

# Compilerbau

Sommersemester 2006

10. Übungsblatt

Abgabetermin: Mo. 10. Juli 2006, in der Vorlesung

Aufgabe 1: Parser: Interpreter

20 Punkte

Gegeben sei die Sprache MiniJava, deren Syntax durch die folgende Grammatik definiert wird:

```
program → decl* stmt*
decl   → type name (, name)* ;
type   → int
stmt   → ; | {stmt*} | name = expr; | name = read (); | write (expr); |
        if (cond) stmt | if (cond) stmt else stmt | while (cond) stmt
expr   → number | name | (expr) | unop expr | expr binop expr
unop   → -
binop  → + | -
cond   → true | false | (cond) | expr comp expr |
        bunop cond | cond bbinop cond
comp   → == | <
bunop  → !
bbinop → &&
number → sign? digit digit*
sign   → + | -
digit  → 0 | ... | 9
name   → letter (letter|digit)*
letter → $ | - | a | ... | z | A | ... | Z
```

Schreibe einen Interpreter für die MiniJava Sprache. Die Semantik ist die von Java, wobei `read` ein Integer von der Standardeingabe liest, und `write` ein Integer auf der Standardausgabe ausgibt. Der Interpreter soll statisch überprüfen, dass jede Variable, die benutzt wird, vorher deklariert wurde. Der Name der MiniJava Eingabedatei soll aus der Kommandozeile gelesen werden.

Sende per E-Mail alle benötigten Dateien inklusive CUP- und JFLEX-Spezifikationen. Spezielle Compileroptionen sind ebenfalls anzugeben.

**Hinweis:** Die Lösung besteht im Wesentlichen aus zwei Teilen:

1. Aufbauen des Syntaxbaumes

2. Interpretierung des Programms laut Syntaxbaum.

Eine sinnvolle Modellierung des Syntaxbaumes könnte dabei wie folgt aussehen:

1. Es gibt eine abstrakte Oberklasse **Statement** mit einer abstrakten Methode **execute**. Ein Objekt einer konkreten Unterklasse repräsentiert ein Statement. Ein Aufruf der Methode **execute** für ein solches Objekt führt das Statement aus.  
Statements wie z.B. das **if**-Statement leiten sich also von der Klasse **Statement** ab und repräsentieren den **then**- und **else**-Zweig durch Referenzen auf entsprechende **Statement**-Objekte (Rekursive Komposition).
2. Ähnlich, gibt es eine abstrakte Oberklasse **Expr** mit einer abstrakten Methode **eval**.