

Compilerbau

Sommersemester 2005

9. Übungsblatt

Abgabetermin: 27. Juni 2005, 12 Uhr

Aufgabe 1: Parser: Syntax-Highlighting

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 Parser, der eine einheitliche Einrückung für MiniJava-Programme erzeugt. Die Namen der Ein- und Ausgabedateien sollen aus der Kommandozeile gelesen werden. Verwende zur Implementierung den **Scanner Generator JFlex** sowie den **Parser Generator CUP**! Ein Beispiel-Parser, der als Grundlage für diese Aufgabe verwendet werden kann, sowie Muster für die Eingabe und die erwartete Ausgabe sind auf der Vorlesungsseite zu finden (<http://www2.in.tum.de/lehre/vorlesungen/SS05/compiler/>).

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/>

Eingabebeispiel:

```
int x,y;int n,n1,n2;
x=read();y=read();
n=-1;n1=-1;n2=-1;
while (x!=y){n=n+1;
if (n<1000) if (x<y){n1=n1+1;y=y-x;}
else{n2=n2+1;x=x-y;}
}
write(n);write(n1);write(n2);write(x);
```

Erwartete formatierte Ausgabe:

```
int x,y;
int n,n1,n2;

x = read();
y = read();
n = -1;
n1 = -1;
n2 = -1;
while (x != y){
    n = n + 1;
    if (n < 1000)
        if (x < y){
            n1 = n1 + 1;
            y = y - x;
        }
        else{
            n2 = n2 + 1;
            x = x - y;
        }
}
write(n);
write(n1);
write(n2);
write(x);
```