

Prüfungsfragen Algorithmische Sprachen 2000

Syntaxanalyse, Compilertechnik,
Grundlagen objektorientierter Softwareentwicklung (GOOSE)
Prof. Schneider, Beisitzer: G. Kokai
Oktober 2000

Bemerkungen zu Prüfung und Prüfer

- Ergebnis: 1,0
- Prüfer gibt Hilfestellungen
- Schwerpunkt liegt auf Verständnis, es wird wenig Detailwissen, keine formalen Definitionen, Beweise, etc. gefragt.
- Das Skript reicht zum Lernen, für Compiler-technik ist Wilhelm/Maurer zu empfehlen.

Syntaxanalyse

- Welche Verfahren können alle kf. Grammatiken parsen? (CKY, Earley)
- Earley-Verfahren erläutern (Scanner, Predictor, Completer-Schritte)
- In welcher Reihenfolge werden diese Schritte ausgeführt? (Predictor- und Completer evtl. mehrmals)
- Kann man die mehrmalige Ausführung von Predictor- und Completer- Schritten verhindern? (Ja, Epsilon-Ableitung merken)
- Komplexität des Verfahrens? ($O(n^3)$)
- Worin unterscheiden sich bei den in der Praxis verwendeten Verfahren top-down und bottom-up Vorgehen? (LL/LR-Verfahren, Ableitungs-, Reduktions- und Shiftklassen)
- Welches Verfahren wollen Sie erklären? (Ich habe mich für LL entschieden)
- Warum brauchen wir Oberklassen?
- Voraussetzung an Oberklassen? (Müssen disjunkt sein)

- Was bedeutet es, wenn Ableitungsklassen nicht disjunkt sind? (Grammatik ist mehrdeutig)

Compilertechnik

- Was liegt bei klassischen Programmiersprachen auf dem Stack? (Verwaltungsinformationen, Parameter, Variablen)
- Was ist der Extreme Pointer?
- Was sind Static und Dynamic Link?
- Wozu braucht man einen Static Link? Der Compiler weiss doch welche Variable verwendet werden muss. (Aber nicht wo die Daten der entsprechenden Prozedur im Speicher liegen).
- Wie haben die Static Links C beeinflusst? (Alle Prozeduren müssen auf oberster Ebenen liegen, daher keine SL nötig).
- Wie wird bei oo. Sprachen Einfachvererbung realisiert? (Methodentabelle, verschiedenen Sichten)
- Probleme der Mehrfachvererbung? (Namenskonflikte, abhängige/unabhängige Vererbung)

Grundlagen objektorientierter Softwaretechnik (Goose)

- Was sind Use Cases?
- Was haben Use Cases mit OO zu tun? (Beschreiben Zusammenspiel der Objekte)
- Aus welchen Richtungen stammen Java und Eiffel? (basieren auf C bzw. Pascal, Eiffel ist stark durch formale Spez. beeinflusst)

Damit wir auch in Zukunft aktuelle Prüfungsfragen haben, sind wir auf Deine Mithilfe angewiesen. Bitte maile uns die Fragen Deiner Prüfung, ein Formular dazu findest Du auf unserer Homepage.
--