

**Praktische Informatik 1**  
(Vorlesung von Prof. Bothe)**WS 05/06**

## Übungsblatt 5

Abgabe: 5. - 7.12.05 in den Übungen

Die Lösung dieses Übungsblatts soll in Gruppen von je 2 Personen erfolgen. Die Lösungen sind inklusive Lösungsweg auf Papier abzugeben. Auf jedem Lösungsblatt sind die Matrikelnummern beider Gruppenmitglieder anzugeben!

**Aufgabe 1 (2 Punkte)**

Java ist eine Sprache mit strengem Typsystem (*strongly typed language*). Assemblersprachen haben üblicherweise kein Typsystem. Worin liegt der Unterschied, und was sind die Vorteile der strengen Typisierung? Gibt es Nachteile?

**Aufgabe 2 (2 Punkte)**

Es seien  $x$  und  $y$  vom gleichen Typ. Welcher Typ muss das sein, damit möglichst viele der folgenden Ausdrücke korrekt sind? Welche Werte haben diese korrekten Ausdrücke? Welche Ausdrücke sind in keinem Fall korrekt?

- a)  $x==y=x$
- b)  $(x==y)=x$
- c)  $x==(y=x)$
- d)  $x=y==x$
- e)  $(x=y)==x$
- f)  $x=(y==x)$

**Aufgabe 3 (5 Punkte)**

Schreiben Sie ein Java-Programm, das drei ganze Zahlen einliest und verarbeitet. Die Summe, das Produkt, das arithmetische Mittel und das Maximum der drei Zahlen sind auszugeben. Die Zahlen können wie in den Vorlesungsbeispielen mit `Keyboard.readInt()` eingelesen werden.

**Aufgabe 4 (5 Punkte)**

Leiten Sie mit Hilfe der auf der nächsten Seite angegebenen Grammatik das folgende Programmfragment ab. Verwenden Sie jeweils nur direkte Ableitungen. Sie dürfen jedoch pro Schritt mehrere Nicht-Terminalsymbole ersetzen.

Beispiel: Angenommen Sie hätten (für ein anderes Programmfragment) bereits

```
... -> Eingabe ';' Ausgabe ';'
abgeleitet, dann wäre ein möglicher nächster Schritt
-> 'READ' '(' Variable ')' ';' 'WRITE' '(' Ausdruck ')' ';'.
```

Programmfragment:

```
BEGIN
  READ (z) ;
  FOR i:=1 TO z DO
    WRITE (i) ;
  ENDFOR ;
END ;
```

Grammatik:

**G = [A, M, Block, R]**

**A** und **M** ergeben sich aus den nachfolgenden Regeln. Elemente aus **A** sind in Hochkommas eingefasst (z.B. ':='). Die restlichen Symbole (außer natürlich [, ], | und .) bilden die Metasymbole (Elemente von **M**).

**R:**

```
Block ::= 'BEGIN' Anweisung 'END' ';' .
Anweisung ::= Test ';' | Schleife ';' | Eingabe ';' | Ausgabe ';'
            | Zuweisung ';' | Anweisung Anweisung .
Zuweisung ::= Variable ':=' Ausdruck .
Ausdruck ::= Variable | ganzeZahl | reelleZahl
            | Variable ArithmOperator Variable | Ausdruck ArithmOperator Ausdruck
            | Ausdruck LogOperator Ausdruck | '(' Ausdruck ')'.
Variable ::= 'x' | 'y' | 'z' | 'i' | 'k' | 'm'.
ArithmOperator ::= '+' | '-' | '*' | '/' | 'DIV' | 'MOD'.
LogOperator ::= 'AND' | 'OR'.
VergOperator ::= '=' | '<>' | '<' | '>' | '<=' | '>='.
Vergleich ::= Ausdruck VergOperator Ausdruck | Vergleich LogOperator
            Vergleich | 'NOT' Vergleich | '(' Vergleich ')'.
Test ::= 'IF' Vergleich 'THEN' Anweisung ['ELSE' Anweisung] 'ENDIF'.
Schleife ::= 'WHILE' Vergleich 'DO' Anweisung 'ENDWHILE'
            | 'REPEAT' Anweisung 'UNTIL' Vergleich
            | 'FOR' Variable ':=' Ausdruck 'TO' Ausdruck 'DO' Anweisung 'ENDFOR'.
Eingabe ::= 'READ' '(' Variable ')'.
Ausgabe ::= 'WRITE' '(' Ausdruck ')'.

```

### Aufgabe 5 (6 Punkte)

Das folgende Programm soll eine nichtnegative ganze Zahl  $n$  und einen String  $s$  einlesen und  $s$   $n$ -mal ausgeben. Das Programm enthält (mindestens) 7 Fehler. Finden und nennen Sie wenigstens 6 davon und klassifizieren Sie sie in folgende Kategorien (siehe auch Kapitel 7, Folie 28):

- Lexikalische Fehler
- Fehler bezüglich der kontextfreien Syntax
- Fehler bezüglich der kontextabhängigen Syntax
- Semantische Fehler

```
class Papagei {
    public static void Main(String[] args){
        System.out.println("Ich wiederhole alles ganz oft.");
        System.out.println("Zahl:");
        int z1 = Keyboard.readDouble();
        System.out.println("String: ");
        String s = Keyboard.readString();
        String result = "";
        While(z1>0){
            if(z1 % 2 == 1)
                result += s
            z1 /= 2;
            s += s;
        }
        System.out.println(s);
    }
}
```