

Zuname _____ Vorname _____ Matr.-Nr. _____

Übungsgruppe

- | | | |
|----------------------------|--------------|---------------------------------------|
| <input type="checkbox"/> 1 | (Löberbauer) | Do 10 ¹⁵ -11 ⁴⁵ |
| <input type="checkbox"/> 2 | (Löberbauer) | Do 13 ⁴⁵ -15 ¹⁵ |
| <input type="checkbox"/> 3 | (Würthinger) | Do 10 ¹⁵ -11 ⁴⁵ |

Punkte _____ korr. _____

Letzter Abgabetermin

Mittwoch, 11.11.2009, 18⁰⁰ 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 4.2, Seite 7) muss 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 die drei in der Vorlesung vorgestellten Methoden *scan()*, *check()* und *error()*. Zur 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 *Errors* ausgegeben haben.

Im Vorlesungs-Skriptum sind alle Methoden der Klasse *Parser* als *static* deklariert. In der Übung verwenden wir dynamische Methoden, zB um die JUnit-Testfälle zu vereinfachen. Daher muss ein Objekt der Klasse *Parser* angelegt werden, dem im Konstruktor der *Scanner* übergeben wird.

Abgabe

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

- Ausgedruckt auf Papier: *Parser.java* (nur diese Datei muss verändert werden).
- Elektronisch in das Repository: **Alle** Quellcode-Dateien, die zum **Ausführen** des Compilers benötigt werden (Packages *ssw.mj*, *ssw.mj.codegen* und *ssw.mj.symtab*), also auch alle Klassen der Angabe. Die Verzeichnis-Struktur muss erhalten bleiben.
- Taggen Sie die Abgabe als: Abgabe UE3