

## Syntaxanalyse (Parser)

(24 Punkte)

Implementieren Sie die Syntaxanalyse für *MicroJava* im rekursiven Abstieg. Verwenden Sie dabei das vorgegebene Gerüst in der Klasse *ParserImpl*. Jede Regel der Grammatik (siehe *MicroJava* Grammatik auf der Übungshomepage) muss durch eine eigene Methode vertreten sein, welche die Top-Down-Erkennung realisiert.

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

Nutzen 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 *ScannerImpl* übergeben wird (bereits programmiert).

Für die Implementierung sollen ausschließlich die Subklassen im *ssw.mj.impl* Package verändert werden.

## 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.syntab*), also auch alle Klassen der Angabe. Die Verzeichnis-Struktur muss erhalten bleiben.

`svn://ssw.jku.at/2017W/UB/k<MatrNr>/branches/UE3`

JUnit Testfälle: *ScannerTest*, *ParserTest*