

# Programmierkurs Tag 1 - Lösungen

## Hinweise zur Musterlösung

Die Lösungsvorschläge zu den einzelnen Aufgaben beinhalten sofern nicht anders angegeben jeweils nur den lösungsrelevanten Quelltext. Dieser muss sich innerhalb der `public static void main(String[] args)` Methode befinden.

## Aufgabe 1.1 (Hello World)

```
System.out.println("Hello World");
```

## Aufgabe 1.2

Das Programm soll den Text ausgeben

„Ich bin im 1. Semester.“

Wobei das Semester aus einer Variablen (z.B. `int semester;`) ausgegeben werden soll.

```
int semester = 1;
System.out.println("Ich bin im " + semester + ". Semester.");
```

## Aufgabe 1.3

Schreibe ein Programm, das Meter in Fuß umrechnet.

Das Programm soll auch Kommazahlen akzeptieren. Die Meter, die in Fuß umgerechnet werden sollen, stehen wieder in einer entsprechenden Variable. (Keine interaktive Eingabe.)

Ein Fuß (1 ft) = 30,48 cm.

```
double meter = 4.2;
System.out.println(meter / 0.3048);
```

## Aufgabe 1.4

Ein Mobiltarif kostet einen Grundbetrag (13,37 €) und je Minute 42 Cent.

Schreibe ein Programm, welches für eine von euch gewählte Anzahl von Minuten (soll in eine Variable geschrieben werden, eine graphische Eingabe ist nicht verlangt) den Gesamtbetrag berechnet.

```
int minuten = 15;
System.out.println("Gesamtbetrag: " + (13.37 + 0.42 *
    minuten) + "€");
```

## Aufgabe 1.5

Lege ein Programm an, in dem es zunächst folgende Integer Variablen gibt: *einser*, *zweier*, *dreier*, *vierer*, *fuenfer*

Diese fünf Variablen beinhalten jeweils die Anzahl der Studenten, die die entsprechende Note bei einer Klausur erreicht haben. Berechne nun aus diesen Noten den Notendurchschnitt der Klausur.

```
int einser = 3, zweier = 4, dreier = 6, vierer = 7,
    fuenfer = 2;
double durchschnitt =
    (einser * 1 + zweier * 2 + dreier * 3 + vierer * 4 +
     fuenfer * 5)
    / (double)(einser + zweier + dreier + vierer +
            fuenfer);
System.out.println("Durchschnitt:␣" + durchschnitt);
```

## Aufgabe 1.6

Welchen Wertetyp und welchen Wert hat das Ergebnis folgender Ausdrücke? Den Wert kannst du prüfen indem du den Ausdruck einfach von einem Programm ausgeben lässt. Die Lösung zu den Wertetypen werden morgen mit der Musterlösung besprochen.

- (a)  $5 / (\text{int})2.0 * 3 = 2 * 3 = 6$  (**int**)
- (b)  $(\text{double})(3 / 2) / 2 = (\text{double})1 / 2 = 0.5$  (**double**)
- (c)  $((\text{int})(1.2 * 1.3)) * (\text{short})10 = 1 * 10 = 10$  (**int**)
- (d)  $1 + 2.0 / 5.0f - 10 = 1 + 0.6 - 10 = -8.6$  (**double**)

## Aufgabe 1.7

Was ist an den folgenden Zeilen falsch?

```
int dieAntwort = "zweiundvierzig"; // eine Zeichenkette
    kann nicht in eine Variable vom Typ "int" (Ganzzahl)
    gespeichert werden
int einkommenIn€ = 1337; // € ist unzulässig
int riesigeZahl = 5400000000; // Maximum für int ist
    2147438647
int 42istdieAntwort = 23; // Variablen-Namen dürfen
    nicht mit Zahlen beginnen
int berechnung = "23" - 42; // hier ist 23 ein string
    (eine Zeichenkette). Von dieser kann keine Zahl
    subtrahiert werden. Auch können Strings nicht in
    Variablen eines Zahlentypes gespeichert werden.
```

## Aufgabe 1.8

Beschreibe in eigenen Worten was die folgende Funktion tut: (Gehe der Einfachheit halber davon aus, dass jeder Student jedes Fach belegt)

```
int studenten = 150;
int faecher = 8;
int durchfallquoteInProzent = 20;
int ergebnis =
    studenten*faecher*durchfallquoteInProzent/100;
System.out.println(ergebnis);
```

### SCHLECHT:

Es werden die Variablen `studenten`, `faecher` und `durchfallquote` mit vorgegebenen Werten initialisiert. Danach wird die Anzahl der zu wiederholenden Kurse berechnet und in die Variable `ergebnis` gespeichert. Danach wird die Variable `ergebnis` ausgegeben.

### GUT:

Es wird berechnet wie viele Klausuren insgesamt nicht bestanden werden. Dazu wird die Anzahl der Studenten mit der Anzahl der Fächer multipliziert (= Gesamtzahl an Klausuren) und diese dann mit der Durchfallquote multipliziert, dies ergibt dann die Summe aller nicht bestanden Klausuren in allen Fächern.

## Aufgabe 1.9 (Denkaufgabe)

Schreibe ein Programm, das in einer Variablen `days` die Tage seit dem 1.1.1970 gespeichert hat. Konkret geht es darum, um die in `days` spezifizierte Anzahl Tage auf den 1.1.1970 zu addieren. Als Lösung soll das neue Datum auf der Konsole ausgegeben werden.

Zur Vereinfachung gehe davon aus, dass jeder Monat 30 Tage hat und jedes Jahr aus 12 Monaten besteht.

```
int days = 361;
int year = days / 360 + 1970;
int month = (days % 360) / 30 + 1;
int day = ((days % 360) % 30) + 1; // theoretisch reicht
    hier (days % 30), da (360 % 30) == 0 ist
System.out.println(day + "." + month + "." + year);
```