

Compilerbau

Sommersemester 2005

10. Übungsblatt

Abgabetermin: 4. Juli 2005, 12 Uhr

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    → ; | {decl* 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 Email alle benötigten Dateien inklusive CUP- und JFLEX-Spezifikationen, sowie einem Skript zur vollständigen Kompilierung des Parsers (von den Spezifikationen bis hin zum ausführbaren Java-Programm)!

Links zu Software und Dokumentation:

Scanner-Generator JFLEX: <http://jflex.de/>

Parser-Generator CUP: <http://www2.in.tum.de/projects/cup/>