

Zuname _____ Vorname _____ Matr.-Nr. _____

Übungsgruppe

- | | | |
|--------------------------|--------------|--------------|
| <input type="checkbox"/> | 1 (Kotzmann) | Do 1015-1145 |
| <input type="checkbox"/> | 2 (Kotzmann) | Do 1345-1515 |
| <input type="checkbox"/> | 3 (Wimmer) | Do 1015-1145 |

Punkte _____ korr. _____

Letzter Abgabetermin

Mittwoch, 09.11.2005, 20¹⁵ Uhr

Syntaxanalysator (Parser)

(24 Punkte)

Implementieren Sie den Syntaxanalysator für *MicroJava* im rekursiven Abstieg. Verwenden Sie dabei das vorgegebene Gerüst in der Klasse *Parser*. Jede Regel der Grammatik (siehe VO-Skriptum, Kapitel 5.2, Seite 8) soll dabei durch eine eigene Methode vertreten sein, welche die Top-Down-Erkennung realisiert.

Die Schnittstelle des Parsers nach außen ist durch die Methode *Parser.parse()* definiert, mit der man die Analyse startet.

Benutzen Sie außerdem die drei in der Vorlesung vorgestellten Methoden *scan()*, *check()* und *error()*. Bei der Fehlerbehandlung verwenden Sie (vorläufig) die „Panic Mode“-Strategie, d.h. Sie brechen die Syntaxanalyse beim ersten Fehler ab (nachdem Sie eine entsprechende Fehlermeldung über die Klasse *Error* ausgegeben haben).

Abgabe

Die Abgabe der Übungen 2 – 6 muss auf Papier und elektronisch erfolgen. Geben Sie folgende Dateien ab:

- Ausgedruckt auf Papier: *Parser.java* (andere Dateien müssen für diese Übung nicht verändert werden).
- Elektronisch als ZIP-Datei: **Alle** Dateien, die zum **Ausführen** des Compilers benötigt werden (Packages *ssw.mj*, *ssw.mj.codegen* und *ssw.mj.syntab*), also auch alle Klassen der Angabe. Auf die Datei *messages.properties* nicht vergessen. Die Verzeichnis-Struktur muss in der ZIP-Datei erhalten bleiben.
- **Nicht abzugeben:** JUnit-Testfälle, *.class*-Dateien, Projekt-Dateien von IDEs.