

## Codeerzeugung – Teil 2

(24 Punkte)

Vervollständigen Sie Ihren Compiler, indem Sie die fehlenden Teile der Codeerzeugung gemäß der in den Unterlagen ausgegebenen Spezifikation der *MicroJava*-VM (siehe VO-Skriptum, Kapitel 5, Seiten 12-16) hinzufügen.

In dieser Übung müssen Sie die Codeerzeugung um Sprünge erweitern, dazu sind Sprungmarken (*Labels*) nötig. Ein Label verwaltet ein Sprungziel oder solange das Sprungziel unbekannt ist eine Liste von Sprüngen zu diesem Sprungziel. Vervollständigen Sie die Klasse *Label* mit den Methoden *put()* und *here()*. Verwenden Sie diese Klasse für die Codeerzeugung von Bedingungen und Schleifen.

Wegen *InlineVarDecl* ist die maximale Anzahl der lokalen Variablen nicht vor dem Parsen der Methode bekannt. Um dennoch genügend Platz im Aktivierungssatz zu reservieren muss der zweite Operand der *enter* Instruktion im nachhinein gepatcht werden. Merken Sie sich dazu die Codeadresse des Operanden und schreiben Sie diese nachdem die Methode verarbeitet wurde.

Prüfen Sie alle Kontext- und sonstigen Nebenbedingungen der *MicroJava*-Sprachdefinition und geben Sie entsprechende Fehlermeldungen aus.

Testen Sie Ihren Generator ausführlich. Lassen Sie Ihre Testprogramme auch auf der *MicroJava*-VM laufen (*java ssw.mj.Run <obj-Datei>*).

Auf der Übungsseite finden Sie die Testprogramme *TestProgram.mj*, *StudentList.mj* und *Trap.mj*. *TestProgram.obj* gibt "1234" aus. *StudentList.obj* legt eine Liste von Studenten an und durchsucht diese. *Trap.obj* bricht mit einem Fehler ab, weil in der Funktion *int Trap()* das *return*-Statement fehlt.

## Abgabe und Hinweise

Die Abgabe der Übungen muss elektronisch erfolgen. Geben Sie folgende Dateien ab:

- Elektronisch in das Repository: Alle Quellcode-Dateien, die zum Ausführen des Compilers benötigt werden (Packages *ssw.mj*, *ssw.mj.codegen*, *ssw.mj.impl* und *ssw.mj.symtab*), also auch alle Klassen der Abgabe. Die Verzeichnis-Struktur muss erhalten bleiben.
- `svn://ssw.jku.at/2016W/UB/k<MatrNr>/branches/UE6`

JUnit Testfälle: `ScannerTest`, `ParserTest`, `SymbolTableTest`, `SimpleCodeGenerationTest`,  
`CodeGenerationTest`