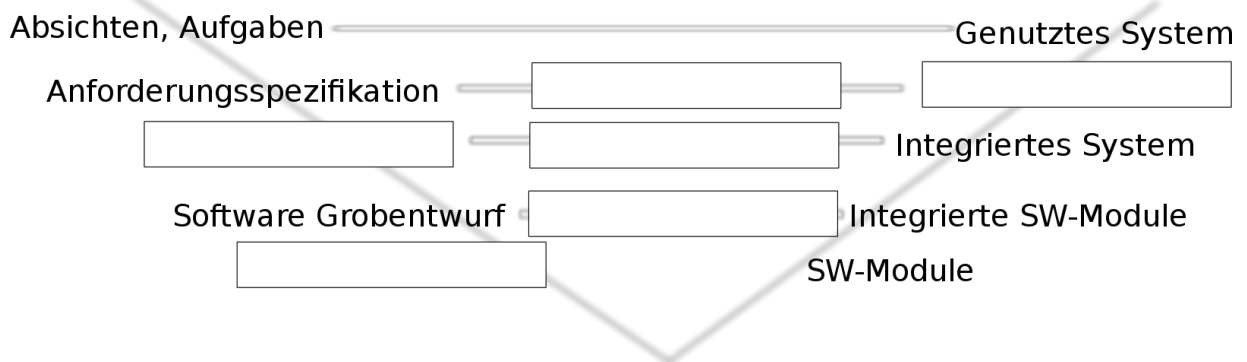


# Braindump Softwareentwicklung in Großprojekten WS 16/17

8. April 2019

## 1 Theoriefragen

a) Vervollständigen Sie die fehlenden Phasen und Testarten des V-Modells:



b) Erklären Sie die folgenden Begriffe:

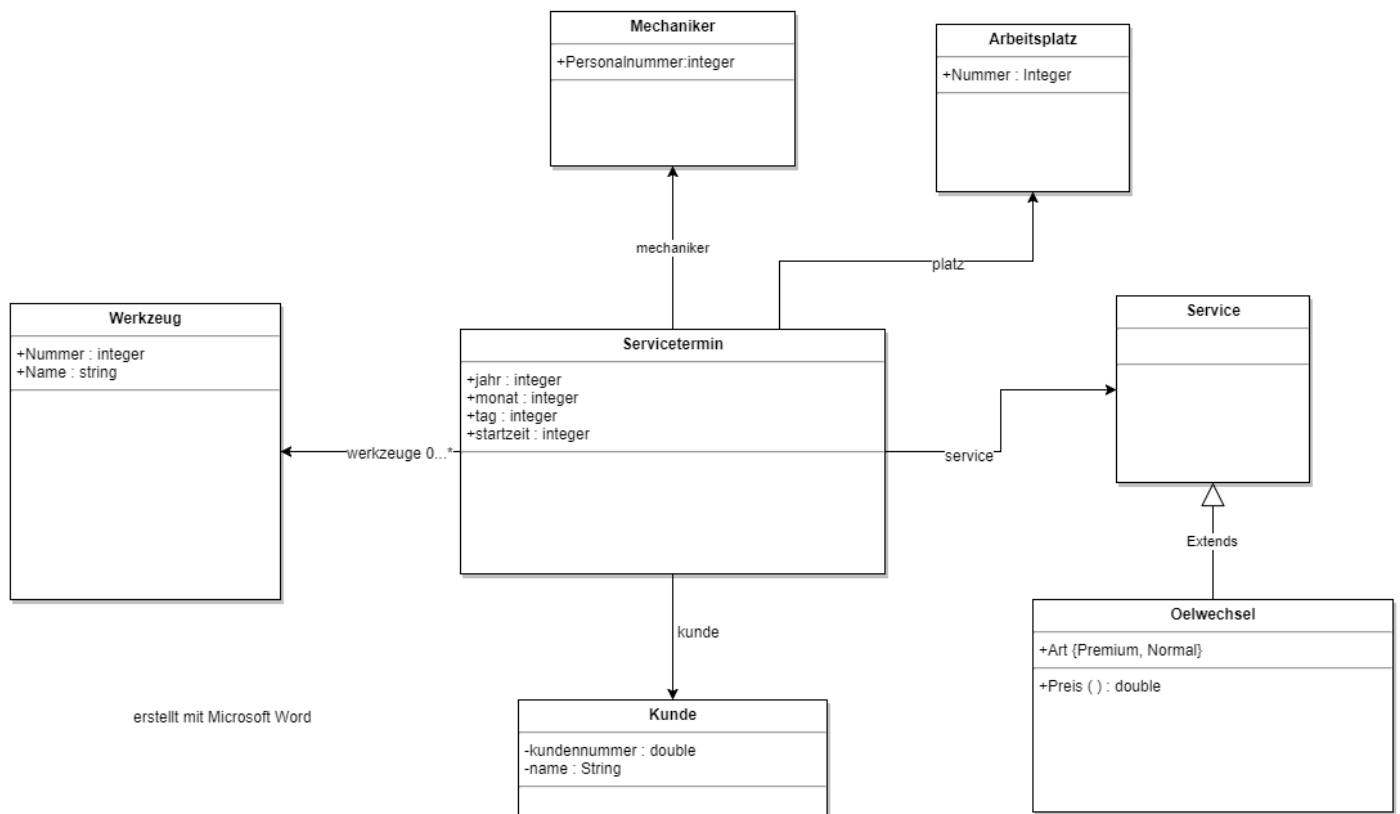
Verifikation

Validierung

c) In welchen Phasen kann trotz Verifikation ein Fehler nicht erkannt werden?

d) Erklären Sie die aus der Vorlesung bekannten Arten zur Wartung von Software.

## 2 OCL



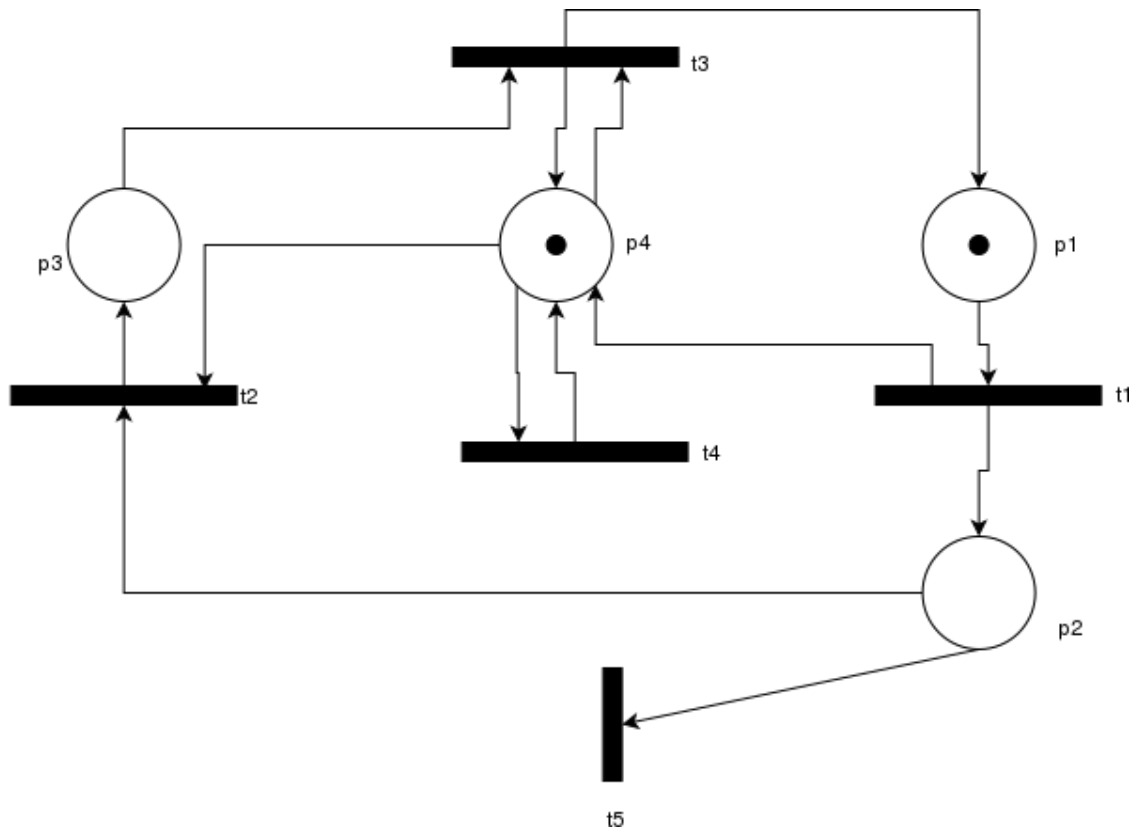
- Es gibt genau 3 Arbeitsplätze
- Von den Werkzeugen, die in einem Servicetermin vorkommen, existiert ein Werkzeug mit der Nummer 22.
- Die Kundennummer liegt zwischen 200 000 und 800 000
- Jeder Mechaniker besitzt eine eindeutige Personalnummer, d. h. es gibt keine zwei Mechaniker mit derselben Personalnummer.
- Je zwei gleichzeitig stattfindenden Servicetermin (selbes Jahr, Monat, Tag), die zur selben Anfangsstunde beginnen, haben weder denselben Mechaniker eingeteilt noch nutzen sie denselben Platz.
- Der Ölwechsel kostet bei normalem Öl 45€, bei Premiumöl 55€.

## 3 Entwurfsmuster

Für die Implementierung eines vereinfachten Dateisystems wurde die Klasse File mit der Methode write-Data(d: int) entwickelt. Weiterhin gibt es die Klasse Login mit den Methoden getCurrentUserId():int und getAllAuthorizedUserIDs(). Von der Klasse Login gibt es systemweit nur ein einziges Objekt. Ohne die Implementierung von File zu ändern, soll nun die Software so erweitert werden, dass Schreibzugriffe nur von autorisierten Klienten tatsächlich ausgeführt werden.

- Welches Entwurfsmuster eignet sich am besten dafür?
- Nennen Sie Klasse und Gültigkeitsbereich dieses Entwurfsmusters.
- Zeichnen Sie ein entsprechendes UML-Diagramm (inkl. Entwurfshinweise/Kommentare), das die Umsetzung dieses Muster in diesem konkreten Anwendungsfall zeigt.

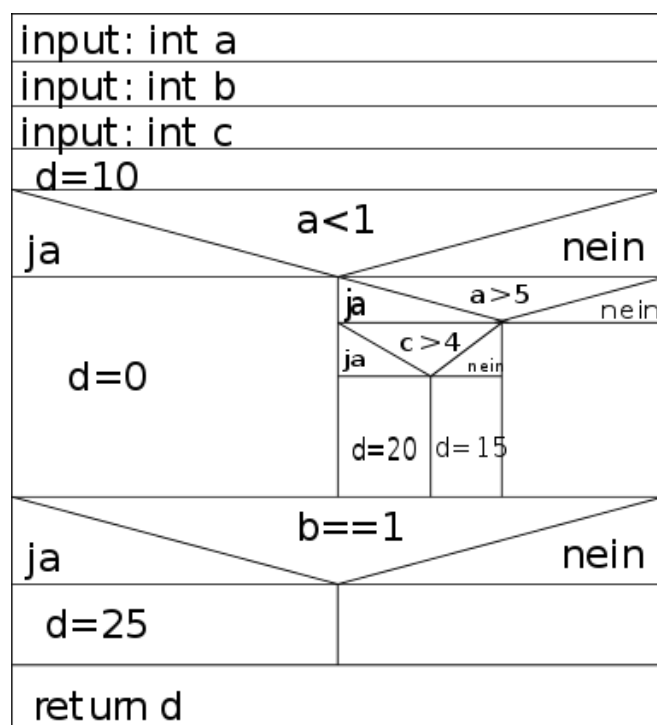
## 4 Petrinetz



Geben Sie den Erreichbarkeitsgraphen an.

## 5 Testen

Geben Sie in den folgenden Aufgaben jeweils eine Mengen von Tupeln der Form  $(a, b, c)$  mit  $a, b, c \in \{0, 1, 2, 3, 4, 5, 6\}$  an.



a) Geben Sie die minimale Testfallmenge für vollständige Anweisungsüberdeckung an.

- b) Geben Sie die minimale Testfallmenge für vollständige Verzweigungsüberdeckung an.
- c) Geben Sie die minimale Testfallmenge für vollständige Pfadüberdeckung an.

## 6 Refactoring

---

```
public class Employee {
    public String firstName;
    public String lastName;
    public String employeeType;
    public int employeeId;
    public String street;
    public String city;
    public String state;
    public int postalCode;
    private List<EmployeeOrder> employeeOrders = new ArrayList<>();

    public void addEmployeeOrder(EmployeeOrder eo) {
        employeeOrders.add(eo);
        System.out.println("Employee order added");

        System.out.println("Employee: First name: " + firstName + " Last Name: " + lastName);
        System.out.println("Employee: ID: " + employeeId);
        System.out.println("Number of employee orders: " + employeeOrders.size());
    }

    public void removeEmployeeOrder(int index) {
        employeeOrders.remove(index);
        System.out.println("Employee order removed");

        System.out.println("Employee: First name: " + firstName + " Last Name: " + lastName);
        System.out.println("Employee: ID: " + employeeId);
        System.out.println("Number of employee orders: " + employeeOrders.size());
    }

    public double getBonus() {
        switch (employeeType) {
            case "Manager":
                return getYearlySalary() * 0.05 + 500;
            case "Engineer":
                if (employeeOrders.size() > 50)
                    return getYearlySalary() * 0.04 + 400;
                else
                    return getYearlySalary() * 0.03 + 300;
            case "other employee":
                if (employeeOrders.size() > 50)
                    return getYearlySalary() * 0.02 + 200;
                else
                    return getYearlySalary() * 0.01 + 100;
        }
        return 0.0;
    }

    public void printAddress() {
```

```
    System.out.println("Street: " + street);
    System.out.println("City: " + city);
    System.out.println("State: " + state);
    System.out.println("Postal Code: " + postalCode);
}
}
```

---

- a) Nennen Sie drei aus der Vorlesung bekannte Arten des Refactorings, die auf diese Klasse sinnvoll angewendet werden können.
- b) Zeichnen Sie ein UML-Klassendiagramm, welches das Softwaresystem nach dem Refactoring beschreibt.