



Universität Karlsruhe (TH)

Lehrstuhl für Programmierparadigmen

Compilerpraktikum SS 2009

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

Betreuer: Matthias Braun

Betreuer: Jürgen Graf

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

snelting@ipd.info.uni-karlsruhe.de

braun@ipd.info.uni-karlsruhe.de

graf@ipd.info.uni-karlsruhe.de

Übungsblatt 7

Ausgabe: 24.06.2009

Besprechung: 01.07.2009

Aufgabe 1: Jasmin Assembler

Die an Javabytecode angelehnte Assemblersprache Jasmin finden Sie unter <http://jasmin.sf.net>.

- Installieren Sie den Jasmin Assembler und lesen Sie sich in die Dokumentation ein. Die meisten Befehle sind relativ wenig dokumentiert, da sie sich exakt wie die gleichnamigen Befehle aus der *Java Virtual Machine Specification* verhalten. Sie müssen zum Einarbeiten folglich beide Dokumente verwenden.
- Schreiben Sie die Programme Opt1.java und Opt2.java aus dem Wiki als Assemblerprogramm.
- Die Unterschiede zu Bytecode wurden schon teilweise angesprochen. Gibt es noch weitere Unterschiede?
- Finden Sie heraus wie man Jasmin am Besten in ihren bestehenden Compiler integrieren kann.
- Schreiben Sie die Klasse welche die main Methode Ihres Compilers enthält in Jasmin Assembler und verwenden Sie die resultierende .class Datei testweise in ihrem Compiler. Funktioniert noch alles?

Aufgabe 2: Firm nach Jasmin Assembler

Ihr endgültiges Ziel wird das Erzeugen von Jasmin Assemblercode sein. Dazu sind einige vorbereitende Schritte notwendig.

- Erzeugen Sie passende Deskriptoren für alle Methoden und Felder in ihren Firmgraphen. Der Aufbau dieser Deskriptoren ist in der *Java Virtual Machine Specification* (Abschnitt 4.3) beschrieben.
- Erzeugen Sie Jasmin Assembler Code für alle Klassen, Methoden und Felder. Die Methodenrümpfe können Sie diese Woche noch leer lassen.
- Überlegen Sie sich wie Sie für einen einzelnen Block im Firmgraph Assembler Code erzeugen würden. Worin sehen Sie problematische Stellen?