

Zuname _____

Matr. Nr. _____

Übungsgruppe

- 1 (Wöß) Do 10¹⁵-11⁴⁵
- 2 (Wöß) Do 12⁴⁵-14¹⁵
- 3 (Rammerstorfer) Do 14³⁰-16¹⁵

Punkte _____

korr. _____

Letzter Abgabetermin

Donnerstag, 8.11.2001, 8¹⁵ Uhr

Syntaxanalysator

a) Rekursiver Abstieg (26 Punkte)

Implementieren Sie den Syntaxanalysator *Parser.java* für *MicroJava* im rekursiven Abstieg. Jede Regel der Grammatik soll dabei durch eine eigene Methode vertreten sein, welche die top-down-Erkennung realisiert. Die Schnittstelle des Parsers nach außen ist durch die Methode *parse* definiert, mit der man die Analyse startet. Benutzen Sie außerdem die drei in der Vorlesung vorgestellten Methoden *scan*, *check* und *synError*.

Verwenden Sie für Ihren Parser folgende Schnittstelle:

```
package MJ;

public class Parser {
    private static Token t;           // last recognized token
    private static Token la;         // lookahead token (not recognized)
    private static int sym;         // always holds la.kind

    private static void scan() {...} // read ahead one symbol
    private static void check(int expected) {...} // verify symbol and read ahead
    private static void synError(String msg) {...} // print error message

    private static void Program() {...} // recognition of rule 'Program'
    private static void ConstDecl() {...} // recognition of rule 'ConstDecl'
    ...
    private static void Mulop() {...} // recognition of rule 'Mulop'

    public static void parse() {...} // starts the analysis
}
```

b) Compiler-Hauptprogramm + Testen (6 Punkte)

Schreiben Sie einen Testtreiber *Compiler.java* für Ihren Syntaxanalysator, den Sie in weiterer Folge auch als Hauptprogramm des gesamten Compilers verwenden. Beim Aufruf soll man per Programmargument die zu analysierende Datei (*xxx.mic*) angeben können. Sämtliche Initialisierungen des Compilers (bisher nur Initialisierung des Scanners und Parsers) und der Start der Übersetzung müssen im Hauptprogramm enthalten sein.

```
package MJ;

public class Compiler {
    public static void main(String [] arg) {...}
}
```

Testen Sie den Parser ausführlich mit einem umfangreichen *MicroJava*-Programm! Testen Sie auch fehlerhafte Programme!