

Übung 5

Ein- und Ausgaben, Programmfluss-Steuerung, Algorithmen

Aufgabe 1: Inkrement / Dekrement (1 Punkt)

Welche Werte haben die folgenden Ausdrücke und Variablen in JAVA nach der Anweisung (bzw. nach dem Semikolon), wenn vor jeder Anweisung **a** den Startwert 2 und **b** den Startwert 12 hat?

1. `a--;`
2. `--b;`
3. `c = b-- - ++a;`
4. `a = ++a + --b;`
5. `a = a++ + --b;`

Aufgabe 2: Wertebereiche, Typkonversion (2 Punkte)

- Schreiben Sie ein Java-Programm, das Ihnen die jeweils größt- bzw. kleinstmögliche **BYTE**, **SHORT**, **INT**, **LONG** Zahl auf dem Bildschirm ausgibt. Addieren Sie anschließend zur größten **LONG-Zahl** den Wert 1 und geben das Ergebnis auf dem Bildschirm aus. Was passiert?
- Warum führt das folgenden Code-Beispiel zu Problemen?

```
int z1 = 0;
long z2 = 1000;
z1 = z2;
float z3 = 1.0;
```

Aufgabe 3: Switch (2 Punkte)

Mina Ralwasser möchte ein Programm entwickeln, welches ihr nach Eingabe einer Klausurnote (als Zahl) die entsprechende Note (als Text), also **1** → **Sehr gut**; **2** → **Gut** ..., ausgibt. Wird eine Note schlechter als 4 erzielt, soll „Durchgefallen“ ausgegeben werden.

Helfen Sie Mina bei der Umsetzung. Verwenden Sie zur Realisierung die **switch-Anweisung**.

Aufgabe 4: Schaltjahr (1 Punkt)

Schreiben Sie ein Java-Programm, das für ein Jahr n prüft, ob es sich um ein Schaltjahr handelt. Recherchieren Sie dazu im Internet, unter welchen Bedingungen n ein Schaltjahr ist.

Aufgabe 5: Schleifen (4 Punkte)

Ermitteln Sie mit Hilfe einer passenden Schleifenkonstruktion:

- Die Summe der Zahlen zwischen 1 und n die **nicht** durch 3 teilbar sind. (**for-Schleife**)

Eingabe $n = 6 \rightarrow 1 + 2 + 4 + 5 = 12$

- Die Quadratzahlen zwischen 1 und n . (**while-Schleife**)

Eingabe $n = 3 \rightarrow 1^2 = 1; 2^2 = 4; 3^2 = 9$

- Alle Primzahlen $\leq n$.

Eingabe $n = 10 \rightarrow 2, 3, 5, 7$

Die Zahl n soll jeweils von der Konsole eingelesen werden!

Aufgabe 6: Was gibt das Programm aus und warum? (2 Punkte)

```
public class Traceme {
    public static void main(String[] args) {
        int x = 1; int y = 2; int z = 1;
        x = y++++--z-++x;
        while(x <= 10){
            x+=20;
            z*=2;
            y=y++ + ++y;
        }
        x = 10;
        x = x++;
        x /= 3;
        y *= 2;
        z += y *= x;
        x*=x+=3;
        y = x++ - x;
        System.out.println("x = " + x + " y = " + y + " z = " + z);
    }
}
```

- Vollziehen Sie die einzelnen Anweisungen „**manuell**“ nach und fertigen Sie einen sogenannten **Trace** an, z.B. mit Hilfe von **Breakpoints** und dem **Debug-Modus** von **eclipse**. Ein Trace hält in tabellarischer Form die Veränderung aller Variablen nach jeder Code-Zeile fest (das bedeutet, in jeder Zeile stehen die aktuellen Werte aller jeweils gültigen Variablen).

Welche Werte werden am Ende auf dem Bildschirm ausgegeben?