



# Universität Karlsruhe (TH)

## Lehrstuhl für Programmierparadigmen

Compilerpraktikum SS 2010

Dozent: Prof. Dr.-Ing. G. Snelting

Betreuer: Matthias Braun

Betreuer: Sebastian Buchwald

<http://pp.info.uni-karlsruhe.de/>

[snelting@ipd.info.uni-karlsruhe.de](mailto:snelting@ipd.info.uni-karlsruhe.de)

[matthias.braun@kit.edu](mailto:matthias.braun@kit.edu)

[sebastian.buchwald@kit.edu](mailto:sebastian.buchwald@kit.edu)

Übungsblatt 10

Ausgabe: 30.06.2010

Besprechung: 07.07.2010

### Aufgabe 1: Backend Rahmen

- Für einige Typen werden Sie Java-Deskriptoren erzeugen müssen. Schreiben Sie eine Funktion die für einen gegebenen Firm-Typen einen Java-Deskriptor ausgibt.
- Schreiben Sie eine Funktion die für alle Klassen in ihrem Programm `emitClass(...)` aufruft.
- Implementieren Sie `emitClass`: Jasmin Code für den Header und Standardkonstruktor der Klasse soll ausgegeben werden. Danach `emitField(...)` und `emitMethod(...)` für alle Felder und Methoden aufgerufen werden.

### Aufgabe 2: Übersetzen von Methoden

#### 2.1 Analysephase

- Die Analyse durchläuft den Firm-Graph (`Graph.walkTopological`) und erstellen dabei Befehlslisten für jeden Grundblock.
- Bedenken Sie, dass Sprungbefehle stets als letzte in einem Grundblock ausgeführt werden müssen (dies ist nicht explizit über Datenabhängigkeiten gegeben).
- Teilen Sie Variablennummern zu wie auf den Praktikumsfolien empfohlen.
- Erstellen Sie für jeden Block eine Liste mit  $\phi$ -Instruktion im Nachfolgerblock.

#### 2.2 Code für einen Grundblock erzeugen

- Implementieren Sie die Funktionen `pushValue` und `createValue` wie auf den Folien angedeutet.
- Implementieren Sie die Codeausgabe für Wurzelknoten (nutzen Sie die Reihenfolge die in den Befehlslisten vorgegeben wird).
- Erzeugen Sie die `Store`-Befehle für die  $\phi$ -Instruktionen im Nachfolgerblock.
- Initial genügt es `.limit stack 100` zu benutzen.

#### 2.3 Verbesserungen (Optional)

- Bei einer geschickten Anordnung der Grundblöcke kann man sich `goto` Befehle sparen, falls von einem Block direkt in den Nächsten gesprungen wird. Wie sieht eine gute Blockanordnung aus und wie findet man sie?
- Die maximale Höhe des Stacks wird im Moment mit 100 angegeben. Mit Hilfe einer Stacksimulation lässt sie sich für einen Grundblock genau bestimmen.
- Wenden Sie die Methoden aus der Compiler 1 Vorlesung (Stichwort starkes Normalformtheorem) an um die maximale Höhe möglichst niedrig zu halten.