z.B. Seismik erklären
  - die Entwicklung des Lebens und der Atmosphäre erläutern
  - in Gruppen Minerale und Gesteine kooperativ und verantwortlich bestimmen
  - komplexe fachbezogene Inhalte über Minerale und Gesteine in der Übung klar und zielgruppengerecht präsentieren und argumentativ vertreten
  - Minerale und Gesteine erkennen und nach vorgegebenen Kriterien bestimmen
- 

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

**[1] Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Nebenfach | Nebenfach Geowissenschaften)

Dieses Modul ist daneben auch in den Studienfächern "Energietechnik (Master of Science)", "Geowissenschaften im Lehramt (Zusatzstudien)", "Informatik (Master of Science)", "Mathematik (Bachelor of Science)", "Modulstudien Naturale (keine Abschlussprüfung angestrebt bzw. möglich)" verwendbar.

---

**Studien-/Prüfungsleistungen:**

Das System Erde (Prüfungsnummer: 49801)

Prüfungsleistung, Präsentation

Anteil an der Berechnung der Modulnote: 100% Prüfungssprache: Deutsch

Erstablingung: WS 2019/2020, 1. Wdh.: WS 2019/2020 (nur für Wiederholer)

1. Prüfer: Anette Regelous (080211)

---

---

**Modulbezeichnung: Produktionstechnik I + II (PT I+II)** **5 ECTS**  
 (Production Engineering I + II)

Modulverantwortliche/r: Marion Merklein

Lehrende: Marion Merklein, Dietmar Drummer, Jörg Franke, Michael Schmidt, Nico Hanenkamp

Startsemester: WS 2019/2020

Dauer: 2 Semester

Turnus: jährlich (WS)

Präsenzzeit: 60 Std.

Eigenstudium: 90 Std.

Sprache: Deutsch

**Lehrveranstaltungen:**

Produktionstechnik I (WS 2019/2020, Vorlesung, 2 SWS, Marion Merklein et al.)

Produktionstechnik II (SS 2020, Vorlesung, 2 SWS, Nico Hanenkamp et al.)

**Inhalt:**
**Produktionstechnik I:**

Basierend auf der DIN 8580 werden in der Vorlesung Produktionstechnik I die aktuellen Technologien sowie die dabei eingesetzten Maschinen in den Bereichen Urformen, Umformen, Trennen, Fügen, Beschichten und das Ändern der Stoffeigenschaften behandelt. Hierbei werden sowohl die Prozessketten als auch die spezifischen Eigenschaften der Produktionstechniken aufgezeigt und anhand von praxisrelevanten Bauteilen erläutert. Zum besseren Verständnis der Verfahren werden zunächst metallkundliche Grundlagen, wie der mikrostrukturelle Aufbau von metallischen Werkstoffen und ihr plastisches Verhalten, erläutert. Anschließend werden die Urformverfahren Gießen und Pulvermetallurgie dargestellt. Im weiteren Verlauf der Vorlesung erfolgt eine Gegenüberstellung der Verfahren der Massivumformung Stauchen, Schmieden, Fließpressen und Walzen. Im Rahmen des Kapitels Blechumformung wird die Herstellung von Bauteilen durch Tiefziehen, Streckziehen und Biegen betrachtet. Der Fokus in der Vorstellung der Verfahrensgruppe Trennen liegt auf den Prozessen des Zerteilens und Spanens. Die Vorlesungseinheit des Bereichs Fügen behandelt die Herstellung von Verbindungen mittels Umformen, Schweißen und Löten. Abschließend werden verschiedene strahlbasierte Fertigungsverfahren aus den sechs Bereichen vorgestellt. Im Fokus stehen hierbei laserbasierte Fertigungsverfahren, wie zum Beispiel Schweißen, Schneiden oder Additiven Fertigung. Eine zusätzlich angebotene Übung dient der Vertiefung und der Anwendung des Vorlesungsinhaltes.

**Produktionstechnik II:**

Die Vorlesung beschäftigt sich inhaltlich mit der Verarbeitung von Kunststoffen (Spritzgießen, Erzeugung von duroplastischen / thermoplastischen Faserverbunden) und Metallen mit dem Fokus auf strahlbasierten Verfahren (Schneiden, Schweißen und Additive Fertigung mittels Wasser-, Elektronen- und Laserstrahl). Des Weiteren werden die Grundlagen zu Werkzeugmaschinen und dem Werkzeugmaschinenbau (Maschinenkomponenten, Funktionalitäten, Anwendungs- / Einsatzmöglichkeiten) sowie zu Montagetechnologien und Verbindungstechniken (Auslegung von Verbindungen, prozesstechnische Umsetzung und Realisierung) vermittelt. Einen weiteren Schwerpunkt stellen der Elektromaschinenbau und die Elektronikproduktion (Funktionsweise und Herstellung von elektronischen Antriebseinheiten, Auslegung und Herstellung von elektronischen Komponenten) dar.

**Lernziele und Kompetenzen:**
*Fachkompetenz*
*Wissen*

- Die Studierenden erwerben grundlegende Kenntnisse in der Metallkunde und der Verarbeitung von Metallen.
- Die Studierenden erhalten einen Überblick über die Produktionsverfahren Urformen, Umformen, Fügen, Trennen, ihre Untergruppen
- Die Studierenden erwerben ein grundlegendes Prozessverständnis hinsichtlich der wirkenden Mechanismen.
- Die Studierenden erwerben Wissen über die Prozessführung sowie spezifische Eigenschaften der Produktionsverfahren.
- Die Studierenden erwerben grundlegendes Verständnis zu den Eigenschaften von Kunststoffen und deren Verarbeitung



- Die Studierenden erwerben Kenntnisse über werkstoffwissenschaftliche Aspekte und Werkstoffeigenschaften sowie Werkstoffverhalten vor und nach den jeweiligen Bearbeitungsprozessen
- Die Studierenden erwerben fundamentale Kenntnisse zu Multi-Materialien-Verbunden.
- Die Studierenden erlangen grundlegende Kenntnisse zur Funktionsweise von elektrischen Antriebseinheiten und deren Herstellung sowie die Herstellung von elektrischen Komponenten (MID)
- Die Studierenden erhalten grundlegende Kenntnisse im Bereich der Produktentwicklung und Produktauslegung (Verfahrensmöglichkeiten, Verfahrensgrenzen, Designeinschränkungen, etc.)

#### *Verstehen*

- Die Studierenden sind in der Lage die grundlegenden Prinzipien von Fertigungsprozessen und der Systemauslegung zu verstehen
- Die Studierenden verstehen die Grundlagen des Anlagen- und Werkzeugbaus

#### *Anwenden*

- Die Studierenden können geeignete Fertigungsverfahren zur Herstellung technischer Produkte bestimmen (Schwerpunkte: Urformen, Umformen, Fügen, Trennen).

#### *Analysieren*

- Die Studierenden können die verschiedenen Fertigungsverfahren erkennen und normgerecht differenzieren

---

### **Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

#### **[1] Informatik (Bachelor of Science): ab 3. Semester**

(Po-Vers. 2009s | Nebenfach | Nebenfach Maschinenbau | Fertigungsautomatisierung und Produktionstechnik)

Dieses Modul ist daneben auch in den Studienfächern "Berufspädagogik Technik (Bachelor of Science)", "Berufspädagogik Technik (Master of Education)", "Informatik (Master of Science)", "Maschinenbau (Bachelor of Science)", "Mechatronik (Bachelor of Science)", "Medizintechnik (Bachelor of Science)", "Wirtschaftsingenieurwesen (Bachelor of Science)" verwendbar.

---

### **Studien-/Prüfungsleistungen:**

Vorlesung Produktionstechnik I und II (Prüfungsnummer: 45701)

(englische Bezeichnung: Lecture: Production Engineering I and II)

Prüfungsleistung, Klausur, Dauer (in Minuten): 120

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Produktionstechnik I
- Produktionstechnik II

Erstablesung: SS 2020, 1. Wdh.: WS 2020/2021

1. Prüfer: Franke/Merkl./M.Schm./Drummer/Hane. (ps0554)

---

---

**Modulbezeichnung: Automatisierte Produktionsanlagen (APA)** **5 ECTS**  
 (Automated Manufacturing Systems)

Modulverantwortliche/r: Jörg Franke  
 Lehrende: Jörg Franke

---

Startsemester: WS 2019/2020	Dauer: 1 Semester	Turnus: jährlich (WS)
Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Deutsch

---

**Lehrveranstaltungen:**

Automatisierte Produktionsanlagen (WS 2019/2020, Vorlesung, 2 SWS, Jörg Franke)  
 Übung zu Automatisierte Produktionsanlagen (WS 2019/2020, Übung, 2 SWS, Jörg Franke)

---

**Inhalt:**

Die Vorlesung „Automatisierte Produktionsanlagen“ richtet sich an Studierende der Informatik, des Maschinenbaus, der Mechatronik, der Medizintechnik und des Wirtschaftsingenieurwesens. Es werden Inhalte zum Aufbau und Betrieb Automatisierter Produktionsanlagen gelehrt. Zu Beginn wird grundlegendes Wissen bezüglich Elektromaschinen, Fluidantrieben, Sensoren und der Steuerungs- und Kommunikationstechnik vermittelt. Darauf aufbauend werden Systeme zur Vereinzelnung, Ordnung und Handhabung von Werkstücken sowie Werkzeugmaschinen vorgestellt. Des Weiteren sind Lösungen zur Realisierung eines automatisierten Materialflusses sowie flexible Fertigungssysteme Inhalte der Vorlesung. Schließlich werden Softwarekomponenten zur rechnergestützten Diagnose und Qualitätssicherung und Auftragssteuerung betrachtet. Somit kann der Hörer die Komponenten einer Automatisierten Produktionsanlage bewerten und die ebenfalls in dieser Vorlesung vermittelten Methoden zur Planung, Optimierung und Inbetriebnahme Automatisierter Produktionsanlagen optimal anwenden.

Übersicht der Vorlesungen und Übungen:

- Industrie 4.0 und Industrial Internet of Things
- Elektrische Maschinen
- Fluidtechnische Antriebe
- Sensoren
- Steuerungs- und Kommunikationstechnik
- Industrieroboter
- Werkzeugmaschinen
- Produktionslogistik und Zuführtechnik
- Flexible Fertigungssysteme
- Planung und Optimierung von Automatisierten Produktionsanlagen
- Auftragssteuerung
- Inbetriebnahme und Betrieb von Automatisierten Produktionsanlagen
- Rechnergestützte Diagnose

**Lernziele und Kompetenzen:**

- Kenntnis von Einsatzfeldern, Definition, Nutzen, Leistungsfähigkeit und technischen Neuerungen für die Zukunft von APA
  - Bewertung der verschiedenen Komponenten von APA hinsichtlich Leistungsfähigkeit, Kosten, Vor- und Nachteilen, möglicher Alternativen
  - Kenntnis der Möglichkeiten zur Vernetzung der einzelnen Komponenten (Schnittstellen: mechanisch, elektrisch, informationstechnisch etc.)
  - Beherrschung von Methoden und Werkzeugen zur Planung, Inbetriebnahme, Betrieb und Optimierung von APA
  - Berechnung der Wirtschaftlichkeit von APA
- 

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

**[1] Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Nebenfach | Nebenfach Maschinenbau | Fertigungsautomatisierung und Produktionstechnik)

## [2] Informatik (Bachelor of Science)

(Po-Vers. 2009s | Nebenfach | Nebenfach Maschinenbau | Computerintegrierte Produktion)

Dieses Modul ist daneben auch in den Studienfächern "Berufspädagogik Technik (Bachelor of Science)", "Berufspädagogik Technik (Master of Education)", "Computational Engineering (Master of Science with Honours)", "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Master of Science)", "Informatik (Master of Science)", "International Production Engineering and Management (Bachelor of Science)", "Internationales Projektmanagement Großanlagenbau/International Project Management in Systems Engineering (Master of Science)", "Maschinenbau (Bachelor of Science)", "Maschinenbau (Master of Science)", "Mechatronik (Bachelor of Science)", "Mechatronik (Master of Science)", "Medizintechnik (Master of Science)", "Wirtschaftsingenieurwesen (Bachelor of Science)", "Wirtschaftsingenieurwesen (Master of Science)" verwendbar.

---

### Studien-/Prüfungsleistungen:

Automatisierte Produktionsanlagen (Prüfungsnummer: 73001)

(englische Bezeichnung: Lecture/Tutorial: Automated Manufacturing Systems)

Prüfungsleistung, Klausur, Dauer (in Minuten): 120

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Automatisierte Produktionsanlagen
- Übung zu Automatisierte Produktionsanlagen

weitere Erläuterungen:

Klausur kann einen Anteil an Fragen im Antwort-Wahl-Verfahren (Multiple-Choice) enthalten.

Erstabllegung: WS 2019/2020, 1. Wdh.: SS 2020

1. Prüfer: Jörg Franke (100615)

---

### Organisatorisches:

weitere Informationen bei:

M.Sc. Marco Ziegler

Für die Prüfung sind ausschließlich folgende Hilfsmittel zugelassen:

- nicht programmierbarer Taschenrechner
- dokumentenechter Stift
- Textmarker
- Lineal, Geodreieck, Zirkel
- Namensstempel

Darüber hinaus sind keine weiteren Hilfsmittel erlaubt (dies gilt insbesondere für Uhren, Mobiltelefone oder sonstige elektronische Geräte).

**Modulbezeichnung:** **Dynamik starrer Körper (3V+2Ü+2T) (DSK)** **7.5 ECTS**  
(Dynamics (3L+2E+2T))

Modulverantwortliche/r: Sigrid Leyendecker  
Lehrende: Sigrid Leyendecker

Startsemester: WS 2019/2020      Dauer: 1 Semester      Turnus: jährlich (WS)  
Präsenzzeit: 105 Std.      Eigenstudium: 120 Std.      Sprache: Deutsch

**Lehrveranstaltungen:**

Dynamik starrer Körper (WS 2019/2020, Vorlesung, 3 SWS, Sigrid Leyendecker)  
Tutorium zur Dynamik starrer Körper (WS 2019/2020, Tutorium, 2 SWS, Denisa Martonová et al.)  
Übungen zur Dynamik starrer Körper (WS 2019/2020, Übung, 2 SWS, David Holz et al.)

**Empfohlene Voraussetzungen:**

Kenntnisse aus dem Modul "*Statik, Elastostatik und Festigkeitslehre*" bzw. "*Statik und Festigkeitslehre*"

**Vorhergehende Module:**

Statik und Festigkeitslehre (3V+2Ü+2T)  
Statik, Elastostatik und Festigkeitslehre  
Statik und Festigkeitslehre

**Inhalt:**

- Kinematik von Punkten und starren Körpern
- Relativkinematik von Punkten und starren Körpern
- Kinetik des Massenpunktes
- Newton'sche Axiome
- Energiesatz
- Stoßvorgänge
- Kinetik des Massenpunktsystems
- Lagrange'sche Gleichungen 2. Art
- Kinetik des starren Körpers
- Trägheitstensor
- Kreiselgleichungen
- Schwingungen

**Lernziele und Kompetenzen:**

- Die Studierenden
- sind vertraut mit den grundlegenden Begriffen und Axiomen der Dynamik;
  - können Bewegungen von Massepunkten und starren Körpern in verschiedenen Koordinatensystemen beschreiben;
  - können die Bewegungsgleichungen von Massepunkten und starren Körpern mittels der Newtonschen Axiome oder mittels der Lagrangeschen Gleichungen aufstellen;
  - können die Bewegungsgleichungen für einfache Stoßprobleme lösen;
  - können die Bewegungsgleichung für einfache Schwingungsprobleme analysieren.

**Literatur:**

Gross, Hauger, Schnell, Wall: Technische Mechanik 3, Berlin:Springer, 2006

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

[1] **Informatik (Bachelor of Science): ab 3. Semester**

(Po-Vers. 2009s | Nebenfach | Nebenfach Maschinenbau | Technische Mechanik)

Dieses Modul ist daneben auch in den Studienfächern "Berufspädagogik Technik (Bachelor of Science)", "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Bachelor of Science)", "Informatik (Master of Science)", "International Production Engineering and Management (Bachelor of Science)", "Maschinenbau

(Bachelor of Science)", "Mechatronik (Bachelor of Science)", "Medizintechnik (Bachelor of Science)", "Medizintechnik (Master of Science)", "Technomathematik (Bachelor of Science)", "Wirtschaftsingenieurwesen (Bachelor of Science)" verwendbar.

---

### **Studien-/Prüfungsleistungen:**

Vorlesung Dynamik starrer Körper (Prüfungsnummer: 45001)

(englische Bezeichnung: Lecture: Dynamics of Rigid Bodies)

Prüfungsleistung, Klausur, Dauer (in Minuten): 90

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Dynamik starrer Körper
- Übungen zur Dynamik starrer Körper

Erstablingung: WS 2019/2020, 1. Wdh.: SS 2020

1. Prüfer: Sigrid Leyendecker (100447)

---

**Modulbezeichnung:** Statik und Festigkeitslehre (SUF) 7.5 ECTS  
(Statics and Strength of Materials)

Modulverantwortliche/r: Kai Willner

Lehrende: Kai Willner, Gunnar Possart, Martin Jerschl, Maximilian Volkan Baloglu, Lucie Spannraft

Startsemester: WS 2019/2020	Dauer: 1 Semester	Turnus: halbjährlich (WS+SS)
Präsenzzeit: 105 Std.	Eigenstudium: 120 Std.	Sprache: Deutsch

**Lehrveranstaltungen:**

Statik und Festigkeitslehre (WS 2019/2020, Vorlesung, 3 SWS, Kai Willner)  
 Übungen zur Statik und Festigkeitslehre (WS 2019/2020, Übung, 2 SWS, Gunnar Possart et al.)  
 Tutorium zur Statik und Festigkeitslehre (WS 2019/2020, Tutorium, 2 SWS, Gunnar Possart et al.)

**Inhalt:**

- Kraft- und Momentenbegriff, Axiome der Statik
- ebene und räumliche Statik
- Flächenmomente 1. und 2. Ordnung
- Haft- und Gleitreibung
- Spannung, Formänderung, Stoffgesetz
- überbestimmte Stabwerke, Balkenbiegung
- Torsion
- Elastizitätstheorie und Festigkeitsnachweis
- Stabilität

**Lernziele und Kompetenzen:**

*Fachkompetenz*

*Wissen*

Die Studierenden kennen

- die axiomatischen Grundlagen der Technischen Mechanik sowie die entsprechenden Fachtermini.
- das Schnittprinzip und die Einteilung der Kräfte in eingeprägte und Reaktionskräfte bzw. in äußere und innere Kräfte.
- die Gleichgewichtsbedingungen am starren Körper.
- das Phänomen der Haft- und Gleitreibung.
- die Begriffe der Verzerrung und Spannung sowie das linear-elastische Stoffgesetz.
- den Begriff der Hauptspannungen sowie das Konzept der Vergleichsspannung und Festigkeitshypothesen.
- das Problem der Stabilität und speziell die vier Eulerschen Knickfälle für ein schlankes Bauteil unter Drucklast.

*Verstehen*

Die Studierenden

- können Kräfte nach verschiedenen Kriterien klassifizieren.
- können verschiedene Lagerungsarten unterscheiden und die entsprechenden Lagerreaktionen angeben.
- können den Unterschied zwischen statisch bestimmten und unbestimmten Systemen erklären.
- können den Unterschied zwischen Haft- und Gleitreibung erläutern.
- können das linear-elastische, isotrope Materialgesetz angeben und die Bedeutung der Konstanten erläutern.
- können die Voraussetzungen der Euler-Bernoulli-Theorie schlanker Balken erklären.
- verstehen die Idee der Vergleichsspannung und können verschiedene Festigkeitshypothesen erklären.

*Anwenden*

- Die Studierenden können den Schwerpunkt eines Körpers bestimmen.
- Die Studierenden können ein System aus mehreren Körpern geeignet freischneiden und die entsprechenden eingepägten Kraftgrößen und die Reaktionsgrößen eintragen.

- Die Studierenden können für ein statisch bestimmtes System die Reaktionsgrößen aus den Gleichgewichtsbedingungen ermitteln.
- Die Studierenden können die Schnittreaktionen für Stäbe und Balken bestimmen.
- Die Studierenden können die Spannungen im Querschnitt schlanker Bauteile (Stab, Balken) unter verschiedenen Belastungen (Zug, Biegung, Torsion) ermitteln.
- Die Studierenden können die Verformungen schlanker Bauteile ermitteln.
- Die Studierenden können aus einem gegebenen, allgemeinen Spannungszustand die Hauptspannungen sowie verschiedene Vergleichsspannungen ermitteln.
- Die Studierenden können die kritische Knicklast für einen gegebenen Knickfall bestimmen.

#### *Analysieren*

- Die Studierenden können ein geeignetes Modell für schlanke Bauteile anhand der Belastungsart und Geometrie auswählen.
- Die Studierenden können ein problemangepasstes Berechnungsverfahren zur Ermittlung von Reaktionsgrößen und Verformungen auch an statisch unbestimmten Systemen wählen.
- Die Studierenden können eine geeignete Festigkeitshypothese wählen.
- Die Studierenden können den relevanten Knickfall für gegebene Randbedingungen identifizieren.

#### *Evaluieren (Beurteilen)*

- Die Studierenden können den Spannungszustand in einem Bauteil hinsichtlich Aspekten der Festigkeit bewerten.
- Die Studierenden können den Spannungszustand in einem schlanken Bauteil hinsichtlich Aspekten der Stabilität bewerten.

#### **Literatur:**

- Gross, Hauger, Schnell, Wall: Technische Mechanik 1, Berlin:Springer 2006
- Gross, Hauger, Schnell, Wall: Technische Mechanik 2, Berlin:Springer 2007

#### **Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

##### **[1] Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Nebenfach | Nebenfach Maschinenbau | Technische Mechanik)

Dieses Modul ist daneben auch in den Studienfächern "Berufspädagogik Technik (Bachelor of Science)", "Chemical Engineering - Nachhaltige Chemische Technologien (Bachelor of Science)", "Chemie- und Bioingenieurwesen (Bachelor of Science)", "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Bachelor of Science)", "Energietechnik (Bachelor of Science)", "Informatik (Master of Science)", "International Production Engineering and Management (Bachelor of Science)", "Life Science Engineering (Bachelor of Science)", "Materialwissenschaft und Werkstofftechnik (Bachelor of Science)", "Mechatronik (Bachelor of Science)", "Medizintechnik (Bachelor of Science)", "Technomathematik (Bachelor of Science)", "Werkstoffwissenschaften (Bachelor of Science)", "Wirtschaftsingenieurwesen (Bachelor of Science)" verwendbar.

#### **Studien-/Prüfungsleistungen:**

Statik und Festigkeitslehre (Prüfungsnummer: 46601)

(englische Bezeichnung: Statics and Strength of Materials)

Prüfungsleistung, Klausur, Dauer (in Minuten): 90

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Statik und Festigkeitslehre
- Tutorium zur Statik und Festigkeitslehre
- Übungen zur Statik und Festigkeitslehre

Erstablesung: WS 2019/2020, 1. Wdh.: keine Angabe

1. Prüfer: Willner/Leyendecker (ps1091)

#### **Organisatorisches:**

Organisatorisches, Termine & Downloads auf StudOn



**Modulbezeichnung: Grundlagen der Messtechnik (GMT)** **5 ECTS**  
(Fundamentals of Metrology)

Modulverantwortliche/r: Tino Hausotte  
Lehrende: Tino Hausotte, Assistenten

Startsemester: SS 2019	Dauer: 1 Semester	Turnus: halbjährlich (WS+SS)
Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Deutsch und Englisch

**Lehrveranstaltungen:**

Fundamentals of Metrology - Grundlagen der Messtechnik (SS 2019, Vorlesung, 2 SWS, Tino Hausotte)  
Fundamentals of Metrology - Grundlagen der Messtechnik - Übung (SS 2019, Übung, 2 SWS, Tino Hausotte et al.)

**Inhalt:**

**Allgemeine Grundlagen**

- **Was ist Metrologie:** Metrologie und Teilgebiete, Einsatzbereiche, historische Entwicklung des Einheitssystems, SI-Einheitensystem - SI-Einheiten (cd, K, kg, m, s, A, mol) - Größe, Größenwert - Extensive und intensive Größen - Messung, Messgröße, Maßeinheit, Messergebnis, Messwert, Gebrauch und korrekte Angabe der Einheiten, Schreibweisen von Größenwerten, Angabe von Einheiten - Grundvoraussetzungen für das Messen - Rückführung der Einheiten
- **Messprinzipien, Messmethoden und Messverfahren:** Messprinzip, Messmethode, Messverfahren - Einteilung der Messmethoden, Ausschlagmessmethode, Differenzmessmethode, Substitutionsmessmethode und Nullabgleichsmethode (Kompensationsmethode) - Prinzip eines Messgerätes, direkte und indirekte Messmethoden - Kennlinie und Kennlinienarten, analoge und digitale Messmethoden, kontinuierliche und diskontinuierliche Messung, Auflösung, Empfindlichkeit, Messbereich - absolute und inkrementelle Messmethoden
- **Statistik - Auswertung von Messreihen:** Berechnung eines Messergebnisses anhand von Messreihen - Grundbegriffe der deskriptiven Statistik - Darstellung und Interpretation von Messwertverteilungen (Histogramme) - Häufigkeit (absolute, relative, kumulierte, relative kumulierte) - Berechnung und Interpretation grundlegender Parameter: Lage (Mittelwert, Median, Modus), Streuung (Spannweite, Varianz, Standardabweichung, Variationskoeffizient), Form (Schiefe, Kurtosis bzw. Exzess) - Grundbegriffe der Stochastik, Wahrscheinlichkeiten, Verteilungen (Rechteck-, U- und Normalverteilung), Zentraler Grenzwertsatz, statistische Momente - Grundbegriffe der analytischen Statistik, statistische Tests und statistische Schätzverfahren - Korrelation und Regression
- **Messabweichungen und Messunsicherheit:** Messwert, wahrer Wert, Ringvergleich, vereinbarter Wert - Einflüsse auf die Messung (Ishikawa-Diagramm) - Messabweichung (absolute, relative, systematische, zufällige) - Umgang mit Messabweichungen, Korrektur bekannter systematischer Messabweichungen - Kalibrierung, Verifizierung, Eichung - Messpräzision, Messgenauigkeit, Messrichtigkeit - Wiederholbedingungen/-präzision, Vergleichsbedingungen/-präzision, Erweiterte Vergleichsbedingungen/-präzision - Fehlerfortpflanzungsgesetz (altes Konzept), Messunsicherheit, Eigenunsicherheit, Übersicht über Standardverfahren des GUM (Messunsicherheit), korrekte Angabe eines Messergebnisses

**Messgrößen des SI-Einheitensystems**

- **Messen elektrischer Größen und digitale Messtechnik:** SI-Basiseinheit Ampere, Widerstands- und Spannungsnormale, Messung von Strom und Spannung, Lorentzkraft, Drehspulmesswerk, Bereichsanpassung - Widerstandsmessung, strom- und spannungsrichtige Messung, Wheatstone'sche Brückenschaltung (Viertel-, Halb- und Vollbrücke, Differenzmethode und Kompensationsmethode) - Charakteristische Werte sinusförmiger Wechselgrößen, Dreheisenmesswerk, Wechselspannungsbrücke - Messsignale, dynamische Kennfunktionen und Kennwerte, Übertragungsfunktionen (Frequenzgänge) - Digitalisierungskette, Zeit- und Wertdiskretisierung, Alias-Effekte, Shannon's Abtasttheorem, Filter, Operationsverstärker (Invertierender Verstärker, Nichtinvertierender Verstärker, Impedanzwandler, invertierender Addierer, Differenzverstärker, Integrierer, Differenzierer, Instrumentenverstärker), Abtast-Halte-Glied, Analog-Digital-Wandlung, Abweichungen bei der Analog-Digital-

Wandlung - Universelle Messgeräte (Digitalmultimeter, analoge und digitale Oszilloskope)

- **Messen optischer Größen:** Licht und Eigenschaften des Lichtes - Empfindlichkeitsspektrum des Auges - Radiometrie und Photometrie - SI-Basiseinheit Candela (cd, Lichtstärke) - Strahlungsfluss, radiometrisches (fotometrisches) Grundgesetz, photometrische und radiometrische Größen - Strahlungsgesetze - Fotodetektoren (Fotowiderstände, Fotodioden, Betriebsarten, Bauformen, CCD- und CMOS-Sensoren)
- **Messen von Temperaturen:** Temperatur, SI-Basiseinheit Kelvin, Definition, Wärmeübertragung (Wärmeleitung, Konvektion, Wärmestrahlung) - Thermodynamische Temperatur - Primäre und sekundäre Temperaturmessverfahren, praktische Temperaturskalen, Fixpunkte (Tripelpunkte, Erstarrungspunkte), Fixpunktzellen, klassische Temperaturskalen, internationale Temperaturskala (ITS-90) - Berührungsthermometer, thermische Messabweichungen, thermische Ausdehnung, Gasthermometer, Flüssigkeitsglasthermometer, Bimetall-Thermometer, Metall-Widerstandsthermometer (Kennlinie, Genauigkeit, Bauformen, Messschaltungen), Thermoelemente (Seebeck-Effekt, Bauformen, Ausgleichsleitungen, Messschaltungen) - Strahlungsthermometer (Prinzip, Strahlungsgesetze, Pyrometer, Messabweichungen)
- **Zeit und Frequenz:** SI - Basiseinheit Sekunde, Zeitmessung (Aufgaben, Historie, mechanische Uhren, Quarzuhren, Atomuhr) - Darstellung der Zeit - Verbreitung der Zeitskala UTC - Globales Positionssystem (GPS) - Frequenz- und Phasenwinkelmessung
- **Längenmesstechnik:** SI - Basiseinheit Meter - Messschieber, Abbe'sches Komparatorprinzip, Bügelmessschraube, Abweichungen 1.- und 2.-Ordnung - Längenmessung mit Linearencodern (Bewegungsrichtung, Ausgangssignale, Differenzsignale, Demodulation) - Absolutkodierung (V-Scannen und Gray Code) - Interferometrie, Michelson-Interferometer, transversale elektromagnetische Wellen, Grundlagen der Interferenz, destruktive und konstruktive Interferenz, Homodynprinzip, Heterodynprinzip, Interferenz am Homodyninterferometer, Demodulation am Homodyn- und Heterodyninterferometer, Einfluss Luftbrechzahl, Realisierung der Meterdefinition, Reflektoren und Aufbau von Interferometern, induktive Längenmessung, kapazitive Längenmessung, Laufzeitmessung
- **Masse, Kraft und Drehmoment:** SI - Basiseinheit Kilogramm, Definition Masse, Kraft und Drehmoment - Massenormale (Vergleiche, Bauformen und Abweichungsgrenzen), Prinzip der Masseableitung, Stabilität der Einheit und Neudefinition - Messprinzipien von Waagen, Einflussgrößen bei Massebestimmung (lokale Erdbeschleunigung, Luftauftrieb), Balkenwaage (unterschälige Waagen, Empfindlichkeit, Bauformen, oberhalbige Waagen, Ecklastabhängigkeit), Federwaage, DMS, Verformungskörper, DMS-Waage, EMK-Waage, Massekomparatoren - Drehmomentmessung (Reaktions- und Aktionsdrehmoment)

#### Teilgebiete der industriellen Messtechnik

- **Prozessmesstechnik:** Messgrößen der Prozessmesstechnik - Definition des Druckes, Druckarten (Absolutdruck, Überdruck, Differenzdruck) - Druckwaage (Kolbenmanometer), U-Rohrmanometer und -Barometer, Rohrfedermanometer, Plattenfedermanometer - Drucksensoren (mit DMS, piezoresistiv, kapazitiv, piezoelektrisch) - Durchflussmessung (Volumenstrom und Massestrom, Strömung von Fluiden) - volumetrische Verfahren, Wirkdruckverfahren, magnetisch-induktive Durchflussmessung, Ultraschall-Durchflussmessung - Massedurchflussmessung (Coriolis, thermisch)
- **Fertigungsmesstechnik:** Aufgaben, Methoden, Ziele und Bereiche der Fertigungsmesstechnik - Gestaltparameter von Werkstücken (Mikro- und Makrogestalt), Geometrische Produktspezifikation (GPS), Gestaltabweichungsarten - Geräte und Hilfsmittel der Fertigungsmesstechnik, Gegenüberstellung klassische Fertigungsmesstechnik und Koordinatenmesstechnik, Auswertung - Bauarten und Grundstruktur von Koordinatenmessgeräten - Vorgehensweise bei Messen mit einem Koordinatenmessgerät

#### Inhalt (Übung)

- Grundlagen der Elektrotechnik (Wiederholung von Grundlagen)
- Statistik - Auswertung von Messreihen (Histogramme, Hypothesentest, Konfidenzintervalle, statistischen Maßzahlen)
- Korrelation und Regression (Korrelationskoeffizient, Fehlerfortpflanzung, Residuenanalyse)

- Messabweichungen, Einführung in die Messunsicherheitsberechnung (Kompensation systematischer Abweichungen, Messunsicherheitsanalyse einer einfachen Messung)
- Elektrische Größen, Messelektronik und Analog-Digital-Umsetzung (Abweichungsberechnung bei der Strommessung, Anpassungsnetzwerk für ein Drehspulinstrument, Bereichsanpassung mit einem Operationsverstärker)
- Anwendung der Wheatstone'schen Brückenschaltung bei Messungen mit Dehnungsmessstreifen
- Messungen mit Fotodioden bei unterschiedlichen Betriebsarten
- Temperaturmesstechnik (Aufgaben zu Metall-Widerstandsthermometern und Pyrometern)
- Längenmesstechnik (Abbe'sche Prinzip, Induktivität eines Eisenkerns mit Luftspalt, Foliendickenmessung mittels einer kapazitiven Messeinrichtung)
- Messen von Kraft und Masse (Massewirkung, Balkenwaage, Federwaage, piezoelektrischer Kraftsensor)
- Prozessmesstechnik (Druck- und Durchflussmessung, U-Rohrmanometer, Corioliskraftmessung, Ultraschallmessverfahren, Turbinenzähler)
- Fertigungsmesstechnik (Standardgeometrieelemente, Angabe von Toleranzen, Prüfen von Rundheitsabweichungen mit Hilfe eines Feinzeigers)

## Contents (Lecture)

### General basics

- **What is metrology:** Metrology and branches, application fields, historical development of the unit system, SI unit system - Definitions of SI units (cd, K, kg, m, s, A, mol) - Quantity, quantity value - Extensive and intensive quantities - Measurement, measurand, measurement unit, measurement result, measured quantity value - Correct use and notation of units and of quantity values - Basic requirements for the measurement - Traceability
- **Principles, methods and procedures of measurement:** Principles, methods and procedures of measurement - Classification of measurement methods, deflection, differential, substitution and compensation measurement methods - Principle of a measuring instrument, direct and indirect measurement methods - Characteristic curve, types of characteristic curves, analogue and digital measurement methods, continuous and discontinuous measurement, resolution, sensitivity, measuring interval - Absolute and incremental measurement methods
- **Statistics - Evaluation of measurements series:** Calculation of a measurement result based on measurement series - Basic terms of descriptive statistics - Presentation and interpretation of measured value distributions (histograms) - Frequency (absolute, relative, cumulative, relative cumulative) - Calculation and interpretation of basic parameters: location (mean, median, mode), dispersion (range, variance, standard deviation, coefficient of variation), shape (skewness, excess, kurtosis) - Basic terms of stochastics, probabilities, distributions (rectangle, U and normal distribution), central limit theorem, statistical moments - Basic terms of analytical statistics, statistical tests and statistical estimation methods - Correlation and regression
- **Measurement errors and measurement uncertainty:** Measured value, true value, key comparison, conventional quantity value - Influences on the measurement (Ishikawa diagram) - Measurement error (absolute, relative, systematic, random) - Handling of errors, correction of known systematic measurement errors - Calibration, verification, legal verification - Measurement precision, accuracy and trueness - Repeatability conditions and repeatability, intermediate precision condition and measurement precision, reproducibility condition of measurement and reproducibility - Error propagation law (old concept), measurement uncertainty, definitional uncertainty, overview of standard method of the GUM (measurement uncertainty), correct specification of a measurement result

### Mesurands of the SI system of units

- **Measurement of electrical quantities:** SI base unit Ampere, resistance and voltage standards, measurement of current and voltage, Lorentz force, moving coil instrument, range adjustment - Resistance measurement, current and voltage correct measurement, Wheatstone bridge circuit (quarter, half and full bridge, differential method and compensation method) - Characteristic values of

sinusoidal alternating quantities, moving iron instrument, alternating voltage bridge - Measuring signals, dynamic characteristic functions and characteristics, transfer functions (frequency responses) - Digitalisation chain, time and value discretization, aliasing, Shannon's sampling theorem, filter, operational amplifier (inverting amplifier, non-inverting amplifier, impedance converter, inverting summing amplifier, differential amplifier, integrating amplifier, differentiating amplifier, instrumentation amplifier), sample-and-hold device, analogue-digital conversion, errors of analogue-to-digital conversion - Universal measuring devices (digital multimeter, analogue and digital oscilloscopes)

- **Measurement of optical quantities:** Light and properties of light - Sensitivity spectra of the eye - Radiometry and photometry - SI base unit candela (cd, luminous intensity) - Radiant flux, radiometric (photometric) fundamental law, photometric and radiometric quantities - Radiation laws - Photo detectors (photo resistors, photo diodes, modes of operation, designs, CCD and CMOS sensors)
- **Measurement of temperatures:** Temperature, SI base unit Kelvin, definition, heat transfer (conduction, convection, radiation) - Thermodynamic temperature - Primary and secondary temperature measurement methods, practical temperature scales, fixpoints (triple points, freezing points), fixpoint cells, classical temperature scales, International Temperature Scale (ITS-90) - Contact thermometers, thermal measurement errors, thermal expansion, gas thermometer, liquid thermometer, bimetal thermometer, metal resistance thermometers (characteristic curve, accuracy, designs, circuits), thermocouples (Seebeck effect, designs, extension wires, measurement circuits) - Radiation thermometer (principle, radiation laws, pyrometers, measurement errors)
- **Time and frequency:** SI base unit second, time measurement (tasks, history, mechanical clocks, quartz clock, atomic clock) - Representation of time - Propagation of UTC - Global Positioning System (GPS) - Frequency and phase angle measurement
- **Length:** SI base unit metre - Calliper, Abbe comparator principle, micrometer, errors 1st and 2nd order - Length measurement with linear encoders (motion direction, output signals, differential signals, demodulation) - Absolute coding (V-Scan and Gray code) - Interferometry, Michelson interferometer, transversal electromagnetic waves, basics of interference, destructive and constructive interference, homodyne principle, heterodyne principle, interference on homodyne interferometer, demodulation at homodyne and heterodyne interferometer, influence of air refractive index, realisation of the metre definition, reflectors and assembly of interferometers, inductive length measurement, capacitive length measurement, time of flight measurement
- **Mass, force and torque:** SI - base unit kilogram, definition of mass, force and torque - Mass standards (comparisons, types, deviation limits), principle of mass dissemination, stability of the unit and redefinition - Measurement principles of weighing, influences for mass determination (local gravitational acceleration, air buoyancy), beam balance (hanging pan balances, sensitivity, types, top pan balances, corner load sensitivity), spring balance, DMS, deformation elements, DMS balance, EMC balance, mass comparators - Measurement of torque (reactive and active)

### Branches of industrial metrology

- **Process measurement technology:** Quantities of process measurement technology - Definition of pressure, pressure types (absolute pressure, overpressure, differential pressure) - Deadweight tester (piston manometer), U-tube manometer and barometer, bourdon tube gauge, diaphragm pressure gauge - Pressure sensors (with DMS, piezoresistive, capacitive, piezoelectric) - Flow measurement (volume flow and mass flow, flow of fluids) - Volumetric method, differential pressure method, magneto-inductive flowmeter, ultrasonic flow measurement - Mass flow rate measurement (Coriolis, thermal)
- **Manufacturing metrology:** Tasks, methods, objectives and branches of manufacturing metrology - Form parameters of workpieces (micro-and macro-shape), geometrical product specification (GPS), geometrical tolerances - Comparison of classical manufacturing metrology and coordinate metrology, evaluation - Designs and basic structure of coordinate measuring machines - Procedure for measuring with a coordinate measuring machine

### Lernziele und Kompetenzen:

*Fachkompetenz*

*Wissen*

- Die Studierenden kennen das Basiswissen zu Grundlagen der Messtechnik und messtechnischen Tätigkeiten.
- Die Studierenden haben Grundkenntnisse zur methodisch-operativen Herangehensweise an Aufgaben des Messens statischer Größen, zum Lösen einfacher Messaufgaben und zum Ermitteln von Messergebnissen aus Messwerten.

#### *Verstehen*

- Die Studierenden können die Eigenschaften von Messeinrichtungen und Messprozessen beschreiben.
- Die Studierenden können das Internationale Einheitensystem und die Rückführung von Messergebnissen beschreiben.

#### *Anwenden*

- Die Studierenden können einfache Messungen statischer Größen durchführen.

#### *Evaluieren (Beurteilen)*

- Die Studierenden können Messeinrichtungen, Messprozesse und Messergebnisse bewerten.

#### **Literatur:**

- International Vocabulary of Metrology - Basic and General Concepts and Associated Terms, VIM, 3rd edition, JCGM 200:2008, <http://www.bipm.org/en/publications/guides/vim.html>
- DIN e.V. (Hrsg.): Internationales Wörterbuch der Metrologie - Grundlegende und allgemeine Begriffe und zugeordnete Benennungen (VIM) ISO/IEC-Leitfaden 99:2007. Korrigierte Fassung 2012, Beuth Verlag GmbH, 4. Auflage 2012
- Hoffmann, Jörg: Handbuch der Messtechnik. 4. Auflage, Carl Hanser Verlag München, 2012 - ISBN 978-3-446-42736-5
- Lerch, Reinhard: Elektrische Messtechnik. 6. Auflage, Springer-Verlag Berlin Heidelberg, 2012 - ISBN 978-3-642-22608-3
- Richter, Werner: Elektrische Meßtechnik. 3. Auflage, Verlag Technik Berlin, 1994 - ISBN 3-341-01106-4
- Kohlrausch, Friedrich: Praktische Physik : zum Gebrauch für Unterricht, Forschung und Technik. Band 1-3, 24. Auflage, Teubner Verlag, 1996 - ISBN 3-519-23001-1, 3-519-23002-X, 3-519-23000-3
- H. Czichos (Hrsg.): Das Ingenieurwissen Gebundene. 7. Auflage, Springer Verlag, 2012, ISBN 978-3-642-22849-0.
- Ernst, Alfons: Digitale Längen- und Winkelmesstechnik. 4. Auflage, Verlag Moderne Industrie, 2001 - ISBN 3-478-93264-5
- Pfeifer, Tilo: Fertigungsmeßtechnik. R. Oldenbourg Verlag München Wien, 1998 - ISBN 3-486-24219-9
- Keferstein, Claus P.: Fertigungsmesstechnik. 7. Auflage, Vieweg+Teubner Verlag, 2011 - ISBN 978-3-8348-0692-5
- Warnecke, H.-J.; Dutschke, W.: Fertigungsmeßtechnik. Springer-Verlag Berlin Heidelberg New York Tokyo, 1984 - ISBN 3-540-11784-9

---

#### **Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

##### **[1] Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Nebenfach | Nebenfach Maschinenbau | Qualitätsmanagement und Messtechnik)

Dieses Modul ist daneben auch in den Studienfächern "Berufspädagogik Technik (Bachelor of Science)", "Berufspädagogik Technik (Master of Education)", "Energietechnik (Bachelor of Science)", "Informatik (Master of Science)", "Maschinenbau (Bachelor of Science)", "Mechatronik (Bachelor of Science)", "Medizintechnik (Bachelor of Science)", "Technomathematik (Bachelor of Science)", "Wirtschaftsingenieurwesen (Bachelor of Science)", "Wirtschaftsingenieurwesen (Master of Science)" verwendbar.

---

#### **Studien-/Prüfungsleistungen:**

Grundlagen der Messtechnik (Prüfungsnummer: 45101)

(englische Bezeichnung: Fundamentals of Metrology)

Prüfungsleistung, Klausur, Dauer (in Minuten): 60

Anteil an der Berechnung der Modulnote: 100%

weitere Erläuterungen:

Die Klausur kann teilweise Multiple-Choice Aufgaben enthalten.

**Prüfungstermine**, eine **allgemeine Regel der Prüfungstagvergabe** und **Termine der Klausureinsicht** finden Sie auf StudOn: Prüfungstermine und Termine der Klausureinsicht

Erstablingung: SS 2019, 1. Wdh.: WS 2019/2020

1. Prüfer: Tino Hausotte (100620)

---

### Organisatorisches:

- Unterlagen zur Lehrveranstaltung werden auf der Lernplattform StudOn ([www.studon.uni-erlangen.de](http://www.studon.uni-erlangen.de)) bereitgestellt. Das Passwort wird in der ersten Vorlesung bekannt gegeben.

**Modulbezeichnung: Grundlagen der Messtechnik (GMT)**  
 (Fundamentals of Metrology)

**5 ECTS**

Modulverantwortliche/r: Tino Hausotte

Lehrende: Tino Hausotte, Andreas Gröschl, Martin Heint

Startsemester: WS 2019/2020

Dauer: 1 Semester

Turnus: halbjährlich (WS+SS)

Präsenzzeit: 60 Std.

Eigenstudium: 90 Std.

Sprache: Deutsch und Englisch

### Lehrveranstaltungen:

Grundlagen der Messtechnik (WS 2019/2020, Vorlesung, 2 SWS, Tino Hausotte)

Grundlagen der Messtechnik - Übung (WS 2019/2020, Übung, 2 SWS, Tino Hausotte et al.)

### Inhalt:

#### Allgemeine Grundlagen

- **Was ist Metrologie:** Metrologie und Teilgebiete, Einsatzbereiche, historische Entwicklung des Einheitssystems, SI-Einheitensystem - SI-Einheiten (cd, K, kg, m, s, A, mol) - Größe, Größenwert - Extensive und intensive Größen - Messung, Messgröße, Maßeinheit, Messergebnis, Messwert, Gebrauch und korrekte Angabe der Einheiten, Schreibweisen von Größenwerten, Angabe von Einheiten - Grundvoraussetzungen für das Messen - Rückführung der Einheiten
- **Messprinzipien, Messmethoden und Messverfahren:** Messprinzip, Messmethode, Messverfahren - Einteilung der Messmethoden, Ausschlagmessmethode, Differenzmessmethode, Substitutionsmessmethode und Nullabgleichsmethode (Kompensationsmethode) - Prinzip eines Messgerätes, direkte und indirekte Messmethoden - Kennlinie und Kennlinienarten, analoge und digitale Messmethoden, kontinuierliche und diskontinuierliche Messung, Auflösung, Empfindlichkeit, Messbereich - absolute und inkrementelle Messmethoden
- **Statistik - Auswertung von Messreihen:** Berechnung eines Messergebnisses anhand von Messreihen - Grundbegriffe der deskriptiven Statistik - Darstellung und Interpretation von Messwertverteilungen (Histogramme) - Häufigkeit (absolute, relative, kumulierte, relative kumulierte) - Berechnung und Interpretation grundlegender Parameter: Lage (Mittelwert, Median, Modus), Streuung (Spannweite, Varianz, Standardabweichung, Variationskoeffizient), Form (Schiefe, Kurtosis bzw. Exzess) - Grundbegriffe der Stochastik, Wahrscheinlichkeiten, Verteilungen (Rechteck-, U- und Normalverteilung), Zentraler Grenzwertsatz, statistische Momente - Grundbegriffe der analytischen Statistik, statistische Tests und statistische Schätzverfahren - Korrelation und Regression
- **Messabweichungen und Messunsicherheit:** Messwert, wahrer Wert, Ringvergleich, vereinbarter Wert - Einflüsse auf die Messung (Ishikawa-Diagramm) - Messabweichung (absolute, relative, systematische, zufällige) - Umgang mit Messabweichungen, Korrektur bekannter systematischer Messabweichungen - Kalibrierung, Verifizierung, Eichung - Messpräzision, Messgenauigkeit, Messrichtigkeit - Wiederholbedingungen/-präzision, Vergleichsbedingungen/-präzision, Erweiterte Vergleichsbedingungen/-präzision - Fehlerfortpflanzungsgesetz (altes Konzept), Messunsicherheit, Eigenunsicherheit, Übersicht über Standardverfahren des GUM (Messunsicherheit), korrekte Angabe eines Messergebnisses

#### Messgrößen des SI-Einheitensystems

- **Messen elektrischer Größen und digitale Messtechnik:** SI-Basiseinheit Ampere, Widerstands- und Spannungsnormale, Messung von Strom und Spannung, Lorentzkraft, Drehspulmesswerk, Bereichsanpassung - Widerstandsmessung, strom- und spannungsrichtige Messung, Wheatstone'sche Brückenschaltung (Viertel-, Halb- und Vollbrücke, Differenzmethode und Kompensationsmethode) - Charakteristische Werte sinusförmiger Wechselgrößen, Dreheisenmesswerk, Wechselspannungsbrücke - Messsignale, dynamische Kennfunktionen und Kennwerte, Übertragungsfunktionen (Frequenzgänge) - Digitalisierungskette, Zeit- und Wertdiskretisierung, Alias-Effekte, Shannon's Abtasttheorem, Filter, Operationsverstärker (Invertierender Verstärker, Nichtinvertierender Verstärker, Impedanzwandler, invertierender Addierer, Differenzverstärker, Integrierer, Differenzierer, Instrumentenverstärker), Abtast-Halte-Glied, Analog-Digital-Wandlung, Abweichungen bei der Analog-Digital-Wandlung - Universelle Messgeräte (Digitalmultimeter, analoge und digitale Oszilloskope)

- **Messen optischer Größen:** Licht und Eigenschaften des Lichtes - Empfindlichkeitsspektrum des Auges - Radiometrie und Photometrie - SI-Basiseinheit Candela (cd, Lichtstärke) - Strahlungsfluss, radiometrisches (fotometrisches) Grundgesetz, photometrische und radiometrische Größen - Strahlungsgesetze - Fotodetektoren (Fotowiderstände, Fotodioden, Betriebsarten, Bauformen, CCD- und CMOS-Sensoren)
- **Messen von Temperaturen:** Temperatur, SI-Basiseinheit Kelvin, Definition, Wärmeübertragung (Wärmeleitung, Konvektion, Wärmestrahlung) - Thermodynamische Temperatur - Primäre und sekundäre Temperaturmessverfahren, praktische Temperaturskalen, Fixpunkte (Tripelpunkte, Erstarrungspunkte), Fixpunktzellen, klassische Temperaturskalen, internationale Temperaturskala (ITS-90) - Berührungsthermometer, thermische Messabweichungen, thermische Ausdehnung, Gasthermometer, Flüssigkeitsglaskthermometer, Bimetall-Thermometer, Metall-Widerstandsthermometer (Kennlinie, Genauigkeit, Bauformen, Messschaltungen), Thermoelemente (Seebeck-Effekt, Bauformen, Ausgleichsleitungen, Messschaltungen) - Strahlungsthermometer (Prinzip, Strahlungsgesetze, Pyrometer, Messabweichungen)
- **Zeit und Frequenz:** SI - Basiseinheit Sekunde, Zeitmessung (Aufgaben, Historie, mechanische Uhren, Quarzuhren, Atomuhr) - Darstellung der Zeit - Verbreitung der Zeitskala UTC - Globales Positionssystem (GPS) - Frequenz- und Phasenwinkelmessung
- **Längenmesstechnik:** SI - Basiseinheit Meter - Messschieber, Abbe'sches Komparatorprinzip, Bügelmessschraube, Abweichungen 1.- und 2.-Ordnung - Längenmessung mit Linearencodern (Bewegungsrichtung, Ausgangssignale, Differenzsignale, Demodulation) - Absolutkodierung (V-Scannen und Gray Code) - Interferometrie, Michelson-Interferometer, transversale elektromagnetische Wellen, Grundlagen der Interferenz, destruktive und konstruktive Interferenz, Homodynprinzip, Heterodynprinzip, Interferenz am Homodyninterferometer, Demodulation am Homodyn- und Heterodyninterferometer, Einfluss Luftbrechzahl, Realisierung der Meterdefinition, Reflektoren und Aufbau von Interferometern, induktive Längenmessung, kapazitive Längenmessung, Laufzeitmessung
- **Masse, Kraft und Drehmoment:** SI - Basiseinheit Kilogramm, Definition Masse, Kraft und Drehmoment - Massenormale (Vergleiche, Bauformen und Abweichungsgrenzen), Prinzip der Masseableitung, Stabilität der Einheit und Neudefinition - Messprinzipien von Waagen, Einflussgrößen bei Massebestimmung (lokale Erdbeschleunigung, Luftauftrieb), Balkenwaage (unterschälige Waagen, Empfindlichkeit, Bauformen, oberhalbige Waagen, Ecklastabhängigkeit), Federwaage, DMS, Verformungskörper, DMS-Waage, EMK-Waage, Massekomparatoren - Drehmomentmessung (Reaktions- und Aktionsdrehmoment)

### Teilgebiete der industriellen Messtechnik

- **Prozessmesstechnik:** Messgrößen der Prozessmesstechnik - Definition des Druckes, Druckarten (Absolutdruck, Überdruck, Differenzdruck) - Druckwaage (Kolbenmanometer), U-Rohrmanometer und -Barometer, Rohrfedermanometer, Plattenfedermanometer - Drucksensoren (mit DMS, piezoresistiv, kapazitiv, piezoelektrisch) - Durchflussmessung (Volumenstrom und Massestrom, Strömung von Fluiden) - volumetrische Verfahren, Wirkdruckverfahren, magnetisch-induktive Durchflussmessung, Ultraschall-Durchflussmessung - Massedurchflussmessung (Coriolis, thermisch)
- **Fertigungsmesstechnik:** Aufgaben, Methoden, Ziele und Bereiche der Fertigungsmesstechnik - Gestaltparameter von Werkstücken (Mikro- und Makrogestalt), Geometrische Produktspezifikation (GPS), Gestaltabweichungsarten - Geräte und Hilfsmittel der Fertigungsmesstechnik, Gegenüberstellung klassische Fertigungsmesstechnik und Koordinatenmesstechnik, Auswertung - Bauarten und Grundstruktur von Koordinatenmessgeräten - Vorgehensweise bei Messen mit einem Koordinatenmessgerät

### Inhalt (Übung)

- Grundlagen der Elektrotechnik (Wiederholung von Grundlagen)
- Statistik - Auswertung von Messreihen (Histogramme, Hypothesentest, Konfidenzintervalle, statistischen Maßzahlen)
- Korrelation und Regression (Korrelationskoeffizient, Fehlerfortpflanzung, Residuenanalyse)
- Messabweichungen, Einführung in die Messunsicherheitsberechnung (Kompensation systematischer Abweichungen, Messunsicherheitsanalyse einer einfachen Messung)



- Elektrische Größen, Messelektronik und Analog-Digital-Umsetzung (Abweichungsberechnung bei der Strommessung, Anpassungsnetzwerk für ein Drehspulinstrument, Bereichsanpassung mit einem Operationsverstärker)
- Anwendung der Wheatstone'schen Brückenschaltung bei Messungen mit Dehnungsmessstreifen
- Messungen mit Fotodioden bei unterschiedlichen Betriebsarten
- Temperaturmesstechnik (Aufgaben zu Metall-Widerstandsthermometern und Pyrometern)
- Längenmesstechnik (Abbe'sche Prinzip, Induktivität eines Eisenkerns mit Luftspalt, Foliendickenmessung mittels einer kapazitiven Messeinrichtung)
- Messen von Kraft und Masse (Massewirkung, Balkenwaage, Federwaage, piezoelektrischer Kraftsensor)
- Prozessmesstechnik (Druck- und Durchflussmessung, U-Rohrmanometer, Corioliskraftmessung, Ultraschallmessverfahren, Turbinenzähler)
- Fertigungsmesstechnik (Standardgeometrielemente, Angabe von Toleranzen, Prüfen von Rundheitsabweichungen mit Hilfe eines Feinzeigers)

## Contents (Lecture)

### General basics

- **What is metrology:** Metrology and branches, application fields, historical development of the unit system, SI unit system - Definitions of SI units (cd, K, kg, m, s, A, mol) - Quantity, quantity value - Extensive and intensive quantities - Measurement, measurand, measurement unit, measurement result, measured quantity value - Correct use and notation of units and of quantity values - Basic requirements for the measurement - Traceability
- **Principles, methods and procedures of measurement:** Principles, methods and procedures of measurement - Classification of measurement methods, deflection, differential, substitution and compensation measurement methods - Principle of a measuring instrument, direct and indirect measurement methods - Characteristic curve, types of characteristic curves, analogue and digital measurement methods, continuous and discontinuous measurement, resolution, sensitivity, measuring interval - Absolute and incremental measurement methods
- **Statistics - Evaluation of measurements series:** Calculation of a measurement result based on measurement series - Basic terms of descriptive statistics - Presentation and interpretation of measured value distributions (histograms) - Frequency (absolute, relative, cumulative, relative cumulative) - Calculation and interpretation of basic parameters: location (mean, median, mode), dispersion (range, variance, standard deviation, coefficient of variation), shape (skewness, excess, kurtosis) - Basic terms of stochastics, probabilities, distributions (rectangle, U and normal distribution), central limit theorem, statistical moments - Basic terms of analytical statistics, statistical tests and statistical estimation methods - Correlation and regression
- **Measurement errors and measurement uncertainty:** Measured value, true value, key comparison, conventional quantity value - Influences on the measurement (Ishikawa diagram) - Measurement error (absolute, relative, systematic, random) - Handling of errors, correction of known systematic measurement errors - Calibration, verification, legal verification - Measurement precision, accuracy and trueness - Repeatability conditions and repeatability, intermediate precision condition and measurement precision, reproducibility condition of measurement and reproducibility - Error propagation law (old concept), measurement uncertainty, definitional uncertainty, overview of standard method of the GUM (measurement uncertainty), correct specification of a measurement result

### Mesurands of the SI system of units

- **Measurement of electrical quantities:** SI base unit Ampere, resistance and voltage standards, measurement of current and voltage, Lorentz force, moving coil instrument, range adjustment - Resistance measurement, current and voltage correct measurement, Wheatstone bridge circuit (quarter, half and full bridge, differential method and compensation method) - Characteristic values of sinusoidal alternating quantities, moving iron instrument, alternating voltage bridge - Measuring signals, dynamic characteristic functions and characteristics, transfer functions (frequency responses) - Digitalisation chain, time and value discretization, aliasing, Shannon's sampling theorem, filter,

operational amplifier (inverting amplifier, non-inverting amplifier, impedance converter, inverting summing amplifier, differential amplifier, integrating amplifier, differentiating amplifier, instrumentation amplifier), sample-and-hold device, analogue-digital conversion, errors of analogue-to-digital conversion - Universal measuring devices (digital multimeter, analogue and digital oscilloscopes)

- **Measurement of optical quantities:** Light and properties of light - Sensitivity spectra of the eye - Radiometry and photometry - SI base unit candela (cd, luminous intensity) - Radiant flux, radiometric (photometric) fundamental law, photometric and radiometric quantities - Radiation laws - Photo detectors (photo resistors, photo diodes, modes of operation, designs, CCD and CMOS sensors)
- **Measurement of temperatures:** Temperature, SI base unit Kelvin, definition, heat transfer (conduction, convection, radiation) - Thermodynamic temperature - Primary and secondary temperature measurement methods, practical temperature scales, fixpoints (triple points, freezing points), fixpoint cells, classical temperature scales, International Temperature Scale (ITS-90) - Contact thermometers, thermal measurement errors, thermal expansion, gas thermometer, liquid thermometer, bimetal thermometer, metal resistance thermometers (characteristic curve, accuracy, designs, circuits), thermocouples (Seebeck effect, designs, extension wires, measurement circuits) - Radiation thermometer (principle, radiation laws, pyrometers, measurement errors)
- **Time and frequency:** SI base unit second, time measurement (tasks, history, mechanical clocks, quartz clock, atomic clock) - Representation of time - Propagation of UTC - Global Positioning System (GPS) - Frequency and phase angle measurement
- **Length:** SI base unit metre - Calliper, Abbe comparator principle, micrometer, errors 1st and 2nd order - Length measurement with linear encoders (motion direction, output signals, differential signals, demodulation) - Absolute coding (V-Scan and Gray code) - Interferometry, Michelson interferometer, transversal electromagnetic waves, basics of interference, destructive and constructive interference, homodyne principle, heterodyne principle, interference on homodyne interferometer, demodulation at homodyne and heterodyne interferometer, influence of air refractive index, realisation of the metre definition, reflectors and assembly of interferometers, inductive length measurement, capacitive length measurement, time of flight measurement
- **Mass, force and torque:** SI - base unit kilogram, definition of mass, force and torque - Mass standards (comparisons, types, deviation limits), principle of mass dissemination, stability of the unit and redefinition - Measurement principles of weighing, influences for mass determination (local gravitational acceleration, air buoyancy), beam balance (hanging pan balances, sensitivity, types, top pan balances, corner load sensitivity), spring balance, DMS, deformation elements, DMS balance, EMC balance, mass comparators - Measurement of torque (reactive and active)

### Branches of industrial metrology

- **Process measurement technology:** Quantities of process measurement technology - Definition of pressure, pressure types (absolute pressure, overpressure, differential pressure) - Deadweight tester (piston manometer), U-tube manometer and barometer, bourdon tube gauge, diaphragm pressure gauge - Pressure sensors (with DMS, piezoresistive, capacitive, piezoelectric) - Flow measurement (volume flow and mass flow, flow of fluids) - Volumetric method, differential pressure method, magneto-inductive flowmeter, ultrasonic flow measurement - Mass flow rate measurement (Coriolis, thermal)
- **Manufacturing metrology:** Tasks, methods, objectives and branches of manufacturing metrology - Form parameters of workpieces (micro-and macro-shape), geometrical product specification (GPS), geometrical tolerances - Comparison of classical manufacturing metrology and coordinate metrology, evaluation - Designs and basic structure of coordinate measuring machines - Procedure for measuring with a coordinate measuring machine

### Lernziele und Kompetenzen:

#### Wissen

- Die Studierenden kennen grundlegende statistische Methoden zur Beurteilung von Messergebnissen und Ermittlung von Messunsicherheiten.
- Die Studierenden kennen grundlegende Messverfahren zur Erfassung der Messgrößen aller SI-Einheiten.

- Die Studierenden kennen das Basiswissen zu Grundlagen der Messtechnik und messtechnischen Tätigkeiten.
- Die Studierenden haben Grundkenntnisse zur methodisch-operativen Herangehensweise an Aufgaben des Messens statischer Größen, zum Lösen einfacher Messaufgaben und zum Ermitteln von Messergebnissen aus Messwerten.

#### **Verstehen**

- Die Studierenden können die Eigenschaften von Messeinrichtungen und Messprozessen beschreiben.
- Die Studierenden können das Internationale Einheitensystem und die Rückführung von Messergebnissen beschreiben.

#### **Anwenden**

- Die Studierenden können einfache Messungen statischer Größen durchführen.
- Die Studierenden können Messunsicherheiten komplexer Messeinrichtungen bei gegebenen Eingangsgrößen berechnen.

#### **Evaluieren (Beurteilen)**

- Die Studierenden können Messeinrichtungen, Messprozesse und Messergebnisse bewerten.

#### **Learning targets and competences:**

##### **Remembering**

- The students know basic statistical methods for the evaluation of measurement results and the determination of measurement uncertainties.
- The students know basic measuring methods for the record of measured values &#8203;&#8203;for all SI units.
- The students have basic knowledge of fundamentals of metrology and metrology activities.
- The students have fundamental knowledge for methodological and operational approach to measuring tasks of static measurement types, to solve basic measurement tasks and to establishing measurement results from measurement values.

##### **Understanding**

- The students are able to describe the characteristics of measuring instruments and measurement processes.
- The students are able to describe the international system of units (SI) and the traceability of measurement results

##### **Applying**

- The students are able to run basic measurements of static measurands.

##### **Evaluating**

- The students are able to evaluate measuring systems, measurement processes and measurement results.
- Students are able to calculate the measurement uncertainty of complex measuring systems for given input variables.

#### **Literatur:**

- International Vocabulary of Metrology - Basic and General Concepts and Associated Terms, VIM, 3rd edition, JCGM 200:2008, <http://www.bipm.org/en/publications/guides/vim.html>
- DIN e.V. (Hrsg.): Internationales Wörterbuch der Metrologie - Grundlegende und allgemeine Begriffe und zugeordnete Benennungen (VIM) ISO/IEC-Leitfaden 99:2007. Korrigierte Fassung 2012, Beuth Verlag GmbH, 4. Auflage 2012
- Hoffmann, Jörg: Handbuch der Messtechnik. 4. Auflage, Carl Hanser Verlag München, 2012 - ISBN 978-3-446-42736-5
- Lerch, Reinhard: Elektrische Messtechnik. 6. Auflage, Springer-Verlag Berlin Heidelberg, 2012 - ISBN 978-3-642-22608-3
- Richter, Werner: Elektrische Meßtechnik. 3. Auflage, Verlag Technik Berlin, 1994 - ISBN 3-341-01106-4
- Kohlrausch, Friedrich: Praktische Physik : zum Gebrauch für Unterricht, Forschung und Technik. Band 1-3, 24. Auflage, Teubner Verlag, 1996 - ISBN 3-519-23001-1, 3-519-23002-X, 3-519-23000-3
- H. Czichos (Hrsg.): Das Ingenieurwissen Gebundene. 7. Auflage, Springer Verlag, 2012, ISBN 978-3-642-22849-0.

- Ernst, Alfons: Digitale Längen- und Winkelmesstechnik. 4. Auflage, Verlag Moderne Industrie, 2001 - ISBN 3-478-93264-5
- Pfeifer, Tilo: Fertigungsmeßtechnik. R. Oldenbourg Verlag München Wien, 1998 - ISBN 3-486-24219-9
- Keferstein, Claus P.: Fertigungsmesstechnik. 7. Auflage, Vieweg+Teubner Verlag, 2011 - ISBN 978-3-8348-0692-5
- Warnecke, H.-J.; Dutschke, W.: Fertigungsmeßtechnik. Springer-Verlag Berlin Heidelberg New York Tokyo, 1984 - ISBN 3-540-11784-9

---

### Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

#### [1] Informatik (Bachelor of Science)

(Po-Vers. 2009s | Nebenfach | Nebenfach Maschinenbau | Qualitätsmanagement und Messtechnik)

Dieses Modul ist daneben auch in den Studienfächern "Berufspädagogik Technik (Bachelor of Science)", "Berufspädagogik Technik (Master of Education)", "Energietechnik (Bachelor of Science)", "Informatik (Master of Science)", "Maschinenbau (Bachelor of Science)", "Mechatronik (Bachelor of Science)", "Medizintechnik (Bachelor of Science)", "Technomathematik (Bachelor of Science)", "Wirtschaftsingenieurwesen (Bachelor of Science)", "Wirtschaftsingenieurwesen (Master of Science)" verwendbar.

---

### Studien-/Prüfungsleistungen:

Grundlagen der Messtechnik (Prüfungsnummer: 45101)

(englische Bezeichnung: Fundamentals of Metrology)

Prüfungsleistung, Klausur mit MultipleChoice, Dauer (in Minuten): 60

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Grundlagen der Messtechnik
- Grundlagen der Messtechnik - Übung

weitere Erläuterungen:

- **Prüfungstermine**, eine **allgemeine Regel der Prüfungstagvergabe** und **Termine der Klausureinsicht** finden Sie auf StudOn: Prüfungstermine und Termine der Klausureinsicht
- Die Lehrveranstaltungen *Grundlagen der Messtechnik [GMT]* im Wintersemester und *Fundamentals of Metrology [FoM]* im Sommersemester sind **inhaltlich identisch**. Beide Lehrveranstaltungen werden **bilingual** (Vorlesungsunterlagen: englisch-deutsch, Vortragssprache: deutsch) gehalten.
- Die **Prüfungen** über *Grundlagen der Messtechnik [GMT]* (Prüfungnr. 45101) und *Fundamentals of Metrology [FoM]* (Prüfungnr. 47701) sind **inhaltlich identisch**. Die Aufgabenstellung der Prüfung über *GMT* ist nur **in Deutsch**, während die Aufgabenstellung der Prüfung über *FoM* **bilingual** (englisch-deutsch) ist.

Erstablingung: WS 2019/2020, 1. Wdh.: SS 2020, 2. Wdh.: WS 2020/2021

1. Prüfer: Tino Hausotte (100620)

---

### Organisatorisches:

- Unterlagen zur Lehrveranstaltung werden auf der Lernplattform StudOn ([www.studon.uni-erlangen.de](http://www.studon.uni-erlangen.de)) bereitgestellt. Das Passwort wird in der Einführungsveranstaltung bekannt gegeben.

---

**Modulbezeichnung:** Fertigungsmesstechnik II (FMT II) 5 ECTS  
 (Manufacturing Metrology II)

Modulverantwortliche/r: Tino Hausotte

Lehrende: Tino Hausotte, Tino Hausotte, Sebastian Metzner

---

Startsemester: SS 2019	Dauer: 1 Semester	Turnus: jährlich (SS)
Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Deutsch

---

**Lehrveranstaltungen:**

Fertigungsmesstechnik II (SS 2019, Vorlesung, 2 SWS, Tino Hausotte)

Fertigungsmesstechnik II - Übung (SS 2019, Übung, 2 SWS, Tino Hausotte et al.)

---

**Inhalt:**

- **Optische Oberflächenmesstechnik:** Überblick Oberflächenabweichungen und Oberflächenmessprinzipien, Wechselwirkungen, Einteilung der optischen Oberflächenmessverfahren, Mikroskope und Komponenten, Messmikroskope, Numerische Apertur, Punktverwaschungsfunktion, Auflösungsvermögen, Modulations-Transfer-Funktion - Fokusvariation, Fokusvariation mit strukturierter Beleuchtung, Flying Spot Mikroskop, konfokales Mikroskop (Aufbau, Prinzip, Kennlinie, Nipkow-Scheibe, Scanspiegel, Mikrolinsenarray), Laserscanningmikroskop, konfokaler zwei Wellenlängenfasersensor, chromatischer Weißlichtsensor, Laser-Autofokusverfahren, Interferenzmikroskope (Michelson, Mirau, Linnik, Phasenschieber), Weißlichtinterferometer - Streulichtmessung
- **Taktile Formmesstechnik:** Grundlagen der Formmesstechnik, Prinzip, Charakteristika, Messaufgaben - Bauarten von taktilen Formmessgeräten (Drehtisch-, Drehspindelgeräte, Universalmessgeräte, Tastsysteme) - Messabweichungen (Einflussfaktoren, Abweichungen der Drehführung und deren Bestimmung, Abweichungen der Geradführungen) - Kalibrierung von Formmessgeräten (Flick-Normale, Vergrößerungsnormale, Kugelnormale, Mehrwellennormale) - Mehrlagenverfahren, Umschlagverfahren
- **Optische Formmesstechnik:** Interferometrische Formmessung (Interferenz gleicher Neigung und gleicher Dicke, Mehrstrahlinterferenz, Fabry-Perot und Fizeauinterferometer, Interferenzfilter, Newton'sche Ringe, Phasenschiebeinterferometer, Demodulation mit Phasenschiebung, synthetische Wellenlänge, Anwendung der Fizeau-Interferometrie, Einfluss der Referenzfläche, Dreiplattentest, Interferometrie streifendem Einfall, Twyman-Green Interferometer, Einsatzgrenzen) - Deflektometrische Formmessung (Überblick Deflektometrie, Grundprinzip, Extended Shear Angle Difference Methode, flächenhafte Deflektometrie, Einsatzgrenzen)
- **Photogrammetrie:** Grundprinzip, Stereophotogrammetrie, passive Triangulation, Grundlagen, aktive Triangulation (Punkttriangulation, linienhafte und flächenhafte Triangulation) - Streifenlichtprojektion (strukturierte Beleuchtung, Grundprinzip Ein- und Zweikerasysteme, Kodierung Gray Code, Phasenschiebung, Kombinierte Beleuchtung aus Gray Code und Phasenschiebung, Anwendung, Datenverarbeitung, Einsatzgrenzen)
- **Röntgen-Computertomografie:** Röntgenstrahlung, Grundprinzip der Röntgen-Computertomografie, Aufbau und Scanvarianten, Röntgenstrahlquellen, Strahlungsspektrum, Wechselwirkung mit Material (Photoelektrischer Effekt, Compton Streuung), Detektoren, Vergrößerung, Rekonstruktion (Radontransformation, algebraische Rekonstruktion, gefilterte Rückprojektion, Artefakte (Strahlauhfärtung, Ringartefakte, Streustrahlung, Scannerausrichtung), Schwellwertfindung, Anwendung (Defekterkennung, Micro- und Nano-CT, Hochenergie-CT, Multimaterial), Rückführung
- **Spezifikation und Messung optischer Komponenten:** Zeichnungen für optische Elemente und Systeme, Materialspezifikation, Spezifikation von Oberflächenformtoleranzen, Prüfung der Oberflächenformabweichungen (Passe) mit Probegläsern, Oberflächenbehandlungen und Beschichtungen, Messung geometrischer Spezifikationen
- **Mikro- und Nanomesstechnik:** Positioniersysteme (Führungen und Antriebe, Gewichtskraftkompensation), metrologischer Rahmen und Gerätekoordinatensysteme, Antastprinzipien und Messsystem (Rasterelektronenmikroskop, Rastertunnelmikroskop, Rasterkraftmikroskope, Nahfeldmikrosko-

pe, mikrotaktile Antastung), Mikro- und Nanokoordinatenmesssysteme, Einflussgrößen, Kalibrierung und Rückführung

## Lernziele und Kompetenzen:

### *Fachkompetenz*

#### *Wissen*

- Die Studierenden kennen relevante Definitionen, Fachbegriffe und Kriterien der Fertigungsmesstechnik.
- Die Studierenden können einen Überblick zur Gerätetechnik der Fertigungsmesstechnik sowie deren Funktionsweise und Einsatzgebiete wiedergeben
- Die Studierenden wissen um die operative Herangehensweise an Aufgaben der messtechnischen Erfassung von dimensionellen und geometrischen.

#### *Verstehen*

- Die Studierenden sind in der Lage die, den vorgestellten Messgeräten der Fertigungsmesstechnik, zugrundeliegenden Messprinzipien in eigenen Worten zu erläutern.
- Die Studierenden können Messaufgaben beschreiben und interpretieren, und Schwachstellen in der Planung und Durchführung erkennen.
- Die Studierenden können Messergebnisse und die zugrunde liegenden Verfahren angemessene kommunizieren und interpretieren.

#### *Anwenden*

- Die Studierenden können eigenständig geeignete Verfahren im Bereich Fertigungsmesstechnik auswählen.
- Die Studierenden können das Erlernte auf unbekannte, aber ähnliche Messaufgaben transferieren.

#### *Evaluieren (Beurteilen)*

- Die Studierenden können Messaufgaben in der Fertigungsmesstechnik beurteilen und strukturell analysieren.
- Die Studierenden sind in der Lage Messergebnisse zu hinterfragen und auf dieser Basis die Funktionalität des Messsystems sowie die zum Zeitpunkt der Messung vorherrschenden Messbedingungen zu bewerten.

#### *Erschaffen*

- Die Studierenden können die Eignungsuntersuchungen verschiedener Messprinzipien zur Erfüllung neuer Messaufgaben erstellen und auf deren Basis adaptierte Messsysteme konzipieren.

## Literatur:

- International Vocabulary of Metrology - Basic and General Concepts and Associated Terms, VIM, 3rd edition, JCGM 200:2008, <http://www.bipm.org/en/publications/guides/vim.html>
- DIN e.V. (Hrsg.): Internationales Wörterbuch der Metrologie - Grundlegende und allgemeine Begriffe und zugeordnete Benennungen (VIM) ISO/IEC-Leitfaden 99:2007. Korrigierte Fassung 2012, Beuth Verlag GmbH, 4. Auflage 2012
- Pfeifer, Tilo: Fertigungsmeßtechnik. R. Oldenbourg Verlag München Wien, 1998 - ISBN 3-486-24219-9
- Keferstein, Claus P.: Fertigungsmesstechnik. 7. Auflage, Vieweg+Teubner Verlag, 2011 - ISBN 978-3-8348-0692-5
- Warnecke, H.-J.; Dutschke, W.: Fertigungsmeßtechnik. Springer-Verlag Berlin Heidelberg New York Tokyo, 1984 - ISBN 3-540-11784-9
- Christoph, Ralf; Neumann, Hans Joachim: Multisensor-Koordinatenmesstechnik. 3. Auflage, Verlag Moderne Industrie, 2006 - ISBN 978-3-937889-51-2
- Neumann, Hans Joachim: Koordinatenmesstechnik im der industriellen Einsatz. Verlag Moderne Industrie, 2000 - ISBN 3-478-93212-2
- Weckenmann, A.: Koordinatenmesstechnik: Flexible Strategien für funktions- und fertigungsgerechtes Prüfen, 2. Auflage, Carl Hanser Verlag München, 2012

- Hausotte, Tino: Nanopositionier- und Nanomessmaschinen - Geräte für hochpräzise makro- bis nanoskalige Oberflächen- und Koordinatenmessungen. Pro Business Verlag, 2011 - ISBN 978-3-86805-948-9
- David J. Whitehouse: Handbook of Surface and Nanometrology, Crc Pr Inc., 2010 - ISBN 978-1420082012

---

### Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

#### [1] Informatik (Bachelor of Science)

(Po-Vers. 2009s | Nebenfach | Nebenfach Maschinenbau | Qualitätsmanagement und Messtechnik)

Dieses Modul ist daneben auch in den Studienfächern "Berufspädagogik Technik (Bachelor of Science)", "Berufspädagogik Technik (Master of Education)", "Energietechnik (Master of Science)", "Informatik (Master of Science)", "Maschinenbau (Bachelor of Science)", "Maschinenbau (Master of Science)", "Mechatronik (Bachelor of Science)", "Mechatronik (Master of Science)", "Medizintechnik (Master of Science)", "Wirtschaftsingenieurwesen (Master of Science)" verwendbar.

---

### Studien-/Prüfungsleistungen:

Fertigungsmesstechnik II (Prüfungsnummer: 69251)

(englische Bezeichnung: Manufacturing Metrology II)

Prüfungsleistung, Klausur mit MultipleChoice, Dauer (in Minuten): 60

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Fertigungsmesstechnik II
- Fertigungsmesstechnik II - Übung

weitere Erläuterungen:

**Prüfungstermine**, eine **allgemeine Regel der Prüfungstagvergabe** und **Termine der Klausureinsicht** finden Sie auf StudOn: Prüfungstermine und Termine der Klausureinsicht

Prüfungssprache: Deutsch

Erstablingung: SS 2019, 1. Wdh.: WS 2019/2020

1. Prüfer: Tino Hausotte (100620)

---

### Organisatorisches:

- Unterlagen zur Lehrveranstaltung werden auf der Lernplattform StudOn ([www.studon.uni-erlangen.de](http://www.studon.uni-erlangen.de)) bereitgestellt. Das Passwort wird in der Einführungsveranstaltung bekannt gegeben.

---

**Modulbezeichnung:** Hauptseminar Messtechnik (HS MT) 2.5 ECTS  
 (Advanced Seminar Manufacturing Metrology)

Modulverantwortliche/r: Tino Hausotte  
 Lehrende: Tino Hausotte, Assistenten

---

Startsemester: SS 2019	Dauer: 1 Semester	Turnus: halbjährlich (WS+SS)
Präsenzzeit: 30 Std.	Eigenstudium: 45 Std.	Sprache: Deutsch

---

**Lehrveranstaltungen:**

Hauptseminar Messtechnik (SS 2019, Hauptseminar, 2 SWS, Anwesenheitspflicht, Tino Hausotte et al.)

---

**Inhalt:**

**Ablauf des Seminars**

**1. Voranmeldung StudOn**

- Onlineanmeldung über StudOn (Zeitraum siehe oben).

**2. Anmeldung zum Seminarthema**

- Zeitraum, in dem die Anmeldung zu einem bestimmten Seminarthemen möglich ist, wird angemeldeten Teilnehmern des StudOn-Kurses **per E-Mail** mitgeteilt.
- Hierfür wird eine Liste der Seminarthemen mit zugeordnete StudOn-Gruppen bereit gestellt.
- Die Anmeldung zu einem bestimmten Thema erfolgt durch selbstständige Anmeldung zur zugeordneten StudOn-Gruppe.
- Freigabe der gewünschten Einteilung durch den Koordinator spätestens eine Woche nach Anmeldezeitraum.

**3. Vorbereitung des Seminarthema**

- Kontaktaufnahme mit dem Betreuer, der das gewählte Seminarthema betreut. Klärung von Ziel, Auftrag und Kontext.
- Recherche, Auswahl der Informationen.
- Grobe Ablaufplanung der Präsentation (Begrüßung und Themenübersicht, Einstieg ins Thema, Transport der Inhalte, Themenbegrenzung), Ausstieg, Fragen und Diskussion).
- Feine Ablaufplanung: Detaillierung der Inhalte (Sinnvolle Gliederung, Inhaltlichen Fortgang visualisieren, Zum Thema immer wieder zurückkehren, Gedankensprünge vermeiden, Foliensprünge vermeiden, Layout für den „roten Faden“, Ringschluss zwischen Anfang und Ende schaffen).
- Erstellen der Präsentation (Vorlage auf StudOn beachten).
- Erstellen eines Seminar-Papers (Vorlage auf StudOn beachten).

**4. Durchführung der Präsentation**

- Terminplan der Präsentationen wird vom Koordinator festgelegt und per E-Mail mitgeteilt (Termine sind in der Regel gegen Ende der Vorlesungszeit).
- Termin zur Abgabe von Präsentation und Seminar-Paper wird vom Koordinator festgelegt und per E-Mail mitgeteilt (Termin ist in der Regel 3 Werktage vor dem ersten Präsentationstermin).
- Durchführung der Präsentation (Präsentationsdauer 20 min. + 10 min. Diskussion)
- Teilnahme an 8 weiteren Vorträgen.

**5. Notenbekanntgabe**

- Notenbekanntgabe über StudOn.
- Koordinator schickt den ausgestellten Schein direkt an das Prüfungsamt.
- Auf Anfrage Feedback vom Betreuer (sofern gewünscht).

**Lernziele und Kompetenzen:**

Die Studierenden:

- erlangen grundlegender Kenntnisse in Recherche, Themenaufbereitung und Präsentationstechniken,
- erarbeiten Schwerpunkte technischer Zusammenhänge bei einem gegebenen Thema,
- vertiefen eigenständig einen technischen Schwerpunkt an Hand eines konkreten Beispiels der Fertigungsmesstechnik,
- erlernen die Fähigkeit, sich in unbekannte Probleme einzuarbeiten und diese verständlich zu präsentieren,



- erlernen die Fähigkeit, als Zuhörer aktiv Fragen zu formulieren und technische Sachverhalte zu diskutieren,

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

**[1] Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Nebenfach | Nebenfach Maschinenbau | Qualitätsmanagement und Messtechnik)

Dieses Modul ist daneben auch in den Studienfächern "Maschinenbau (Master of Science)", "Mechatronik (Bachelor of Science)", "Wirtschaftsingenieurwesen (Bachelor of Science)", "Wirtschaftsingenieurwesen (Master of Science)" verwendbar.

---

**Studien-/Prüfungsleistungen:**

Hauptseminar Messtechnik (Prüfungsnummer: 607629)

(englische Bezeichnung: Advanced Seminar Manufacturing Metrology)

Prüfungsleistung, Seminararbeit+Vortrag

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Hauptseminar Messtechnik

weitere Erläuterungen:

Die Gesamtmodulnote errechnet sich aus 25% der Seminararbeits- und 75% der Vortragsleistung.

Erstablingung: SS 2019, 1. Wdh.: WS 2019/2020

1. Prüfer: Tino Hausotte (100620)

---

---

**Modulbezeichnung:** Qualitätsmanagement II - Phasenübergreifendes Qualitätsmanagement (QM II) (Life-Cycle Oriented Quality Management) 2.5 ECTS

Modulverantwortliche/r: N.N.  
 Lehrende: N.N.

---

Startsemester: SS 2019	Dauer: 1 Semester	Turnus: jährlich (SS)
Präsenzzeit: 30 Std.	Eigenstudium: 45 Std.	Sprache: Deutsch

---

**Lehrveranstaltungen:**

Die Präsenzveranstaltung Qualitätsmanagement II entfällt im Sommersemester 2019. Zur Vorbereitung auf die Prüfung QM II oder auf die Modulabschlussprüfung QM können Sie jedoch den als äquivalent geltenden E-Learning-Kurs Qualitätsmanagement - QMaK über StudOn ([https://www.studon.fau.de/studon/login.php?target=crs\\_2493916](https://www.studon.fau.de/studon/login.php?target=crs_2493916)) belegen. Qualitätsmanagement II - Phasenübergreifendes Qualitätsmanagement (SS 2019, Vorlesung mit Übung, 2 SWS, N.N.)

---

**Inhalt:**

- Qualitätsmanagementsystem - Auditierung und Zertifizierung
- Total Quality Management und EFQM-Modell
- Rechnerunterstützung im Qualitätsmanagement
- Ausbildung und Motivation
- Kontinuierliche Verbesserungsprogramme und Benchmarking
- Problemlösungstechniken und Qualitätszirkel
- Qualitätsbewertung
- Qualität und Wirtschaftlichkeit
- Six Sigma
- Qualität und Umwelt, Umweltmanagement
- Qualität und Recht, Sicherheit
- *Qualitätsbewertung (Übung)*
- *Qualitätsbezogene Kosten und Wirtschaftlichkeit (Übung)*
- *Ökobilanzierung (Übung)*

**Lernziele und Kompetenzen:**

Nach dem Besuch der Veranstaltung sind die Teilnehmenden in der Lage,

Wissen:

- Motivation, Ziele, Grundsätze und Strategien des prozessorientierten Qualitätsmanagements wiederzugeben

Verstehen:

- Wissen zu Qualitätsmanagement als unternehmens- und produktlebenszyklusübergreifende Strategie zu veranschaulichen
- Anforderungen, Aufbau, Einführung sowie die Auditierung und Zertifizierung von Qualitätsmanagementsystemen darzustellen

Anwenden:

- Business Excellence anhand Total Quality Management (TQM), Unternehmensbewertungsmodelle wie EFQM und kontinuierlicher Verbesserungsprozesse im Unternehmen auszuführen
- die Interaktion von Qualitätsmanagement mit rechtlichen und sicherheitsrelevanten Themen zu erklären
- den Zusammenhang zwischen Qualitätsmanagement und Umweltmanagement zu beschreiben
- die Wirtschaftlichkeit von Qualitätsverbesserungsmaßnahmen zu demonstrieren
- die Methodik „Six Sigma“ zu beschreiben und dem Kontext der Qualitätsverbesserung zuzuordnen

Analysieren:

- Handlungsgrundlagen hinsichtlich Ausbildungs-, Motivations- und Organisationsverbesserung zu ermitteln

Evaluiieren:

- die Qualität mit etablierten Vorgehensweisen zu bewerten

## Literatur:

- Kamiske, G. F.; Brauer, J.-P.: Qualitätsmanagement von A - Z, Carl Hanser Verlag, München 2005
- Masing, W.: Handbuch Qualitätsmanagement, Carl Hanser Verlag, München 2007
- Wagner, K. W.; Patzak, G.: Performance Excellence - Der Praxisleitfaden zum effektiven Prozessmanagement, Carl Hanser Verlag, München 2007
- Zink, K. J.: Mitarbeiterbeteiligung bei Verbesserungs- und Veränderungsprozessen, Carl Hanser Verlag, München 2007

---

## Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

### [1] Informatik (Bachelor of Science)

(Po-Vers. 2009s | Nebenfach | Nebenfach Maschinenbau | Qualitätsmanagement und Messtechnik)

Dieses Modul ist daneben auch in den Studienfächern "Energietechnik (Master of Science)", "Informatik (Master of Science)", "Maschinenbau (Bachelor of Science)", "Maschinenbau (Master of Science)", "Mechatronik (Master of Science)", "Medizintechnik (Bachelor of Science)", "Wirtschaftsingenieurwesen (Bachelor of Science)", "Wirtschaftsingenieurwesen (Master of Science)" verwendbar.

---

## Studien-/Prüfungsleistungen:

Qualitätsmanagement II - Phasenübergreifendes Qualitätsmanagement (Prüfungsnummer: 30301)

(englische Bezeichnung: Life-Cycle Oriented Quality Management)

Prüfungsleistung, Klausur, Dauer (in Minuten): 60

Anteil an der Berechnung der Modulnote: 100%

weitere Erläuterungen:

**Prüfungstermine, eine allgemeine Regel der Prüfungstagvergabe und Termine der Klausureinsicht** finden Sie auf StudOn: Prüfungstermine und Termine der Klausureinsicht

Erstablingung: SS 2019, 1. Wdh.: WS 2019/2020

1. Prüfer: Tino Hausotte (100620)

---

---

**Modulbezeichnung: Rechnergestützte Messtechnik (RMT)** **5 ECTS**  
 (Computer-Aided Metrology)

 Modulverantwortliche/r: Tino Hausotte  
 Lehrende: Tino Hausotte

---

Startsemester: SS 2019	Dauer: 1 Semester	Turnus: jährlich (SS)
Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Deutsch

---

**Lehrveranstaltungen:**

 Rechnergestützte Messtechnik (SS 2019, Vorlesung, 2 SWS, Tino Hausotte)  
 Rechnergestützte Messtechnik - Übung (SS 2019, Übung, 2 SWS, Tino Hausotte et al.)

**Inhalt:**

- **Grundlagen:** Grundbegriffe (Größe, Größenwert, Messgröße, Maßeinheit, Messprinzip, Messung, Messkette, Messsignal, Informationsparameter, analoges und digitales Signal) - Prinzip eines Messgerätes, direkte und indirekte Messmethode, Kennlinie und Kennlinienarten, analoge und digitale Messmethoden, kontinuierliche und diskontinuierliche Messung, Zeit- und Wertdiskretisierung, Auflösung, Empfindlichkeit, Messbereich - Signal, Messsignal, Klassifizierung von Signalen (Informationsparameter) - Signalbeschreibung, Fourierreihen und Fouriertransformation - Fourieranalyse - DFT und FFT (praktische Realisierung) - Aliasing und Shannon's-Abtasttheorem - Übertragungsverhalten (Antwortfunktionen, Frequenzgang, Übertragungsfunktion) - Laplace-Transformation, Digitalisierungskette, Z-Transformation und Wavelet-Transformation
- **Verarbeitung und Übertragung analoger Signale:** Messverstärker, Operationsverstärker (idealer und realer, Rückkopplung) - Kenngrößen von Operationsverstärkern - Frequenzabhängige Verstärkung von Operationsverstärkern - Operationsverstärkertypen - Rückkopplung und Grundsaltungen (Komparator, Invertierender Verstärker, Nichtinvertierender Verstärker, Impedanzwandler, Strom-Spannungs-Wandler, Differenzverstärker, Integrierer, Differenzierer, invertierender Addierer, Subtrahierer, Logarithmierer, e-Funktionsgeneratoren, Instrumentenverstärker) - OPV mit differentiellen Ausgang - analoge Filter (Tiefpassfilter, Hochpassfilter, Bandpassfilter, Bandsperrfilter, Bodeplot, Phasenschiebung, aktive analoge Filter) - Messsignalübertragung (Einheitssignale, Anschlussvarianten) - Spannungs-Frequenz-Wandler - Galvanische Trennung und optische Übertragung - Modulatoren und Demulatoren - Multiplexer und Demultiplexer - Abtast-Halte-Glied
- **A/D- und D/A-Umsetzer:** Digitale und analoge Signale - Digitalisierungskette - A/D-Umsetzer (Nachlauf ADU, Wägeverfahren, Rampen-A/D-Umsetzer, Dual Slope-Verfahren, Charge-Balancing-A/D-Umsetzverfahren, Parallel-A/D-Umsetzer, Kaskaden-A/D-Umsetzverfahren, Pipeline-A/D-Umsetzer, Delta-Sigma-A/D-Umsetzer / 1-Bit- bis N-Bit-Umsetzer, Einsatzbereiche, Kennwerte, Abweichungen, Signal-Rausch-Verhältnis) - Digital-Analog-Umsetzungskette - D/A-Umsetzer (Direkt bzw. Parallelumsetzer, Wägeumsetzer, Zählverfahren, Pulsweitenmodulation, Delta-Sigma-Umsetzer / 1-Bit- bis N-Bit-Umsetzer)
- **Verarbeitung digitaler Signale:** digitale Codes - Schaltnetze (Kombinatorische Schaltungslogik) - Schaltalgebra und logische Grundverknüpfungen - Schaltwerke (Sequentielle Schaltnetze) - Speicherglieder (Flip-Flops, Sequentielle Grundsaltungen), Halbleiterspeicher (statische und dynamische, FIFO) - Anwendungsspezifische integrierte Schaltungen (ASICs) - Programmierbare logische Schaltung (PLDs, Programmierbarkeit, Vorteile, Anwendungen, Programmierung) - Rechnerarten
- **Bussysteme:** Bussysteme (Master, Slave, Arbiter, Routing, Repeater) - Arbitrierung - Topologien (physikalische und logische Topologie, Kennwerte, Punkt-zu-Punkt-Topologie, vermaschtes Netz, Stern-Topologie, Ring-Topologie, Bus-Topologie, Baum-Topologie, Zell-Topologie) - Übertragungsmedien (Mehrdrahtleitung, Koaxialkabel, Lichtwellenleiter) - ISO-OSI-Referenzmodell - Physikalische Schnittstellenstandards (RS-232C, RS-422, RS-485) - Feldbussysteme, GPIB (IEC-625-Bus), Messgerätebusse
- **USB Universal Serial Bus:** Struktur des Busses - Verbindung der Geräte, Transceiver, Geschwindigkeitserkennung, Signalkodierung - Übertragungsarten (Control-Transfer, Bulk-Transfer, Isochrone-Transfer, Interrupt-Transfer, Datenübertragung mit Paketen) - Frames und Mikroframes, Geschwin-

digkeiten, Geschwindigkeitsumsetzung mit Hub - Deskriptoren und Software - Layer Entwicklungstools - Compliance Test - USB 3.0

- **Digitale Filter:** Analoge Filter - Eigenschaften und Charakterisierung von digitalen Filtern - Digitale Filter (Implementierung, Topologien, IIR-Filter und FIR-Filter) und Formen - Messwert-Dezimirer, digitaler Mittelwertfilter, Gaußfilter - Fensterfunktionen, Gibbs-Phänomen - Realisierung mit MATLAB - Vor- und Nachteile digitaler Filter
- **Messdatenauswertung:** Absolute, relative, zufällige und systematische Messabweichungen, Umgang mit Messabweichungen, Kalibrierung - Korrelationsanalyse - Kennlinienabweichungen und Methoden zu deren Ermittlung - Regressionsanalyse - Kennlinienkorrektur - Approximation, Interpolation, Extrapolation - Arten der Kennlinienkorrektur - Messpräzision, Messgenauigkeit, Messrichtigkeit, Fehlerfortpflanzungsgesetz (altes Konzept), Messunsicherheit und deren Bestimmung - Vorgehensweise zur Ermittlung der Unsicherheit, Monte-Carlo-Methode
- **Schaltungs- und Leiterplattenentwurf:** Leiterplatten - Leiterplattenmaterial - Leiterplattenarten - Durchkontaktierungen - Leiterplattenentwurf und -entflechtung - Software - Leiterplattenherstellung

## Contents

- **Basics:** Terms (quantity, quantity value, measurand, measurement unit, principle of measurement, measurement, measuring chain, measurement signal, information parameter, analogue and digital signal) - Principle of a measuring instrument, direct and indirect measurement, characteristic curves and characteristic curve types, analogue and digital measuring methods, continuous and discontinuous measurement, time and value discretisation, resolution, sensitivity, measuring interval (range) - Signal, measurement signal, classification of signals (information parameter) - Signal description, Fourier series and Fourier transformation - Fourier analysis - DFT and FFT (practical realization) - Aliasing and Shannon's sampling theorem - Transfer behaviour (response functions, frequency response, transfer function) - Laplace transform, digitisation chain, Z-transform and wavelet transform
- **Processing and transmission of analogue signals:** Measuring amplifiers, operational amplifiers (ideal and real, feedback) - Characteristics of operational amplifiers - Frequency-dependent gain of operational amplifiers - Operational amplifier types - Feedback and basic circuits (comparator, inverting amplifier, non-inverting amplifier, impedance converter, current-voltage converter, differential amplifier, integrator, differentiator, inverting adder, subtractor, logarithmic, exponential function generators, instrumentation amplifier) - OPV with differential output - Analogue filter (low pass filter, high pass filter, band pass filter, band elimination filter, Bodeplot, phase shifting, active analogue filters) - Measurement signal transmission (standard signals, connection variants) - Voltage-frequency converters - Galvanic isolation and optical transmission - modulators and demodulators - multiplexers and demultiplexers - sample-and-hold amplifier
- **A/D and D/A converter:** Digital and analogue signals - Digitisation chain - A/D converter (follow-up ADC, weighing method, ramp A/D converter, dual slope method, charge-balancing ADC, parallel ADC, cascade ADC, pipeline A/D converter, the delta-sigma A/D converter / 1-bit to N-bit converter, application, characteristics, deviations, signal-to-noise ratio) - Digital-to-analogue conversion chain - D/A converter (direct or parallel converters, weighing method, counting method, pulse width modulation, delta-sigma converter / 1-bit to N-bit converter)
- **Digital signal processing:** Digital codes - Switching networks (combinatorial circuit logic) - Boolean algebra and basic logic operations - Sequential circuit (sequential switching networks) - Storage elements (flip-flops, sequential basic circuits), semiconductor memory (static and dynamic, FIFO) - Application Specific Integrated Circuits (ASICs) - The programmable logic device (PLD, programmability, benefits, applications, programming) - computer types
- **Data bus systems:** Bus systems (master, slave, arbiter, routing, repeater) - Arbitration - Topologies (physical and logical topology, characteristics, point-to-point topology, mesh network, star topology, ring topology, bus topology, tree topology, cell topology) - Transmission media (multi-wire cable, coaxial cable, fibre optic cable) - ISO OSI reference model - Physical interface standards (RS-232C, RS-422, RS-485) - Fieldbus systems, GPIB (IEC-625 bus) , Measuring device buses
- **USB Universal Serial Bus:** Bus structure - Connection of the devices, transceiver, speed detection, signal coding - Transfer types (control transfer, bulk transfer, isochronous transfer, interrupt transfer,

data transfer with packages) - Frames and micro-frames, speeds, speed conversion with hubs - Descriptors and software - Layer development tools - Compliance test - USB 3.0

- **Digital filters:** Analogue filter - Properties and characterization of digital filters - Digital Filter (implementation, topologies, IIR filters and FIR filters) and forms - Measurement value decimator, digital averaging filter, Gaussian filter - Window functions, Gibbs phenomenon - Realisation with MATLAB - Advantages and disadvantages of digital filters
- **Data analysis:** Absolute, relative, random and systematic errors, handling of measurement errors, calibration - Correlation analysis - Characteristic curve deviations and methods for their determination - Regression analysis - Characteristic curve correction - Approximation, interpolation, extrapolation - Kinds of characteristic curve correction - Measurement precision, measurement accuracy, measurement trueness, error propagation law (old concept), uncertainty and their estimation - Procedure for determining the uncertainty, Monte Carlo method
- **Circuit and PCB design:** Printed circuit boards (PCB) - PCB material - PCB types - Vias - PCB design and deconcentration - Software - PCB production

### Lernziele und Kompetenzen:

#### *Fachkompetenz*

##### *Wissen*

- Die Studierenden können einen Überblick zur rechnergestützten Messtechnik sowie deren Einsatzgebiete wiedergeben.
- Die Studierenden können Wissen zur rechnergestützten Messdatenerfassung, -auswertung, -analyse und -visualisierung als Grundlage für zielorientierte, effiziente Entwicklung und für kontinuierliche Produkt- und Prozessverbesserung abrufen

##### *Verstehen*

- Die Studierenden können Konzepte zur Sensorintegration und Datenfusion beschreiben

##### *Evaluieren (Beurteilen)*

- Die Studierenden können rechnergestützte Werkzeuge für die Messdatenerfassung, -auswertung, -analyse und -visualisierung auswählen und bewerten.

### Literatur:

- International Vocabulary of Metrology - Basic and General Concepts and Associated Terms, VIM, 3rd edition, JCGM 200:2008, <http://www.bipm.org/en/publications/guides/vim.html>
- DIN e.V. (Hrsg.): Internationales Wörterbuch der Metrologie - Grundlegende und allgemeine Begriffe und zugeordnete Benennungen (VIM) ISO/IEC-Leitfaden 99:2007. Korrigierte Fassung 2012, Beuth Verlag GmbH, 4. Auflage 2012
- Hoffmann, Jörg: Handbuch der Messtechnik. 4. Auflage, Carl Hanser Verlag München, 2012 - ISBN 978-3-446-42736-5
- Lerch, Reinhard: Elektrische Messtechnik. 6. Auflage, Springer-Verlag Berlin Heidelberg, 2012 - ISBN 978-3-642-22608-3
- Richter, Werner: Elektrische Meßtechnik. 3. Auflage, Verlag Technik Berlin, 1994 - ISBN 3-341-01106-4
- H. Czichos (Hrsg.): Das Ingenieurwissen Gebundene. 7. Auflage, Springer Verlag, 2012, ISBN 978-3-642-22849-0.
- Best, Roland: Digitale Meßwertverarbeitung. Oldenbourg München, 1991 - ISBN 3-486-21573-6.
- E DIN IEC 60050-351:2013-07 International Electrotechnical Vocabulary - Part 351: Control technology / Internationales Elektrotechnisches Wörterbuch - Teil 351: Leittechnik.
- DIN 44300:1982-03 Informationsverarbeitung; Begriffe.
- DIN 44300-1:1995-03 Informationsverarbeitung - Begriffe - Teil 1: Allgemeine Begriffe.
- DIN 40900-12:1992-09 Graphische Symbole für Schaltungsunterlagen; Binäre Elemente.

---

### Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

## [1] Informatik (Bachelor of Science)

(Po-Vers. 2009s | Nebenfach | Nebenfach Maschinenbau | Qualitätsmanagement und Messtechnik)

Dieses Modul ist daneben auch in den Studienfächern "Berufspädagogik Technik (Bachelor of Science)", "Berufspädagogik Technik (Master of Education)", "Informatik (Master of Science)", "Maschinenbau (Bachelor of Science)", "Maschinenbau (Master of Science)", "Mechatronik (Bachelor of Science)", "Mechatronik (Master of Science)", "Medizintechnik (Master of Science)", "Wirtschaftsingenieurwesen (Master of Science)" verwendbar.

---

### Studien-/Prüfungsleistungen:

Rechnergestützte Messtechnik (Prüfungsnummer: 69301)

(englische Bezeichnung: Computer-Aided Metrology)

Prüfungsleistung, Klausur mit MultipleChoice, Dauer (in Minuten): 60

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Rechnergestützte Messtechnik
- Rechnergestützte Messtechnik - Übung

weitere Erläuterungen:

**Prüfungstermine**, eine **allgemeine Regel der Prüfungstagvergabe** und **Termine der Klausureinsicht** finden Sie auf StudOn: Prüfungstermine und Termine der Klausureinsicht

Prüfungssprache: Deutsch

Erstablingung: SS 2019, 1. Wdh.: WS 2019/2020

1. Prüfer: Tino Hausotte (100620)

---

**Modulbezeichnung:** Fertigungsmesstechnik I (FMT I) **5 ECTS**  
(Manufacturing Metrology)

Modulverantwortliche/r: Tino Hausotte  
Lehrende: Tino Hausotte

Startsemester: WS 2019/2020      Dauer: 1 Semester      Turnus: jährlich (WS)  
Präsenzzeit: 60 Std.      Eigenstudium: 90 Std.      Sprache: Deutsch

**Lehrveranstaltungen:**

Fertigungsmesstechnik I (WS 2019/2020, Vorlesung, 2 SWS, Tino Hausotte)  
Fertigungsmesstechnik I - Übung (WS 2019/2020, Übung, 2 SWS, Tino Hausotte et al.)

**Empfohlene Voraussetzungen:**

Für eine optimale Vorbereitung empfiehlt sich eine Belegung des Moduls "Grundlagen der Messtechnik". Dies ist jedoch keine Teilnahmevoraussetzung für das Modul "Fertigungsmesstechnik I".

**Vorhergehende Module:**

Grundlagen der Messtechnik

**Inhalt:**

- **Grundlagen, Begriffe, Größen und Aufgaben der FMT:** Teilgebiete der industriellen Messtechnik - Fertigungsmesstechnik, Grundaufgaben und Ziele - Messen, Prüfen, Überwachen, Lehren - Begriffsdefinition: Messgröße, Messwert, Messunsicherheit, wahrer Wert, vereinbarter Wert, Messergebnis, Prüfung, Messung, Messprinzip, Messmethode, Messverfahren, Nennmaß, Grenzmaß, Grenzabmaß - Grundeinteilung der Mess- und Prüfmittel in der FMT - Messschieber, Messschrauben, Messuhr - Taylorscher Grundsatz, Lehren - Endmaße, Sinustisch oder Sinuslineal, Maßverkörperungen, Winkelmaß
- **Geometrische Produktspezifikation und Verifikation (GPS) - Basis der Messaufgabenbeschreibung und -durchführung:** Geometrischen Produktspezifikation (GPS) - Dualitätsprinzip und Operationen - Begriffsdefinition von Geometrieelementen (Nenn-, wirkliches, erfasstes und zugeordnetes Geometrieelement) - Standardgeometrieelemente - Gestaltparameter an Werkstücken (Grobgestalt, Feingestalt, Maß, Abstand, Lage, Form, Welligkeit, Rauheit) - Systematik der Gestaltabweichungsarten (Maß-, Form-, Lageabweichungen und Abweichung der Oberflächenbeschaffenheit) - Toleranzbegriff - Form- und Lagetoleranzen - Systematik der Tolerierung von Unabhängigkeitsprinzip Werkstücken (Unabhängigkeitsprinzip, Hüllprinzip)
- **Grundlagen der Längenmesstechnik (Maßstäbe und Interferometer):** Messprinzipien zur Längenmessung - Abbe Komparator, Maßstäbe mit Skalen - Eppensteinprinzip - Linearencoder, Gitterabtastung, Richtungserkennung, Ausgangssignale, Demodulation, Differenzsignalerfassung, Referenzmarken, Abtastung (abbildend, interferometrisch, Durchlicht, Auflicht) - Demodulationsabweichungen: Quantisierungs-, Amplituden-, Offset- und Phasenabweichungen, Heydemannkorrektur - absolut codierte Maßstäbe: V- und U-Abtastung und Gray Code - Transversale elektromagnetische Welle, Überlagerung von Wellen, konstruktive und destruktive Interferenz Polarisation des Lichtes, Voraussetzungen für die Interferenz, Interferenz von Lichtwellen - Interferenz (Homodynprinzip und Heterodynprinzip), Interferenz am Michelson-Interferometern, Einteilung von Interferometern, Luftbrechzahl, Demodulation am Homodyninterferometer, Demodulation am Heterodyninterferometer - Einteilung von Interferometern, Luftbrechzahl, zeitliche und räumliche Kohärenz - Laser, He-Ne-Laser - Aufbau von Interferometern, Anwendung der Interferometer
- **Koordinatenmesstechnik:** Prinzip, Koordinatensysteme, Grundanordnung, Bauarten - Tastsysteme (Erzeugung der Antastkraft, Messung der Auslenkung, Integration mehrerer Achsen, Kinematik, weitere Achse, Umwelt, Arten von Tastsystemen, Taststiftbiegung, Taster) - Einzelpunktantastung, Scanning - Beschreiben und Festlegen der Messaufgabe - Feststellen Einflüsse auf das Messergebnis - Vorbereitung der Messung - Auswahl und Einmessen des Tasters - Festlegen der Messstrategie - Auswertung der Messergebnisse (Ausgleichsverfahren) - Spezifikation, Parameter und Prüfung
- **Formprüftechnik:** Prinzip, Charakteristika, Messaufgaben, Bauarten (Drehtisch-, Drehspindelgeräte) - Abweichungen der Drehführung von der idealen Achse und deren Bestimmung - Kalibrierung



von Formessgeräten - Mehrlagenverfahren, Umschlagverfahren

- **Oberflächenmesstechnik:** Oberflächenmessprinzipien - Tastschnittgeräte, optische Oberflächenmessgeräte, Fokusvariation, Konfokales Mikroskop, Laser-Autofokusverfahren, Interferenzmikroskope, Weißlichtinterferometer - Oberflächenparameter Normenreihe DIN EN ISO (Profil, Flächen) - Profilauswertung entsprechend DIN EN ISO 3274 und DIN EN ISO 4287 - Profilkenngrößen (Rauheits-, Welligkeit- und Struktur-Kenngrößen): Filterung, Senkrecht-, Waagrechtkenngößen, gemischte Kenngrößen - Kenngrößen aus Materialanteil-Kurve (ISO 13565-2 und ISO 13565-3) - Flächenparameter (Höhenparameter, räumliche Parameter, flächenhafte Materialanteilkurve, topographischen Elemente) - Streulichtmessung, Streulichtparameter

#### Content:

- **Basics, Terms, Dimensions and Tasks of the Manufacturing Metrology:** Parts of the industrial measurement technology - Manufacturing Metrology, Tasks and Aims - Measure, Inspect, Control, Gauge - Terms: Measurand, measurement value, measurement uncertainty, true value, measurement result, inspection, measurement, measurement principle, measurement method, basic size, limiting size, limiting dimension - Classification of measurement and inspection equipment - Caliper, micrometer screw, indicator - Basic principle of Taylor, gauge - Gauge block, sinus table, sinus ruler, material measure, angle gauge block
- **Geometrical product specification and verification (GPS) - Basis of the measurement task description and execution:** Geometrical product specification and verification (GPS) - Duality principle and operations - Definition of terms of geometry elements - Standard geometry elements - Shape parameter on workpieces - System of shape deviations - Terms of tolerance - Form tolerance and position tolerance - System of toleration with the principle of independence
- **Basics of dimension measurement (scale and interferometry):** Principle of dimension measurement - Abbe comparator, scales - Principle of Eppenstein - Linear encoder, lattice sampling, direction detection, output signals, demodulation, detection of signal difference, reference marks, sampling - Demodulation deviation: Deviation of quantification, amplitude, offset and phases, Heydemann correction - Absolute coded scales; V- and U-sampling, gray code - Transversal electromagnetic weave, overlap of weaves, constructive and destructive interferences, polarization of light, requirements for interference, interference of light waves - Interference (homodyne principle, heterodyne principle), interference with the Michelson interferometer, classification of interferometer, index of refraction, demodulation on the homodyne and heterodyne interferometer - Classification of interferometer, index of refraction, temporal and spatial coherence - Laser, He-Ne-laser - Setup of interferometer, field of application of interferometer
- **Coordinate measuring technology:** Principle, coordinate system, setup, designs - Caliper systems - Single point measurement, scanning - Description of measurement tasks - Definition of influences on the measurement result - Preparation of the measurement - Right choice of caliper, calibration of caliper - Definition of a measurement strategy - Evaluation of the measurement results - Specifications, parameters and inspection
- **Form inspection technique:** Principle, characteristics, measurement tasks, designs - Deviation of the swivel guide from an ideal axis - Calibration of form measurement systems
- **Surface measurements:** Principles of surface measurements - Profilometer, optical surface measurement systems, focus variation, confocal microscope, laser-auto focus variation, interference microscope, white light interferometer - Surface parameters in DIN EN ISO - Profile analysis according to DIN EN ISO 3274 and DIN EN ISO 4287 - Profile parameters - Parameters of the material-curve (ISO 13565-2 and ISO 13565-3) - Area parameters - Scattered light measurement, scattered light parameters

#### Lernziele und Kompetenzen:

- Motivation, Ziele, Grundsätze und Strategien der Fertigungsmesstechnik erfassen.
- Beurteilen und strukturelle Analyse von Messaufgaben in der Fertigungsmesstechnik. Transfer des Erlernten auf unbekannte, aber ähnliche Messaufgaben.
- Verständnis um die operative Herangehensweise an Aufgaben der messtechnischen Erfassung von dimensionellen und geometrischen Größen an Werkstücken.
- Eigenständige Auswahl geeigneter Verfahren im Bereich Fertigungsmesstechnik.

- Beschreiben von Messaufgaben, Durchführen, Auswerten von Messungen.
- Selbstständiges Erkennen von Schwachstellen in der Planung und Durchführung.
- Bewerten von Messergebnissen aus dem Bereich der Fertigungsmesstechnik.
- Angemessene Kommunikation und Interpretation von Messergebnissen und der zugrunde liegenden Verfahren.

#### Literatur:

- Kohlrausch, Friedrich: Praktische Physik : zum Gebrauch für Unterricht, Forschung und Technik. Band 1-3, 24. Auflage, Teubner Verlag, 1996 - ISBN 3-519-23001-1, 3-519-23002-X, 3-519-23000-3
- DIN e.V. (Hrsg.): Internationales Wörterbuch der Metrologie - Grundlegende und allgemeine Begriffe und zugeordnete Benennungen (VIM) ISO/IEC-Leitfaden 99:2007. Beuth Verlag GmbH, 3. Auflage 2010
- Pfeifer, Tilo: Fertigungsmeßtechnik. R. Oldenbourg Verlag München Wien, 1998 - ISBN 3-486-24219-9
- Keferstein, Claus P.: Fertigungsmesstechnik. 7. Auflage, Vieweg+Teubner Verlag, 2011 - ISBN 978-3-8348-0692-5
- Warnecke, H.-J.; Dutschke, W.: Fertigungsmeßtechnik. Springer-Verlag Berlin Heidelberg New York Tokyo, 1984 - ISBN 3-540-11784-9
- Christoph, Ralf; Neumann, Hans Joachim: Multisensor-Koordinatenmesstechnik. 3. Auflage, Verlag Moderne Industrie, 2006 - ISBN 978-3-937889-51-2
- Neumann, Hans Joachim: Koordinatenmesstechnik im der industriellen Einsatz. Verlag Moderne Industrie, 2000 - ISBN 3-478-93212-2
- Ernst, Alfons: Digitale Längen- und Winkelmesstechnik. 4. Auflage, Verlag Moderne Industrie, 2001 - ISBN 3-478-93264-5
- Joza, Jan: Messen großer Längen. VEB Verlag Technik Berlin, 1969
- Henzold, Georg: Form und Lage. 3. Auflage, Beuth Verlag GmbH Berlin, 2011 - ISBN 978-3-410-21196-9
- Weckenmann, A.: Koordinatenmesstechnik: Flexible Strategien für funktions- und fertigungsgerechtes Prüfen, 2. Auflage, Carl Hanser Verlag München, 2012

#### Internetlinks für weitere Information zum Thema Messtechnik

- Video des VDI: Messtechnik - Unsichtbare Präzision überall
- Multisensor-Koordinatenmesstechnik
- E-Learning Kurs AUKOM Stufe 1

---

#### Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

##### [1] Informatik (Bachelor of Science)

(Po-Vers. 2009s | Nebenfach | Nebenfach Maschinenbau | Qualitätsmanagement und Messtechnik | Wahlpflichtmodule Qualitätsmanagement und Messtechnik für Nebenfach Informatik)

Dieses Modul ist daneben auch in den Studienfächern "Berufspädagogik Technik (Bachelor of Science)", "Berufspädagogik Technik (Master of Education)", "Energietechnik (Master of Science)", "Informatik (Master of Science)", "Maschinenbau (Bachelor of Science)", "Maschinenbau (Master of Science)", "Materialwissenschaft und Werkstofftechnik (Master of Science)", "Mechatronik (Bachelor of Science)", "Mechatronik (Master of Science)", "Medizintechnik (Master of Science)", "Wirtschaftsingenieurwesen (Bachelor of Science)", "Wirtschaftsingenieurwesen (Master of Science)" verwendbar.

---

#### Studien-/Prüfungsleistungen:

Fertigungsmesstechnik I (Prüfungsnummer: 72471)

(englische Bezeichnung: Manufacturing Metrology)

Prüfungsleistung, Klausur mit MultipleChoice, Dauer (in Minuten): 60

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Fertigungsmesstechnik I
- Fertigungsmesstechnik I - Übung

weitere Erläuterungen:

**Prüfungstermine**, eine **allgemeine Regel der Prüfungstagvergabe** und **Termine der Klausureinsicht** finden Sie auf StudOn: Prüfungstermine und Termine der Klausureinsicht

Prüfungssprache: Deutsch

Erstablingung: WS 2019/2020, 1. Wdh.: SS 2020

1. Prüfer: Tino Hausotte (100620)

---

**Organisatorisches:**

- Unterlagen zur Lehrveranstaltung werden auf der Lernplattform StudOn ([www.studon.uni-erlangen.de](http://www.studon.uni-erlangen.de)) bereitgestellt. Das Passwort wird in der Einführungsveranstaltung bekannt gegeben.

---

**Modulbezeichnung:** Fertigungsmesstechnik II (FMT II) 5 ECTS  
 (Manufacturing Metrology II)

Modulverantwortliche/r: Tino Hausotte

Lehrende: Tino Hausotte, Tino Hausotte, Sebastian Metzner

Startsemester: SS 2019

Dauer: 1 Semester

Turnus: jährlich (SS)

Präsenzzeit: 60 Std.

Eigenstudium: 90 Std.

Sprache: Deutsch

---

**Lehrveranstaltungen:**

Fertigungsmesstechnik II (SS 2019, Vorlesung, 2 SWS, Tino Hausotte)

Fertigungsmesstechnik II - Übung (SS 2019, Übung, 2 SWS, Tino Hausotte et al.)

---

**Inhalt:**

- **Optische Oberflächenmesstechnik:** Überblick Oberflächenabweichungen und Oberflächenmessprinzipien, Wechselwirkungen, Einteilung der optischen Oberflächenmessverfahren, Mikroskope und Komponenten, Messmikroskope, Numerische Apertur, Punktverwaschungsfunktion, Auflösungsvermögen, Modulations-Transfer-Funktion - Fokusvariation, Fokusvariation mit strukturierter Beleuchtung, Flying Spot Mikroskop, konfokales Mikroskop (Aufbau, Prinzip, Kennlinie, Nipkow-Scheibe, Scanspiegel, Mikrolinsenarray), Laserscanningmikroskop, konfokaler zwei Wellenlängenfasersensor, chromatischer Weißlichtsensor, Laser-Autofokusverfahren, Interferenzmikroskope (Michelson, Mirau, Linnik, Phasenschieber), Weißlichtinterferometer - Streulichtmessung
- **Taktile Formmesstechnik:** Grundlagen der Formmesstechnik, Prinzip, Charakteristika, Messaufgaben - Bauarten von taktilen Formmessgeräten (Drehtisch-, Drehspindelgeräte, Universalmessgeräte, Tastsysteme) - Messabweichungen (Einflussfaktoren, Abweichungen der Drehführung und deren Bestimmung, Abweichungen der Geradführungen) - Kalibrierung von Formmessgeräten (Flick-Normale, Vergrößerungsnormale, Kugelnormale, Mehrwellennormale) - Mehrlagenverfahren, Umschlagverfahren
- **Optische Formmesstechnik:** Interferometrische Formmessung (Interferenz gleicher Neigung und gleicher Dicke, Mehrstrahlinterferenz, Fabry-Perot und Fizeauinterferometer, Interferenzfilter, Newton'sche Ringe, Phasenschiebeinterferometer, Demodulation mit Phasenschiebung, synthetische Wellenlänge, Anwendung der Fizeau-Interferometrie, Einfluss der Referenzfläche, Dreiplattentest, Interferometrie streifendem Einfall, Twyman-Green Interferometer, Einsatzgrenzen) - Deflektometrische Formmessung (Überblick Deflektometrie, Grundprinzip, Extended Shear Angle Difference Methode, flächenhafte Deflektometrie, Einsatzgrenzen)
- **Photogrammetrie:** Grundprinzip, Stereophotogrammetrie, passive Triangulation, Grundlagen, aktive Triangulation (Punkttriangulation, linienhafte und flächenhafte Triangulation) - Streifenlichtprojektion (strukturierte Beleuchtung, Grundprinzip Ein- und Zweikerasysteme, Kodierung Gray Code, Phasenschiebung, Kombinierte Beleuchtung aus Gray Code und Phasenschiebung, Anwendung, Datenverarbeitung, Einsatzgrenzen)
- **Röntgen-Computertomografie:** Röntgenstrahlung, Grundprinzip der Röntgen-Computertomografie, Aufbau und Scanvarianten, Röntgenstrahlquellen, Strahlungsspektrum, Wechselwirkung mit Material (Photoelektrischer Effekt, Compton Streuung), Detektoren, Vergrößerung, Rekonstruktion (Radontransformation, algebraische Rekonstruktion, gefilterte Rückprojektion, Artefakte (Strahlauhfärtung, Ringartefakte, Streustrahlung, Scannerausrichtung), Schwellwertfindung, Anwendung (Defekterkennung, Micro- und Nano-CT, Hochenergie-CT, Multimaterial), Rückführung
- **Spezifikation und Messung optischer Komponenten:** Zeichnungen für optische Elemente und Systeme, Materialspezifikation, Spezifikation von Oberflächenformtoleranzen, Prüfung der Oberflächenformabweichungen (Passe) mit Probegläsern, Oberflächenbehandlungen und Beschichtungen, Messung geometrischer Spezifikationen
- **Mikro- und Nanomesstechnik:** Positioniersysteme (Führungen und Antriebe, Gewichtskraftkompensation), metrologischer Rahmen und Gerätekoordinatensysteme, Antastprinzipien und Messsystem (Rasterelektronenmikroskop, Rastertunnelmikroskop, Rasterkraftmikroskope, Nahfeldmikrosko-

pe, mikrotaktile Antastung), Mikro- und Nanokoordinatenmesssysteme, Einflussgrößen, Kalibrierung und Rückführung

## Lernziele und Kompetenzen:

### *Fachkompetenz*

#### *Wissen*

- Die Studierenden kennen relevante Definitionen, Fachbegriffe und Kriterien der Fertigungsmesstechnik.
- Die Studierenden können einen Überblick zur Gerätetechnik der Fertigungsmesstechnik sowie deren Funktionsweise und Einsatzgebiete wiedergeben
- Die Studierenden wissen um die operative Herangehensweise an Aufgaben der messtechnischen Erfassung von dimensionellen und geometrischen.

#### *Verstehen*

- Die Studierenden sind in der Lage die, den vorgestellten Messgeräten der Fertigungsmesstechnik, zugrundeliegenden Messprinzipien in eigenen Worten zu erläutern.
- Die Studierenden können Messaufgaben beschreiben und interpretieren, und Schwachstellen in der Planung und Durchführung erkennen.
- Die Studierenden können Messergebnisse und die zugrunde liegenden Verfahren angemessene kommunizieren und interpretieren.

#### *Anwenden*

- Die Studierenden können eigenständig geeignete Verfahren im Bereich Fertigungsmesstechnik auswählen.
- Die Studierenden können das Erlernte auf unbekannte, aber ähnliche Messaufgaben transferieren.

#### *Evaluieren (Beurteilen)*

- Die Studierenden können Messaufgaben in der Fertigungsmesstechnik beurteilen und strukturell analysieren.
- Die Studierenden sind in der Lage Messergebnisse zu hinterfragen und auf dieser Basis die Funktionalität des Messsystems sowie die zum Zeitpunkt der Messung vorherrschenden Messbedingungen zu bewerten.

#### *Erschaffen*

- Die Studierenden können die Eignungsuntersuchungen verschiedener Messprinzipien zur Erfüllung neuer Messaufgaben erstellen und auf deren Basis adaptierte Messsysteme konzipieren.

## Literatur:

- International Vocabulary of Metrology - Basic and General Concepts and Associated Terms, VIM, 3rd edition, JCGM 200:2008, <http://www.bipm.org/en/publications/guides/vim.html>
- DIN e.V. (Hrsg.): Internationales Wörterbuch der Metrologie - Grundlegende und allgemeine Begriffe und zugeordnete Benennungen (VIM) ISO/IEC-Leitfaden 99:2007. Korrigierte Fassung 2012, Beuth Verlag GmbH, 4. Auflage 2012
- Pfeifer, Tilo: Fertigungsmeßtechnik. R. Oldenbourg Verlag München Wien, 1998 - ISBN 3-486-24219-9
- Keferstein, Claus P.: Fertigungsmesstechnik. 7. Auflage, Vieweg+Teubner Verlag, 2011 - ISBN 978-3-8348-0692-5
- Warnecke, H.-J.; Dutschke, W.: Fertigungsmeßtechnik. Springer-Verlag Berlin Heidelberg New York Tokyo, 1984 - ISBN 3-540-11784-9
- Christoph, Ralf; Neumann, Hans Joachim: Multisensor-Koordinatenmesstechnik. 3. Auflage, Verlag Moderne Industrie, 2006 - ISBN 978-3-937889-51-2
- Neumann, Hans Joachim: Koordinatenmesstechnik im der industriellen Einsatz. Verlag Moderne Industrie, 2000 - ISBN 3-478-93212-2
- Weckenmann, A.: Koordinatenmesstechnik: Flexible Strategien für funktions- und fertigungsgerechtes Prüfen, 2. Auflage, Carl Hanser Verlag München, 2012

- Hausotte, Tino: Nanopositionier- und Nanomessmaschinen - Geräte für hochpräzise makro- bis nanoskalige Oberflächen- und Koordinatenmessungen. Pro Business Verlag, 2011 - ISBN 978-3-86805-948-9
- David J. Whitehouse: Handbook of Surface and Nanometrology, Crc Pr Inc., 2010 - ISBN 978-1420082012

---

### Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

#### [1] Informatik (Bachelor of Science)

(Po-Vers. 2009s | Nebenfach | Nebenfach Maschinenbau | Qualitätsmanagement und Messtechnik | Wahlpflichtmodule Qualitätsmanagement und Messtechnik für Nebenfach Informatik)

Dieses Modul ist daneben auch in den Studienfächern "Berufspädagogik Technik (Bachelor of Science)", "Berufspädagogik Technik (Master of Education)", "Energietechnik (Master of Science)", "Informatik (Master of Science)", "Maschinenbau (Bachelor of Science)", "Maschinenbau (Master of Science)", "Mechatronik (Bachelor of Science)", "Mechatronik (Master of Science)", "Medizintechnik (Master of Science)", "Wirtschaftsingenieurwesen (Master of Science)" verwendbar.

---

### Studien-/Prüfungsleistungen:

Fertigungsmesstechnik II (Prüfungsnummer: 69251)

(englische Bezeichnung: Manufacturing Metrology II)

Prüfungsleistung, Klausur mit MultipleChoice, Dauer (in Minuten): 60

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Fertigungsmesstechnik II
- Fertigungsmesstechnik II - Übung

weitere Erläuterungen:

**Prüfungstermine**, eine **allgemeine Regel der Prüfungstagvergabe** und **Termine der Klausureinsicht** finden Sie auf StudOn: Prüfungstermine und Termine der Klausureinsicht

Prüfungssprache: Deutsch

Erstablingung: SS 2019, 1. Wdh.: WS 2019/2020

1. Prüfer: Tino Hausotte (100620)

---

### Organisatorisches:

- Unterlagen zur Lehrveranstaltung werden auf der Lernplattform StudOn ([www.studon.uni-erlangen.de](http://www.studon.uni-erlangen.de)) bereitgestellt. Das Passwort wird in der Einführungsveranstaltung bekannt gegeben.

---

**Modulbezeichnung:** Hauptseminar Messtechnik (HS MT) 2.5 ECTS  
 (Advanced Seminar Manufacturing Metrology)

Modulverantwortliche/r: Tino Hausotte  
 Lehrende: T. Hausotte, Assistenten

---

Startsemester: WS 2019/2020	Dauer: 1 Semester	Turnus: halbjährlich (WS+SS)
Präsenzzeit: 30 Std.	Eigenstudium: 45 Std.	Sprache: Deutsch

---

**Lehrveranstaltungen:**

Hauptseminar Messtechnik (WS 2019/2020, Hauptseminar, 2 SWS, Anwesenheitspflicht, Tino Hausotte et al.)

---

**Inhalt:**

**Ablauf des Seminars**

**1. Voranmeldung StudOn**

- Onlineanmeldung über StudOn (Zeitraum siehe oben).

**2. Anmeldung zum Seminarthema**

- Zeitraum, in dem die Anmeldung zu einem bestimmten Seminarthemen möglich ist, wird angemeldeten Teilnehmern des StudOn-Kurses **per E-Mail** mitgeteilt.
- Hierfür wird eine Liste der Seminarthemen mit zugeordnete StudOn-Gruppen bereit gestellt.
- Die Anmeldung zu einem bestimmten Thema erfolgt durch selbstständige Anmeldung zur zugeordneten StudOn-Gruppe.
- Freigabe der gewünschten Einteilung durch den Koordinator spätestens eine Woche nach Anmeldezeitraum.

**3. Vorbereitung des Seminarthema**

- Kontaktaufnahme mit dem Betreuer, der das gewählte Seminarthema betreut. Klärung von Ziel, Auftrag und Kontext.
- Recherche, Auswahl der Informationen.
- Grobe Ablaufplanung der Präsentation (Begrüßung und Themenübersicht, Einstieg ins Thema, Transport der Inhalte, Themenbegrenzung), Ausstieg, Fragen und Diskussion).
- Feine Ablaufplanung: Detaillierung der Inhalte (Sinnvolle Gliederung, Inhaltlichen Fortgang visualisieren, Zum Thema immer wieder zurückkehren, Gedankensprünge vermeiden, Foliensprünge vermeiden, Layout für den „roten Faden“, Ringschluss zwischen Anfang und Ende schaffen).
- Erstellen der Präsentation (Vorlage auf StudOn beachten).
- Erstellen eines Seminar-Papers (Vorlage auf StudOn beachten).

**4. Durchführung der Präsentation**

- Terminplan der Präsentationen wird vom Koordinator festgelegt und per E-Mail mitgeteilt (Termine sind in der Regel gegen Ende der Vorlesungszeit).
- Termin zur Abgabe von Präsentation und Seminar-Paper wird vom Koordinator festgelegt und per E-Mail mitgeteilt (Termin ist in der Regel 3 Werktage vor dem ersten Präsentationstermin).
- Durchführung der Präsentation (Präsentationsdauer 20 min. + 10 min. Diskussion)
- Teilnahme an 8 weiteren Vorträgen.

**5. Notenbekanntgabe**

- Notenbekanntgabe über StudOn.
- Koordinator schickt den ausgestellten Schein direkt an das Prüfungsamt.
- Auf Anfrage Feedback vom Betreuer (sofern gewünscht).

**Lernziele und Kompetenzen:**

Die Studierenden:

- erlangen grundlegender Kenntnisse in Recherche, Themenaufbereitung und Präsentationstechniken,
- erarbeiten Schwerpunkte technischer Zusammenhänge bei einem gegebenen Thema,
- vertiefen eigenständig einen technischen Schwerpunkt an Hand eines konkreten Beispiels der Fertigungsmesstechnik,
- erlernen die Fähigkeit, sich in unbekannte Probleme einzuarbeiten und diese verständlich zu präsentieren,

- erlernen die Fähigkeit, als Zuhörer aktiv Fragen zu formulieren und technische Sachverhalte zu diskutieren,

---

### **Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

#### **[1] Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Nebenfach | Nebenfach Maschinenbau | Qualitätsmanagement und Messtechnik | Wahlpflichtmodule Qualitätsmanagement und Messtechnik für Nebenfach Informatik)

Dieses Modul ist daneben auch in den Studienfächern "Informatik (Master of Science)", "Maschinenbau (Bachelor of Science)", "Maschinenbau (Master of Science)", "Mechatronik (Bachelor of Science)", "Mechatronik (Master of Science)", "Wirtschaftsingenieurwesen (Bachelor of Science)", "Wirtschaftsingenieurwesen (Master of Science)" verwendbar.

---

### **Studien-/Prüfungsleistungen:**

Hauptseminar Messtechnik (Prüfungsnummer: 607629)

(englische Bezeichnung: Advanced Seminar Manufacturing Metrology)

Prüfungsleistung, Seminararbeit+Vortrag

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Hauptseminar Messtechnik

weitere Erläuterungen:

Die Gesamtmodulnote errechnet sich aus 25% der Seminararbeits- und 75% der Vortragsleistung.

Erstablingung: WS 2019/2020, 1. Wdh.: SS 2020

1. Prüfer: Tino Hausotte (100620)

---



---

**Modulbezeichnung:** Qualitätsmanagement II - Phasenübergreifendes Qualitätsmanagement (QM II) (Life-Cycle Oriented Quality Management) 2.5 ECTS

Modulverantwortliche/r: N.N.  
 Lehrende: N.N.

---

Startsemester: SS 2019	Dauer: 1 Semester	Turnus: jährlich (SS)
Präsenzzeit: 30 Std.	Eigenstudium: 45 Std.	Sprache: Deutsch

---

**Lehrveranstaltungen:**

Die Präsenzveranstaltung Qualitätsmanagement II entfällt im Sommersemester 2019. Zur Vorbereitung auf die Prüfung QM II oder auf die Modulabschlussprüfung QM können Sie jedoch den als äquivalent geltenden E-Learning-Kurs Qualitätsmanagement - QMaK über StudOn ([https://www.studon.fau.de/studon/login.php?target=crs\\_2493916](https://www.studon.fau.de/studon/login.php?target=crs_2493916)) belegen. Qualitätsmanagement II - Phasenübergreifendes Qualitätsmanagement (SS 2019, Vorlesung mit Übung, 2 SWS, N.N.)

---

**Inhalt:**

- Qualitätsmanagementsystem - Auditierung und Zertifizierung
- Total Quality Management und EFQM-Modell
- Rechnerunterstützung im Qualitätsmanagement
- Ausbildung und Motivation
- Kontinuierliche Verbesserungsprogramme und Benchmarking
- Problemlösungstechniken und Qualitätszirkel
- Qualitätsbewertung
- Qualität und Wirtschaftlichkeit
- Six Sigma
- Qualität und Umwelt, Umweltmanagement
- Qualität und Recht, Sicherheit
- *Qualitätsbewertung (Übung)*
- *Qualitätsbezogene Kosten und Wirtschaftlichkeit (Übung)*
- *Ökobilanzierung (Übung)*

**Lernziele und Kompetenzen:**

Nach dem Besuch der Veranstaltung sind die Teilnehmenden in der Lage,

Wissen:

- Motivation, Ziele, Grundsätze und Strategien des prozessorientierten Qualitätsmanagements wiederzugeben

Verstehen:

- Wissen zu Qualitätsmanagement als unternehmens- und produktlebenszyklusübergreifende Strategie zu veranschaulichen
- Anforderungen, Aufbau, Einführung sowie die Auditierung und Zertifizierung von Qualitätsmanagementsystemen darzustellen

Anwenden:

- Business Excellence anhand Total Quality Management (TQM), Unternehmensbewertungsmodelle wie EFQM und kontinuierlicher Verbesserungsprozesse im Unternehmen auszuführen
- die Interaktion von Qualitätsmanagement mit rechtlichen und sicherheitsrelevanten Themen zu erklären
- den Zusammenhang zwischen Qualitätsmanagement und Umweltmanagement zu beschreiben
- die Wirtschaftlichkeit von Qualitätsverbesserungsmaßnahmen zu demonstrieren
- die Methodik „Six Sigma“ zu beschreiben und dem Kontext der Qualitätsverbesserung zuzuordnen

Analysieren:

- Handlungsgrundlagen hinsichtlich Ausbildungs-, Motivations- und Organisationsverbesserung zu ermitteln

Evaluiieren:

- die Qualität mit etablierten Vorgehensweisen zu bewerten

## Literatur:

- Kamiske, G. F.; Brauer, J.-P.: Qualitätsmanagement von A - Z, Carl Hanser Verlag, München 2005
- Masing, W.: Handbuch Qualitätsmanagement, Carl Hanser Verlag, München 2007
- Wagner, K. W.; Patzak, G.: Performance Excellence - Der Praxisleitfaden zum effektiven Prozessmanagement, Carl Hanser Verlag, München 2007
- Zink, K. J.: Mitarbeiterbeteiligung bei Verbesserungs- und Veränderungsprozessen, Carl Hanser Verlag, München 2007

---

## Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

### [1] Informatik (Bachelor of Science)

(Po-Vers. 2009s | Nebenfach | Nebenfach Maschinenbau | Qualitätsmanagement und Messtechnik | Wahlpflichtmodule Qualitätsmanagement und Messtechnik für Nebenfach Informatik)

Dieses Modul ist daneben auch in den Studienfächern "Energietechnik (Master of Science)", "Informatik (Master of Science)", "Maschinenbau (Bachelor of Science)", "Maschinenbau (Master of Science)", "Mechatronik (Master of Science)", "Medizintechnik (Bachelor of Science)", "Wirtschaftsingenieurwesen (Bachelor of Science)", "Wirtschaftsingenieurwesen (Master of Science)" verwendbar.

---

## Studien-/Prüfungsleistungen:

Qualitätsmanagement II - Phasenübergreifendes Qualitätsmanagement (Prüfungsnummer: 30301)

(englische Bezeichnung: Life-Cycle Oriented Quality Management)

Prüfungsleistung, Klausur, Dauer (in Minuten): 60

Anteil an der Berechnung der Modulnote: 100%

weitere Erläuterungen:

**Prüfungstermine, eine allgemeine Regel der Prüfungstagvergabe und Termine der Klausureinsicht** finden Sie auf StudOn: Prüfungstermine und Termine der Klausureinsicht

Erstblegung: SS 2019, 1. Wdh.: WS 2019/2020

1. Prüfer: Tino Hausotte (100620)

---

---

**Modulbezeichnung:** Prozess- und Temperaturmesstechnik (PTMT) 5 ECTS  
 (Process and Temperature Metrology)

Modulverantwortliche/r: Tino Hausotte  
 Lehrende: Tino Hausotte

---

Startsemester: WS 2019/2020	Dauer: 1 Semester	Turnus: jährlich (WS)
Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Deutsch

---

**Lehrveranstaltungen:**

Prozess- und Temperaturmesstechnik (WS 2019/2020, Vorlesung, 2 SWS, Tino Hausotte)  
 Prozess- und Temperaturmesstechnik - Übung (WS 2019/2020, Übung, 2 SWS, Tino Hausotte et al.)

---

**Empfohlene Voraussetzungen:**

- Der Besuch der Grundlagen-Vorlesungen *Grundlagen der Messtechnik* (GMT) wird empfohlen.
- 

**Inhalt:**

- **Temperaturmesstechnik:** Messgröße Temperatur: (thermodynamische Temperatur, Symbole, Einheiten, Temperatur als intensive Größe, Prinzip eines Messgerätes, direkte Messung und Voraussetzungen, indirekte Temperaturmessung und Voraussetzungen, Überblick primäre Temperaturmessverfahren, unmittelbar und mittelbare Temperaturmessung) - Prinzipielle Einteilung der Temperaturmessverfahren - Temperaturskalen: praktische Temperaturskalen (Tripelpunkte, Schmelz- und Erstarrungspunkte), klassische Temperaturskalen (Benennung und Fixpunkte), ITS 90 (Bereich, Fixpunkte, Interpolationsinstrumente - Mechanische Berührungsthermometer - Widerstandsthermometer (Pt100, NTC, PTC, Kennlinie, Messschaltungen) - Thermoelemente (Grundlagen, Aufbau, Vergleichsstelle, Bauformen) - Spezielle Temperaturmessverfahren (Rauschtemperaturmessung, Quarz-Thermometer) - Strahlungsthermometer - Statik und Dynamik thermischer Sensoren
- **Druck- und Durchflussmesstechnik:** Definition des Druckes, Druckarten, Fluide im Schwerfeld - Druckwaage (Kolbenmanometer) - Druckmessung mit Sperrflüssigkeit (U-Rohrmanometer und U-Rohrbarometer, Gefäßmanometer, Schrägrohrmanometer, Ringwaage) - Rohrfedermanometer, Plattenfedermanometer, Kapselfedermanometer - Druckmessumformer (DMS-Drucksensoren, Piezoresistive Drucksensoren, Kapazitive Drucksensoren) - Druckmittler (Druckvorlagen oder Trennvorlagen)
- **Füllstand und Grenzstand:** Füllstandsmessung, Grenzstandmessung - Peilstäbe, Schaugläser, Schwimmermessgeräte - Elektromechanische Lotsysteme, Tastplattenmessung, Vedrängergeräte - Hydrostatische Füllstandsmessung - Behälterwägung - Kapazitive Messverfahren - Radiometrische Messung - Laufzeitmessung
- **Messumformertechnik**

**Content**

- **Temperature measurement:** Measure "temperature" (thermodynamic temperature, symbols, units, temperature and intensive quantity, principle of a measuring instrument, and direct measurement conditions, indirect temperature measurement and conditions Overview primary temperature measurement methods, direct and indirect temperature measurement) - Basic classification of temperature measurement methods - Temperature scales: practical temperature scales (triple points, melting and solidification points), classical temperature scales (naming and fixed points), ITS 90 (range, fixed points, interpolating instruments) - Mechanical contact thermometers - Resistance thermometer (Pt100, NTC, PTC, characteristic, measurement circuits) - Thermocouples (foundations, structure, junction, mounting positions) - Special methods of temperature measurement (noise temperature measurement, quartz thermometer) - Pyrometer - Static and dynamic thermal sensors
- **Pressure and Flow Measurement:** Definition of stress, pressure types, fluids in the gravitational field - Pressure balance (Deadweight) - Pressure measurement with barrier fluid (U-tube manometer and U-tube barometer, tube manometer, Inclined, ring horizontally) - Bourdon tube pressure gauge, Diaphragm, Capsule spring manometer - Pressure transducer (strain gauge pressure sensors, piezo resistive pressure sensors, capacitive pressure sensors) - Pressure Transmitter (print templates or templates release)

- **Level and point level:** Level measurement, point level measurement - Dipsticks, sight glasses, float gauges - Electromechanical normal systems, touch plate measurement, displacement body devices - Hydrostatic level measurement - Vessel Weighing - Capacitive measuring method - Radiometric measurement - Acoustical logging
- **Converter Technology**

#### Lernziele und Kompetenzen:

- Motivation, Ziele, Grundsätze und Strategien der Prozessmesstechnik erfassen.
- Beurteilen und strukturelle Analyse von Messaufgaben in den genannten Bereichen. Transfer des Erlernten auf unbekannte, aber ähnliche Messaufgaben.
- Verständnis um die operative Herangehensweise an Aufgaben der messtechnischen Erfassung von nicht-geometrischen Prozessgrößen.
- Eigenständige Auswahl geeigneter Verfahren im Bereich Prozess- und Temperaturmesstechnik.
- Beschreiben von Messaufgaben, Durchführen, Auswerten von Messungen.
- Selbstständiges Erkennen von Schwachstellen in der Planung und Durchführung.
- Bewerten von Messergebnissen aus dem Bereich Prozessmesstechnik.
- Angemessene Kommunikation und Interpretation von Messergebnissen und der zugrunde liegenden Verfahren.

#### Literatur:

- Hoffmann, Jörg: Handbuch der Messtechnik. 4. Auflage, Carl Hanser Verlag München, 2012 - ISBN 978-3-446-42736-5
- Bernhard, Frank: Technische Temperaturmessung. Springer-Verlag Berlin Heidelberg New York, 2004 - ISBN 3-540-62672-7
- Freudenberger, Adalbert: Prozeßmeßtechnik. Vogel Buchverlag, 2000 - ISBN 978-3802317538
- Kohlrausch, Friedrich: Praktische Physik : zum Gebrauch für Unterricht, Forschung und Technik. Band 1-3, 24. Auflage, Teubner Verlag, 1996 - ISBN 3-519-23001-1, 3-519-23002-X, 3-519-23000-3
- DIN e.V. (Hrsg.): Internationales Wörterbuch der Metrologie - Grundlegende und allgemeine Begriffe und zugeordnete Benennungen (VIM) ISO/IEC-Leitfaden 99:2007. Beuth Verlag GmbH, 3. Auflage 2010

#### Internetlinks für weitere Information zum Thema Messtechnik

- Video des VDI: Messtechnik - Unsichtbare Präzision überall

#### Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

##### [1] Informatik (Bachelor of Science)

(Po-Vers. 2009s | Nebenfach | Nebenfach Maschinenbau | Qualitätsmanagement und Messtechnik | Wahlpflichtmodule Qualitätsmanagement und Messtechnik für Nebenfach Informatik)

Dieses Modul ist daneben auch in den Studienfächern "Berufspädagogik Technik (Bachelor of Science)", "Berufspädagogik Technik (Master of Education)", "Energietechnik (Master of Science)", "Informatik (Master of Science)", "Maschinenbau (Bachelor of Science)", "Maschinenbau (Master of Science)", "Materialwissenschaft und Werkstofftechnik (Master of Science)", "Mechatronik (Bachelor of Science)", "Mechatronik (Master of Science)", "Medizintechnik (Master of Science)", "Wirtschaftsingenieurwesen (Bachelor of Science)", "Wirtschaftsingenieurwesen (Master of Science)" verwendbar.

#### Studien-/Prüfungsleistungen:

Prozess- und Temperaturmesstechnik (Prüfungsnummer: 72481)

(englische Bezeichnung: Process and Temperature Metrology)

Prüfungsleistung, Klausur mit MultipleChoice, Dauer (in Minuten): 60

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Prozess- und Temperaturmesstechnik
- Prozess- und Temperaturmesstechnik - Übung

weitere Erläuterungen:

**Prüfungstermine**, eine **allgemeine Regel der Prüfungstagvergabe** und **Termine der Klausureinsicht** finden Sie auf StudOn: Prüfungstermine und Termine der Klausureinsicht

Erstablingung: WS 2019/2020, 1. Wdh.: SS 2020

1. Prüfer: Tino Hausotte (100620)

---

**Organisatorisches:**

- Unterlagen zur Lehrveranstaltung werden auf der Lernplattform StudOn ([www.studon.uni-erlangen.de](http://www.studon.uni-erlangen.de)) bereitgestellt. Das Passwort wird in der Einführungsveranstaltung bekannt gegeben.

---

**Modulbezeichnung:** Qualitätsmanagement I - Qualitätstechniken für die Produktentstehung (QM I) 2.5 ECTS  
 (Quality Management I - Quality Techniques for Product Development and Manufacturing)

Modulverantwortliche/r: Tino Hausotte  
 Lehrende: Tino Hausotte

---

Startsemester: WS 2019/2020	Dauer: 1 Semester	Turnus: jährlich (WS)
Präsenzzeit: 30 Std.	Eigenstudium: 45 Std.	Sprache: Deutsch

---

**Lehrveranstaltungen:**

- Die virtuelle Lehrveranstaltung QTeK gilt als äquivalent zur **ehemaligen** Präsenzvorlesung Qualitätsmanagement I - Qualitätstechniken für die Produktentstehung (QM I). Eine Prüfungsleistung über die Lehrveranstaltung kann nur einmal eingebracht werden (entweder QTeK oder QM I).
  - Eine Kombination der einzeln abgelegten Prüfungsleistungen QTeK und QMaK als Wahlpflichtmodul **Qualitätsmanagement (QM)** ist nicht möglich.
- Qualitätstechniken - QTeK - vhb (WS 2019/2020, Vorlesung, 2 SWS, Tamara Reuter et al.)
- 

**Inhalt:**

- Einführung und Begriffe
- Grundwerkzeuge des Qualitätsmanagements
- Erweiterte Werkzeuge des Qualitätsmanagements
- Qualitätsmanagement in der Produktplanung (QFD)
- Qualitätsmanagement in der Entwicklung und Konstruktion (DR, FTA, ETA, FMEA)
- Versuchsmethodik
- Maschinen- und Prozessfähigkeit, Qualitätsregelkarten
- Zuverlässigkeitstechniken
- Qualitätsmanagementsystem - Aufbau und Einführung
- *Grundwerkzeuge des QM (Übung)*
- *QFD und FMEA (Übung)*
- *Versuchsmethodik (Übung)*
- *SPC (Übung)*

**Lernziele und Kompetenzen:**

Nach dem Besuch der Veranstaltung sind die Teilnehmenden in der Lage,

Wissen:

- die Motivation, Ziele, Grundsätze und Strategien des prozessorientierten Qualitätsmanagements darzulegen

Verstehen:

- die Werkzeuge, Techniken und Methoden des Qualitätsmanagements entlang des Produktlebenszyklus darzustellen
- die Zuverlässigkeit von Systemen zu beschreiben
- den Aufbau und die Einführung von Qualitätsmanagementsystemen darzustellen

Anwenden:

- die grundlegenden Qualitätsmethoden, -techniken und -werkzeuge auf ein anderes Problem zu übertragen
- Prozesse mit Hilfe der statistischen Prozesslenkung (SPC), Qualitätsregelkarten und Prozessfähigkeitsindizes zu beschreiben

Analysieren

- mit Hilfe der Qualitätsmethoden, -techniken und -werkzeugen Probleme zu analysieren
- statistische Versuchspläne auf praktische Probleme zu übertragen und aus den Ergebnissen die Zusammenhänge und Einflüsse der Faktoren zu interpretieren

Evaluiieren:

- statistische Auswertungen zu interpretieren und neue Probleme auf statistische Auffälligkeiten zu testen

**Literatur:**

- DGQ e.V. (Hrsg.): DGQ-Schrift 11-04: Managementsysteme Begriffe, Beuth Verlag, Berlin 2002
- DIN (Hrsg.): Internationales Wörterbuch der Metrologie, Beuth-Verlag, Berlin 1994
- Masing, W.: Handbuch Qualitätsmanagement, Carl HanserVerlag, München 2007

---

### Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

#### [1] Informatik (Bachelor of Science)

(Po-Vers. 2009s | Nebenfach | Nebenfach Maschinenbau | Qualitätsmanagement und Messtechnik | Wahlpflichtmodule Qualitätsmanagement und Messtechnik für Nebenfach Informatik)

Dieses Modul ist daneben auch in den Studienfächern "Energietechnik (Master of Science)", "Informatik (Master of Science)", "Maschinenbau (Bachelor of Science)", "Maschinenbau (Master of Science)", "Materialwissenschaft und Werkstofftechnik (Master of Science)", "Mechatronik (Bachelor of Science)", "Mechatronik (Master of Science)", "Medizintechnik (Bachelor of Science)", "Wirtschaftsingenieurwesen (Bachelor of Science)", "Wirtschaftsingenieurwesen (Master of Science)" verwendbar.

---

### Studien-/Prüfungsleistungen:

Qualitätsmanagement I - Qualitätstechniken für die Produktentstehung (Prüfungsnummer: 59401)

(englische Bezeichnung: Quality Management I - Quality Techniques for Product Development and Manufacturing)

Prüfungsleistung, Klausur mit MultipleChoice, Dauer (in Minuten): 60

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Qualitätstechniken - QTeK - vhb

weitere Erläuterungen:

**Prüfungstermine**, eine **allgemeine Regel der Prüfungstagvergabe** und **Termine der Klausureinsicht** finden Sie auf StudOn: Prüfungstermine und Termine der Klausureinsicht

Erstablingung: WS 2019/2020, 1. Wdh.: SS 2020

1. Prüfer: Tino Hausotte (100620)

---

---

**Modulbezeichnung: Rechnergestützte Messtechnik (RMT)** **5 ECTS**  
 (Computer-Aided Metrology)

 Modulverantwortliche/r: Tino Hausotte  
 Lehrende: Tino Hausotte

---

Startsemester: SS 2019	Dauer: 1 Semester	Turnus: jährlich (SS)
Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Deutsch

---

**Lehrveranstaltungen:**

 Rechnergestützte Messtechnik (SS 2019, Vorlesung, 2 SWS, Tino Hausotte)  
 Rechnergestützte Messtechnik - Übung (SS 2019, Übung, 2 SWS, Tino Hausotte et al.)

**Inhalt:**

- **Grundlagen:** Grundbegriffe (Größe, Größenwert, Messgröße, Maßeinheit, Messprinzip, Messung, Messkette, Messsignal, Informationsparameter, analoges und digitales Signal) - Prinzip eines Messgerätes, direkte und indirekte Messmethode, Kennlinie und Kennlinienarten, analoge und digitale Messmethoden, kontinuierliche und diskontinuierliche Messung, Zeit- und Wertdiskretisierung, Auflösung, Empfindlichkeit, Messbereich - Signal, Messsignal, Klassifizierung von Signalen (Informationsparameter) - Signalbeschreibung, Fourierreihen und Fouriertransformation - Fourieranalyse - DFT und FFT (praktische Realisierung) - Aliasing und Shannon's-Abtasttheorem - Übertragungsverhalten (Antwortfunktionen, Frequenzgang, Übertragungsfunktion) - Laplace-Transformation, Digitalisierungskette, Z-Transformation und Wavelet-Transformation
- **Verarbeitung und Übertragung analoger Signale:** Messverstärker, Operationsverstärker (idealer und realer, Rückkopplung) - Kenngrößen von Operationsverstärkern - Frequenzabhängige Verstärkung von Operationsverstärkern - Operationsverstärkertypen - Rückkopplung und Grundsaltungen (Komparator, Invertierender Verstärker, Nichtinvertierender Verstärker, Impedanzwandler, Strom-Spannungs-Wandler, Differenzverstärker, Integrierer, Differenzierer, invertierender Addierer, Subtrahierer, Logarithmierer, e-Funktionsgeneratoren, Instrumentenverstärker) - OPV mit differentiellen Ausgang - analoge Filter (Tiefpassfilter, Hochpassfilter, Bandpassfilter, Bandsperrfilter, Bodeplot, Phasenschiebung, aktive analoge Filter) - Messsignalübertragung (Einheitssignale, Anschlussvarianten) - Spannungs-Frequenz-Wandler - Galvanische Trennung und optische Übertragung - Modulatoren und Demulatoren - Multiplexer und Demultiplexer - Abtast-Halte-Glied
- **A/D- und D/A-Umsetzer:** Digitale und analoge Signale - Digitalisierungskette - A/D-Umsetzer (Nachlauf ADU, Wägeverfahren, Rampen-A/D-Umsetzer, Dual Slope-Verfahren, Charge-Balancing-A/D-Umsetzverfahren, Parallel-A/D-Umsetzer, Kaskaden-A/D-Umsetzverfahren, Pipeline-A/D-Umsetzer, Delta-Sigma-A/D-Umsetzer / 1-Bit- bis N-Bit-Umsetzer, Einsatzbereiche, Kennwerte, Abweichungen, Signal-Rausch-Verhältnis) - Digital-Analog-Umsetzungskette - D/A-Umsetzer (Direkt bzw. Parallelumsetzer, Wägeumsetzer, Zählverfahren, Pulsweitenmodulation, Delta-Sigma-Umsetzer / 1-Bit- bis N-Bit-Umsetzer)
- **Verarbeitung digitaler Signale:** digitale Codes - Schaltnetze (Kombinatorische Schaltungslogik) - Schaltalgebra und logische Grundverknüpfungen - Schaltwerke (Sequentielle Schaltnetze) - Speicherglieder (Flip-Flops, Sequentielle Grundsaltungen), Halbleiterspeicher (statische und dynamische, FIFO) - Anwendungsspezifische integrierte Schaltungen (ASICs) - Programmierbare logische Schaltung (PLDs, Programmierbarkeit, Vorteile, Anwendungen, Programmierung) - Rechnerarten
- **Bussysteme:** Bussysteme (Master, Slave, Arbitr, Routing, Repeater) - Arbitrierung - Topologien (physikalische und logische Topologie, Kennwerte, Punkt-zu-Punkt-Topologie, vermaschtes Netz, Stern-Topologie, Ring-Topologie, Bus-Topologie, Baum-Topologie, Zell-Topologie) - Übertragungsmedien (Mehrdrahtleitung, Koaxialkabel, Lichtwellenleiter) - ISO-OSI-Referenzmodell - Physikalische Schnittstellenstandards (RS-232C, RS-422, RS-485) - Feldbussysteme, GPIB (IEC-625-Bus), Messgerätebusse
- **USB Universal Serial Bus:** Struktur des Busses - Verbindung der Geräte, Transceiver, Geschwindigkeitserkennung, Signalkodierung - Übertragungsarten (Control-Transfer, Bulk-Transfer, Isochrone-Transfer, Interrupt-Transfer, Datenübertragung mit Paketen) - Frames und Mikroframes, Geschwin-



digkeiten, Geschwindigkeitsumsetzung mit Hub - Deskriptoren und Software - Layer Entwicklungstools - Compliance Test - USB 3.0

- **Digitale Filter:** Analoge Filter - Eigenschaften und Charakterisierung von digitalen Filtern - Digitale Filter (Implementierung, Topologien, IIR-Filter und FIR-Filter) und Formen - Messwert-Dezimierer, digitaler Mittelwertfilter, Gaußfilter - Fensterfunktionen, Gibbs-Phänomen - Realisierung mit MATLAB - Vor- und Nachteile digitaler Filter
- **Messdatenauswertung:** Absolute, relative, zufällige und systematische Messabweichungen, Umgang mit Messabweichungen, Kalibrierung - Korrelationsanalyse - Kennlinienabweichungen und Methoden zu deren Ermittlung - Regressionsanalyse - Kennlinienkorrektur - Approximation, Interpolation, Extrapolation - Arten der Kennlinienkorrektur - Messpräzision, Messgenauigkeit, Messrichtigkeit, Fehlerfortpflanzungsgesetz (altes Konzept), Messunsicherheit und deren Bestimmung - Vorgehensweise zur Ermittlung der Unsicherheit, Monte-Carlo-Methode
- **Schaltungs- und Leiterplattenentwurf:** Leiterplatten - Leiterplattenmaterial - Leiterplattenarten - Durchkontaktierungen - Leiterplattenentwurf und -entflechtung - Software - Leiterplattenherstellung

## Contents

- **Basics:** Terms (quantity, quantity value, measurand, measurement unit, principle of measurement, measurement, measuring chain, measurement signal, information parameter, analogue and digital signal) - Principle of a measuring instrument, direct and indirect measurement, characteristic curves and characteristic curve types, analogue and digital measuring methods, continuous and discontinuous measurement, time and value discretisation, resolution, sensitivity, measuring interval (range) - Signal, measurement signal, classification of signals (information parameter) - Signal description, Fourier series and Fourier transformation - Fourier analysis - DFT and FFT (practical realization) - Aliasing and Shannon's sampling theorem - Transfer behaviour (response functions, frequency response, transfer function) - Laplace transform, digitisation chain, Z-transform and wavelet transform
- **Processing and transmission of analogue signals:** Measuring amplifiers, operational amplifiers (ideal and real, feedback) - Characteristics of operational amplifiers - Frequency-dependent gain of operational amplifiers - Operational amplifier types - Feedback and basic circuits (comparator, inverting amplifier, non-inverting amplifier, impedance converter, current-voltage converter, differential amplifier, integrator, differentiator, inverting adder, subtractor, logarithmic, exponential function generators, instrumentation amplifier) - OPV with differential output - Analogue filter (low pass filter, high pass filter, band pass filter, band elimination filter, Bodeplot, phase shifting, active analogue filters) - Measurement signal transmission (standard signals, connection variants) - Voltage-frequency converters - Galvanic isolation and optical transmission - modulators and demodulators - multiplexers and demultiplexers - sample-and-hold amplifier
- **A/D and D/A converter:** Digital and analogue signals - Digitisation chain - A/D converter (follow-up ADC, weighing method, ramp A/D converter, dual slope method, charge-balancing ADC, parallel ADC, cascade ADC, pipeline A/D converter, the delta-sigma A/D converter / 1-bit to N-bit converter, application, characteristics, deviations, signal-to-noise ratio) - Digital-to-analogue conversion chain - D/A converter (direct or parallel converters, weighing method, counting method, pulse width modulation, delta-sigma converter / 1-bit to N-bit converter)
- **Digital signal processing:** Digital codes - Switching networks (combinatorial circuit logic) - Boolean algebra and basic logic operations - Sequential circuit (sequential switching networks) - Storage elements (flip-flops, sequential basic circuits), semiconductor memory (static and dynamic, FIFO) - Application Specific Integrated Circuits (ASICs) - The programmable logic device (PLD, programmability, benefits, applications, programming) - computer types
- **Data bus systems:** Bus systems (master, slave, arbiter, routing, repeater) - Arbitration - Topologies (physical and logical topology, characteristics, point-to-point topology, mesh network, star topology, ring topology, bus topology, tree topology, cell topology) - Transmission media (multi-wire cable, coaxial cable, fibre optic cable) - ISO OSI reference model - Physical interface standards (RS-232C, RS-422, RS-485) - Fieldbus systems, GPIB (IEC-625 bus) , Measuring device buses
- **USB Universal Serial Bus:** Bus structure - Connection of the devices, transceiver, speed detection, signal coding - Transfer types (control transfer, bulk transfer, isochronous transfer, interrupt transfer,

data transfer with packages) - Frames and micro-frames, speeds, speed conversion with hubs - Descriptors and software - Layer development tools - Compliance test - USB 3.0

- **Digital filters:** Analogue filter - Properties and characterization of digital filters - Digital Filter (implementation, topologies, IIR filters and FIR filters) and forms - Measurement value decimator, digital averaging filter, Gaussian filter - Window functions, Gibbs phenomenon - Realisation with MATLAB - Advantages and disadvantages of digital filters
- **Data analysis:** Absolute, relative, random and systematic errors, handling of measurement errors, calibration - Correlation analysis - Characteristic curve deviations and methods for their determination - Regression analysis - Characteristic curve correction - Approximation, interpolation, extrapolation - Kinds of characteristic curve correction - Measurement precision, measurement accuracy, measurement trueness, error propagation law (old concept), uncertainty and their estimation - Procedure for determining the uncertainty, Monte Carlo method
- **Circuit and PCB design:** Printed circuit boards (PCB) - PCB material - PCB types - Vias - PCB design and deconcentration - Software - PCB production

### Lernziele und Kompetenzen:

#### *Fachkompetenz*

##### *Wissen*

- Die Studierenden können einen Überblick zur rechnergestützten Messtechnik sowie deren Einsatzgebiete wiedergeben.
- Die Studierenden können Wissen zur rechnergestützten Messdatenerfassung, -auswertung, -analyse und -visualisierung als Grundlage für zielorientierte, effiziente Entwicklung und für kontinuierliche Produkt- und Prozessverbesserung abrufen

##### *Verstehen*

- Die Studierenden können Konzepte zur Sensorintegration und Datenfusion beschreiben

##### *Evaluieren (Beurteilen)*

- Die Studierenden können rechnergestützte Werkzeuge für die Messdatenerfassung, -auswertung, -analyse und -visualisierung auswählen und bewerten.

### Literatur:

- International Vocabulary of Metrology - Basic and General Concepts and Associated Terms, VIM, 3rd edition, JCGM 200:2008, <http://www.bipm.org/en/publications/guides/vim.html>
- DIN e.V. (Hrsg.): Internationales Wörterbuch der Metrologie - Grundlegende und allgemeine Begriffe und zugeordnete Benennungen (VIM) ISO/IEC-Leitfaden 99:2007. Korrigierte Fassung 2012, Beuth Verlag GmbH, 4. Auflage 2012
- Hoffmann, Jörg: Handbuch der Messtechnik. 4. Auflage, Carl Hanser Verlag München, 2012 - ISBN 978-3-446-42736-5
- Lerch, Reinhard: Elektrische Messtechnik. 6. Auflage, Springer-Verlag Berlin Heidelberg, 2012 - ISBN 978-3-642-22608-3
- Richter, Werner: Elektrische Meßtechnik. 3. Auflage, Verlag Technik Berlin, 1994 - ISBN 3-341-01106-4
- H. Czichos (Hrsg.): Das Ingenieurwissen Gebundene. 7. Auflage, Springer Verlag, 2012, ISBN 978-3-642-22849-0.
- Best, Roland: Digitale Meßwertverarbeitung. Oldenbourg München, 1991 - ISBN 3-486-21573-6.
- E DIN IEC 60050-351:2013-07 International Electrotechnical Vocabulary - Part 351: Control technology / Internationales Elektrotechnisches Wörterbuch - Teil 351: Leittechnik.
- DIN 44300:1982-03 Informationsverarbeitung; Begriffe.
- DIN 44300-1:1995-03 Informationsverarbeitung - Begriffe - Teil 1: Allgemeine Begriffe.
- DIN 40900-12:1992-09 Graphische Symbole für Schaltungsunterlagen; Binäre Elemente.

---

### Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

## [1] Informatik (Bachelor of Science)

(Po-Vers. 2009s | Nebenfach | Nebenfach Maschinenbau | Qualitätsmanagement und Messtechnik | Wahlpflichtmodule Qualitätsmanagement und Messtechnik für Nebenfach Informatik)

Dieses Modul ist daneben auch in den Studienfächern "Berufspädagogik Technik (Bachelor of Science)", "Berufspädagogik Technik (Master of Education)", "Informatik (Master of Science)", "Maschinenbau (Bachelor of Science)", "Maschinenbau (Master of Science)", "Mechatronik (Bachelor of Science)", "Mechatronik (Master of Science)", "Medizintechnik (Master of Science)", "Wirtschaftsingenieurwesen (Master of Science)" verwendbar.

---

### Studien-/Prüfungsleistungen:

Rechnergestützte Messtechnik (Prüfungsnummer: 69301)

(englische Bezeichnung: Computer-Aided Metrology)

Prüfungsleistung, Klausur mit MultipleChoice, Dauer (in Minuten): 60

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Rechnergestützte Messtechnik
- Rechnergestützte Messtechnik - Übung

weitere Erläuterungen:

**Prüfungstermine**, eine **allgemeine Regel der Prüfungstagvergabe** und **Termine der Klausureinsicht** finden Sie auf StudOn: Prüfungstermine und Termine der Klausureinsicht

Prüfungssprache: Deutsch

Erstablingung: SS 2019, 1. Wdh.: WS 2019/2020

1. Prüfer: Tino Hausotte (100620)

---

---

**Modulbezeichnung:** Technische Darstellungslehre II (TD II) 2.5 ECTS  
 (Technical Drawing II)

Modulverantwortliche/r: Sandro Wartzack  
 Lehrende: Benjamin Gerschütz

---

Startsemester: SS 2019	Dauer: 1 Semester	Turnus: jährlich (SS)
Präsenzzeit: 25 Std.	Eigenstudium: 50 Std.	Sprache: Deutsch

---

**Lehrveranstaltungen:**

Technische Darstellungslehre II (SS 2019, Praktikum, 2 SWS, Benjamin Gerschütz et al.)

---

**Inhalt:**

- Technologie des Computer Aided Design
- Einführung in die virtuelle Produktentwicklung mit CAD-Systemen
- Grundlagen des CAD: Arten von 3D-Modellierern, Systemmodule und Eigenschaften von Modellen
- Modellierungsstrategien, Vorgehensweise bei der Modellierung, Grundprinzipien, Besondere Modellierungsvereinfachungen im Zusammenhang mit genormten Darstellungen
- Rechnerübung mit Hausübung an CAD-Systemen zum Anfertigen von Bauteilen, Baugruppen und technischen Zeichnungen

**Lernziele und Kompetenzen:**

*Fachkompetenz*

*Wissen*

Verständnis für Funktion, Aufbau und Bedienung von im industriellen Umfeld eingesetzten, vollparametrischen 3D-CAD-Systemen und Verständnis für die Bedeutung von CAD-Systemen als zentralem Synthesewerkzeug des rechnerunterstützten Produktentwicklungsprozesses im Maschinenbau und in verwandten Disziplinen, hierzu

- Grundwissen über die einzelnen Phasen des Produktlebenszyklus und die Möglichkeiten der Rechnerunterstützung (CAx)
- Wissen über den Einsatz von CAD zur Definition der Produktgestalt im Hinblick auf eine durchgängige Verwendung der erzeugten Daten als Grundlage für weitere CAx-Werkzeuge sowie für die Ableitung normgerechter Zeichnungen und Stücklisten
- Wissen über die Geometrieverarbeitung auf Rechnersystemen: Historische Entwicklung, Stand der Technik, Grundfunktionalitäten moderner CAD-Systeme, Parametrik, Assoziative Datenspeicherung, Features und Konstruktionselemente, historienbasierte und direkte Modellierung.

*Erschaffen*

Erstellen von Einzelteilen durch Modellieren von Volumenkörpern unter Berücksichtigung einer robusten Modellierungsstrategie, hierzu

- Definieren von Geometriereferenzen und zweidimensionalen Skizzen als Grundlage für Konstruktionselemente
- Erzeugen von Volumenkörpern mit Hilfe der Konstruktionselemente Profilextrusion, Rotation, Zug und Verbund
- Kombinieren von Volumenkörpern durch BOOLEsche Operationen zu Rohbauteilen gemäß eines spanenden Fertigungsverfahrens
- Detaillieren von Rohbauteilen durch Hinzufügen von Bohrungen, Fasen und Metainformationen (z. B. Toleranzangaben)
- Nachträgliches Ändern der Geometrie mit Hilfe von Parametrik.

Erstellen von Baugruppen durch Kombination von Einzelteilen unter Verwendung von Normteillbibliotheken, hierzu

- Planen einer Baugruppenhierarchie im Hinblick auf Robustheit
- Verarbeiten von Importgeometrie (Fremdformate)
- Definieren von Montagebedingungen
- Anwenden einfacher Baugruppenanalysefunktionen (z. B. Durchdringung und Masseeigenschaften).

Ableiten norm-, funktions- und fertigungsgerechter Einzelteil- und Zusammenbauzeichnungen aus den 3D-CAD-Modellen, welche den Regeln der Technischen Darstellungslehre folgen, hierbei Aufzeigen von Querverweisen zu den in der Lehrveranstaltung Technische Darstellungslehre I erworbenen Kompetenzen.

Befähigung zum Erstellen auch komplexerer Einzelteile und Baugruppen in 3D-CAD-Systemen und zum Ableiten zugehöriger Technischer Zeichnungen sowie Befähigung, sich Modellierungsmöglichkeiten zu erschließen, die nicht explizit im Rahmen der Lehrveranstaltung behandelt wurden und Befähigung, die gewonnenen Erkenntnisse auf andere als im Rahmen der Lehrveranstaltung eingesetzte 3D-CAD-Systeme übertragen zu können.

---

### **Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

#### **[1] Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Nebenfach | Nebenfach Maschinenbau | Grundlagen der Produktentwicklung)

Dieses Modul ist daneben auch in den Studienfächern "Informatik (Master of Science)", "Medizintechnik (Bachelor of Science)" verwendbar.

---

### **Studien-/Prüfungsleistungen:**

Praktikum Technische Darstellungslehre 2. Teil (Prüfungsnummer: 45902)

(englische Bezeichnung: Laboratory: Engineering Drawing Part 2)

Studienleistung, Praktikumsleistung

Zugeordnete Lehrveranstaltungen:

- Technische Darstellungslehre II

weitere Erläuterungen:

Für den Erwerb des Scheins als Dokumentation der erbrachten Studienleistung müssen insgesamt 8 3D-CAD-Modelle erfolgreich testiert sein. 4 3D-CAD-Modelle hiervon sind im Rechnerraum unter Betreuung eigenständig zu erstellen. Weitere 4 3D-CAD-Modelle sind individuell eigenständig zu erstellen und verbindlich zu vorab definierten Terminen abzugeben. Zu den Übungen im Rechnerraum besteht Anwesenheitspflicht.

Prüfungssprache: Deutsch

Erstablesung: SS 2019, 1. Wdh.: WS 2019/2020

1. Prüfer: Sandro Wartzack (100617)

---

**Modulbezeichnung:** Technische Produktgestaltung (TPG) 5 ECTS  
 (Technical Product Design)

Modulverantwortliche/r: Sandro Wartzack

Lehrende: Sandro Wartzack, Benjamin Schleich

Startsemester: SS 2019

Dauer: 1 Semester

Turnus: jährlich (SS)

Präsenzzeit: 60 Std.

Eigenstudium: 90 Std.

Sprache: Deutsch

---

**Lehrveranstaltungen:**

Technische Produktgestaltung (SS 2019, Vorlesung mit Übung, 4 SWS, Sandro Wartzack et al.)

---

**Inhalt:**

- Einführung in die Technische Produktgestaltung
- Baustrukturen technischer Produkte
- Fertigungsgerechte Werkstückgestaltung
- toleranzgerechtes Konstruieren
- kostengerechtes Konstruieren
- beanspruchungsgerechtes Konstruieren
- werkstoffgerechtes Konstruieren
- Leichtbau
- umweltgerechtes Konstruieren
- nutzerzentrierte Produktgestaltung

**Lernziele und Kompetenzen:**

*Fachkompetenz*

*Wissen*

Im Rahmen von TPG erwerben die Studierenden Kenntnisse zur Berücksichtigung verschiedener Aspekte des Design-for-X bei der Entwicklung technischer Produkte. Nach der erfolgreichen Teilnahme kennen sie die jeweiligen Gestaltungsrichtlinien und zugehörige Methoden. Dies sind im Einzelnen:

- Wissen über Möglichkeiten zur Umsetzung des Leichtbaus und daraus abgeleitet über spezifische Gestaltungsrichtlinien, die im Rahmen des Leichtbaus zu berücksichtigen sind, hierzu: Beanspruchungsgerechtes Konstruieren (Kraftfluss, Prinzip der konstanten Gestaltfestigkeit, Kerbwirkung, Prinzip der abgestimmten Verformung, Prinzip des Kräfteausgleichs)
- Wissen über werkstoffgerechtes Konstruieren (Anforderungs- und Eigenschaftsprofil, wirtschaftliche Werkstoffauswahl, Auswirkung der Werkstoffwahl auf Fertigung, Lebensdauer und Gewicht)
- Wissen über die Auswirkungen eines Produktes (und insbesondere der vorhergehenden Konstruktion) auf Umwelt, Kosten und den Nutzer, hierzu: Umweltgerechtes Konstruieren (Recycling, Einflussmöglichkeiten in der Produktentwicklung, Strategien zur Berücksichtigung von Umweltaspekten, Life Cycle Assessment, Produktinstandsetzung, Design for Recycling)
- Wissen über kostengerechtes Konstruieren (Beeinflussung der Lebenslauf-, Herstell- und Selbstkosten in der Produktentwicklung, Auswirkungen der Stückzahl und der Fertigungsverfahren, Entwicklungsbegleitende Kalkulation)
- Wissen über nutzerzentrierte Produktentwicklung (Anthropometrie, Nutzerintegration in der Produktentwicklung, Mensch-Maschine-Schnittstellen, Beeinträchtigungen im Alter, Universal Design, Gestaltungsrichtlinien nach dem SENSI-Regelkatalog, etc.)
- Wissen über spezifische Gestaltungsrichtlinien der Fertigungsverfahren des „Urformens“ (Gießen, Pulvermetallurgie, Additive Fertigung)
- Wissen über spezifische Gestaltungsrichtlinien der Fertigungsverfahren des „Umformens“ (Schmieden, Walzen, Biegen, Scheiden, Tiefziehen, Stanzen, Fließpressen)
- Wissen über spezifische Gestaltungsrichtlinien der Fertigungsverfahren des „Trennens“ (Zerteilen, Drehen, Fräsen, Bohren, Schleifen, Erodieren)
- Wissen über spezifische Gestaltungsrichtlinien der Fertigungsverfahren des „Fügens“ (Schweißen, Lötten, Nieten, Durchsetzfügen, Kleben, Fügen durch Urformen)

- Wissen über spezifische Gestaltungsrichtlinien der Fertigungsverfahren des „Beschichtens und Stoffeigenschaften ändern“ (Schmelztauchen, Lackieren, Thermisches Spritzen, Physical Vapour Deposition, Chemical Vapour Deposition, Galvanische Verfahren, Pulverbeschichten, Vergüten, Glühen)
- Wissen über spezifische Gestaltungsrichtlinien des montagegerechten Konstruierens bzgl. der Baustruktur technischer Produkte (Integral-, Differential und Verbundbauweise, Produktstrukturierung, Variantenmanagement, Modularisierung) und des Montageprozesses (Gestaltung der Fügebauteile und Fügestellen, Automatisches Handhaben und Speichern, Toleranzausgleich, DFMA)
- Wissen über spezifische Inhalte des toleranzgerechten Konstruierens (insbesondere Grundlage der geometrischen Tolerierung und die Vorgehensweise zur Vergabe von Toleranzen)

#### *Verstehen*

Nach der erfolgreichen Teilnahme an der Lehrveranstaltung „Technische Produktgestaltung“ verfügen die Studierenden über Verständnisse hinsichtlich der technischen und nicht-technischen Einflussfaktoren und deren Abhängigkeiten bei der Gestaltung technischer Produkte ausgehend von der Produktstruktur bis zur konstruktiven Bauteilgestaltung. Hierbei stehen besonders die folgenden Verständnisse im Fokus:

- Verständnis über die Spezifikation von Toleranzen, Passungen und Oberflächen in Technischen Zeichnungen unter Berücksichtigung deren Auswirkungen auf Fertigung, Montage und den Betrieb des Produktes, hierzu: Verständnis der Vorgehensweise zur Toleranzspezifikation sowie erforderlicher Grundlagen zur Tolerierung von Bauteilen (Allgemeintoleranzen, wirkliche und abgeleitete Geometrielemente, Hüllbedingung, Unabhängigkeitsprinzip, Inklusion verschiedener Toleranzarten, Bezugssysteme und Ausrichtungskonzepte, statistische Toleranzanalyse, etc.)
- Verständnis über Fertigung und Montage sowie über die Bedeutung des Design-for-X und insbesondere des fertigungsgerechten Konstruierens im Produktentwicklungsprozess
- Verständnis über die Berücksichtigung nicht-technischer Faktoren, wie beispielsweise Umwelt-, Kosten- und Nutzeraspekten, und deren Wechselwirkungen bei der Gestaltung technischer Produkte.

#### *Anwenden*

Die Studierenden wenden im Rahmen von Übungsaufgaben Gelerntes an. Dabei werden bestehende Entwürfe und Konstruktionen durch die Studierenden entsprechend der vermittelten Gestaltungsrichtlinien optimiert und neue Konstruktionen unter Einhaltung dieser Gestaltungsrichtlinien erschaffen. Dies beinhaltet im Einzelnen:

- Erstellung der fertigungsgerechten und montagegerechten Tolerierung von Bauteilen. Dies umschließt folgende Tätigkeiten: Bestimmen der zugrundeliegenden Bezugssysteme und Ausrichtungskonzepte; Bestimmen des Tolerierungsgrundsatzes. Integration von, durch Normen definierte Toleranz- und Passungsvorgaben in bestehende Tolerierungen; Zusammenfassen kombinierbarer Form- und Lagetoleranzen zu Zeichnungsvereinfachung; Festlegung der Größen der Toleranzzonen aller vergebenen Toleranzen.
- Optimierung der Tolerierung anhand der statistischen Toleranzanalyse. Dies umschließt folgende Tätigkeiten: Erkennen und Ableiten der analytischen Schließmaßgleichungen; Definition der zugrundeliegenden Toleranzwerten und zugehörigen Wahrscheinlichkeitsverteilungen; Berechnung der resultierenden Wahrscheinlichkeitsverteilungen der Schließmaße; analytische Bestimmung der statistischen Beitragsleistung mittels lokaler Sensitivitätsanalysen; Beurteilung der Ergebnisse und ggf. anschließende Anpassung der Tolerierung der Bauteile; Transfer der Ergebnisse auf zeitabhängige Mechanismen (kinematische Systeme).
- Änderung der Gestaltung von Bauteilen, bedingt durch die Änderung der zu fertigenden Stückzahl der Baugruppe. Dies umschließt die folgenden Tätigkeiten: Bestimmung des konstruktiven Handlungsbedarfs; Anpassung der Gestaltung der Bauteile - insbesondere hinsichtlich der fertigungsgerechten und der montagegerechten Gestaltung. Gestaltung der erforderlichen Werkzeuge zur Fertigung der Bauteile und Bewertung dieser bzgl. der resultierenden Kosten.

#### *Analysieren*

- Aufzeigen von Querverweisen zu den in der Lehrveranstaltung Produktionstechnik zu erwerbenden Kompetenzen über die Hauptgruppen der Fertigungsverfahren nach DIN 8580

- Aufzeigen von Querverweisen zu den in der Lehrveranstaltung Handhabungs- und Montagetechnik zu erwerbenden Kompetenzen über montagegerechtes Konstruieren
- Aufzeigen von Querverweisen zu den in der Lehrveranstaltung Umformtechnik zu erwerbenden Kompetenzen über Fertigungsverfahren der Hauptgruppe Umformen nach DIN 8580
- Aufzeigen von Querverweisen zu den in der Lehrveranstaltung Konstruktionsübung zu erwerbenden Kompetenzen über das Konstruieren von Maschinen und deren konstruktive Auslegung.

#### *Evaluieren (Beurteilen)*

Anhand der erlernten Grundlagen über unterschiedliche Aspekte des Design-for-X, deren Berücksichtigung bei der Gestaltung technischer Produkte durch Gestaltungsrichtlinien, Methoden, und Vorgehensweisen sowie den dargelegten Möglichkeiten zur Rechnerunterstützung können die Studierenden kontextbezogene Richtlinien für die Gestaltung technischer Produkte in unbekanntem Konstruktionsaufgaben auswählen und deren Anwendbarkeit einschätzen. Zudem sind sie in der Lage konträre Gestaltungsrichtlinien aufgabenspezifisch abzuwägen.

#### *Erschaffen*

Die Studierenden werden durch die erlernten Grundlagen befähigt, konkrete Verbesserungsvorschläge zu bestehenden Konstruktionen hinsichtlich unterschiedlicher Design-for-X Aspekte eigenständig zu erarbeiten. Zudem sind sie in der Lage technische Produkte so zu gestalten, dass diese verschiedenste technische und nicht-technische Anforderungen (fertigungsbezogene Anforderungen, Kostenanforderungen, Umweltaforderungen, Nutzeranforderungen, etc.) bedienen. Darüber hinaus werden die Studierenden in die Lage versetzt, Gestaltungsrichtlinien für neuartige Fertigungsverfahren aus grundlegenden Verfahrenseigenschaften abzuleiten und bei der Gestaltung technischer Produkte anzuwenden.

#### *Lern- bzw. Methodenkompetenz*

Befähigung zur selbständigen Gestaltung von Produkten und Prozessen gemäß erlernter Vorgehensweisen und Richtlinien sowie unter verschiedensten Design-for-X-Aspekten sowie zur objektiven Bewertung bestehender Produkte und Prozesse hinsichtlich gestellter Anforderungen des Design-for-X.

#### *Selbstkompetenz*

Befähigung zur selbständigen Arbeitseinteilung und Einhaltung von Meilensteinen. Objektive Beurteilung sowie Reflexion der eigenen Stärken und Schwächen sowohl in fachlicher (u. a. Umsetzung der gelehrten Richtlinien des Design-for-X in der Konstruktion) als auch in sozialer Hinsicht (u. a. Erarbeitung von Lösungen und Kompromissen im interdisziplinären Team).

#### *Sozialkompetenz*

Die Studierenden organisieren selbstständig die Bearbeitung von Übungsaufgaben in kleinen Gruppen und erarbeiten gemeinsam Lösungsvorschläge für die gestellten Übungsaufgaben. In der gemeinsamen Diskussion erarbeiteter Lösungen geben Betreuer und Kommilitonen wertschätzendes Feedback.

### **Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

#### **[1] Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Nebenfach | Nebenfach Maschinenbau | Grundlagen der Produktentwicklung)

Dieses Modul ist daneben auch in den Studienfächern "Berufspädagogik Technik (Bachelor of Science)", "Berufspädagogik Technik (Master of Education)", "Informatik (Master of Science)", "Maschinenbau (Bachelor of Science)", "Maschinenbau (Master of Science)", "Mechatronik (Bachelor of Science)", "Mechatronik (Master of Science)", "Medizintechnik (Bachelor of Science)", "Medizintechnik (Master of Science)", "Wirtschaftsingenieurwesen (Bachelor of Science)", "Wirtschaftsingenieurwesen (Master of Science)" verwendbar.

### **Studien-/Prüfungsleistungen:**

Technische Produktgestaltung (Prüfungsnummer: 71101)

(englische Bezeichnung: Technical Product Design)

Prüfungsleistung, Klausur, Dauer (in Minuten): 120

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:



- Technische Produktgestaltung  
weitere Erläuterungen:  
Die Note ergibt sich zu 100% aus der Klausur.

Erstablingung: SS 2019, 1. Wdh.: WS 2019/2020  
1. Prüfer: Sandro Wartzack (100617)

---

---

**Modulbezeichnung:** Grundlagen der Produktentwicklung (GPE) 7.5 ECTS  
(Basic Principles of Product Development)

Modulverantwortliche/r: Stephan Tremmel  
Lehrende: Stephan Tremmel

---

Startsemester: WS 2019/2020	Dauer: 1 Semester	Turnus: jährlich (WS)
Präsenzzeit: 90 Std.	Eigenstudium: 135 Std.	Sprache: Deutsch

---

**Lehrveranstaltungen:**

Grundlagen der Produktentwicklung (WS 2019/2020, Vorlesung, 4 SWS, Stephan Tremmel)  
Übung zu Grundlagen der Produktentwicklung (WS 2019/2020, Übung, 2 SWS, Michael Jüttner)  
Technische Darstellungslehre für GPE (WS 2019/2020, Vorlesung, Benjamin Schleich)

---

**Inhalt:**

**Einführung in die Produktentwicklung**

- Synthese und Analyse als zentrale Aufgaben
- Vorgehensmodelle im Produktentwicklungsprozess

**Konstruktionswerkstoffe Grundlagen der Bauteilauslegung - Festigkeitslehre**

- Typische Versagenskriterien
- Definition und Aufgaben der Festigkeitslehre, Prinzip
- Ermittlung von Belastungen
- Ermittlung von Beanspruchungen
- Beanspruchungsarten
- Zeitlicher Verlauf der Beanspruchung und Lastannahmen
- Resultierende Spannungen und Vergleichsspannungen
- Kerbwirkung und Stützwirkung
- Weitere Einflussfaktoren auf die Festigkeit von Bauteilen
- Maßgebliche Werkstoffkennwerte
- Bauteildimensionierung und Festigkeitsnachweis

**Einführung in die Technische Produktgestaltung**

- Gestalten von Maschinen
- Fertigungsgerechtes Gestalten
- Sicherheitsgerechtes Gestalten

**Normung, Toleranzen, Passungen und Oberflächen Maschinenelemente**

- Schweißverbindungen
- Passfeder- und Keilwellenverbindungen
- Bolzen- und Stiftverbindungen
- Zylindrische Pressverbindungen
- Kegelverbindungen
- Spannelementverbindungen
- Schraubenverbindungen
- Wälzlager
- Gleitlager
- Dichtungen
- Stirnräder und Stirnradgetriebe
- Kupplungen

**Lernziele und Kompetenzen:**

*Fachkompetenz*

*Wissen*

Verständnis für das Konstruieren von Maschinen als methodischer Prozess unter besonderer Beachtung von Synthese und Analyse als zentrale Aufgaben der Produktentwicklung und auf Basis der Begriffe Merkmale und Eigenschaften nach der Definition von Weber Anwendung von Vorgehensmodellen in Produktentwicklungsprozessen mit Fokus auf VDI 2221 ff.; hierbei Aufzeigen von Querverweisen zu den in der Lehrveranstaltung Methodisches und rechnerunterstütztes Kon-

struieren zu erwerbenden Kompetenzen.

Verständnis für Konstruktionswerkstoffe, deren spezifische Eigenschaften sowie Möglichkeiten zur Beschreibung des Festigkeits-, Verformungs- und Bruchverhaltens. Unter Konstruktionswerkstoffen werden insbesondere Eisenwerkstoffe, daneben auch Nichteisenmetalle, Polymerwerkstoffe und spezielle neue Werkstoffe, z. B. Verbundwerkstoffe, verstanden. Erschließen von Querverbindungen zu den in der Lehrveranstaltung Werkstoffkunde erworbenen Kompetenzen.

Verständnis für das Gestalten von Maschinenbauteilen unter besonderer Berücksichtigung der Fertigungsgerechtigkeit, hierbei Aufzeigen von Querverweisen zu den in der Lehrveranstaltung Produktionstechnik erworbenen Kompetenzen und zu den in der Lehrveranstaltung Technische Produktgestaltung zu erwerbenden Kompetenzen.

Verständnis für Normen (DIN, EN, ISO), Richtlinien (VDI, FKM) und Standards im Kontext des Maschinenbaus, hierbei Aufzeigen von Querverweisen zu den in der Lehrveranstaltung Technische Darstellungslehre erworbenen Kompetenzen.

Verständnis für herstell- und messbedingte Abweichungen sowie zu vergebende Toleranzen für Maß, Form, Lage und Oberfläche bei Maschinenbauteilen sowie Berechnung von Maßtoleranzen, hierbei Aufzeigen von Querverweisen zu den in der Lehrveranstaltung Messtechnik erworbenen Kompetenzen.

Funktionsorientiertes Verständnis für und Überblick zu gängigen Maschinenelementen sowie Vertiefung einzelner Maschinenelemente unter Berücksichtigung derer spezifischen Merkmale, Eigenschaften und Einsatzbedingungen. Im Einzelnen:

- Gestaltung und Berechnung von Schweißverbindungen sowie Beurteilung der Tragfähigkeit von Schweißverbindungen nach dem Verfahren von NIEMANN
- Gestaltung und Berechnung formschlüssiger Welle-Nabe-Verbindungen, insbesondere Passfederverbindungen auf Basis von DIN 6892 und Keilwellenverbindungen sowie Beurteilung der zugrunde gelegten Berechnungsmodelle im Hinblick auf deren Gültigkeitsgrenzen
- Gestaltung und Berechnung einfacher Bolzen- und Stiftverbindungen sowie Beurteilung der zugrunde gelegten Berechnungsmodelle im Hinblick auf deren Gültigkeitsgrenzen
- Verständnis für reibschlüssige Welle-Nabe-Verbindungen (Wirkprinzip) und Gestaltung, Berechnung und Herstellung von zylindrischen Quer- und Längspressverbänden in Anlehnung an DIN 7190 (elastische Auslegung) sowie von Kegelpressverbänden
- Verständnis für die Elemente von Schraubenverbindungen unter besonderer Berücksichtigung des Maschinenelements Schraube (Gewinde) sowie Überprüfung längs- und querbelasteter, vorgespannter Schraubenverbindungen in Anlehnung an VDI 2230 im Hinblick auf Anziehdrehmoment, Bruch, Fließen und Dauerbruch der Schraube unter Einfluss von Setzvorgängen und Schwankungen beim Anziehen
- Verständnis für rotatorische Wälzlager und Wälzlagerungen, insbesondere Wissen über die gängigen Radial- und Axialwälzlagerbauformen, deren spezifische Merkmale und Eigenschaften sowie deren sachgerechte Einbindung in die Umgebungskonstruktion; Berechnung der Tragfähigkeit von Wälzlagern für statische und dynamische Betriebszustände auf Basis von DIN ISO 76 und DIN ISO 281 (nominelle und erweiterte modifizierte Lebensdauer); Verständnis für die konstruktive Gestaltung von Wälzlagerstellen, insbesondere Passungswahl und Lageranordnungen. Dadurch Befähigung zur Auswahl geeigneter Wälzlager, zur Grobgestaltung von Wälzlagerstellen und zur Einschätzung der konstruktiven Ausführung von Wälzlagerungen; hierbei Aufzeigen von Querverweisen zu den in der Lehrveranstaltung Wälzlagertechnik zu erwerbenden Kompetenzen
- Verständnis für Dichtungen, Klassifizieren statischer und dynamischer Dichtungen und Auswahl von Dichtungen unter Berücksichtigung gegebener technischer Randbedingungen
- Basiswissen über Antriebssysteme, Antriebsstränge und Antriebskomponenten, Verständnis für Last- und Beschleunigungsdrehmomente und zu reduzierende Trägheitsmomente; hierbei Aufzeigen von Querverweisen zu den in den Lehrveranstaltungen Regelungstechnik und Elektrische Antriebstechnik zu erwerbenden Kompetenzen
- Verständnis für Getriebe als wichtige mechanische Komponente in Antriebssträngen, Berechnung von Übersetzungen
- Verständnis für Zahnradgetriebe mit Fokus auf Stirnräder und Stirnradgetriebe, hierbei Ver-

ständnis des Verzahnungsgesetzes und der Geometrie der Evolventenverzahnung für Gerad- und Schrägverzahnung; Analyse der am Zahnrad wirkenden Kräfte und Ermittlung der Zahnfuß- und der Grübchentragsfähigkeit in Anlehnung an DIN 3990

- Verständnis für nicht-schaltbare und schaltbare Kupplungen; Klassifizieren von Kupplungen nach deren Funktions- und Wirkprinzipien; Auswahl von Kupplungen unter Berücksichtigung gegebener technischer Randbedingungen.

#### *Evaluieren (Beurteilen)*

Bewerten und Einschätzen von Maschinenbauteilen im Hinblick auf deren rechnerische Auslegung und konstruktive Gestaltung unter Berücksichtigung des Werkstoffverhaltens, der Geometrie und der auf das Bauteil einwirkenden Lasten. Hierzu:

- Analyse der auf ein Bauteil wirkenden Belastungen und Erschließen von Querverbindungen zu den in der Lehrveranstaltung Statik erworbenen Kompetenzen
- Analyse der aus den Belastungen resultierenden Beanspruchungen und Erschließen von Querverbindungen zu den in der Lehrveranstaltung Elastostatik erworbenen Kompetenzen. Hierbei Fokus auf die Beanspruchung stabförmiger Bauteile, Kontaktbeanspruchung sowie Instabilität stabförmiger Bauteile (Knicken)
- Unterscheidung von Nennspannungen und örtlichen Spannungen
- Analyse und Beurteilung von Lastannahmen sowie des zeitlichen Verlaufs von Beanspruchungen (statisch, dynamisch)
- Verständnis für mehrachsige Beanspruchungszustände und Festigkeitshypothesen in Verbindung mit den werkstoffspezifischen Versagenskriterien, Ermittlung von Vergleichsspannungen
- Verständnis für die Auswirkungen von Kerben auf Maschinenbauteile unter statischer und dynamischer Beanspruchung und Ermittlung von Kerbspannungen auf Basis von Kerbform-, Kerbwirkungszahlen und plastischen Stützzahlen unter Berücksichtigung von Oberflächeneinflüssen
- Verständnis für Werkstoffkennwerte und den Einfluss der Bauteilgröße und des Oberflächenzustandes sowie Gegenüberstellung zu dazugehörigen Versagenskriterien
- Überprüfung der Festigkeit von Maschinenbauteilen im Zuge von Dimensionierungsaufgaben und Tragfähigkeitsnachweisen in Anlehnung an die einschlägige FKM-Richtlinie sowie Beurteilung der durchgeführten Berechnungen unter besonderer Berücksichtigung von Unsicherheiten, welche Ausdruck in der Wahl von Mindestsicherheiten finden.

Auswahl und Beurteilung gängiger Maschinenelemente unter Funktionsgesichtspunkten sowie Auslegen ausgewählter Maschinenelemente.

Befähigung zur Einschätzung und Bewertung von Maschinenelementen, einschließlich der Befähigung, Berechnungsansätze und Gestaltungsgrundsätze auch auf andere Maschinenelemente, die nicht explizit im Rahmen der Lehrveranstaltung behandelt wurden, zu übertragen.

#### **Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

##### **[1] Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Nebenfach | Nebenfach Maschinenbau | Grundlagen der Produktentwicklung)

Dieses Modul ist daneben auch in den Studienfächern "Informatik (Master of Science)", "Materialwissenschaft und Werkstofftechnik (Bachelor of Science)", "Medizintechnik (Bachelor of Science)", "Medizintechnik (Master of Science)" verwendbar.

#### **Studien-/Prüfungsleistungen:**

Grundlagen der Produktentwicklung (Prüfungsnummer: 47111)

(englische Bezeichnung: Foundations of Product Development)

Prüfungsleistung, Klausur, Dauer (in Minuten): 120

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Grundlagen der Produktentwicklung
- Übung zu Grundlagen der Produktentwicklung

Erstablesung: WS 2019/2020, 1. Wdh.: SS 2020, 2. Wdh.: WS 2020/2021

1. Prüfer: Stephan Tremmel (100618)

---

**Organisatorisches:**

Es werden empfohlen:

- Technische Darstellungslehre I
- Statik und Festigkeitslehre

---

**Modulbezeichnung: Automotive Engineering (AutoEng)** **2.5 ECTS**  
(Automotive Engineering)

Modulverantwortliche/r: Ingo Kriebitzsch  
Lehrende: Ingo Kriebitzsch

---

Startsemester: WS 2019/2020	Dauer: 1 Semester	Turnus: jährlich (WS)
Präsenzzeit: 30 Std.	Eigenstudium: 45 Std.	Sprache: Deutsch

---

**Lehrveranstaltungen:**

Automotive Engineering (WS 2019/2020, Vorlesung, 2 SWS, Jörg Franke et al.)

---

**Inhalt:**

Die Vorlesung ist an alle ingenieurwissenschaftliche Studiengänge und Studenten mit Interesse an einer Tätigkeit in der Automobilindustrie oder deren Umfeld gerichtet. Es werden die Themen der Produktentstehung bis zur Fertigung und Vertrieb beleuchtet. Dabei wird der Aspekt des interdisziplinären Agierens aus unterschiedlichen Blickwinkeln dargestellt. Zum einen werden Einblicke in die technische, konstruktive Umsetzung von wesentlichen Elementen eines Automobils gestreift, zum anderen sollen aber auch strategische und betriebswirtschaftlich bestimmende Größen vermittelt und deren Bedeutung für den Ingenieur vertieft werden. Ziel ist es ein Gesamtverständnis für den Komplex der Automobilindustrie zu vermitteln.

Folgende thematischen Schwerpunkte werden in der Vorlesung behandelt:

- Überblick über die Abläufe und Rahmenbedingungen für die Entwicklung in der Automobilindustrie.
- Die Produktentstehung
- Der Produktionsprozess in der Automobilindustrie
- Integrierte Absicherung
- Handlungorganisation: Markteinführung, Marketingkonzepte, Service und Aftermarket Strategien
- Elektrifizierung, Hybrid, alternative Antriebe
- Elektronik im Fahrzeug: Fahrerassistenz, Navigation, Kommunikation
- Neue Technologien für die Herstellung von Karosserien
- Passive und aktive Sicherheit. Trend und Markttendenzen, technische Lösungen
- Entwicklung der Fahrdynamik
- IT-Systeme in der Automobilindustrie
- Spitzenleistungen als faszinierende Herausforderungen (Designstudien, Experimentalfahrzeuge, Rennsport)
- Qualitätsmanagement

**Lernziele und Kompetenzen:**

Das Automobil ist zunehmend eines der komplexesten Industriegüter. Es ist geprägt durch gesellschaftliche Anforderungen, gesetzliche Restriktionen und unterschiedlichste Markt- und Kundenwünschen weltweit. Lernen Sie die Herausforderungen für die Ingenieurwissenschaften in der Automobilindustrie kennen, die Zusammenhänge verstehen und die Lösungen zu erarbeiten.

Nach besuch der Vorlesung sind die Studenten in der Lage:

- Einen Überblick über die Produktentstehung bis hin zur Serienentwicklung zu geben
- Die Produktionsprozesse im Automobilbau zu verstehen
- Supportprozesse wie die integrierte Absicherung zu verstehen
- Die Vor- und Nachteile der unterschiedlichen Antriebstechnologien zu nennen
- Einen Überblick von Elektrik und Elektronik im Fahrzeug zu haben
- Einflüsse auf die Fahrzeugdynamik zu verstehen

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

**[1] Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Nebenfach | Nebenfach Maschinenbau | Computerintegrierte Produktion)

Dieses Modul ist daneben auch in den Studienfächern "Informatik (Master of Science)", "Maschinenbau (Bachelor of Science)", "Maschinenbau (Master of Science)", "Mechatronik (Bachelor of Science)", "Mechatronik

(Master of Science)", "Wirtschaftsingenieurwesen (Bachelor of Science)", "Wirtschaftsingenieurwesen (Master of Science)" verwendbar.

---

**Studien-/Prüfungsleistungen:**

Automotive Engineering (Prüfungsnummer: 53401)

(englische Bezeichnung: Automotive Engineering)

Prüfungsleistung, Klausur, Dauer (in Minuten): 60

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Automotive Engineering

Erstablingung: WS 2019/2020, 1. Wdh.: SS 2020

1. Prüfer: Jörg Franke (100615)

---

**Organisatorisches:**

Prüfung: schriftlich, 60 min

**Modulbezeichnung:** Die Werkzeugmaschine als mechatronisches System (WZM 2.5 ECTS MS)

(Machine Tools as a Mechatronic System)

Modulverantwortliche/r: Siegfried Russwurm

Lehrende: Siegfried Russwurm

Startsemester: WS 2019/2020

Dauer: 1 Semester

Turnus: jährlich (WS)

Präsenzzeit: 35 Std.

Eigenstudium: 40 Std.

Sprache: Deutsch

#### Lehrveranstaltungen:

Die Werkzeugmaschine als mechatronisches System (WS 2019/2020, Vorlesung, 2 SWS, Siegfried Russwurm)

#### Inhalt:

- Bedeutung der Mechatronik im Werkzeugmaschinenbau
- Grundlegende Begrifflichkeiten mit Bezug auf den Werkzeugmaschinenbau zu den Themen Mechanik, Elektrotechnik und Software
- Analyse, Modellierung und Regelung von Werkzeugmaschinen
- CNC-Steuerungstechnik für die Werkzeugmaschine
- Parallelkinematik-Maschinen
- Evolution der Drehmaschinen
- Vertikale und horizontale IT-Integration

#### Lernziele und Kompetenzen:

Nach Besuch der Lehrveranstaltung sollen die Studierenden in der Lage sein folgende Themen zu bearbeiten:

- Wesentliche mechatronische Komponenten der Werkzeugmaschine
- Modellversuche zur elektrischen Antriebstechnik
- Analytische Vorgehensweise zur regelungstechnischen Modellbildung
- Regelungstechnische Möglichkeiten der elektrischen Antriebstechnik
- CNC - Verfahrenskette: vom CAD-Geometriemodell zur Werkzeugposition
- Konsequenzen alternativer Maschinenkonzepte: Parallelkinematiken, modulare Maschinen
- Werkzeugmaschinen als IT-Komponenten: horizontale und vertikale Integration und Kommunikation
- Mechatronische Systeme im allg. Maschinenbau: Übertragung der Konzepte d. Werkzeugmaschine auf andere Maschinenbau-Applikationen

#### Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

##### [1] Informatik (Bachelor of Science)

(Po-Vers. 2009s | Nebenfach | Nebenfach Maschinenbau | Computerintegrierte Produktion)

Dieses Modul ist daneben auch in den Studienfächern "Informatik (Master of Science)", "Maschinenbau (Bachelor of Science)", "Maschinenbau (Master of Science)", "Mechatronik (Bachelor of Science)", "Mechatronik (Master of Science)", "Medizintechnik (Bachelor of Science)", "Wirtschaftsingenieurwesen (Bachelor of Science)", "Wirtschaftsingenieurwesen (Master of Science)" verwendbar.

#### Studien-/Prüfungsleistungen:

Die Werkzeugmaschine als mechatronisches System (Prüfungsnummer: 52701)

(englische Bezeichnung: Machine Tools as a Mechatronic System)

Prüfungsleistung, Klausur, Dauer (in Minuten): 60

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Die Werkzeugmaschine als mechatronisches System

Erstablegung: WS 2019/2020, 1. Wdh.: SS 2020

1. Prüfer: Siegfried Russwurm (100613)



**Organisatorisches:**

Ansprechpartner am Lehrstuhl FAPS: Eva Fischer, M. Sc.

---

**Modulbezeichnung:** Industrie 4.0 - Anwendungsszenarien in Produktion und Service (ASPS4.0) 2.5 ECTS

(Industry 4.0 - Application scenarios in production and service)

Modulverantwortliche/r: Jörg Franke

Lehrende: Ulrich Löwen

---

Startsemester: WS 2019/2020

Dauer: 1 Semester

Turnus: jährlich (WS)

Präsenzzeit: 35 Std.

Eigenstudium: 40 Std.

Sprache: Deutsch

---

**Lehrveranstaltungen:**

Industrie 4.0 - Anwendungsszenarien in Produktion und Service (WS 2019/2020, Vorlesung, 2 SWS, Ulrich Löwen)

---

**Inhalt:**

Die IT-Durchdringung in der produzierenden Industrie nimmt rasant zu. Der nutzenstiftende Einsatz von IT bei der Gestaltung von Wertschöpfungsprozessen hat für Deutschland eine zentrale strategische Bedeutung. Diese Trends werden unter Begriffen wie „Industrie 4.0“ und „Industrial Internet“ bzw. „Internet of Things“ weltweit diskutiert. Dabei treffen doch recht unterschiedliche Sichtweisen aufeinander. In der Vorlesung werden diese Trends und Visionen anhand von ausgewählten Anwendungsszenarien erläutert. Außerdem werden die dafür zum Verständnis notwendigen Grundlagen erklärt.

Ziele:

- Bewusstseinschärfung bezüglich der Auswirkungen der Digitalisierung auf die produzierende Industrie
- Verständnis von Geschäftstreibern, technischen Möglichkeiten und deren Wechselwirkungen in der produzierenden Industrie
- Vermittlung Branchen- und Domänen-übergreifender Prozesse und Methoden in der produzierenden Industrie

**Lernziele und Kompetenzen:**

Den Studierenden sollen die Auswirkungen der Digitalisierung auf die produzierende Industrie verdeutlicht und dadurch ein Bewusstsein für diese Entwicklungen geschaffen werden. Zusätzlich soll ein Verständnis für Geschäftstreiber, technische Möglichkeiten und deren Wechselwirkungen in der produzierenden Industrie sowie branchen- und domänenübergreifender Prozesse und Methoden vermittelt werden.

Die Vorlesung ist auf Basis der folgenden Leitlinien aufgebaut:

- Methodische und konsequente Trennung der Diskussion von Problemperspektive, konzeptioneller Lösungsperspektive und technischer Umsetzungsperspektive
- Umfassendes Gesamtverständnis bezüglich der oft sehr vielschichtigen wirtschaftlichen und technischen Zusammenhänge (zu Lasten eines tiefen technischen Detaildiskussion)
- Betonung des für einen Anwender gestifteten (geschäftlichen) Nutzens und der möglichen Alleinstellungsmerkmale für einen Standort Deutschland

Die Studierenden sind nach Besuch der Lehrveranstaltung in der Lage:

- die kontroversen und vielschichtigen Diskussionen im Umfeld der Digitalisierung in der Produzierenden Industrie in einen konsistenten Gesamtkontext einzuordnen
- anhand repräsentativer Beispiele den Unterschied zu verstehen zwischen dem aktuellen Stand der Technik und Forschung sowie den durch Industrie 4.0 postulierten Innovationshypothesen
- aufgrund der vermittelten Beispiele und Methoden durch eine Hinterfragung von Zielen und des wirtschaftlichen Nutzens die oft stark emotional geführten Diskussionen im Kontext von Industrie 4.0 zu versachlichen

Das im Rahmen dieser Lehrveranstaltung vermittelte Wissen ist in allen Bereichen der industriellen Branchen, so z. B. im Automobilbau, der Informatik und Wirtschaftsinformatik, der Elektrotechnik und Medizintechnik und dem Maschinen- und Anlagenbau erforderlich.

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

## [1] Informatik (Bachelor of Science)

(Po-Vers. 2009s | Nebenfach | Nebenfach Maschinenbau | Computerintegrierte Produktion)

Dieses Modul ist daneben auch in den Studienfächern "Elektrotechnik, Elektronik und Informationstechnik (Bachelor of Science)", "Elektrotechnik, Elektronik und Informationstechnik (Master of Science)", "Informatik (Master of Science)", "Maschinenbau (Bachelor of Science)", "Maschinenbau (Master of Science)", "Mechatronik (Bachelor of Science)", "Mechatronik (Master of Science)", "Wirtschaftsingenieurwesen (Bachelor of Science)", "Wirtschaftsingenieurwesen (Master of Science)" verwendbar.

---

### Studien-/Prüfungsleistungen:

Industrie 4.0 - Anwendungsszenarien und Grundlagen (Prüfungsnummer: 49451)

Prüfungsleistung, Klausur, Dauer (in Minuten): 60

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Industrie 4.0 - Anwendungsszenarien in Produktion und Service

Prüfungssprache: Deutsch

Erstablingung: WS 2019/2020, 1. Wdh.: SS 2020

1. Prüfer: Jörg Franke (100615)

---

### Organisatorisches:

Ansprechpartner am Lehrstuhl FAPS: M.Sc. Jonathan Fuchs

**Modulbezeichnung:** International Supply Chain Management (ISCM) 5 ECTS  
(International Supply Chain Management)

Modulverantwortliche/r: Jörg Franke  
Lehrende: Jörg Franke

Startsemester: WS 2019/2020      Dauer: 1 Semester      Turnus: halbjährlich (WS+SS)  
Präsenzzeit: 60 Std.              Eigenstudium: 90 Std.      Sprache: Englisch

**Lehrveranstaltungen:**

International Supply Chain Management (vhb) (WS 2019/2020, Vorlesung, 4 SWS, Jörg Franke et al.)

**Inhalt:**

Ziel der virtuellen Vorlesung ist ein Überblick über die Aufgaben eines Supply Chain Managers auf dem internationalen Parkett:

- Ziele und Aufgaben
- Methoden und Tools
- Internationales Umfeld
- Erfahrung und Wissen aus der industriellen Praxis
- Aktueller Stand der Wissenschaft im SCM-Umfeld

Der Kurs gliedert sich in folgende Lerneinheiten:

- Integrated logistics, procurement, materials management and production
- Material inventory and material requirements in the enterprise
- Analysis of cost reduction in materials management
- Management of procurement and purchasing
- Procurement strategies
- Warehouse management, picking systems, in-plant material handling, packaging
- Distribution logistics, global tracking and tracing
- Modes of transport in international logistics
- Disposal logistics
- Logistics controlling
- Global logistic structures and value chains
- IT systems in supply chain management
- Sustainable global structures of production and logistics
- Summary

Zur praktischen Vertiefung werden im Rahmen des Kurses 3 Case Studies durchgeführt.

**Lernziele und Kompetenzen:**

Der Studierende ist nach erfolgreichem Abschluss des Kurses in der Lage

- Grundbegriffe des Supply Chain Managements zu definieren,
- wichtige Methoden und Strategien der Beschaffung zu verstehen,
- verschiedene Lagertypen und -strategien zu nennen und einzuordnen,
- Möglichkeiten zur Kostenreduktion in Lieferketten zu analysieren,
- zentrale IT-Systeme des Supply Chain Managements zu kennen und voneinander abzugrenzen,
- Entsorgungs- und Controllingstrategien zu erläutern,
- wesentliche Probleme in internationalen Liefernetzwerken zu erkennen,
- Möglichkeiten der Transformation hin zur nachhaltigen Supply Chain zu erläutern,
- verschiedene Verkehrsträger zu bewerten.

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

[1] **Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Nebenfach | Nebenfach Maschinenbau | Computerintegrierte Produktion)

Dieses Modul ist daneben auch in den Studienfächern "Berufspädagogik Technik (Bachelor of Science)", "Berufspädagogik Technik (Master of Education)", "Informatik (Master of Science)", "International Production Engineering and Management (Bachelor of Science)", "Maschinenbau (Master of Science)", "Mechatronik (Bachelor of Science)", "Mechatronik (Master of Science)", "Wirtschaftsingenieurwesen (Master of Science)" verwendbar.

---

### **Studien-/Prüfungsleistungen:**

International Supply Chain Management (Prüfungsnummer: 49201)

(englische Bezeichnung: International Supply Chain Management)

Prüfungsleistung, Klausur, Dauer (in Minuten): 120

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- International Supply Chain Management (vhb)

weitere Erläuterungen:

Zur Teilnahme ist eine Anmeldung und Registrierung bei der vhb erforderlich!

Klausur kann einen Anteil an Fragen im Antwort-Wahl-Verfahren (Multiple-Choice) enthalten.

Erstablingung: WS 2019/2020, 1. Wdh.: keine Angabe

1. Prüfer: Jörg Franke (100615)

---

---

**Modulbezeichnung:** Einführung in die Programmierung humanoider Roboter (NAORob) 5 ECTS  
 (Basic principles of programming humanoid robots)

Modulverantwortliche/r: Jörg Franke  
 Lehrende: Jörg Franke, Assistenten

---

Startsemester: WS 2019/2020	Dauer: 1 Semester	Turnus: jährlich (WS)
Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Deutsch

---

**Lehrveranstaltungen:**

Einführung in die Programmierung humanoider Roboter (WS 2019/2020, Vorlesung mit Übung, 4 SWS, Sebastian Reitelshöfer et al.)

---

**Empfohlene Voraussetzungen:**

Programmiererfahrung in C++

---

**Inhalt:**

- Roboterkinematik (kinematischer Aufbau von Standard-Robotertypen, Koordinatentransformation)
- Bewegungssteuerung und -planung
- Grundlagen des zweibeinigen Laufens
- Rechnersehen mit OpenCV
- Selbstlokalisierung
- Programmierung verteilter Robotersysteme
- Einführung in das Framework Robot Operating System (ROS)
- Verwendung von ROS zur C++- Programmierung des humanoiden Roboters NAO
- Lösung einer Teamaufgabe im Rahmen der Veranstaltung

**Lernziele und Kompetenzen:**

Nach dem Besuch der Lehrveranstaltung sind die Studierenden in der Lage, eigenständig auch fortgeschrittene Aufgabenstellungen in der Robotik am Beispiel des humanoiden Roboters NAO beziehungsweise an anderen Roboterkinematiken umzusetzen.

Die Vorlesung vermittelt Kenntnisse aus folgenden Bereichen:

- Grundlagen der Robotik in Bezug auf humanoide Systeme
- Roboterkinematik (kinematischer Aufbau von Standard-Robotertypen, Koordinatentransformationen, direkte und inverse Transformation)
- Roboterprogrammierung und Softwareentwicklung
- Umgang mit dem Robot Operating System ROS
- Bewegungssteuerung und -planung
- Selbstlokalisierung in unbekanntem Umgebungen
- Bildverarbeitung (OpenCV)
- Auswertung multimodaler Sensoren

Die Studenten erwerben und trainieren im Rahmen des Praktikums zusätzlich folgende Fähigkeiten:

- Problemlösungsfähigkeit und analytisches Denken
- Projektmanagement und Teamarbeit
- Kommunikationsfähigkeit

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

**[1] Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Nebenfach | Nebenfach Maschinenbau | Computerintegrierte Produktion)

Dieses Modul ist daneben auch in den Studienfächern "Berufspädagogik Technik (Bachelor of Science)", "Berufspädagogik Technik (Master of Education)", "Informatik (Master of Science)", "Maschinenbau (Master of

Science)", "Mechatronik (Bachelor of Science)", "Mechatronik (Master of Science)", "Medizintechnik (Bachelor of Science)", "Wirtschaftsingenieurwesen (Master of Science)" verwendbar.

---

**Studien-/Prüfungsleistungen:**

Einführung in die Programmierung humanoider Roboter (Prüfungsnummer: 71241)

(englische Bezeichnung: Basic principles of programming humanoid robots)

Prüfungsleistung, Klausur, Dauer (in Minuten): 90

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Einführung in die Programmierung humanoider Roboter

weitere Erläuterungen:

Klausur kann einen Anteil an Fragen im Antwort-Wahl-Verfahren (Multiple-Choice) enthalten.

Erstablingung: WS 2019/2020, 1. Wdh.: SS 2020

1. Prüfer: Jörg Franke (100615)

---

---

**Modulbezeichnung:** Technische Grundlagen des ressourcenschonenden und intelligenten Wohnens (TGW) 2.5 ECTS  
 (Technische Grundlagen des ressourcenschonenden und intelligenten Wohnens)

Modulverantwortliche/r: Jörg Franke  
 Lehrende: Jörg Franke, Julian Praß

---

Startsemester: WS 2019/2020	Dauer: 1 Semester	Turnus: halbjährlich (WS+SS)
Präsenzzeit: 35 Std.	Eigenstudium: 40 Std.	Sprache: Deutsch

---

**Lehrveranstaltungen:**

Technische Grundlagen des ressourcenschonenden und intelligenten Wohnens (vhb) (WS 2019/2020, Vorlesung, 2 SWS, Jörg Franke et al.)

---

**Inhalt:**

Ebenso wie die Sektoren Verkehr und Industrie, gerät auch das private Wohnen zunehmend in das Spannungsfeld aus Ressourcenschonung und demografischem Wandel. Mit intelligenter Automatisierungstechnik ist es möglich, diesen Herausforderungen zu begegnen. Eine besondere Beachtung ist hier den soziologischen und ökonomischen Bedarfen zu schenken. Folgende Themenschwerpunkte werden im Rahmen der virtuellen Vorlesung adressiert:

- Energieerzeugung, -speicherung und -verteilung im privaten Umfeld
- Energieeffizient Wohnen mit intelligenter Automatisierungstechnik
- Steigerung von Sicherheit und Komfort durch nutzergerechte Hausautomation
- Betrachtung soziologischer, technologischer und ökonomischer Begleitfaktoren

**Lernziele und Kompetenzen:**

Nach Bearbeitung der Lehrveranstaltung sollen Sie als Studierende folgende Lernziele erreicht haben:

- Der Begriff Smart Home und die Interdependenzen seiner Domänen sind Ihnen bekannt
- Sie kennen die Charakteristiken der technischen Anlagen zur Stromerzeugung und deren physikalischen Grundlagen
- Sie sind fähig je nach Anforderung ein geeignetes Heizsystem auszuwählen
- Sie kennen die Grundlagen zu Transport- und Verteilung elektrischer Energie
- Die Problematik der Anbindung dezentraler, regenerativer Erzeugungsanlagen an das elektrische Versorgungsnetz ist Ihnen bekannt
- Ein Überblick zu vorhandener Sensorik und Aktorik im AAL-Bereich herrscht vor
- Sie kennen die charakteristischen Vor- und Nachteile der verschiedenen etablierten Kommunikationstechnologien im Smart-Home-Umfeld
- Sie können Prozesse und Methoden aufzählen und erklären, die für eine technische Realisierung eines sich selbst organisierenden Smart Homes wichtig sind
- Sie haben einen Überblick gewonnen, wie die Geräteklassen zur Realisierung ganzheitlicher Anwendungsszenarien verknüpft werden können
- Sie kennen die grundlegenden Begriffe aus dem Innovationsmanagement und der Innovationsforschung
- Der Begriff Akzeptanz ist Ihnen in seinen unterschiedlichen Dimensionen bekannt

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

**[1] Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Nebenfach | Nebenfach Maschinenbau | Computerintegrierte Produktion)

Dieses Modul ist daneben auch in den Studienfächern "Elektrotechnik, Elektronik und Informationstechnik (Bachelor of Science)", "Elektrotechnik, Elektronik und Informationstechnik (Master of Science)", "Informatik (Master of Science)", "Maschinenbau (Bachelor of Science)", "Maschinenbau (Master of Science)", "Mechatronik (Bachelor of Science)", "Mechatronik (Master of Science)", "Wirtschaftsingenieurwesen (Bachelor of Science)", "Wirtschaftsingenieurwesen (Master of Science)" verwendbar.



**Studien-/Prüfungsleistungen:**

Technische Grundlagen des ressourcenschonenden und intelligenten Wohnens (Prüfungsnummer: 49401)

Prüfungsleistung, Klausur, Dauer (in Minuten): 60

Anteil an der Berechnung der Modulnote: 100% Prüfungssprache: Deutsch

Erstablesung: WS 2019/2020, 1. Wdh.: SS 2020

1. Prüfer: Jörg Franke (100615)

---

**Organisatorisches:**

Fragen zur Vorlesung bitte an Julian Praß richten.

---

**Modulbezeichnung: Produktionssystematik (PS)** **5 ECTS**  
 (Production Systematics)

Modulverantwortliche/r: Jörg Franke  
 Lehrende: Jörg Franke

---

Startsemester: SS 2019	Dauer: 1 Semester	Turnus: jährlich (SS)
Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Deutsch

---

**Lehrveranstaltungen:**

Produktionssystematik (SS 2019, Vorlesung, 2 SWS, Jörg Franke et al.)  
 Übung zu Produktionssystematik (SS 2019, Übung, 2 SWS, Jörg Franke et al.)

---

**Inhalt:**

Ziel dieser Vorlesung Produktionssystematik ist es, dem Studenten die gesamte Bandbreite der technischen Betriebsführung von der Planung, Organisation und technischen Auftragsabwicklung bis hin zu Fragen des Management und der Personalführung, Entlohnung sowie Kosten- und Wirtschaftlichkeitsrechnung näherzubringen. Die Übung zur Vorlesung vertieft diese Themen.

**Lernziele und Kompetenzen:**

Nach einem Besuch der Vorlesung Produktionssystematik sollen die Studenten in der Lage sein:

- Ziele, Strategien, Vision und Mission der Unternehmen beurteilen zu können;
  - sich in der Aufbau- und Ablauforganisation eines Unternehmen zurecht zu finden;
  - die Inhalte der wesentlichen Kernprozesse produzierender Unternehmen zu kennen;
  - die technische und administrative Auftragsabwicklung nachzuvollziehen.
- 

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

**[1] Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Nebenfach | Nebenfach Maschinenbau | Computerintegrierte Produktion)

Dieses Modul ist daneben auch in den Studienfächern "Berufspädagogik Technik (Bachelor of Science)", "Berufspädagogik Technik (Master of Education)", "Informatik (Master of Science)", "International Production Engineering and Management (Bachelor of Science)", "Maschinenbau (Bachelor of Science)", "Maschinenbau (Master of Science)", "Mechatronik (Bachelor of Science)", "Mechatronik (Master of Science)", "Medizintechnik (Master of Science)", "Wirtschaftsingenieurwesen (Bachelor of Science)", "Wirtschaftsingenieurwesen (Master of Science)" verwendbar.

---

**Studien-/Prüfungsleistungen:**

Produktionssystematik (Vorlesung mit Übung) (Prüfungsnummer: 71011)

(englische Bezeichnung: Production Systematics)

Prüfungsleistung, Klausur, Dauer (in Minuten): 120

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Produktionssystematik
- Übung zu Produktionssystematik

Erstabelleung: SS 2019, 1. Wdh.: WS 2019/2020

1. Prüfer: Jörg Franke (100615)

---

**Organisatorisches:**

Für die Prüfung sind ausschließlich folgende Hilfsmittel zugelassen:

- nicht programmierbarer Taschenrechner
- dokumentenechter Stift
- Textmarker

- Lineal, Geodreieck, Zirkel
- Namensstempel

Darüber hinaus sind keine weiteren Hilfsmittel erlaubt (dies gilt insbesondere für Uhren, Mobiltelefone oder sonstige elektronische Geräte).

---

**Modulbezeichnung:** **Lineare und Kombinatorische Optimierung (LKOpt)** **10 ECTS**  
 (Linear and Combinatorial Optimization)

Modulverantwortliche/r: Alexander Martin  
 Lehrende: Dieter Weninger

---

Startsemester: WS 2019/2020	Dauer: 1 Semester	Turnus: jährlich (WS)
Präsenzzeit: 90 Std.	Eigenstudium: 210 Std.	Sprache:

---

**Lehrveranstaltungen:**

Lineare und Kombinatorische Optimierung (WS 2019/2020, Vorlesung, 4 SWS, Dieter Weninger)  
 Übung zur Linearen und Kombinatorischen Optimierung (WS 2019/2020, Übung, 2 SWS, Dieter Weninger)

---

**Empfohlene Voraussetzungen:**

Lineare Algebra

**Vorhergehende Module:**

Lineare Algebra II  
 Lineare Algebra I

---

**Inhalt:**

Schwerpunkt dieser Vorlesung ist die Theorie und Lösung kombinatorischer und in diesem Kontext linearer Optimierungsprobleme. Wir behandeln klassische Probleme auf Graphen, wie das Kürzeste-Wege-Problem, das Aufspannende-Baum-Problem oder das Max-Flow-Min-Cut-Theorem. Zum Vorlesungsumfang gehört auch das Simplexverfahren für lineare Programme und das Studium algorithmischer Grundprinzipien wie Sortieren, Greedy, Tiefen- und Breitensuche sowie Heuristiken.

**Lernziele und Kompetenzen:**

Die Studierenden

- erkennen und analysieren selbstständig kombinatorische Optimierungsprobleme;
- erläutern algorithmische Grundprinzipien und wenden diese zielorientiert an;
- klassifizieren komplexe Verfahren des Lerngebietes;
- sammeln und bewerten relevante Informationen und stellen Zusammenhänge her

**Literatur:**

- Vorlesungsskript zu diesem Modul
- Schrijver: Combinatorial Optimization Vol. A - C; Springer, 2003
- Korte, J. Vygen: Combinatorial Optimization; Springer, 2005

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

**[1] Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Nebenfach | Nebenfach Mathematik)

Dieses Modul ist daneben auch in den Studienfächern "Informatik (Master of Science)", "Mathematik (Bachelor of Science)", "Technomathematik (Bachelor of Science)", "Wirtschaftsmathematik (Bachelor of Science)" verwendbar.

---

**Studien-/Prüfungsleistungen:**

Vorlesung Kombinatorische Optimierung (Prüfungsnummer: 50461)

(englische Bezeichnung: Lecture: Combinatorial Optimisation)

Prüfungsleistung, Klausur, Dauer (in Minuten): 90

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Lineare und Kombinatorische Optimierung

Erstablesung: WS 2019/2020, 1. Wdh.: SS 2020

1. Prüfer: Dieter Weninger (060186)

## Übungsleistung Kombinatorische Optimierung (Prüfungsnummer: 50462)

(englische Bezeichnung: Tutorial Achievement: Combinatorial Optimisation)

Studienleistung, Übungsleistung

Zugeordnete Lehrveranstaltungen:

- Übung zur Linearen und Kombinatorischen Optimierung

weitere Erläuterungen:

Hausaufgaben (wöchentlich ein Übungsblatt)

Erstabelleung: WS 2019/2020, 1. Wdh.: keine Angabe

1. Prüfer: Dieter Weninger (060186)

---

### Organisatorisches:

Neben der vierstündigen Vorlesung werden zweistündige Übungen angeboten. Anhand von Präsenz- und Hausaufgaben werden wesentliche Lerninhalte geübt. Zusätzlich werden kleinere Softwareübungen angeboten.

### Bemerkungen:

Pflichtmodul in

- B. Sc. Wirtschaftsmathematik

Wahlpflichtmodul in

- B. Sc. Mathematik und Technomathematik

---

**Modulbezeichnung:** **Optimierung für Ingenieure (mit Praktikum) (OptIngVP)** 7.5 ECTS  
(Optimization for Engineers (with Laboratory))

Modulverantwortliche/r: Johannes Hild, Martin Gugat

Lehrende: Johannes Hild, Martin Gugat

---

Startsemester: SS 2019

Dauer: 1 Semester

Turnus: jährlich (SS)

Präsenzzeit: 60 Std.

Eigenstudium: 165 Std.

Sprache: Englisch

---

**Lehrveranstaltungen:**

Optimierung für Ingenieure (Optimization for Engineers) (SS 2019, Vorlesung mit Übung, 3 SWS, Johannes Hild)

Praktikum zu Optimierung für Ingenieure (Laboratory for Optimization for Engineers) (SS 2019, Übung, 1 SWS, Johannes Hild)

---

**Empfohlene Voraussetzungen:**

Requires contents of the lecture Mathematics for Engineers I, II and III. Especially:

- Linear algebra
- Analysis of real valued functions
- Differential and integral calculus in multi dimensional spaces

Requires basic knowledge in the implementation of algorithms and data structures in a development environment.

---

**Inhalt:**

**Introduction to continuous optimization problems and methods with and without constraints**

- Classification of problem types
- Optimality conditions and termination criterions
- Descent directions and line search methods
- Convergence analysis

**Unconstrained optimization**

- Steepest descent and conjugate gradient
- Newton-type methods
- Nonlinear Least Squares

**Constrained optimization**

- Projection methods
- Trust Region
- Barrier and penalty methods

**Outlook**

- Linear programming and simplex method
- Integer programming
- Noisy functions

**Laboratory**

- Introduction to MATLAB
- Implementation of optimization algorithms
- Algorithmic optimization of test problems

**Lernziele und Kompetenzen:**

*Fachkompetenz*

*Wissen*

- Students list requirements, strengths and weaknesses of common optimization methods.
- Students recognize crucial components in existence and convergence proofs in the context of minimizing sequences.
- Students identify optimization routines written in a programming language.

*Verstehen*

- Students explain the different components of optimization methods.

- Students describe the relationship between requirements and conclusions of existence and convergence theorems in the context of minimizing sequences.

#### Anwenden

- Students check feasibility, well-posedness and constraint qualifications of optimization problems.
- Students formulate and solve optimality conditions analytically.
- Students apply optimization algorithms to optimization problems.
- Students translate mathematical descriptions into flow control routines.

#### Analysieren

- Students analyse uncommon optimization approaches and extract their requirements, strengths and weaknesses.
- Students observe the behavior of common optimization algorithms applied to numerical test problems.

#### Evaluieren (Beurteilen)

- Students evaluate the class and structure of unsolved optimization problems.
- Students choose suitable algorithmic approaches for unsolved optimization problems.

#### Erschaffen

- Students formulate optimization problems using mathematical methods and structures.
- Students modify and combine common optimization routines to create project-specific algorithms for unsolved optimization problems.
- Students plan and implement optimization algorithms in a programming language.

#### Literatur:

Nocedal, Jorge and Wright, Stephen J.: Numerical Optimization. Springer Serie in Operations Research, 2006.

Kelley, C. T.: Iterative Methods for Optimization. Frontiers in Applied Mathematics 18, SIAM Philadelphia 1999;

Polak, E.: Optimization. Algorithms and Consistent Approximations. Applied Mathematical Sciences, Volume 124, Springer-Verlag New York, 1997.

Jarre, F.: Optimierung, Springer 2003;

Hamacher, H.W. and K. Klamroth, K.: Linear and Network Optimization: bilingual textbook. Vieweg 2000

#### Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

##### [1] Informatik (Bachelor of Science)

(Po-Vers. 2009s | Nebenfach | Nebenfach Mathematik)

Dieses Modul ist daneben auch in den Studienfächern "Computational Engineering (Master of Science with Honours)", "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Master of Science)", "Informatik (Master of Science)" verwendbar.

#### Studien-/Prüfungsleistungen:

Optimierung für Ingenieure (Prüfungsnummer: 40501)

(englische Bezeichnung: Optimization for Engineers)

Prüfungsleistung, Klausur, Dauer (in Minuten): 90

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Optimierung für Ingenieure (Optimization for Engineers)

Prüfungssprache: Englisch

Erstablesung: SS 2019, 1. Wdh.: WS 2019/2020

1. Prüfer: Johannes Hild (060100)

## Optimierung für Ingenieure - Übungen (Prüfungsnummer: 40602)

(englische Bezeichnung: Optimization for Engineers - Laboratory)

Studienleistung, Übungsleistung

Zugeordnete Lehrveranstaltungen:

- Praktikum zu Optimierung für Ingenieure (Laboratory for Optimization for Engineers)

weitere Erläuterungen:

Aquired by completing and testing optimization algorithms in MATLAB. Templates are provided by the lecturer, solution files are returned using StudOn. Laboratory is passed, if 66% of the solution files are correct.

Prüfungssprache: Englisch

Erstablingung: SS 2019, 1. Wdh.: keine Angabe

1. Prüfer: Johannes Hild (060100)

---

### Bemerkungen:

This module aims at students of the Faculty of Engineering of all disciplines and is suitable as an elective subject in the Bachelor's and Master's degree.



---

**Modulbezeichnung:** **Numerik I für Ingenieure (NumIng1)** **5 ECTS**  
 (Numerics for Engineers I)

Modulverantwortliche/r: Wilhelm Merz, J. Michael Fried

Lehrende: Wilhelm Merz, J. Michael Fried, Nicolas Neuß, u.a.

Startsemester: WS 2019/2020

Dauer: 1 Semester

Turnus: jährlich (WS)

Präsenzzeit: 60 Std.

Eigenstudium: 90 Std.

Sprache: Deutsch

---

**Lehrveranstaltungen:**

Numerik I für Ingenieure (WS 2019/2020, Vorlesung, 2 SWS, Nicolas Neuß)

Übungen zur Numerik I für Ingenieure (WS 2019/2020, Praktikum, 2 SWS, Nicolas Neuß)

---

**Empfohlene Voraussetzungen:**

Kurs Mathematik für Ingenieure I, II und III

---

**Inhalt:**

**Elementare Numerik**

Direkte und iterative Lösungsverfahren bei linearen Gleichungssystemen, Interpolation mit Newton-Polynomen und Splines, Quadratur mit Newton-Côtes-Formeln, Extrapolation nach Romberg

**Numerik gewöhnlicher Differentialgleichungen:**

Verschiedene Runge-Kutta Methoden als Einschrittverfahren, Konsistenz, Stabilität- und Konvergenzaussage, Mehrschrittverfahren

**Lernziele und Kompetenzen:**

Die Studierenden lernen

- verschiedene numerische Methoden zur Lösung linearer Gleichungssysteme
- verschiedene Methoden zu beurteilen
- Interpolationstechniken und Güte der Approximation
- grundlegende Quadraturverfahren und die Beurteilung solcher
- grundlegende Diskretisierungsmethoden bei gewöhnlichen Differentialgleichungen
- Beurteilung dieser Methoden und Verfahren
- algorithmische Umsetzung o.g. Verfahren als Grundlage für Computer-Codes

**Literatur:**

Skripte des Dozenten

H.-R. Schwarz, N. Köckler: *Numerische Mathematik*, Teubner

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

**[1] Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Nebenfach | Nebenfach Mathematik)

Dieses Modul ist daneben auch in den Studienfächern "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Bachelor of Science)", "Informatik (Master of Science)", "Maschinenbau (Bachelor of Science)", "Maschinenbau (Master of Science)", "Mechatronik (Bachelor of Science)", "Medizintechnik (Bachelor of Science)", "Medizintechnik (Master of Science)", "Physische Geographie (Bachelor of Science)" verwendbar.

---

**Studien-/Prüfungsleistungen:**

Vorlesung Numerik I für Ingenieure (Prüfungsnummer: 46201)

(englische Bezeichnung: Lecture: Numerics for Engineers I)

Prüfungsleistung, Klausur, Dauer (in Minuten): 60

Anteil an der Berechnung der Modulnote: 100%

Erstablesung: WS 2019/2020, 1. Wdh.: SS 2020 (nur für Wiederholer)

1. Prüfer: Nicolas Neuß (060215)

---

**Modulbezeichnung:** Numerik II für Ingenieure (NumIng2) 5 ECTS  
 (Numerics for Engineers II)

Modulverantwortliche/r: Wilhelm Merz  
 Lehrende: Wilhelm Merz

---

Startsemester: SS 2019	Dauer: 1 Semester	Turnus: jährlich (SS)
Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Deutsch

---

**Lehrveranstaltungen:**

Numerik II für Ingenieure (SS 2019, Vorlesung, 2 SWS, J. Michael Fried)  
 Übungen zur Numerik II für Ingenieure (SS 2019, Übung, 2 SWS, J. Michael Fried)

---

**Inhalt:**

**Numerik partieller Differentialgleichungen**

Finite Differenzenmethode, Stabilität, Konsistenz, Konvergenz, Einführung finite Elementmethode bei elliptischen Problemen, Fehlerschätzer

**Lernziele und Kompetenzen:**

Die Studierenden

- erklären verschiedene Diskretisierungsmethoden
- beurteilen diese Diskretisierungsmethoden
- leiten Finite Elemente Diskretisierungen elliptischer Probleme her
- folgern Aussagen anhand grundlegender Beweistechniken aus oben genannten Bereichen
- konstruieren Algorithmen zu Finite Elemente Diskretisierungen
- erklären Fehlerschätzer

**Literatur:**

Skripte des Dozenten  
 H. Jung, M. Langer, Methode der Finiten Elemente, Teubner  
 P. Knabner, L. Angermann, Numerik partieller Differentialgleichungen, Springer

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

**[1] Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Nebenfach | Nebenfach Mathematik)

Dieses Modul ist daneben auch in den Studienfächern "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Bachelor of Science)", "Informatik (Master of Science)", "Maschinenbau (Master of Science)", "Medizintechnik (Bachelor of Science)", "Medizintechnik (Master of Science)" verwendbar.

---

**Studien-/Prüfungsleistungen:**

Vorlesung Numerik II für Ingenieure (Prüfungsnummer: 46311)

(englische Bezeichnung: Lecture: Numerics for Engineers II)

Prüfungsleistung, Klausur, Dauer (in Minuten): 60

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Numerik II für Ingenieure
- Übungen zur Numerik II für Ingenieure

Erstablingung: SS 2019, 1. Wdh.: WS 2019/2020

1. Prüfer: J. Michael Fried (060210)

---

---

**Modulbezeichnung:** **Discrete Optimization (DiscOpt I)** **5 ECTS**  
 (Discrete Optimization)

Modulverantwortliche/r: Alexander Martin  
 Lehrende: Andreas Bärmann

---

Startsemester: WS 2019/2020	Dauer: 1 Semester	Turnus: unregelmäßig
Präsenzzeit: 45 Std.	Eigenstudium: 105 Std.	Sprache:

---

**Lehrveranstaltungen:**

Diskrete Optimierung I (WS 2019/2020, Vorlesung, 2 SWS, Andreas Bärmann)

---

**Empfohlene Voraussetzungen:**

Recommended: Linear and Combinatorial Optimization

---

**Inhalt:**

Theoretical and practical fundamentals of solving difficult mixed-integer linear optimization problems (MIPs) constitute the main focus of this lecture. At first, the concept of NP-completeness and a selection of common NP-complete problems will be presented. As for polyhedral theory, fundamentals concerning the structure of faces of convex polyhedra will be covered. Building upon these fundamentals, cutting plane algorithms as well as branch-and-cut algorithms for solving MIPs will be taught. Finally, some typical problems of discrete optimization, e.g., the knapsack problem, the traveling salesman problem or the set packing problem will be discussed.

**Lernziele und Kompetenzen:**

Students

- will gain basic theoretical knowledge of solving mixed-integer linear optimization problems (MIPs),
- are able to solve MIPs with the help of state-of-the-art optimization software

**Literatur:**

- Lecture notes
  - Conforti, Cornuéjols, Zambelli: Integer Programming, Springer 2014
  - B. Grünbaum, Convex Polytopes, Springer, 2003
  - B. Korte, J. Vygen: Combinatorial Optimization, Springer 2005
  - G. L. Nemhauser, L.A. Wolsey: Integer and Combinatorial Optimization, Wiley 1994
  - A. Schrijver: Theory of Linear and Integer Programming, Wiley 1986
  - L.A. Wolsey: Integer Programming, Wiley 1998
  - G. Ziegler, Lectures on Polytopes, Springer, 1995
- 

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

**[1] Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Nebenfach | Nebenfach Mathematik)

Dieses Modul ist daneben auch in den Studienfächern "Computational and Applied Mathematics (Master of Science)", "Informatik (Master of Science)" verwendbar.

---

**Studien-/Prüfungsleistungen:**

Discrete optimization I (Prüfungsnummer: 59171)

(englische Bezeichnung: Discrete optimization I)

Prüfungsleistung, mündliche Prüfung, Dauer (in Minuten): 15

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Diskrete Optimierung I

Erstablingung: WS 2019/2020, 1. Wdh.: WS 2019/2020

1. Prüfer: Andreas Bärmann (060187)

---

---

**Modulbezeichnung:** Diskrete Optimierung I (DiskOpt I) 5 ECTS  
 (Discrete optimization I)

Modulverantwortliche/r: Alexander Martin  
 Lehrende: Andreas Bärmann

---

Startsemester: WS 2019/2020	Dauer: 1 Semester	Turnus: jährlich (WS)
Präsenzzeit: 45 Std.	Eigenstudium: 105 Std.	Sprache: Deutsch

---

**Lehrveranstaltungen:**

Diskrete Optimierung I (WS 2019/2020, Vorlesung, 2 SWS, Andreas Bärmann)  
 Tutorial zu Diskrete Optimierung I (WS 2019/2020, Übung, 1 SWS, Andreas Bärmann)

---

**Empfohlene Voraussetzungen:**

Lineare Algebra, Kombinatorische Optimierung

---

**Inhalt:**

Die Vorlesung behandelt theoretische und praktische Grundlagen zur Lösung schwieriger gemischt-ganzzahliger linearer Optimierungsprobleme (MIPs). Zunächst werden Kerndefinitionen der NP-Vollständigkeit behandelt und einige der bekannten NP-vollständigen Probleme vorgestellt. Im Bereich der Polyedertheorie werden die Grundlagen der Seitenstruktur konvexer Polyeder behandelt. Darauf aufbauend werden Schnittebenenverfahren sowie Branch-and-Cut Verfahren zur Lösung von MIPs gelehrt. Abschließend studieren wir einige klassische Probleme der Diskreten Optimierung wie das Rucksack-Problem, das Traveling-Salesman-Problem oder das Set-Packing-Problem.

**Lernziele und Kompetenzen:**

Die Studierenden

- verfügen über grundlegende theoretische Erkenntnisse zur Lösung gemischt-ganzzahliger linearer Optimierungsprobleme (MIPs),
- können MIPs mittels verfügbarer Standard Software lösen.

**Literatur:**

- Vorlesungsskript zu diesem Modul
- Conforti, Cornuéjols, Zambelli: Integer Programming, Springer 2014
- B. Grünbaum, Convex Polytopes, Springer, 2003
- B. Korte, J. Vygen: Combinatorial Optimization, Springer 2005
- G. L. Nemhauser, L.A. Wolsey: Integer and Combinatorial Optimization, Wiley 1994
- A. Schrijver: Theory of Linear and Integer Programming, Wiley 1986
- L.A. Wolsey: Integer Programming, Wiley 1998
- G. Ziegler, Lectures on Polytopes, Springer, 1995

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

[1] **Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Nebenfach | Nebenfach Mathematik)

Dieses Modul ist daneben auch in den Studienfächern "Computational and Applied Mathematics (Master of Science)", "Informatik (Master of Science)" verwendbar.

---

**Studien-/Prüfungsleistungen:**

Discrete optimization I (Prüfungsnummer: 59171)

(englische Bezeichnung: Discrete optimization I)

Prüfungsleistung, mündliche Prüfung, Dauer (in Minuten): 15

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Diskrete Optimierung I

Erstablingung: WS 2019/2020, 1. Wdh.: WS 2019/2020

1. Prüfer: Andreas Bärmann (060187)

---

**Organisatorisches:**

Neben der Vorlesung werden Übungen angeboten, in denen die Studierenden von einem Übungsgruppenleiter betreut werden. Anhand von Präsenz- und Hausaufgaben werden wesentliche Lerninhalte geübt.

Bis WS 14/15 hieß das Modul "Theoretische Grundlagen der Diskreten Optimierung"!

**Bemerkungen:**

- Wahlmodul: Master Mathematik, Technomathematik und Wirtschaftsmathematik
- Kern-/Forschungsmodul Master Mathematik Studienrichtung „Modellierung, Simulation, Optimierung“, Master Technomathematik Studienrichtung „Optimierung“, Master Wirtschaftsmathematik Studienrichtung „Optimierung und Prozessmanagement“

---

**Modulbezeichnung:** **Lineare und Kombinatorische Optimierung (LKOpt)** **10 ECTS**  
 (Linear and Combinatorial Optimization)

Modulverantwortliche/r: Alexander Martin  
 Lehrende: Dieter Weninger

---

Startsemester: WS 2019/2020	Dauer: 1 Semester	Turnus: jährlich (WS)
Präsenzzeit: 90 Std.	Eigenstudium: 210 Std.	Sprache:

---

**Lehrveranstaltungen:**

Lineare und Kombinatorische Optimierung (WS 2019/2020, Vorlesung, 4 SWS, Dieter Weninger)  
 Übung zur Linearen und Kombinatorischen Optimierung (WS 2019/2020, Übung, 2 SWS, Dieter Weninger)

---

**Empfohlene Voraussetzungen:**

Lineare Algebra

**Vorhergehende Module:**

Lineare Algebra II  
 Lineare Algebra I

---

**Inhalt:**

Schwerpunkt dieser Vorlesung ist die Theorie und Lösung kombinatorischer und in diesem Kontext linearer Optimierungsprobleme. Wir behandeln klassische Probleme auf Graphen, wie das Kürzeste-Wege-Problem, das Aufspannende-Baum-Problem oder das Max-Flow-Min-Cut-Theorem. Zum Vorlesungsumfang gehört auch das Simplexverfahren für lineare Programme und das Studium algorithmischer Grundprinzipien wie Sortieren, Greedy, Tiefen- und Breitensuche sowie Heuristiken.

**Lernziele und Kompetenzen:**

Die Studierenden

- erkennen und analysieren selbstständig kombinatorische Optimierungsprobleme;
- erläutern algorithmische Grundprinzipien und wenden diese zielorientiert an;
- klassifizieren komplexe Verfahren des Lerngebietes;
- sammeln und bewerten relevante Informationen und stellen Zusammenhänge her

**Literatur:**

- Vorlesungsskript zu diesem Modul
- Schrijver: Combinatorial Optimization Vol. A - C; Springer, 2003
- Korte, J. Vygen: Combinatorial Optimization; Springer, 2005

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

**[1] Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Nebenfach | Nebenfach Mathematik)

Dieses Modul ist daneben auch in den Studienfächern "Informatik (Master of Science)", "Mathematik (Bachelor of Science)", "Technomathematik (Bachelor of Science)", "Wirtschaftsmathematik (Bachelor of Science)" verwendbar.

---

**Studien-/Prüfungsleistungen:**

Übungsleistung: Lineare und Kombinatorische Optimierung (Prüfungsnummer: 51602)

Studienleistung, Übungsleistung

Zugeordnete Lehrveranstaltungen:

- Übung zur Linearen und Kombinatorischen Optimierung

weitere Erläuterungen:

Hausaufgaben (wöchentlich ein Übungsblatt)

Erstablesung: WS 2019/2020, 1. Wdh.: keine Angabe

1. Prüfer: Dieter Weninger (060186)

Klausur: Lineare und Kombinatorische Optimierung (Prüfungsnummer: 51601)

Prüfungsleistung, Klausur, Dauer (in Minuten): 90

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Lineare und Kombinatorische Optimierung

Erstablingung: WS 2019/2020, 1. Wdh.: SS 2020

1. Prüfer: Dieter Weninger (060186)

---

**Organisatorisches:**

Neben der vierstündigen Vorlesung werden zweistündige Übungen angeboten. Anhand von Präsenz- und Hausaufgaben werden wesentliche Lerninhalte geübt. Zusätzlich werden kleinere Softwareübungen angeboten.

**Bemerkungen:**

Pflichtmodul in

- B. Sc. Wirtschaftsmathematik

Wahlpflichtmodul in

- B. Sc. Mathematik und Technomathematik



---

**Modulbezeichnung: Algebra (Alg)** **10 ECTS**  
(Algebra)

Modulverantwortliche/r: Friedrich Knop  
Lehrende: Friedrich Knop

---

Startsemester: WS 2019/2020	Dauer: 1 Semester	Turnus: jährlich (WS)
Präsenzzeit: 105 Std.	Eigenstudium: 195 Std.	Sprache: Deutsch

---

**Lehrveranstaltungen:**

Algebra (WS 2019/2020, Vorlesung, 4 SWS, Friedrich Knop)  
Übungen zur Algebra (WS 2019/2020, Übung, 2 SWS, Friedrich Knop)  
Tutorium zur Algebra (WS 2019/2020, Tutorium, Friedrich Knop)

---

**Empfohlene Voraussetzungen:**

Lineare Algebra I und II

---

**Inhalt:**

- Gruppentheorie: Untergruppen, Quotienten, Operationen von Gruppen, endlich erzeugte abelsche Gruppen
- Ringtheorie: Ideale, Quotienten, Polynomringe, maximale Ideale,
- Irreduzibilität
- Elementare Zahlentheorie: Restklassenringe, Eulersche phi-Funktion, Chinesischer Restsatz, quadratisches Reziprozitätsgesetz

**Lernziele und Kompetenzen:**

Die Studierenden

- nennen und erklären algebraische Strukturen anhand von Gruppen, Ringen und Körpern und verwenden diese;
- behandeln auch komplexe Symmetrien mittels Gruppentheorie selbständig;
- lösen geometrische und zahlentheoretische Probleme mittels Ringtheorie und Zahlentheorie;
- sammeln und bewerten relevante Informationen und erkennen Zusammenhänge.

**Literatur:**

- M. Artin: Algebra
- G. Fischer: Algebra
- N. Jacobson: Basic Algebra I,II
- S. Lang: Algebra

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

**[1] Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Nebenfach | Nebenfach Mathematik)

Dieses Modul ist daneben auch in den Studienfächern "Informatik (Master of Science)", "Mathematik (1. Staatsprüfung für das Lehramt an Gymnasien)", "Mathematik (Bachelor of Science)", "Wirtschaftsmathematik (Bachelor of Science)" verwendbar.

---

**Studien-/Prüfungsleistungen:**

Modulabschlussprüfung: Algebra (Prüfungsnummer: 53111)

Prüfungsleistung, Klausur mit Übungsleistung, Dauer (in Minuten): 90

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Algebra
- Übungen zur Algebra
- Tutorium zur Algebra

weitere Erläuterungen:

Klausur 90 Min. (benotet; 100 %) und Hausaufgaben (wöchentlich 1 Übungsblatt) (unbenotet)

Erstablingung: WS 2019/2020, 1. Wdh.: SS 2020

1. Prüfer: Friedrich Knop (060112)

---

**Organisatorisches:**

Die Präsentation des Stoffes erfolgt in Vorlesungsform. Die weitere Aneignung der wesentlichen Begriffe und Techniken erfolgt durch wöchentliche Hausaufgaben.

**Bemerkungen:**

Wahlpflichtmodul in

- B. Sc. Mathematik, Wirtschaftsmathematik und für das Lehramt vertieft
- Nachklausur jeweils zu Beginn des kommenden Sommersemesters

---

**Modulbezeichnung:** Kryptographie I (Kry I) 10 ECTS  
(Cryptography I)

Modulverantwortliche/r: Wolfgang Ruppert  
Lehrende: Wolfgang Ruppert

---

Startsemester: WS 2019/2020	Dauer: 1 Semester	Turnus: unregelmäßig
Präsenzzeit: 90 Std.	Eigenstudium: 210 Std.	Sprache:

---

**Lehrveranstaltungen:**

Kryptographie I (WS 2019/2020, Vorlesung, 4 SWS, Wolfgang Ruppert)  
Übungen zur Kryptographie I (WS 2019/2020, Übung, 2 SWS, Wolfgang Ruppert)

---

**Empfohlene Voraussetzungen:**

Grundkenntnisse aus den Modulen Analysis I und Lineare Algebra I

---

**Inhalt:**

- Einführung in die Kryptographie
- Klassische Chiffrierverfahren
- Grundeigenschaften der Ringe  $Z$  und  $Z/nZ$
- Primzahltests
- Public-Key-Kryptosysteme - RSA
- Die Pollard-rho-Methode zur Faktorisierung
- Kryptographische Anwendungen diskreter Logarithmen
- Kryptographische Hashfunktionen
- Digitale Signaturen
- Methoden zur Berechnung diskreter Logarithmen
- Enigma

**Lernziele und Kompetenzen:**

Die Studierenden

- erklären wichtige kryptographische Verfahren und wenden diese praktisch an;
- nützen Software wie Maple, Python3 oder Sage zur Ver- und Entschlüsselung sowie zur Kryptoanalyse;
- erläutern wichtige zahlentheoretische Algorithmen, ihre theoretischen Hintergründe und ihre Funktion bei der Konstruktion von Public-Key-Kryptosystemen.

**Literatur:**

- Vorlesungsskript zum Modul
- J. Buchmann: Einführung in die Kryptographie
- J. Hoffstein, J. Pipher, J. H. Silvermann: An Introduction to Mathematical Cryptography

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

**[1] Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Nebenfach | Nebenfach Mathematik)

Dieses Modul ist daneben auch in den Studienfächern "Informatik (Master of Science)", "Mathematik (1. Staatsprüfung für das Lehramt an Gymnasien)", "Mathematik (Bachelor of Science)", "Wirtschaftsmathematik (Bachelor of Science)" verwendbar.

---

**Studien-/Prüfungsleistungen:**

Vorlesung Angewandte Mathematik (Prüfungsnummer: 56011)

(englische Bezeichnung: Lecture: Applied Mathematics)

Untertitel: Kryptografie I Prüfungsleistung, Klausur, Dauer (in Minuten): 90

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Kryptographie I

Erstablingung: WS 2019/2020, 1. Wdh.: SS 2020

1. Prüfer: Wolfgang Ruppert (060148)

Übung Angewandte Mathematik (Prüfungsnummer: 56012)

(englische Bezeichnung: Tutorial: Applied Mathematics)

Untertitel: Kryptographie I Studienleistung, Übungsleistung

Zugeordnete Lehrveranstaltungen:

- Übungen zur Kryptographie I

weitere Erläuterungen:

erfolgreiche Bearbeitung wöchentlicher Hausaufgaben

Erstablingung: WS 2019/2020, 1. Wdh.: keine Angabe

1. Prüfer: Wolfgang Ruppert (060148)

---

**Organisatorisches:**

Die Präsentation des Stoffes erfolgt in Vorlesungsform. Die weitere Aneignung der wesentlichen Begriffe und Techniken erfolgt durch wöchentliche Hausaufgaben.

**Bemerkungen:**

Wahlpflichtmodul in

- B. Sc. Mathematik, Technomathematik, Wirtschaftsmathematik und für das Lehramt vertieft

---

**Modulbezeichnung:** Discrete Optimization II (DiscOptII) 10 ECTS  
 (Discrete Optimization II)

Modulverantwortliche/r: Alexander Martin  
 Lehrende: Alexander Martin

---

Startsemester: SS 2019	Dauer: 1 Semester	Turnus: jährlich (SS)
Präsenzzeit: 90 Std.	Eigenstudium: 210 Std.	Sprache: Englisch

---

**Lehrveranstaltungen:**

Diskrete Optimierung II (SS 2019, Vorlesung, 4 SWS, Alexander Martin)  
 Übung Diskrete Optimierung II (SS 2019, Übung, 2 SWS, Alexander Martin et al.)

---

**Empfohlene Voraussetzungen:**

Knowledge in linear and combinatorial optimization, discrete optimization I

---

**Inhalt:**

In this lecture we cover theoretical aspects and solution strategies for difficult integer and mixed-integer optimization problems. First, we point out the equivalence between separation and optimization. Second, fundamental results of integral polyhedra, lattices and lattice polyhedra as well as its importance to discrete optimization are discussed. Furthermore, we introduce solution strategies for large-scale optimization problems, e.g., decomposition methods or approximation algorithms and heuristics based on linear programming. In addition, we discuss applications arising in engineering, finance, energy management or public transport.

**Lernziele und Kompetenzen:**

Students

- use basic terms of discrete optimization,
- model real-world discrete optimization problems, determine their complexity and solve them with appropriate mathematical methods.

**Literatur:**

- Lecture notes
- D. Bertsimas, R. Weismantel: Optimization over Integers, Dynamic Ideas, 2005
- Conforti, Cornuéjols, Zambelli: Integer Programming, Springer 2014
- G. L. Nemhauser, L.A. Wolsey: Integer and Combinatorial Optimization, Wiley 1994
- A. Schrijver: Combinatorial optimization Vol. A C, Springer 2003
- A. Schrijver: Theory of Linear and Integer Programming, Wiley, 1986 -L.A. Wolsey: Integer Programming, Wiley

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

[1] **Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Nebenfach | Nebenfach Mathematik)

Dieses Modul ist daneben auch in den Studienfächern "Computational and Applied Mathematics (Master of Science)", "Informatik (Master of Science)" verwendbar.

---

**Studien-/Prüfungsleistungen:**

Discrete optimization II (Prüfungsnummer: 59221)

(englische Bezeichnung: Discrete optimization II)

Prüfungsleistung, mündliche Prüfung, Dauer (in Minuten): 20

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Diskrete Optimierung II

Erstabelleung: SS 2019, 1. Wdh.: SS 2019

1. Prüfer: Alexander Martin (060159)



---

**Modulbezeichnung:** Diskretisierung und numerische Optimierung (DnO) 10 ECTS  
(Discretization and Numerical Optimization)

Modulverantwortliche/r: Peter Knabner, Günter Leugering, Eberhard Bänsch

Lehrende: Martin Burger

---

Startsemester: SS 2019	Dauer: 1 Semester	Turnus: jährlich (SS)
Präsenzzeit: 90 Std.	Eigenstudium: 210 Std.	Sprache: Deutsch

---

**Lehrveranstaltungen:**

Diskretisierung und numerische Optimierung (SS 2019, Vorlesung, 4 SWS, Martin Burger et al.)  
 Übungen zu Diskretisierung und numerische Optimierung (SS 2019, Übung, 2 SWS, Martin Burger et al.)

---

**Empfohlene Voraussetzungen:**

Die Module Analysis, Lineare Algebra, Programmierung und Einführung Numerik.

---

**Inhalt:**

Teil 1: Diskretisierung

Ein- und Mehrschrittverfahren für Anfangswertaufgaben gewöhnlicher Differentialgleichungen:

- explizite und implizite Runge-Kutta-Verfahren, BDF, Extrapolation
- asymptotische Stabilität (Nullstabilität), Konsistenz, Konvergenz
- Steifheit und Stabilität bei fester Schrittweite
- Schrittweiten- und Ordnungsadaptivität
- Randwertaufgaben für gewöhnliche Differentialgleichungen
- Einführung in Finite-Element-Verfahren

Teil 2: Unrestringierte Optimierung

- Abstiegsverfahren
- CG-Verfahren (mit Vorkonditionierung, CG-Newton)
- Quadratische Optimierungsprobleme
- Penalty- und Barriereverfahren

**Lernziele und Kompetenzen:**

Die Studierenden

- verwenden algorithmische Zugänge zu Problemen, die mittels gewöhnlicher Differentialgleichungen beschrieben werden können oder von unrestringierten endlichdimensionalen Optimierungsproblemen herkommen, und erklären und bewerten diese;
- urteilen über die Stabilität und Effizienz eines numerischen Verfahrens;
- setzen mit eigener oder gegebener Software Verfahren um und bewerten deren Ergebnisse kritisch;
- erläutern und verwenden ein breites Problem- und Verfahrensspektrum: Differenzenverfahren für Anfangs- und Randwertaufgaben, Finite-Element-Verfahren für 2-Punkt-Randwertaufgaben
- übertragen die erlangten Fachkompetenzen auf die Behandlung partieller Differentialgleichungen, Abstiegs- und CG-Verfahren bis zum Barriereverfahren;
- sammeln und bewerten relevante Informationen und erkennen Zusammenhänge.

**Literatur:**

- P. Deuffhard und F. Bornemann: Numerische Mathematik II; de Gruyter, Berlin 2002
- J. Stoer und R. Bulirsch: Numerische Mathematik II; Springer, Berlin, 2005
- K. Strehmel und R. Weiner: Numerik gewöhnlicher Differentialgleichungen; Teubner, Stuttgart 1995
- A. Quarteroni, R. Sacco und F. Saleri: Numerische Mathematik I, II; Springer, Berlin 2002
- Vorlesungsskriptum auf der Homepage des Bereichs Modellierung, Simulation und Optimierung des Departments Mathematik, ständig neu an die Vorlesung angepasst

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

[1] **Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Nebenfach | Nebenfach Mathematik)

Dieses Modul ist daneben auch in den Studienfächern "Informatik (Master of Science)", "Mathematik (Bachelor of Science)", "Technomathematik (Bachelor of Science)", "Wirtschaftsmathematik (Bachelor of Science)" verwendbar.

---

**Studien-/Prüfungsleistungen:**

Klausur: Diskretisierung und Numerische Optimierung (Prüfungsnummer: 52311)

Prüfungsleistung, Klausur, Dauer (in Minuten): 90

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Diskretisierung und numerische Optimierung
- Übungen zu Diskretisierung und numerische Optimierung

Erstablingung: SS 2019, 1. Wdh.: WS 2019/2020

1. Prüfer: Martin Burger (060190)

Übungsleistung: Diskretisierung und Numerische Optimierung (Prüfungsnummer: 52312)

Studienleistung, Übungsleistung

Zugeordnete Lehrveranstaltungen:

- Übungen zu Diskretisierung und numerische Optimierung

weitere Erläuterungen:

erfolgreiche Bearbeitung wöchentlicher Hausaufgaben

Erstablingung: SS 2019, 1. Wdh.: keine Angabe

1. Prüfer: Martin Burger (060190)

---

**Organisatorisches:**

Die Präsentation des Stoffes erfolgt in Vorlesungsform. Die weitere Aneignung der wesentlichen Begriffe und Techniken erfolgt durch wöchentliche Hausaufgaben.

**Bemerkungen:**

Wahlpflichtmodul in

- B. Sc. Mathematik, Technomathematik und Wirtschaftsmathematik



---

**Modulbezeichnung: Elementare Zahlentheorie (EZth) 5 ECTS**

(Elementary Number Theory)

Modulverantwortliche/r: Friedrich Knop, Yasmine Sanderson

Lehrende: Christina Birkenhake

Startsemester: WS 2019/2020

Dauer: 1 Semester

Turnus: jährlich (WS)

Präsenzzeit: 60 Std.

Eigenstudium: 90 Std.

Sprache: Deutsch

---

**Lehrveranstaltungen:**

Elementare Zahlentheorie (WS 2019/2020, Vorlesung, 4 SWS, Christina Birkenhake)

---

**Empfohlene Voraussetzungen:**

Elemente der Linearen Algebra I und II sowie Elemente der Analysis I und II

---

**Inhalt:**

 Anwendung der vollständigen Induktion, Division mit Rest, Untergruppen von  $\mathbb{Z}$ , ggT und kgV, euklidischer Algorithmus, Teilbarkeitslehre, Begriff der Primzahl und Fundamentalsatz der Arithmetik, Primzahlen und Primzahlprobleme, Diophantik mit Anwendungen, Prime Restklassengruppe, Dezimalbruch-Entwicklung, Algebraische und transzendente Zahlen

**Lernziele und Kompetenzen:**

Die Studierenden erklären grundlegende Begriffe der elementaren Zahlentheorie und wenden sie auf klassische mathematische Probleme an.

**Literatur:**

Vorlesungsskript zu diesem Modul

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

**[1] Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Nebenfach | Nebenfach Mathematik)

Dieses Modul ist daneben auch in den Studienfächern "Berufspädagogik Technik (Master of Education)", "Informatik (Master of Science)", "Mathematik (1. Staatsprüfung für das Lehramt an Grundschulen)", "Mathematik (1. Staatsprüfung für das Lehramt an Gymnasien)", "Mathematik (1. Staatsprüfung für das Lehramt an Hauptschulen)", "Mathematik (1. Staatsprüfung für das Lehramt an Mittelschulen)", "Mathematik (1. Staatsprüfung für das Lehramt an Realschulen)", "Mathematik (Bachelor of Education)" verwendbar.

---

**Studien-/Prüfungsleistungen:**

Elementare Zahlentheorie (Prüfungsnummer: 55801)

Prüfungsleistung, Klausur, Dauer (in Minuten): 60

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Elementare Zahlentheorie

Erstablegung: WS 2019/2020, 1. Wdh.: SS 2020

1. Prüfer: Christina Birkenhake (060111)

Elementare Zahlentheorie (Prüfungsnummer: 55802)

Studienleistung, Übungsleistung

Zugeordnete Lehrveranstaltungen:

- Elementare Zahlentheorie

weitere Erläuterungen:

erfolgreiche Bearbeitung wöchentlicher Übungsblätter

Erstablegung: WS 2019/2020, 1. Wdh.: keine Angabe

1. Prüfer: Christina Birkenhake (060111)

**Organisatorisches:**

Wahlpflichtmodul für alle nicht-vertieften Lehramtsstudiengänge

**Bemerkungen:**

Lehrform: Die Präsentation des Stoffes erfolgt in Vorlesungsform. Die weitere Aneignung der wesentlichen Begriffe und Techniken erfolgt durch wöchentliche Hausaufgaben.

---

**Modulbezeichnung:** **Funktionalanalysis I (FA1)** **10 ECTS**  
 (Functional analysis I)

Modulverantwortliche/r: Günther Grün  
 Lehrende: Hermann Schulz-Baldes

---

Startsemester: SS 2019	Dauer: 1 Semester	Turnus: jährlich (SS)
Präsenzzeit: 90 Std.	Eigenstudium: 210 Std.	Sprache:

---

**Lehrveranstaltungen:**

Funktionalanalysis I (SS 2019, Vorlesung, 4 SWS, Hermann Schulz-Baldes)  
 Übung zu Funktionalanalysis I (SS 2019, Übung, 2 SWS, Hermann Schulz-Baldes)

---

**Empfohlene Voraussetzungen:**

Drei der vier Module Lineare Algebra I und II, Analysis I und II müssen bestanden sein.

---

**Inhalt:**

Grundlagen zu folgenden Themen:

- Hilbert- und Banach-Räume
- Sobolev-Räume
- Lineare Operatoren
- Lineare Funktionale und der Satz von Hahn-Banach
- Prinzip der gleichmäßigen Beschränktheit
- Kompakte Operatoren
- Lösbarkeit linearer Gleichungen (inklusive Fredholm'sche Alternative)
- Spektraltheorie kompakter Operatoren und Anwendungen

**Lernziele und Kompetenzen:**

Die Studierenden

- nennen und erklären die Grundprinzipien der linearen Funktionalanalysis und verwenden diese;
- kennen und erklären die Topologien von Hilbert- und Banachräumen, weisen Konvergenz von Folgen in unterschiedlichen Topologien nach (stark, schwach) und zeigen Implikationen aus kompakten Einbettungen auf;
- beweisen Aussagen zu Existenz und Eindeutigkeit von Lösungen linearer Operatorgleichungen und zeigen insbesondere die Existenz schwacher Lösungen zu Randwertproblemen bei linearen elliptischen Differentialgleichungen;
- treffen Aussagen zur Integrierbarkeit bzw. Glattheit von Sobolev-Funktionen.

**Literatur:**

- Vorlesungsskripte zu diesem Modul
- H.W. Alt: Lineare Funktionalanalysis; Springer-Verlag
- D. Werner: Funktionalanalysis; Springer-Verlag

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

**[1] Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Nebenfach | Nebenfach Mathematik)

Dieses Modul ist daneben auch in den Studienfächern "Informatik (Master of Science)", "Mathematik (Bachelor of Science)", "Technomathematik (Bachelor of Science)", "Wirtschaftsmathematik (Bachelor of Science)" verwendbar.

---

**Studien-/Prüfungsleistungen:**

Klausur: Funktionalanalysis (Prüfungsnummer: 51101)

Prüfungsleistung, Klausur, Dauer (in Minuten): 90

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Funktionalanalysis I

Erstablingung: SS 2019, 1. Wdh.: WS 2019/2020

1. Prüfer: Hermann Schulz-Baldes (060126)

Übungsleistung: Funktionalanalysis (Prüfungsnummer: 51102)

Studienleistung, Übungsleistung

weitere Erläuterungen:

Erfolgreiche Bearbeitung wöchentlicher Hausaufgaben

Erstablingung: SS 2019, 1. Wdh.: WS 2019/2020

1. Prüfer: Hermann Schulz-Baldes (060126)

---

**Organisatorisches:**

Die Präsentation des Stoffes erfolgt in Vorlesungsform. Die weitere Aneignung der wesentlichen Begriffe und Techniken erfolgt durch wöchentliche Hausaufgaben.

**Bemerkungen:**

Wahlpflichtmodul (Vertiefungsmodul) in

- B. Sc. Mathematik, Technomathematik und Wirtschaftsmathematik

---

**Modulbezeichnung: Funktionentheorie I (FThI)** **5 ECTS**  
(Complex Analysis I)

Modulverantwortliche/r: Hermann Schulz-Baldes

Lehrende: Yasmine Sanderson

---

Startsemester: SS 2019

Dauer: 1 Semester

Turnus: jährlich (SS)

Präsenzzeit: 45 Std.

Eigenstudium: 105 Std.

Sprache: Deutsch

---

**Lehrveranstaltungen:**

Funktionentheorie (SS 2019, Vorlesung, 2 SWS, Yasmine Sanderson)

Übungen zur Funktionentheorie (SS 2019, Übung, 2 SWS, Yasmine Sanderson)

---

**Empfohlene Voraussetzungen:**

Analysis I und II

---

**Inhalt:**

Grundlagen zu folgenden Themen:

- Holomorphe Abbildungen
- Cauchy-Riemann'sche Differentialgleichungen
- Wegintegrale und der Cauchy'sche Integralsatz
- Satz von Liouville
- Laurent-Reihen
- Residuenkalkül

**Lernziele und Kompetenzen:**

Die Studierenden

- erklären die Grundprinzipien der Funktionentheorie und wenden diese an;
- erkennen die Gemeinsamkeiten und Unterschiede zwischen reell und komplex differenzierbaren Funktionen und erklären diese;
- wenden komplex-analytische Methoden zur Lösung von Problemen der reellen Analysis selbständig an.

**Literatur:**

- Freitag, Busam: Funktionentheorie I
- Remmert: Funktionentheorie

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

**[1] Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Nebenfach | Nebenfach Mathematik)

Dieses Modul ist daneben auch in den Studienfächern "Informatik (Master of Science)", "Mathematik (1. Staatsprüfung für das Lehramt an Gymnasien)", "Mathematik (Bachelor of Science)", "Wirtschaftsmathematik (Bachelor of Science)" verwendbar.

---

**Studien-/Prüfungsleistungen:**

Funktionentheorie (Prüfungsnummer: 53511)

Prüfungsleistung, Klausur mit Übungsleistung, Dauer (in Minuten): 60

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Funktionentheorie
- Übungen zur Funktionentheorie

Erstabelleung: SS 2019, 1. Wdh.: SS 2019

1. Prüfer: Yasmine Sanderson (060114)

**Organisatorisches:**

Die Präsentation des Stoffes erfolgt in Vorlesungsform. Die weitere Aneignung der wesentlichen Begriffe und Techniken erfolgt durch wöchentliche Hausaufgaben.

**Bemerkungen:**

B. Sc. Mathematik, Technomathematik, Wirtschaftsmathematik, vertieftes Lehramt Mathematik

---

**Modulbezeichnung: Geometrie (Geom)** **5 ECTS**  
 (Geometry)

Modulverantwortliche/r: Friedrich Knop, Karl-Hermann Neeb

Lehrende: Yasmine Sanderson

---

Startsemester: WS 2019/2020	Dauer: 1 Semester	Turnus: jährlich (SS)
Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Deutsch

---

### Lehrveranstaltungen:

Diese Veranstaltung ist sowohl "Vertiefungsmodul Geometrie" bzw. "Geometrie" im Bachelorstudiengang, also auch "Vertiefte Geometrie (Geom)" fuer das Lehramt.

---

### Empfohlene Voraussetzungen:

die Module Lineare Algebra und Analysis

---

### Inhalt:

(dieser Modul wird mit wechselnden Schwerpunkten angeboten):

- A) Euklidische, hyperbolische, sphärische und projektive Geometrie (Symmetriegruppen geometrischer Strukturen, Invarianten, Geodäten, Dreiecke, Krümmung)
- B) Elementare Differentialgeometrie: Kurventheorie (ebene Kurven, Raumkurven), Flächentheorie (Fundamentalformen, Krümmung, Integration, spezielle Klassen, Riemannsche Metriken)
- C) Algebraische Geometrie: Kommutative Algebra, Nullstellensatz, Affine Varietäten, Projektive Varietäten, Normalisierung, Singularitäten, Algebraische Gruppen

### Lernziele und Kompetenzen:

Die Studierenden

- wenden Methoden einer der Vertiefungsrichtungen der Geometrie an
- analysieren konkrete Beispiele systematisch und behandeln diese im Rahmen der allgemeinen Theorie.

### Literatur:

werden in der Vorlesung bekannt gegeben

---

### Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

#### [1] Informatik (Bachelor of Science)

(Po-Vers. 2009s | Nebenfach | Nebenfach Mathematik)

Dieses Modul ist daneben auch in den Studienfächern "Informatik (Master of Science)", "Mathematik (1. Staatsprüfung für das Lehramt an Gymnasien)", "Mathematik (Bachelor of Science)", "Technomathematik (Bachelor of Science)", "Wirtschaftsmathematik (Bachelor of Science)" verwendbar.

---

### Organisatorisches:

- Vertiefungsmodul im Bachelorstudiengang Mathematik
- *Modul Geom (vertiefte Geometrie) im Lehramt*

---

**Modulbezeichnung: Topologie (Top)** **5 ECTS**  
 (Topology)

Modulverantwortliche/r: Karl-Hermann Neeb  
 Lehrende: Karl-Hermann Neeb, Catherine Meusburger

---

Startsemester: SS 2019	Dauer: 1 Semester	Turnus: jährlich (SS)
Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Deutsch

---

**Lehrveranstaltungen:**

Topologie (Querschnittmodul) (SS 2019, Vorlesung, 4 SWS, Karl-Hermann Neeb et al.)  
 Übungen zu Topologie (Querschnittmodul) (SS 2019, Übung, 2 SWS, Karl-Hermann Neeb et al.)

---

**Inhalt:**

- Stetige Funktionen, Zusammenhang, Trennungsaxiome
- Erzeugung von Topologien (initiale, finale, Quotienten etc.)
- Konvergenz in topologischen Räumen (Filter, Netze)
- Kompaktheit (Satz von Tychonov, kompakte metrische Räume, lokalkompakte Räume)
- Anwendung auf Funktionenräume (Satz von Stone-Weierstraß, Satz von Ascoli)
- Überlagerungstheorie (Fundamentalgruppen, Hochhebung von Abbildungen)

**Lernziele und Kompetenzen:**

- Die Studierenden
- wenden die Methoden der allgemeinen Topologie, die in den Grundvorlesungen nur am Rande vorkommt, an;
  - ordnen die topologischen Grundbegriffe in einen größeren Kontext ein;
  - erklären und verwenden wichtige Resultate, die in vielen Bereichen der Mathematik zum Handwerkzeug gehören.

**Literatur:**

- Vorlesungsskript zu diesem Modul
  - Bredon: Geometry and Topology
  - Skript auf StudOn bereitgestellt und auch unter [www.studium.math.fau.de/lehrveranstaltungen/skripten.html](http://www.studium.math.fau.de/lehrveranstaltungen/skripten.html)
- 

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

**[1] Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Nebenfach | Nebenfach Mathematik)

Dieses Modul ist daneben auch in den Studienfächern "Informatik (Master of Science)", "Mathematik (1. Staatsprüfung für das Lehramt an Gymnasien)", "Mathematik (Bachelor of Science)", "Technomathematik (Bachelor of Science)", "Wirtschaftsmathematik (Bachelor of Science)" verwendbar.

---

**Studien-/Prüfungsleistungen:**

Modulabschlussprüfung: Geometrie (Prüfungsnummer: 56211)  
 Prüfungsleistung, Klausur mit Übungsleistung, Dauer (in Minuten): 60  
 Anteil an der Berechnung der Modulnote: 100%  
 Zugeordnete Lehrveranstaltungen:

- Topologie (Querschnittmodul)
- Übungen zu Topologie (Querschnittmodul)

Prüfungssprache: Deutsch

Erstabledung: SS 2019, 1. Wdh.: SS 2019  
 1. Prüfer: Karl-Hermann Neeb (060160)





---

**Modulbezeichnung:** **Gewöhnliche Differentialgleichungen (GDgl)** **10 ECTS**  
 (Ordinary Differential Equations)

Modulverantwortliche/r: Andreas Knauf  
 Lehrende: Andreas Knauf

---

Startsemester: SS 2019	Dauer: 1 Semester	Turnus: jährlich (SS)
Präsenzzeit: 90 Std.	Eigenstudium: 210 Std.	Sprache: Deutsch

---

**Lehrveranstaltungen:**

Gewöhnliche Differentialgleichungen (SS 2019, Vorlesung, 4 SWS, Andreas Knauf)  
 Übungen zu Gewöhnliche Differentialgleichungen (SS 2019, Übung, 2 SWS, Andreas Knauf)

---

**Empfohlene Voraussetzungen:**

Analysis 1 und 2, Lineare Algebra 1 und 2

---

**Inhalt:**

Grundlagen zu folgenden Themen:

- Typen von Differentialgleichungen und elementare Lösungsmethoden
- Existenz-, Eindeutigkeits- und Stetigkeitssätze für das Anfangswertproblem
- Differentialungleichungen (Lemma von Gronwall)
- Fortsetzung von Lösungen
- lineare und gestörte lineare Systeme
- autonome Systeme und Flüsse
- Stabilität
- Randwertprobleme

**Lernziele und Kompetenzen:**

Die Studierenden

- lösen einfache, insbesondere autonome lineare Differentialgleichungen selbständig;
- erklären und prüfen qualitative Eigenschaften wie Stabilität;
- wenden die relevanten Lösungsmethoden selbstständig an;
- klassifizieren konkrete Probleme und setzen theoretische Modelle zur Behandlung ein;
- überführen die Prinzipien in allgemeineren oder auch einfacheren Kontext.

**Literatur:**

- Vorlesungsskripte zu diesem Modul
- H. Amann: Gewöhnliche Differentialgleichungen. de Gruyter
- V.I. Arnol'd: Gewöhnliche Differentialgleichungen. Springer
- H. Heuser: Gewöhnliche Differentialgleichungen. Teubner
- W. Walter: Gewöhnliche Differentialgleichungen. Springer

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

**[1] Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Nebenfach | Nebenfach Mathematik)

Dieses Modul ist daneben auch in den Studienfächern "Informatik (Master of Science)", "Mathematik (Bachelor of Science)", "Technomathematik (Bachelor of Science)", "Wirtschaftsmathematik (Bachelor of Science)" verwendbar.

---

**Studien-/Prüfungsleistungen:**

Klausur: Gewöhnliche Differentialgleichungen (Prüfungsnummer: 51001)

Prüfungsleistung, Klausur, Dauer (in Minuten): 90

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Gewöhnliche Differentialgleichungen
- Übungen zu Gewöhnliche Differentialgleichungen

Erstablingung: SS 2019, 1. Wdh.: WS 2019/2020

1. Prüfer: Andreas Knauf (060107)

Übungsleistung: Gewöhnliche Differentialgleichungen (Prüfungsnummer: 51002)

Studienleistung, Übungsleistung

Zugeordnete Lehrveranstaltungen:

- Übungen zu Gewöhnliche Differentialgleichungen

weitere Erläuterungen:

erfolgreiche Bearbeitung woechentlicher Hausaufgaben

Erstablingung: SS 2019, 1. Wdh.: keine Angabe

1. Prüfer: Andreas Knauf (060107)

---

**Organisatorisches:**

Wahlpflichtmodul in

- B. Sc. Mathematik, Technomathematik und Wirtschaftsmathematik
- Lehramt vertieft

Hörerinnen und Hörer aus verwandten Studiengängen sind willkommen

**Bemerkungen:**

Die Präsentation des Stoffes erfolgt in der Vorlesung und der integrierten zentralen Übung. Die weitere Aneignung der wesentlichen Begriffe und Techniken erfolgt durch wöchentliche Hausaufgaben.

---

**Modulbezeichnung:** Introduction to Statistics and Statistical Programming (Stat) 5 ECTS  
 (Introduction to Statistics and Statistical Programming)

Modulverantwortliche/r: Christoph Richard  
 Lehrende: Christoph Richard

---

Startsemester: SS 2019	Dauer: 1 Semester	Turnus: jährlich (SS)
Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Deutsch

---

**Lehrveranstaltungen:**

- Introduction to Statistics and Statistical Programming (SS 2019, Vorlesung, 2 SWS, Christoph Richard)
- Computer lab classes "Introduction to Statistics and Statistical Programming" (SS 2019, Übung, 1 SWS, Christoph Richard)
- Review session "Introduction to Statistics and Statistical Programming" (SS 2019, Tutorium, 1 SWS, Christoph Richard)
- Problem session "Introduction to Statistics and Statistical Programming" (SS 2019, Übung, 1 SWS, Christoph Richard)

---

**Empfohlene Voraussetzungen:**

Keine.  
 Empfohlen wird die Teilnahme am Modul Stochastische Modellbildung im gleichen oder in einem vorherigen Semester.

---

**Inhalt:**

- Einführung in die Statistik-Software R und elementares Programmieren
- Beschreibende Statistik: Visualisierung und Kenngrößen kategoriieller und metrischer Daten, qq-Plots, Kurvenanpassung, log- und loglog- Plots, robuste Verfahren
- Schließende Statistik: Schätz- und Testverfahren: parametrische Tests, ausgewählte nichtparametrische Tests, exakte und asymptotische Konfidenzintervalle
- Simulation: Zufallszahlen, Monte-Carlo

**Lernziele und Kompetenzen:**

- Die Studierenden können
- die grundlegenden Verfahren der beschreibenden und schließenden Statistik beschreiben und erläutern;
  - in einer Gruppe ihren Lösungsvorschlag für ein nicht triviales statistisches Problem angemessen erklären und alternative Lösungsvorschläge kritisch vergleichen;
  - statistische Standard-Auswertungen innerhalb eines vorgegebenen Zeitrahmens mit dem Computer bearbeiten und dessen Ausgaben richtig interpretieren;
  - einfache statistische Simulationen durchführen;
  - zu einem Datensatz angemessene Fragen formulieren, adäquate statistische Verfahren zur Beantwortung dieser Fragen wählen und solche Verfahren am Computer umsetzen.

**Literatur:**

- Vorlesungsskript zu diesem Modul
- Rice: Mathematical Statistics and Data Analysis; Thomson, 2007
- [www.cran.r-project.org](http://www.cran.r-project.org)

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

[1] **Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Nebenfach | Nebenfach Mathematik)

Dieses Modul ist daneben auch in den Studienfächern "Informatik (Master of Science)", "Integrated Life Sciences: Biologie, Biomathematik, Biophysik (Master of Science)", "Integrated Life Sciences: Biology, Bioma-

thematics, Biophysics (Master of Science)", "Mathematik (Bachelor of Science)", "Technomathematik (Bachelor of Science)", "Wirtschaftsmathematik (Bachelor of Science)" verwendbar.

---

### **Studien-/Prüfungsleistungen:**

Introduction to Statistics and Statistical Programming (Prüfungsnummer: 80711)

Prüfungsleistung, Klausur, Dauer (in Minuten): 90

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Introduction to Statistics and Statistical Programming

Erstablingung: SS 2019, 1. Wdh.: WS 2019/2020

1. Prüfer: Christoph Richard (060153)

Introduction to Statistics and Statistical Programming (Prüfungsnummer: 80712)

Studienleistung, Übungsleistung

Zugeordnete Lehrveranstaltungen:

- Computer lab classes "Introduction to Statistics and Statistical Programming"
- Problem session "Introduction to Statistics and Statistical Programming"

weitere Erläuterungen:

Bearbeitung wöchentlicher Übungsaufgaben

Erstablingung: SS 2019, 1. Wdh.: keine Angabe

1. Prüfer: Christoph Richard (060153)

---

### **Organisatorisches:**

Theoretische Grundlagen des Stoffes werden in Vorlesungsform, der Umgang mit der Statistik-Software R in den Rechnerübungen vermittelt. Dies wird ergänzt durch wöchentliche Hausaufgaben, welche theoretische Fragestellungen und praktische Probleme beinhalten, die mit R zu lösen sind. Die Hausaufgaben werden in Übungsgruppen diskutiert.

### **Bemerkungen:**

Pflichtmodul oder Schlüsselqualifikation (unbenotet) in

- B. Sc. Wirtschaftsmathematik

Wahlpflichtmodul oder Schlüsselqualifikation (unbenotet) in

- B. Sc. Mathematik

**Modulbezeichnung:** **Mathematische Modellierung Praxis (MaMoPra)** **5 ECTS**  
(Mathematical Modelling in Practice)

Modulverantwortliche/r: Serge Kräutle

Lehrende: Nadja Ray, Maria Neuss-Radu, Serge Kräutle

Startsemester: WS 2019/2020

Dauer: 1 Semester

Turnus: jährlich (WS)

Präsenzzeit: 30 Std.

Eigenstudium: 120 Std.

Sprache: Deutsch

**Lehrveranstaltungen:**

Mathematische Modellierung Praxis (WS 2019/2020, Praktikum, 2 SWS, Nadja Ray et al.)

**Empfohlene Voraussetzungen:**

Die Teilnahme am Modul ist nur in Kombination mit dem Modul Mathematische Modellierung Theorie möglich.

Voraussetzung sind die Module Analysis und Lineare Algebra oder Module einer zwei-semesterigen Mathematikgrundausbildung für nicht-mathematische Studiengänge.

Es wird empfohlen, die Module Numerische Mathematik und Gewöhnliche Differentialgleichungen abgelegt zu haben.

**Inhalt:**

Analyse und Lösung von Problemen aus Ingenieur- und Naturwissenschaften (u.a. Mechanik, Life Sciences)

**Lernziele und Kompetenzen:**

Die Studierenden:

- bearbeiten Modellierungsprojekte im Team;
- modellieren Alltagsprobleme, lösen sie mit analytischen / numerischen Methoden und diskutieren die Ergebnisse kritisch;
- prägen Problemlösungskompetenz aus;
- erwerben Schlüsselkompetenzen: prägen durch die Projektarbeit

Teammanagement aus, sind durch Berichterstattung in den Projekten zu Vortragspräsentation und wissenschaftlichem Schreiben befähigt.

**Literatur:**

- Ch. Eck, H. Garcke, P. Knabner: Mathematische Modellierung; Springer-Verlag, 2. Auflage, Berlin, 2011
- F. Hauser, Y. Luchko: Mathematische Modellierung mit MATLAB; Spektrum Akademischer Verlag, 2011
- G. Strang: Introduction to Applied Mathematics; Wellesley-Cambridge Press, Wellesley 1986

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

**[1] Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Nebenfach | Nebenfach Mathematik)

Dieses Modul ist daneben auch in den Studienfächern "Informatik (Master of Science)", "Technomathematik (Bachelor of Science)" verwendbar.

**Studien-/Prüfungsleistungen:**

Modulabschlussprüfung: Mathematische Modellierung Praktikum (Prüfungsnummer: 52531)

Studienleistung, Projekt-/Praktikumsbericht

Zugeordnete Lehrveranstaltungen:

- Mathematische Modellierung Praxis

Erstablingung: WS 2019/2020, 1. Wdh.: keine Angabe  
1. Prüfer: Nadja Ray (060223)

---

**Modulbezeichnung:** **Mathematische Modellierung Theorie (MaMoThe)** **5 ECTS**  
(Mathematical Modelling Theory)

Modulverantwortliche/r: Serge Kräutle

Lehrende: Nadja Ray

Startsemester: WS 2019/2020	Dauer: 1 Semester	Turnus: halbjährlich (WS+SS)
Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Deutsch

**Lehrveranstaltungen:**

Mathematische Modellierung Theorie (WS 2019/2020, Vorlesung, 2 SWS, Nadja Ray)

Übungen zur Mathematische Modellierung Theorie (WS 2019/2020, Übung, 2 SWS, Nadja Ray et al.)

**Empfohlene Voraussetzungen:**

- Module Analysis und Lineare Algebra oder Module einer zweisemestrigen Mathematikgrundausbildung für nichtmathematische Studiengänge.
- Modul Numerische Mathematik empfohlen.
- Modul Gewöhnliche Differentialgleichungen empfohlen.

**Inhalt:**

- Handwerkzeuge der mathematischen Modellierung: Dimensionsanalyse, asymptotische Entwicklung, Stabilitäts-, Sensitivitätsbetrachtungen, Existenz und Nichtnegativität von Lösungen
- Modelle in Form von linearen Gleichungssystemen (elektrische Netzwerke, Stabwerke, Zusammenhang zu Minimierungsaufgaben), nichtlinearen Gleichungssystemen (chemische Gleichgewichte in reaktiven Mehrspeziessystemen), Anfangswertaufgaben für gewöhnliche Differentialgleichungen (chemische Reaktionen, Populationsmodelle)

**Lernziele und Kompetenzen:**

Die Studierenden:

- nennen und erklären die grundlegenden und vertiefenden Begriffe mathematischer Modellierung und verwenden die zugehörigen Prinzipien;
- erstellen und bewerten, auf Basis exemplarischer Kenntnisse aus Ingenieur- und Naturwissenschaften, deterministische Modelle in Form von Gleichungssystemen und gewöhnlichen Differentialgleichungen selbständig;
- lösen vorgegebene Aufgaben im Team mit analytischen / numerischen Methoden und diskutieren die Ergebnisse kritisch;
- prägen Problemlösungskompetenz aus.

**Literatur:**

- Ch. Eck, H. Garcke, P. Knabner "Mathematische Modellierung". Springer-Verlag, 2. Auflage, Berlin 2011
- F. Hauser, Y. Luchko, "Mathematische Modellierung mit MATLAB". Spektrum Akademischer Verlag 2011
- G. Strang "Introduction to Applied Mathematics". Wellesley-Cambridge Press, Wellesley 1986

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

**[1] Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Nebenfach | Nebenfach Mathematik)

Dieses Modul ist daneben auch in den Studienfächern "Informatik (Master of Science)", "Technomathematik (Bachelor of Science)" verwendbar.

**Studien-/Prüfungsleistungen:**



Modulabschluss mündliche Prüfung: Mathematische Modellierung Theorie (Prüfungsnummer: 52522)

Prüfungsleistung, mündliche Prüfung, Dauer (in Minuten): 15

Anteil an der Berechnung der Modulnote: 50%

Zugeordnete Lehrveranstaltungen:

- Mathematische Modellierung Theorie
- Übungen zur Mathematische Modellierung Theorie

Erstablingung: WS 2019/2020, 1. Wdh.: keine Angabe

1. Prüfer: Nadja Ray (060223)

Modulabschluss Projektbericht mit Vortrag: Mathematische Modellierung Theorie (Prüfungsnummer: 52521)

Prüfungsleistung, Projekt-/Praktikumsbericht

Anteil an der Berechnung der Modulnote: 50%

Zugeordnete Lehrveranstaltungen:

- Mathematische Modellierung Theorie
- Übungen zur Mathematische Modellierung Theorie

Erstablingung: WS 2019/2020, 1. Wdh.: keine Angabe

1. Prüfer: Nadja Ray (060223)

---

**Organisatorisches:**

Pflichtmodul in

- B. Sc. Technomathematik

Wahlpflichtmodul in

- B. Sc. Mathematik und Wirtschaftsmathematik

**Bemerkungen:**

Die Präsentation des Stoffes erfolgt in Vorlesungsform. Die weitere Aneignung der wesentlichen Begriffe und Techniken erfolgt in Projektarbeit in Kleingruppen, die in den Übungen besprochen werden.

---

**Modulbezeichnung:** **Nichtlineare Optimierung (NOpt)** **10 ECTS**  
 (Nonlinear Optimization)

Modulverantwortliche/r: Wolfgang Achtziger  
 Lehrende: Michael Stingl

---

Startsemester: WS 2019/2020	Dauer: 1 Semester	Turnus: jährlich (WS)
Präsenzzeit: 90 Std.	Eigenstudium: 210 Std.	Sprache:

---

**Lehrveranstaltungen:**

Nichtlineare Optimierung (WS 2019/2020, Vorlesung, 4 SWS, Michael Stingl)  
 Übungen zu Nichtlinearer Optimierung (WS 2019/2020, Übung, 2 SWS, Michael Stingl)

---

**Empfohlene Voraussetzungen:**

Abschluss der Module Analysis I, Analysis II, Lineare Algebra I und Lineare Algebra II. Hilfreich ist der Abschluss der Module Lineare und Kombinatorische Optimierung und Numerische Mathematik.

---

**Inhalt:**

- Unrestringierte Probleme der Nichtlinearen Optimierung (Optimalitätsbedingungen, Abstiegsverfahren, Verfahren der konjugierten Richtungen, Variable-Metrik-Methoden und Quasi-Newton-Methoden)
- Restringierte Probleme der Nichtlinearen Optimierung (Optimalitätsbedingungen)

**Lernziele und Kompetenzen:**

Die Studierenden

- nennen und erklären Grundbegriffe der Nichtlinearen Optimierung;
- modellieren und lösen praxisrelevante Probleme mit Hilfe der erlernten Verfahren;
- sammeln und bewerten relevante Informationen und stellen Zusammenhänge her.

**Literatur:**

- Geiger, Ch. Kanzow: Numerische Verfahren zur Lösung unrestringierter Optimierungsaufgaben; Springer, 1999
- Geiger, Ch. Kanzow: Theorie und Numerik restringierter Optimierungsaufgaben; Springer, 2002
- W. Alt: Nichtlineare Optimierung; Vieweg, 2002
- F. Jarre und J. Stoer: Optimierung; Springer, 2004
- M.S. Bazaraa, H.D. Sherali, C.M. Shetty: Nonlinear Programming - Theory and Algorithms; Wiley, New York, 1993

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

**[1] Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Nebenfach | Nebenfach Mathematik)

Dieses Modul ist daneben auch in den Studienfächern "Informatik (Master of Science)", "Mathematik (Bachelor of Science)", "Technomathematik (Bachelor of Science)", "Wirtschaftsmathematik (Bachelor of Science)" verwendbar.

---

**Studien-/Prüfungsleistungen:**

Übungsleistung: Nichtlineare Optimierung (Prüfungsnummer: 51502)

(englische Bezeichnung: Tutorial achievement: Nonlinear optimisation)

Studienleistung, Übungsleistung

Zugeordnete Lehrveranstaltungen:

- Übungen zu Nichtlinearer Optimierung

weitere Erläuterungen:

Hausaufgaben (wöchentlich ein Übungsblatt)

Erstabledung: WS 2019/2020, 1. Wdh.: keine Angabe

1. Prüfer: Michael Stingl (060221)

Klausur: Nichtlineare Optimierung (Prüfungsnummer: 51501)

Prüfungsleistung, Klausur, Dauer (in Minuten): 90

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Nichtlineare Optimierung

Erstabledung: WS 2019/2020, 1. Wdh.: SS 2020

1. Prüfer: Michael Stingl (060221)

---

**Organisatorisches:**

Die Präsentation des Stoffes erfolgt in Vorlesungsform. Die weitere Aneignung der wesentlichen Begriffe und Techniken erfolgt durch wöchentliche Hausaufgaben, die in den Übungen besprochen werden.

**Bemerkungen:**

Wahlpflichtmodul im Modulblock

- Angewandte Mathematik des Bachelor-Studiengangs Mathematik
- PSO des Bachelor-Studiengangs Technomathematik
- WM des Bachelor-Studiengangs Wirtschaftsmathematik

---

**Modulbezeichnung: Numerische Mathematik (NumMath)** **10 ECTS**  
 (Numerical Mathematics)

Modulverantwortliche/r: Peter Knabner  
 Lehrende: Florian Frank

---

Startsemester: WS 2019/2020	Dauer: 1 Semester	Turnus: jährlich (WS)
Präsenzzeit: 105 Std.	Eigenstudium: 195 Std.	Sprache:

---

**Lehrveranstaltungen:**

- Einführung in die Numerik (= Numerische Mathematik) (WS 2019/2020, Vorlesung, 4 SWS, Florian Frank)
- Übungen zur Einführung in die Numerik (WS 2019/2020, Übung, 2 SWS, Florian Frank)
- Tutorium zur Einführung in die Numerik (WS 2019/2020, Übung, 1 SWS, Florian Frank)
- Matlab-Kurs zur Einführung in die Numerik (WS 2019/2020, Übung, Florian Frank)

---

**Empfohlene Voraussetzungen:**

- Die Module Analysis, Lineare Algebra
- Kenntnisse in MATLAB sind zwingend. Diese können in einem jeweils vor Semesterbeginn stattfindenden Kurs erworben werden.

**Vorhergehende Module:**

- Analysis II
- Lineare Algebra II
- Analysis I
- Lineare Algebra I

---

**Inhalt:**

- Direkte Eliminationsverfahren für lineare Gleichungssysteme [Gauß mit Pivotsuche (Erinnerung), Cholesky, LR-Zerlegung für vollbesetzte (Erinnerung) Bandmatrizen]
- Linear stationäre iterative Verfahren: Erinnerung und SOR-Verfahren
- Verfahren für Eigenwertaufgaben (QR-Verfahren)
- Fehleranalyse und Störungsrechnung (Gleitpunktarithmetik, Konditionsanalyse, schlechtgestellte Probleme)
- Lineare Ausgleichsrechnung (Orthogonalisierungsverfahren, Numerik der Pseudoinverse)
- Iterative Verfahren für nicht-lineare Gleichungssysteme (Fixpunktiteration, Newton-Verfahren, Gauß-Newton)
- Interpolation (Polynome, Polynomialsplines, FFT)
- Numerische Integration (Newton-Cotes, Gauß, Extrapolation, Adaption)

**Lernziele und Kompetenzen:**

Die Studierenden

- verwenden algorithmische Zugänge für Probleme der linearen Algebra und Analysis und erklären und bewerten diese;
- urteilen insbesondere über die Stabilität und Effizienz eines numerischen Verfahrens;
- setzen mit eigener oder gegebener Software Verfahren um und bewerten deren Ergebnisse kritisch;
- erläutern und verwenden ein breites Problem- und Verfahrensspektrum: (Direkte und) iterative Verfahren für lineare Gleichungssysteme, nichtlineare Gleichungssysteme, insbesondere Newton-Verfahren, (nicht)lineare Ausgleichsrechnung, Interpolation und Integration, Numerik von Eigenwertaufgaben;
- sammeln und bewerten relevante Informationen und erkennen Zusammenhänge.

**Literatur:**

- R. Schaback und H. Wendland: Numerische Mathematik; Springer, Berlin, 2005
- A. Quarteroni, R. Sacco, F. Saleri: Numerische Mathematik I, II; Springer, Berlin, 2002
- P. Deuffhard und A. Hohmann: Numerische Mathematik I; de Gruyter, Berlin 2002
- J. Stoer: Numerische Mathematik I; Springer, Berlin, 2005

- J. Stoer und R. Bulirsch: Numerische Mathematik I; Springer, Berlin, 2005
- Vorlesungsskript auf der Homepage des Bereichs Modellierung, Simulation und Optimierung des Departments Mathematik, ständig neu an die Vorlesung angepasst

---

### Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

#### [1] Informatik (Bachelor of Science)

(Po-Vers. 2009s | Nebenfach | Nebenfach Mathematik)

Dieses Modul ist daneben auch in den Studienfächern "Informatik (Master of Science)", "Mathematik (Bachelor of Science)", "Technomathematik (Bachelor of Science)", "Wirtschaftsmathematik (Bachelor of Science)" verwendbar.

---

### Studien-/Prüfungsleistungen:

Klausur: Numerische Mathematik (Prüfungsnummer: 51301)

Prüfungsleistung, Klausur, Dauer (in Minuten): 90

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Einführung in die Numerik (= Numerische Mathematik)

Erstablingung: WS 2019/2020, 1. Wdh.: SS 2020

1. Prüfer: Florian Frank (060256)

Übungsleistung: Numerische Mathematik (Prüfungsnummer: 51302)

Studienleistung, Übungsleistung

Zugeordnete Lehrveranstaltungen:

- Übungen zur Einführung in die Numerik
- Tutorium zur Einführung in die Numerik
- Matlab-Kurs zur Einführung in die Numerik

weitere Erläuterungen:

Hausaufgaben (wöchentlich ein Übungsblatt)

Erstablingung: WS 2019/2020, 1. Wdh.: keine Angabe

1. Prüfer: Florian Frank (060256)

---

### Organisatorisches:

Die Präsentation des Stoffes erfolgt in Vorlesungsform. Die weitere Aneignung der wesentlichen Begriffe und Techniken erfolgt durch wöchentliche Hausaufgaben. Im Tutorium wird Softwareerstellung und -bewertung eingeübt.

### Bemerkungen:

Wahlpflichtmodul in

- B. Sc. Mathematik, Technomathematik und Wirtschaftsmathematik

**Modulbezeichnung:** **Optimierung für Ingenieure (OptIngV)** **5 ECTS**  
(Optimization for Engineers)

Modulverantwortliche/r: Johannes Hild, Martin Gugat

Lehrende: Johannes Hild, Martin Gugat

Startsemester: SS 2019

Dauer: 1 Semester

Turnus: jährlich (SS)

Präsenzzeit: 45 Std.

Eigenstudium: 105 Std.

Sprache: Englisch

**Lehrveranstaltungen:**

Optimierung für Ingenieure (Optimization for Engineers) (SS 2019, Vorlesung mit Übung, 3 SWS, Johannes Hild)

**Empfohlene Voraussetzungen:**

Requires contents of the lecture Mathematics for Engineers I, II and III. Especially:

- Linear algebra
- Analysis of real valued functions
- Differential and integral calculus in multi dimensional spaces

**Inhalt:**

**Introduction to continuous optimization problems and methods with and without constraints**

- Classification of problem types
- Optimality conditions and termination criterions
- Descent directions and line search methods
- Convergence analysis

**Unconstrained optimization**

- Steepest descent and conjugate gradient
- Newton-type methods
- Nonlinear Least Squares

**Constrained optimization**

- Projection methods
- Trust Region
- Barrier and penalty methods

**Outlook**

- Linear programming and simplex method
- Integer programming
- Noisy functions

**Lernziele und Kompetenzen:**

*Fachkompetenz*

*Wissen*

- Students list requirements, strengths and weaknesses of common optimization methods.
- Students recognize crucial components in existence and convergence proofs in the context of minimizing sequences.

*Verstehen*

- Students explain the different components of optimization methods.
- Students describe the relationship between requirements and conclusions of existence and convergence theorems in the context of minimizing sequences.

*Anwenden*

- Students check feasibility, well-posedness and constraint qualifications of optimization problems.
- Students formulate and solve optimality conditions analytically.
- Students apply optimization algorithms to optimization problems.

*Analysieren*

- Students analyse uncommon optimization approaches and extract their requirements, strengths and weaknesses.

### *Evaluieren (Beurteilen)*

- Students evaluate the class and structure of unsolved optimization problems.
- Students choose suitable algorithmic approaches for unsolved optimization problems.

### *Erschaffen*

- Students formulate optimization problems using mathematical methods and structures.
- Students modify and combine common optimization routines to create project-specific algorithms for unsolved optimization problems.

### **Literatur:**

Nocedal, Jorge and Wright, Stephen J.: Numerical Optimization. Springer Serie in Operations Research, 2006.  
 Kelley, C. T.: Iterative Methods for Optimization. Frontiers in Applied Mathematics 18, SIAM Philadelphia 1999;  
 Polak, E.: Optimization. Algorithms and Consistent Approximations. Applied Mathematical Sciences, Volume 124, Springer-Verlag New York, 1997.  
 Jarre, F.: Optimierung, Springer 2003;  
 Hamacher, H.W. and K. Klamroth, K.: Linear and Network Optimization: bilingual textbook. Vieweg 2000

### **Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

#### [1] **Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Nebenfach | Nebenfach Mathematik)

Dieses Modul ist daneben auch in den Studienfächern "Chemical Engineering - Nachhaltige Chemische Technologien (Master of Science)", "Chemie- und Bioingenieurwesen (Master of Science)", "Computational Engineering (Master of Science with Honours)", "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Master of Science)", "Informatik (Master of Science)", "Life Science Engineering (Master of Science)", "Medizintechnik (Master of Science)" verwendbar.

### **Studien-/Prüfungsleistungen:**

Optimierung für Ingenieure (Prüfungsnummer: 40501)

(englische Bezeichnung: Optimisation for Engineers)

Prüfungsleistung, Klausur, Dauer (in Minuten): 90

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Optimierung für Ingenieure (Optimization for Engineers)

Prüfungssprache: Englisch

Erstablingung: SS 2019, 1. Wdh.: WS 2019/2020

1. Prüfer: Johannes Hild (060100)

### **Bemerkungen:**

This module aims at students of the Faculty of Engineering of all disciplines and is suitable as an elective subject in the Bachelor's and Master's degree.

---

**Modulbezeichnung:** Graph Routing and applications (GrRaA) 5 ECTS  
 (Graph Routing and applications)

Modulverantwortliche/r: Alexander Martin

Lehrende: Francisco Javier Zaragoza Martínez

Startsemester: WS 2019/2020

Dauer: 1 Semester

Turnus: jährlich (WS)

Präsenzzeit: 36 Std.

Eigenstudium: 114 Std.

Sprache: Englisch

---

**Lehrveranstaltungen:**

Graph Routing and applications (WS 2019/2020, Vorlesung, 4 SWS, Francisco Javier Zaragoza Martínez)

Tutorial to Graph Routing and applications (WS 2019/2020, Übung, 2 SWS, Francisco Javier Zaragoza Martínez)

---

**Inhalt:**

The origins of graph theory are usually traced back to Euler's solution to the famous amusement known as the Königsberg bridges problem, that is, the problem of starting at some point, traversing each bridge exactly once, and returning to the original point. Since then, numerous generalizations and variants of this problem have been proposed. In this course we are going to present some basic graph routing problems (known as the postman problem and the traveling salesman problem), we are going to cover the necessary graph theoretical concepts and mathematical programming techniques used to model these problems (usually as linear integer programs), and we are going to describe algorithmic methods for their solution.

**Lernziele und Kompetenzen:**

At the end of this course, students should be able to (a) recognize the basic arc and edge graph routing problems, (b) solve graph routing problems using well known combinatorial algorithms, (c) model more complex graph routing problems using integer programming, (d) apply mathematical programming techniques to solve practical graph routing problems.

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

**[1] Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Nebenfach | Nebenfach Mathematik)

Dieses Modul ist daneben auch in den Studienfächern "Computational and Applied Mathematics (Master of Science)", "Informatik (Master of Science)", "Mathematik (Bachelor of Science)", "Mathematik (Master of Science)", "Technomathematik (Bachelor of Science)", "Technomathematik (Master of Science)", "Wirtschaftsingenieurwesen (Master of Science)", "Wirtschaftsmathematik (Bachelor of Science)", "Wirtschaftsmathematik (Master of Science)" verwendbar.

---

**Studien-/Prüfungsleistungen:**

Optimization in industry and economy (Prüfungsnummer: 59231)

(englische Bezeichnung: Optimization in industry and economy)

Prüfungsleistung, mündliche Prüfung, Dauer (in Minuten): 20

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Graph Routing and applications

Prüfungssprache: Englisch

Erstablingung: WS 2019/2020, 1. Wdh.: WS 2019/2020

1. Prüfer: Francisco Javier Zaragoza Martínez (060307)

---



---

**Modulbezeichnung:** Optimization in Industry and Economy (OptIW) 5 ECTS  
 (Optimization in Industry and Economy)

Modulverantwortliche/r: Martin Schmidt, Frauke Liers

Lehrende: Martin Schmidt, Frauke Liers, Francisco Javier Zaragoza Martínez

Startsemester: WS 2019/2020

Dauer: 1 Semester

Turnus: jährlich (WS)

Präsenzzeit: 45 Std.

Eigenstudium: 105 Std.

Sprache: Deutsch

---

**Lehrveranstaltungen:**

Optimization in Industry and Economy (Optimierung in Industrie und Wirtschaft) (WS 2019/2020, Vorlesung, 4 SWS, Francisco Javier Zaragoza Martínez)

---

**Empfohlene Voraussetzungen:**

Lineare Algebra, Lineare und Kombinatorische Optimierung

---

**Inhalt:**

Schwerpunkt dieser Vorlesung ist die Modellierung von Optimierungsproblemen aus der industriellen und wirtschaftlichen Praxis. Vor- und Nachteile unterschiedlicher Modellierungen werden diskutiert, und es werden geeignete Reformulierungen in Hinblick auf effektive Lösbarkeit vorgestellt. Numerische Ergebnisse werden diskutiert. Die Studierenden erlernen sowohl eine sinnvolle Darstellung von Optimierungsergebnissen als auch ihre Interpretation und Auswertung für die Praxis. Themen sind beispielsweise die Optimierung von Versorgungsnetzwerken (Gas, Wasser, Strom), Flugplanung oder mathematische Modellierung und Optimierung zur Bewertung von Marktmechanismen im Energiebereich.

**Lernziele und Kompetenzen:**

Die Studierenden

- modellieren komplexe Optimierungsprobleme aus der Praxis mit Blick auf effektive Lösbarkeit;
- klassifizieren die Modelle und nutzen die passenden Lösungsverfahren;
- bewerten die erzielten Ergebnisse.

**Literatur:**

- Vorlesungsskript/-Folien zu diesem Modul
- aktuelle Forschungsliteratur

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

[1] **Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Nebenfach | Nebenfach Mathematik)

Dieses Modul ist daneben auch in den Studienfächern "Computational and Applied Mathematics (Master of Science)", "Informatik (Master of Science)", "Mathematik (Master of Science)", "Technomathematik (Master of Science)", "Wirtschaftsingenieurwesen (Master of Science)", "Wirtschaftsmathematik (Master of Science)" verwendbar.

---

**Studien-/Prüfungsleistungen:**

Optimization in industry and economy (Prüfungsnummer: 59231)

(englische Bezeichnung: Optimization in industry and economy)

Prüfungsleistung, mündliche Prüfung, Dauer (in Minuten): 15

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Optimization in Industry and Economy (Optimierung in Industrie und Wirtschaft)

Erstablingung: WS 2019/2020, 1. Wdh.: WS 2019/2020

1. Prüfer: Francisco Javier Zaragoza Martínez (060307)

---

**Organisatorisches:**

Die Präsentation des Stoffes erfolgt in Vorlesungsform. Die weitere Aneignung der wesentlichen Begriffe und Techniken erfolgt durch wöchentliche Hausaufgaben.

**Bemerkungen:**

- Wahlmodul: Master Mathematik, Technomathematik und Wirtschaftsmathematik
- Kern-/Forschungsmodul Master Wirtschaftsmathematik Studienrichtung „Optimierung und Prozesssteuerung“

---

**Modulbezeichnung:** Partielle Differentialgleichungen I (PDG I) 10 ECTS  
 (Partial Differential Equations 1)

Modulverantwortliche/r: Günther Grün  
 Lehrende: Frank Duzaar

---

Startsemester: WS 2019/2020	Dauer: 1 Semester	Turnus: jährlich (WS)
Präsenzzeit: 90 Std.	Eigenstudium: 210 Std.	Sprache: Deutsch

---

**Lehrveranstaltungen:**

Partielle Differentialgleichungen I (WS 2019/2020, Vorlesung, 4 SWS, Frank Duzaar)  
 Übungen zu Partielle Differentialgleichungen I (WS 2019/2020, Übung, 2 SWS, Frank Duzaar)

---

**Empfohlene Voraussetzungen:**

Analysis-Module des Bachelorstudiums

---

**Inhalt:**

- schwache Existenztheorie elliptischer Gleichungen zweiter Ordnung
- Regularität schwacher Lösungen (Differenzenquotientenmethode, Moser, Harnack)
- Wärmeleitungsgleichung in Hölderräumen, Vergleichssätze

**Lernziele und Kompetenzen:**

Die Studierenden erarbeiten sich einen Überblick über Anwendungsbereiche von PDGen. Sie verwenden einfache explizite Lösungsmethoden und nutzen klassische und „schwache“ Zugänge zu Existenzresultaten.

**Literatur:**

- E. DiBenedetto, Partial Differential Equations, Birkhäuser 2001
- L. C. Evans, Partial Differential Equations, AMS 1997
- D. Gilbarg, N. S. Trudinger, Elliptic Partial Differential Equations, Springer 1983
- Vorlesungsskriptum

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

**[1] Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Nebenfach | Nebenfach Mathematik)

Dieses Modul ist daneben auch in den Studienfächern "Informatik (Master of Science)", "Mathematik (Bachelor of Science)", "Technomathematik (Bachelor of Science)" verwendbar.

---

**Studien-/Prüfungsleistungen:**

Klausur: Partielle Differentialgleichungen I (Prüfungsnummer: 51201)

Prüfungsleistung, Klausur, Dauer (in Minuten): 90

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Partielle Differentialgleichungen I

Erstablingung: WS 2019/2020, 1. Wdh.: SS 2020

1. Prüfer: Frank Duzaar (060108)

Übungsleistung: Partielle Differentialgleichungen I (Prüfungsnummer: 51202)

Studienleistung, Übungsleistung

Zugeordnete Lehrveranstaltungen:

- Übungen zu Partielle Differentialgleichungen I

weitere Erläuterungen:

Hausaufgaben (wöchentlich ein Übungsblatt)

Erstablingung: WS 2019/2020, 1. Wdh.: SS 2020

1. Prüfer: Frank Duzaar (060108)

---

**Organisatorisches:**

Die Präsentation des Stoffes erfolgt in Vorlesungsform. Die weitere Aneignung der wesentlichen Begriffe und Techniken erfolgt durch wöchentliche Hausaufgaben.

**Bemerkungen:**

- Bachelor Mathematik, Technomathematik
- Wahlmodul: Master Mathematik, Technomathematik und Wirtschaftsmathematik
- Kern-/Forschungsmodul Master Mathematik Studienrichtungen „Analysis und Stochastik“ und „Modellierung-Simulation-Optimierung“, Master Technomathematik Studienrichtung „Modellierung und Simulation“

---

**Modulbezeichnung:** Einführung in die Robuste Optimierung (RobOptnv) 5 ECTS  
 (Introduction to Robust Optimisation)

Modulverantwortliche/r: Frauke Liers  
 Lehrende: Frauke Liers

---

Startsemester: SS 2019	Dauer: 1 Semester	Turnus: jährlich (SS)
Präsenzzeit: 45 Std.	Eigenstudium: 105 Std.	Sprache: Deutsch

---

**Lehrveranstaltungen:**

Diese Bachelorveranstaltung ist der erste Teil einer Vorlesung, die durch die Masterveranstaltung "Robuste Optimierung (vertieft)" (mehr Informationen unter Mastervorlesungen) fortgesetzt wird. Sie findet in der ersten Hälfte des Semesters statt. Es können auch beide Teile der Vorlesung besucht werden und für das Bachelorstudium (insgesamt 10 ECTS) angerechnet werden.

Im Master können bis zu 10% der ECTS-Punkte aus dem Bachelorbereich eingebracht werden. Daher kann die volle VL als 4h-Veranstaltung gehört werden und 10 ECTS in das Masterstudium eingebracht werden, falls nicht schon anderweitig Bachelor-ECTS Punkte eingebracht wurden.

Robuste Optimierung (nicht vertieft) (SS 2019, Vorlesung mit Übung, 2 SWS, Frauke Liers)

Übung zu Robuste Optimierung (nicht vertieft) (SS 2019, Übung, 2 SWS, Frauke Liers)

---

**Empfohlene Voraussetzungen:**

Lineare Algebra. Vorteilhaft wären darüber hinaus das Modul Kombinatorische Optimierung oder das Modul Lineare und konvexe Optimierung.

---

**Inhalt:**

Oft sind die Eingabedaten eines mathematischen Optimierungsproblems in der Praxis nicht exakt bekannt. In der robusten Optimierung werden deswegen möglichst gute Lösungen bestimmt, die für alle innerhalb gewisser Toleranzen liegenden Eingabedaten, zulässig sind. Die Vorlesung behandelt die Theorie und Modellierung robuster Optimierungsprobleme, insbesondere die robuste lineare und robuste kombinatorische Optimierung. Darüber hinaus werden anhand von Anwendungsbeispielen aktuelle Konzepte wie z.B. die „wiederherstellbare Robustheit“ gelehrt.

**Lernziele und Kompetenzen:**

Die Studierenden

- erkennen selbstständig Optimierungsprobleme unter Unsicherheit, modellieren die zugehörigen robustifizierten Optimierungsprobleme geeignet und analysieren diese;
- nutzen die passenden Lösungsverfahren und bewerten die erzielten Ergebnisse.

**Literatur:**

- Vorlesungsskript zu diesem Modul
  - Ben-Tal, El Ghaoui, Nemirovski: Robust Optimization. Princeton University Press 2009.
- 

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

**[1] Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Nebenfach | Nebenfach Mathematik)

Dieses Modul ist daneben auch in den Studienfächern "Informatik (Master of Science)", "Mathematik (Bachelor of Science)", "Wirtschaftsmathematik (Bachelor of Science)" verwendbar.

---

**Studien-/Prüfungsleistungen:**

Klausur: Robuste Optimierung (nicht vertieft) (Prüfungsnummer: 51751)

Prüfungsleistung, Klausur, Dauer (in Minuten): 60

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Robuste Optimierung (nicht vertieft)

Erstablingung: SS 2019, 1. Wdh.: SS 2019

1. Prüfer: Frauke Liers (060129)

Übungsleistung: Robuste Optimierung (nicht vertieft) (Prüfungsnummer: 51752)

Studienleistung, Übungsleistung

Zugeordnete Lehrveranstaltungen:

- Übung zu Robuste Optimierung (nicht vertieft)

weitere Erläuterungen:

erfolgreiche Bearbeitung wöchentlicher Hausaufgaben

Erstablingung: SS 2019, 1. Wdh.: keine Angabe

1. Prüfer: Frauke Liers (060129)

---

**Organisatorisches:**

Die Präsentation des Stoffes erfolgt in Vorlesungsform. Die weitere Aneignung der wesentlichen Begriffe und Techniken erfolgt durch wöchentliche Hausaufgaben.

**Bemerkungen:**

- Bachelor Mathematik, Technomathematik oder Wirtschaftsmathematik
- Wahlmodul: Master Mathematik, Technomathematik und Wirtschaftsmathematik
- Kern-/Forschungsmodul Master Wirtschaftsmathematik Studienrichtung „Optimierung und Prozesssteuerung“

---

**Modulbezeichnung: Topologie (Top)** **5 ECTS**  
(Topology)

Modulverantwortliche/r: Karl-Hermann Neeb  
Lehrende: Karl-Hermann Neeb, Catherine Meusburger

---

Startsemester: SS 2019	Dauer: 1 Semester	Turnus: jährlich (SS)
Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Deutsch

---

**Lehrveranstaltungen:**

Topologie (Querschnittmodul) (SS 2019, Vorlesung, 4 SWS, Karl-Hermann Neeb et al.)  
Übungen zu Topologie (Querschnittmodul) (SS 2019, Übung, 2 SWS, Karl-Hermann Neeb et al.)

---

**Inhalt:**

- Stetige Funktionen, Zusammenhang, Trennungsaxiome
- Erzeugung von Topologien (initiale, finale, Quotienten etc.)
- Konvergenz in topologischen Räumen (Filter, Netze)
- Kompaktheit (Satz von Tychonov, kompakte metrische Räume, lokalkompakte Räume)
- Anwendung auf Funktionenräume (Satz von Stone-Weierstraß, Satz von Ascoli)
- Überlagerungstheorie (Fundamentalgruppen, Hochhebung von Abbildungen)

**Lernziele und Kompetenzen:**

- Die Studierenden
- wenden die Methoden der allgemeinen Topologie, die in den Grundvorlesungen nur am Rande vorkommt, an;
  - ordnen die topologischen Grundbegriffe in einen größeren Kontext ein;
  - erklären und verwenden wichtige Resultate, die in vielen Bereichen der Mathematik zum Handwerkzeug gehören.

**Literatur:**

- Vorlesungsskript zu diesem Modul
  - Bredon: Geometry and Topology
  - Skript auf StudOn bereitgestellt und auch unter [www.studium.math.fau.de/lehrveranstaltungen/skripten.html](http://www.studium.math.fau.de/lehrveranstaltungen/skripten.html)
- 

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

**[1] Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Nebenfach | Nebenfach Mathematik)

Dieses Modul ist daneben auch in den Studienfächern "Informatik (Master of Science)", "Mathematik (1. Staatsprüfung für das Lehramt an Gymnasien)", "Mathematik (Bachelor of Science)", "Technomathematik (Bachelor of Science)", "Wirtschaftsmathematik (Bachelor of Science)" verwendbar.

---

**Studien-/Prüfungsleistungen:**

Klausur: Topologie (Prüfungsnummer: 50801)  
(englische Bezeichnung: Written examination: Introduction to Topology)  
Prüfungsleistung, Klausur, Dauer (in Minuten): 60  
Anteil an der Berechnung der Modulnote: 100%  
Zugeordnete Lehrveranstaltungen:

- Topologie (Querschnittmodul)

Erstablingung: SS 2019, 1. Wdh.: SS 2019  
1. Prüfer: Karl-Hermann Neeb (060160)

Übungsleistung: Topologie (Prüfungsnummer: 50802)

Studienleistung, Übungsleistung

Zugeordnete Lehrveranstaltungen:

- Übungen zu Topologie (Querschnittmodul)

weitere Erläuterungen:

Hausaufgaben (wöchentlich ein Übungsblatt)

Prüfungssprache: Deutsch

Erstabledung: SS 2019, 1. Wdh.: keine Angabe

1. Prüfer: Karl-Hermann Neeb (060160)

---



**Modulbezeichnung:** Einführung in die Darstellungstheorie (EDT) 10 ECTS  
(Introduction to representation theory)

Modulverantwortliche/r: Peter Fiebig, Friedrich Knop

Lehrende: Catherine Meusburger

Startsemester: SS 2019

Dauer: 1 Semester

Turnus: jährlich (SS)

Präsenzzeit: 90 Std.

Eigenstudium: 210 Std.

Sprache:

**Lehrveranstaltungen:**

Einführung in die Darstellungstheorie (SS 2019, Vorlesung mit Übung, 4 SWS, Catherine Meusburger)

Übungen zu Einführung in die Darstellungstheorie (SS 2019, Übung, 2 SWS, Catherine Meusburger)

**Empfohlene Voraussetzungen:**

Modul Algebra

**Inhalt:**

- Darstellungen endlicher Gruppen
- Moduln über Ringen
- Halbeinfache Ringe
- Weiterführende Themen wie Darstellungstheorie von Köchern, Darstellungstheorie symmetrischer Gruppen, Schur-Weyl-Dualität, elementare Darstellungstheorie von Lie-Algebren

**Lernziele und Kompetenzen:**

Die Studierenden

- nennen und erläutern die grundlegenden Begriffe der Darstellungstheorie anhand beispielhaft ausgewählter Kapitel und erkennen und erklären deren Zusammenhänge;
- ordnen Methoden aus der Algebra in einen übergreifenden Kontext ein und wenden diese an;
- analysieren und bewerten algebraische Strukturen und erkennen Zusammenhänge;
- klassifizieren und lösen selbstständig algebraische Probleme.

**Literatur:**

- Vorlesungsskript zu diesem Modul
- Lang: Algebra

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

[1] **Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Nebenfach | Nebenfach Mathematik)

Dieses Modul ist daneben auch in den Studienfächern "Informatik (Master of Science)", "Mathematik (Bachelor of Science)", "Wirtschaftsmathematik (Bachelor of Science)" verwendbar.

**Studien-/Prüfungsleistungen:**

Einführung in die Darstellungstheorie (Prüfungsnummer: 50701)

Prüfungsleistung, Klausur, Dauer (in Minuten): 90

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Einführung in die Darstellungstheorie

Erstablingung: SS 2019, 1. Wdh.: WS 2019/2020

1. Prüfer: Catherine Meusburger (060171)

Übungsleistung: Einführung in die Darstellungstheorie (Prüfungsnummer: 50702)

Studienleistung, Übungsleistung

Zugeordnete Lehrveranstaltungen:

- Übungen zu Einführung in die Darstellungstheorie

weitere Erläuterungen:  
erfolgreiche Bearbeitung wöchentlicher Hausaufgaben

Erstablingung: SS 2019, 1. Wdh.: keine Angabe  
1. Prüfer: Catherine Meusburger (060171)

---

**Organisatorisches:**

Die Präsentation des Stoffes erfolgt in Vorlesungsform. Die weitere Aneignung der wesentlichen Begriffe und Techniken erfolgt durch wöchentliche Hausaufgaben.

**Bemerkungen:**

Wahlpflichtfach in

- B. Sc. Mathematik

**Modulbezeichnung:** Einführung in die Medizinische Informatik für Informatik-Nebenfachstudierende (MedInfEinf) (Introduction to Medical Informatics) **5 ECTS**

Modulverantwortliche/r: Wolfgang Rödle

Lehrende: Wolfgang Rödle, Wolfgang Uter, Werner Adler, Oliver Amft, Meik Kunz

Startsemester: WS 2019/2020

Dauer: 1 Semester

Turnus: halbjährlich (WS+SS)

Präsenzzeit: 30 Std.

Eigenstudium: 120 Std.

Sprache: Deutsch

**Lehrveranstaltungen:**

Einführung in die Medizinische Informatik für Informatik-Nebenfachstudierende (WS 2019/2020, Vorlesung mit Übung, 3 SWS, Wolfgang Rödle et al.)

**Inhalt:**

Die Vorlesung soll an medizinischer Informatik interessierten Studenten einen groben thematischen Überblick über verschiedene Schwerpunkte der Medizinischen Informatik geben. Vorgestellte Themenbereiche:

- Informationssysteme im Gesundheitswesen
- Medizinische Biometrie und Epidemiologie
- mHealth
- Bioinformatik

**Lernziele und Kompetenzen:**

Ziel dieser Vorlesung ist es, den Studenten einen ersten Überblick über die verschiedenen methodischen Schwerpunkte der Medizinischen Informatik zu verschaffen, damit diese die Kompetenz erwerben, das Fach nachfolgend anhand ihrer eigenen Interessensgebiete zu vertiefen. Neben den Vorlesungen sind eine Reihe von Übungsaufgaben zu bearbeiten. Diese beinhalten einerseits das selbständige Recherchieren und Erschließen verschiedener Themen/Probleme/Aufgabenstellungen der Medizinischen Informatik, andererseits ermöglichen sie das Vertiefen der in der Vorlesung vermittelten Lehrinhalte.

*Fachkompetenz*

*Wissen*

- nennen die Grundbegriffe im Bereich Epidemiologie und Biometrie
- nennen grundlegende Werkzeuge und Fragestellungen der Bioinformatik

*Verstehen*

- Informationssysteme und ihre Zusammenhänge im Krankenhaus erläutern
- Prozessmodelle im Krankenhaus skizzieren
- Grundlagen zu klinischer Dokumentation erklären
- einen Überblick zu Gesundheitstelematik-Werkzeugen (z.B. eRezept oder eMedikationsplan) veranschaulichen
- Funktionen und Inhalte der verschiedenen elektronischen AKten (eKA, ePA, eGA) aufzeigen
- erläutern die Zielsetzung von mHealth und Digital Health, sowie der wesentlichen Eigenschaften von kontextbewussten Systemen
- erklären den Aufbau und die grundlegenden Eigenschaften von inertialen Messsystemen und akustischen Aufnehmern
- erklären empirische Methoden zur Analyse der Modellanpassung
- erklären grundlegende statistische Tests
- erklären grundlegende Konzepte und Algorithmen der Bioinformatik
- erklären bioinformatische Methoden für die Analyse von DNA-, RNA- und Protein-Sequenzen
- erklären bioinformatische Methoden für die klinische Entscheidungsunterstützung

*Anwenden*

- wenden das Bayes'sche Theorem und Varianten an, sowie die Maximum A-posterior Entscheidungsregel zur Klassifikation von Merkmalen aus Sensordaten
- wenden die Grundkenntnisse auf einfache Problemstellungen in Biostatistik und Epidemiologie an

- wenden grundlegende bioinformatische Analysemethoden auf einfache medizinische Fragestellungen an

#### *Analysieren*

- analysieren Kontexterkenkung, insbesondere in der isolierten Klassifikation von Aktivitäten anhand von inertialen Sensordaten aus körpergetragenen / Wearable Systemen

---

#### **Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

##### **[1] Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Nebenfach | Nebenfach Medizin (Medizinische Informatik) | Schwerpunktübergreifende Module)

Dieses Modul ist daneben auch in den Studienfächern "Informatik (Master of Science)" verwendbar.

---

#### **Studien-/Prüfungsleistungen:**

Einführung in die Medizinische Informatik für Informatik-Nebenfachstudierende (Prüfungsnummer: 33001)

Prüfungsleistung, mündliche Prüfung, Dauer (in Minuten): 20

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Einführung in die Medizinische Informatik für Informatik-Nebenfachstudierende

weitere Erläuterungen:

schriftliche Teilprüfungen im Semesterverlauf Anteil an der Berechnung der Modulnote: 50.0 %

mündliche Prüfung, Dauer (in Minuten): 20, benotet, Anteil an der Berechnung der Modulnote: 50.0 %

Prüfungssprache: Deutsch

Erstablingung: WS 2019/2020, 1. Wdh.: SS 2020

1. Prüfer: Wolfgang Rödle (031108), 2. Prüfer: Wolfgang Uter (031004)

1. Prüfer: Oliver Amft (030111), 2. Prüfer: Hans-Ulrich Prokosch (031003)

---

#### **Organisatorisches:**

Studenten melden sich bitte mit Angabe Ihrer Matrikelnummer, Ihrer StudOn-Kennung und Ihres Abschlusses (Bachelor/Master) per Mail <mailto:martin.ross@fau.de> an.

**Modulbezeichnung:** Wissensbasierte Systeme in der Medizin 1 (MedInfWiss1) **5 ECTS**  
(Knowledge-based Systems in Healthcare)

Modulverantwortliche/r: Stefan Kraus

Lehrende: Stefan Kraus, Dennis Toddenroth, Wolfgang Rödle, Hans-Ulrich Prokosch

Startsemester: SS 2019

Dauer: 1 Semester

Turnus: jährlich (SS)

Präsenzzeit: 60 Std.

Eigenstudium: 90 Std.

Sprache: Deutsch

### Lehrveranstaltungen:

Wissensbasierte Systeme in der Medizin 1 (SS 2019, Vorlesung mit Übung, 4 SWS, Hans-Ulrich Prokosch et al.)

### Inhalt:

Die Studierenden

- grenzen konventionelle Software von wissensbasierten Systemen bzw. medizinischen Expertensystemen ab.
- erklären grundlegende Inferenzstrategien wie Vorwärts- und Rückwärtsverkettung.
- kennen den einzigen verbreiteten Standard für medizinische Wissensrepräsentation.
- nutzen die Arden-Syntax zum Erstellen von Wissensmodulen.
- kennen historisch bedeutsame Expertensysteme und deren Inferenzstrategien.
- erklären grundlegende Aspekte der Arzneimitteltherapiesicherheit.
- diskutieren die Schwierigkeiten bei der Integration wissensbasierter Funktionen in die klinische Routine.

### Lernziele und Kompetenzen:

Die Studierenden

- erklären den Unterschied zwischen konventioneller Software und wissensbasierten Systemen bzw. medizinischen Expertensystemen.
- erklären grundlegende Inferenzstrategien wie Vorwärts- und Rückwärtsverkettung.
- erklären und nutzen den bisher einzigen Standard für medizinische Wissensrepräsentation.
- kennen historisch bedeutsame Expertensysteme und deren Inferenzstrategien.
- erstellen selbständig standardisierte Wissensmodule.
- erklären grundlegende Aspekte der Arzneimitteltherapiesicherheit.
- verstehen die Schwierigkeiten bei der Integration wissensbasierter Funktionen in die klinische Routine.

### Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

#### [1] Informatik (Bachelor of Science)

(Po-Vers. 2009s | Nebenfach | Nebenfach Medizin (Medizinische Informatik) | Schwerpunkt Informationssysteme im Gesundheitswesen)

Dieses Modul ist daneben auch in den Studienfächern "Medizintechnik (Bachelor of Science)" verwendbar.

### Studien-/Prüfungsleistungen:

Wissensbasierte Systeme in der Medizin 1 (Prüfungsnummer: 28501)

(englische Bezeichnung: Medical Knowledge Processing 1)

Prüfungsleistung, mündliche Prüfung, Dauer (in Minuten): 20

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Wissensbasierte Systeme in der Medizin 1

Erstablesung: SS 2019, 1. Wdh.: keine Angabe

1. Prüfer: Hans-Ulrich Prokosch (031003)

**Organisatorisches:**

Bitte melden Sie sich mit Angabe Ihrer Matrikelnummer, Ihres Studienfachs und Ihres Abschlusses (Bachelor/Master) bis zum 8. April 2019 per E-Mail an <mailto:martin.ross@fau.de> an.

---

**Modulbezeichnung:** Grundlagen der Anatomie und Physiologie für Medizintechniker, Naturwissenschaftler und Ingenieure (AnaPhys\_MT)  
(Fundamentals of Anatomy and Physiology) 5 ECTS

Modulverantwortliche/r: Clemens Forster, Michael Eichhorn  
Lehrende: Clemens Forster, Michael Eichhorn

---

Startsemester: SS 2019	Dauer: 2 Semester	Turnus: halbjährlich (WS+SS)
Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Deutsch

---

**Lehrveranstaltungen:**

**Grundlagen der Anatomie & Physiologie für Nichtmediziner**

Grundlagen der Anatomie und Physiologie für Medizintechniker, Naturwissenschaftler und Ingenieure, Teil 2 (Innere Organe) (WS 2019/2020, Vorlesung, Clemens Forster)

Grundlagen der Anatomie und Physiologie für Medizintechniker, Naturwissenschaftler und Ingenieure, Teil 1 Neurophysiologie (SS 2019, Vorlesung, Clemens Forster et al.)

**Clinical Applications of Optical Technologies and Associated Fundamentals of Anatomy**

English-taught equivalent for the German lecture "Grundlagen der Anatomie & Physiologie für Nichtmediziner" (for international students). If you have completed the German version of the module you cannot take the English version again.

Clinical Application of Advanced Optical Technologies and Associated Fundamentals in Anatomy (SS 2019, Vorlesung, Michael Eichhorn)

---

**Inhalt:**

- Wissensvermittlung zu Grundlagen der Anatomie, Physiologie und Pathophysiologie
- Wissensvermittlung von wichtigen medizinischen Fachbegriffen
- Wissensvermittlung von relevanten und häufigen Krankheitsbildern
- Wissensvermittlung von relevanten Methoden beim biologischen und technischen Sehen
- Diskussion von Methoden und Theorieansätzen, um relevante medizinische Fragestellungen erkennen zu können
- Kritische Betrachtung von den wichtigsten bildgebenden Verfahren in wichtigen Krankheitsbildern
- Darstellung der Organisationsstrukturen von diagnostischen Prozessen

**Lernziele und Kompetenzen:**

Die Studierenden

- verstehen die wichtigsten und häufigsten medizinische Fachbegriffe
  - sind vertraut mit den Grundlagen der Anatomie und der Physiologie
  - kennen wichtige Krankheitsbilder
  - verstehen und erklären medizinische Fragestellungen in der Diagnostik und Therapie anhand von Beispielen
- 

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

[1] **Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Nebenfach | Nebenfach Medizin (Medizinische Informatik) | Schwerpunkt Physiologie)

Dieses Modul ist daneben auch in den Studienfächern "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Master of Science)", "Medizintechnik (Bachelor of Science)", "Medizintechnik (Master of Science)", "Physics (Master of Science)", "Physik (Bachelor of Science)", "Physik (Master of Science)" verwendbar.

---

**Studien-/Prüfungsleistungen:**

Grundlagen der Anatomie und Physiologie für Medizintechniker, Naturwissenschaftler und Ingenieure  
(Prüfungsnummer: 28001)

Prüfungsleistung, Klausur, Dauer (in Minuten): 90

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Grundlagen der Anatomie und Physiologie für Medizintechniker, Naturwissenschaftler und Ingenieure, Teil 2 (Innere Organe)
- Grundlagen der Anatomie und Physiologie für Medizintechniker, Naturwissenschaftler und Ingenieure, Teil 1 Neurophysiologie

weitere Erläuterungen:

In der Klausur werden die Inhalte beider Vorlesungssemester abgefragt.

Prüfungssprache: Deutsch

Erstablingung: WS 2019/2020, 1. Wdh.: SS 2020

1. Prüfer: Clemens Forster (030333)

Abschlussklausur Anatomie und Physiologie für Nichtmediziner (Prüfungsnummer: 28001)

(englische Bezeichnung: Final Examination on Anatomy and Physiology for Non-Medical Students)

Untertitel: Clinical Applications of Optical Technologies and Associated Fundamentals of Anatomy

Prüfungsleistung, Klausur, Dauer (in Minuten): 90

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- Clinical Application of Advanced Optical Technologies and Associated Fundamentals in Anatomy

Prüfungssprache: Englisch

Erstablingung: SS 2019, 1. Wdh.: WS 2019/2020

1. Prüfer: Michael Eichhorn (030203)



---

**Modulbezeichnung:** **Nebenfach Skandinavistik (NFSkand)** **15 ECTS**  
 (Scandinavian Studies as Secondary Studies)

Modulverantwortliche/r: Hanna Eglinger  
 Lehrende: Karina Brehm, , Patrick Ledderose

---

Startsemester: WS 2019/2020	Dauer: 2 Semester	Turnus: jährlich (WS)
Präsenzzeit: 150 Std.	Eigenstudium: 300 Std.	Sprache: Deutsch

---

**Lehrveranstaltungen:**

- BM Nord. Erstsprache: Schwedisch: SK I (WS 2019/2020, Seminar, 4 SWS, Karina Brehm)
- BM Nordische Erstsprache II: Schwedisch SK II (SS 2020, Übung, 4 SWS, Karina Brehm)
- BM: Nordistische Literaturwissenschaft 1: Einführungsseminar 1 (NorLi 1) (WS 2019/2020, Seminar, 2 SWS, Patrick Ledderose)
- BM: Nordische Kulturgeschichte 1 (WS 2019/2020, Seminar, 2 SWS, Jeremias Jokisch)

---

**Inhalt:**

Das Nebenfachmodul Skandinavistik für Informatikstudierende vermittelt einen grundlegenden Überblick über Sprachen, Literatur- und Kulturgeschichte Skandinaviens. Für die zwei Pflichtmodule Nordische Erstsprache 1 und 2 wählt man entweder Dänisch, Norwegisch oder Schwedisch. Aus den beiden Wahlpflichtmodulen Literaturwissenschaft 1 und Nordische Kulturgeschichte 1 ist eines auszuwählen, wobei Literaturwissenschaft 1 die Grundzüge der neueren skandinavischen Literaturgeschichte und Nordische Kulturgeschichte 1 die frühe Geschichte des Nordens von der Wikingerzeit bis zur Reformationszeit behandelt.

**Lernziele und Kompetenzen:**

In den Pflichtmodulen wird die gewählte Sprache durch Hörverstehen, Leseverstehen, Sprechen, Schreiben, Grammatik und Wortschatz sowie ausgewählte landeskundliche und kulturspezifische Elemente eingeübt. Das Zielniveau nach dem Gemeinsamen Europäischen Referenzrahmen für Sprachen (CEFR) ist A1 (für Nordische Erstsprache 1) und A2 (für Nordische Erstsprache 2). Die Wahlpflichtmodule ermöglichen grundlegende Fähigkeiten der literar- und kulturhistorischen Einordnung und des literaturwissenschaftlichen Arbeitens mit Texten und ggf. weiteren Medien skandinavischer Provenienz. Weitere Kompetenzen und Lernziele sind Diskussionsfähigkeit und interkulturelles Verständnis sowie Fähigkeiten, zu vorgegebenen wissenschaftlichen Fragestellungen innerhalb einer begrenzten Zeitvorgabe schriftlich Stellung zu beziehen, eine kohärente Argumentation zu führen, Terminologien korrekt und konsistent anzuwenden und Lerninhalte eigenverantwortlich zu erarbeiten und selbstorganisiert auf die Klausur vorzubereiten.

**Literatur:**

Wird in der ersten Sitzung bekannt gegeben

---

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

**[1] Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Nebenfach | Nebenfach Nordische Philologie)

Dieses Modul ist daneben auch in den Studienfächern "Informatik (Master of Science)" verwendbar.

---

**Studien-/Prüfungsleistungen:**

Nebenfach Nordische Philologie für Informatikstudierende (15 ECTS) (Prüfungsnummer: 39704)  
 (englische Bezeichnung: Minor subject: Scandinavian Studies)

Prüfungsleistung, Portfolio

Anteil an der Berechnung der Modulnote: 100%

Zugeordnete Lehrveranstaltungen:

- BM Nord. Erstsprache: Schwedisch: SK I
- BM Nordische Erstsprache II: Schwedisch SK II
- BM: Nordistische Literaturwissenschaft 1: Einführungsseminar 1 (NorLi 1)
- BM: Nordische Kulturgeschichte 1

weitere Erläuterungen:

Für das Modul muss Nordische Erstsprache I und II bestanden werden, sowie eine der beiden Lehrveranstaltungen Nordistische Literaturwissenschaft 1 und Nordische Kulturgeschichte 1. Alle Teilmodule werden separat mit einer 90-minütigen Klausur abgeprüft.

Prüfungssprache: Deutsch

Erstablingung: SS 2020, 1. Wdh.: WS 2020/2021, 2. Wdh.: SS 2021

1. Prüfer: Karina Brehm (050625)

1. Prüfer: Patrick Ledderose (050787)

1. Prüfer: ()

---

### **Organisatorisches:**

Das Nebenfachmodul Skandinavistik ergibt 15 ECTS. Zu wählen sind pflichtgemäß die Teilmodule Nordische Erstsprache 1 (5 ECTS) und Nordische Erstsprache 2 (5 ECTS). Aus den Modulen Literaturwissenschaft 1 (5 ECTS) und Nordische Kulturgeschichte 1 (5 ECTS) ist eines zu wählen. Das Nebenfachmodul Skandinavistik erstreckt sich über mindestens zwei Semester. Die Teilmodule Nordische Erstsprache 1 (5 ECTS), Literaturwissenschaft 1 (5 ECTS) und Nordische Kulturgeschichte 1 (5 ECTS) sind nur im Wintersemester belegbar. Das Teilmodul Nordische Erstsprache 2 (5 ECTS) ist nur im Sommersemester belegbar. Die Anmeldung erfolgt über meincampus.

**Modulbezeichnung:** Recht als Nebenfach für Informatikstudierende (RechtInf) 15 ECTS  
(Law)

Modulverantwortliche/r: Hans Kudlich

Lehrende: Franz-Rudolf Herber, Hans Kudlich, Hans Kudlich, Franz Hofmann, Jan Gadinger,  
Zoi Michalopoulou, Axel Adrian, Franz Hofmann, Hanno Magnus

Startsemester: WS 2019/2020      Dauer: 2 Semester      Turnus: halbjährlich (WS+SS)  
Präsenzzeit: 135 Std.      Eigenstudium: 315 Std.      Sprache: Deutsch

**Lehrveranstaltungen:**

**Pflichtbereich**

Einführung in die Rechtswissenschaft für Studierende der Politikwissenschaft im BA-Studiengang "Öffentliches Recht als Zweitfach" (WS 2019/2020, Vorlesung, 2 SWS, Franz-Rudolf Herber)

**Vertiefungsbereich I: Strafrecht**

Strafrecht I (WS 2019/2020, Vorlesung, Hans Kudlich)

Strafprozessrecht (SS 2020, Übung, Hans Kudlich)

**Vertiefungsbereich II: Zivilrecht und Legal Tech**

Gewerblicher Rechtsschutz (= Patentrecht, Markenrecht, Designrecht) (WS 2019/2020, Vorlesung, Franz Hofmann)

Proseminar: Rechtsprechungsklassiker zum allgemeinen Schadensrecht (§§ 249 ff. BGB) (WS 2019/2020, Proseminar, Jan Gadinger)

Proseminar "Schutz von Geschäftsgeheimnissen" (WS 2019/2020, Proseminar, Anwesenheitspflicht, Zoi Michalopoulou)

Proseminar "Rechtliche Aspekte eines Legal-Tech Start-up-Unternehmens (WS 2019/2020, Proseminar, 2 SWS, Axel Adrian et al.)

Proseminar "AGB-Recht" (WS 2019/2020, Proseminar, Hanno Magnus)

**Inhalt:**

Das Nebenfach umfasst insgesamt 15 ECTS-Punkte. Nach einem verpflichtenden Einführungsmodul, für das 5 ECTS-Punkte vergeben werden, haben die Nebenfachstudierenden die Wahl zwischen den Vertiefungsbereichen „Strafrecht“ und „Zivilrecht und Legal Tech“. Der Vertiefungsbereich „Strafrecht“ umfasst die beiden Module „Strafrecht I“ und „Strafprozessrecht“. Wer „Zivilrecht und Legal Tech“ wählt, belegt ein zivilrechtliches Proseminar und wählt eines der aufgeführten zivilrechtlichen Vertiefungsmodule.

**Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:**

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

[1] **Informatik (Bachelor of Science)**

(Po-Vers. 2009s | Nebenfach | Nebenfach Rechtswissenschaften)

Dieses Modul ist daneben auch in den Studienfächern "Informatik (Master of Science)" verwendbar.