

1. Java Umgebung

- (a) Kontrollieren Sie ob die Java Umgebung bereits installiert wurde und wenn nicht installieren Sie diese.
- (b) Öffnen Sie die zum JDK gehörende Dokumentation und machen Sie sich mit Hilfe eines HTML-Browsers mit ihr vertraut. Schlagen Sie mit deren Hilfe die Methode `println` der Klasse `PrintStream` des Pakets `java.io` nach. Sie sollen lediglich lernen wie Sie das Nachschlagewerk benutzen können.

2. Ausgabe mit `println` und `print`

- (a) Erstellen Sie das *Hello World-Programm* aus der Vorlesung und bringen Sie es zum Laufen.

Lösung: [Aufg2a.java](#)

- (b) Erstellen Sie ein Programm, das folgende veränderte Ausgabe generiert:

```
Hallo Leute,  
  
heute lerne ich ein bisschen Java!
```

Lösung: [Aufg2b.java](#)

Lösung: [Aufg2bWeiteresBeispiel.java](#)

- (c) Markieren Sie im folgenden Programmcode alle Fehler und schreiben Sie dann eine fehlerfreie Version.

[Aufg2c.java](#)

```
class hello{  
    public void main{String args[]};(  
        System.out.Println('Hello, world')  
    )  
}
```

Lösung: [Aufg2c.java](#)

- (d) Schreiben Sie ein Programm, das nur mit einem einzigen Ausdruck folgendes ausgibt (benutzen Sie Java um das Ergebnis zu **berechnen**):

```
5 + 37 = 42
```

Lösung: [Aufg2d.java](#)

- (e) Wo liegt der Unterschied zwischen `System.out.println()` und `System.out.print()`? Schreiben Sie ein Programm, das folgende Ausgabe generiert und verwenden Sie dabei kein `System.out.println()` sondern nur `System.out.print()`. (Zusatz) Versuchen nur **ein** `System.out.print()` zu verwenden.

```
Ich brauche kein println  
  
ich kann das auch mit print
```

Lösung: [Aufg2e.java](#)

3. Literale

- (a) Beschreiben Sie, welche der folgenden literalen Konstanten in Java gültig sind und welche nicht. Geben Sie bei den gültigen Literalen deren Datentyp und bei den ungültigen den Grund für den Fehler an):

- 42
- -17
- 2+3
- $1.1 \cdot 10^3$
- 2.3
- 2,3
- 20
- 'Hello'
- "a"
- TRUE
- b
- false
- 3.141592
- 1.000.000
- 0,524
- 214089067
- $1.1 \cdot 10^{-10}$
- +13
- 1d
- 0x1d
- .34

Lösung: [Aufg3a.java](#)

4. Variablen - Definition und Initialisierung

- (a) Schreiben Sie ein Programm in dem Sie von jedem Ihnen bekannten Datentyp jeweils eine Variable definieren und in der nächsten Zeile initialisieren. Initialisieren Sie die Variablen mit der größtmöglichen Zahl und geben Sie alles auf der Konsole aus.

Lösung: [Aufg4a.java](#)

- (b) Markieren Sie im folgenden Programmcode alle Fehler bzw. schreiben sie eine fehlerfreie Version und geben sie an, was das Programm ausgibt.

Aufg4b.java

```
class Aufg4b{
    public static void main(String args[]){
        int a, b = 4;
        System.out.println(a);
        System.out.println(b);
        int a = 5;
        a = 6;
        int c = d;

        int i = 5; short s = 4;
        boolean b = 1;
        boolean bool1 = true, boolean bool2 = false;
        double d1 = 1.3;
        double d2 = d1 = 1.5;
        System.out.println(d1);
        System.out.println(d2);
        long l = 131231341313;
        float f = 1.3;
    }
}
```

Lösung: [Aufg4b.java](#)

5. Rechnen mit Variablen

- (a) Finden Sie durch Ausprobieren heraus wie oft sie die Zahl $0.1e-23 = 0,1 * 10^{-23}$ (double) mal sich selbst nehmen müssen bis sie 0 erhalten. Warum ist das Ergebnis mathematisch nicht richtig?

Lösung: [Aufg5a.java](#)

- (b) Finden Sie durch Ausprobieren heraus wie oft sie die Zahl $9e100 = 9 * 10^{100}$ (double) nehmen müssen bis *infinity* heraus kommt. Versuchen Sie infinity auch mit dem Datentyp int und einer Zahl Ihrer Wahl heraus zu bekommen. Was stellen Sie fest? Ist das Ergebnis mathematisch korrekt?

Lösung: [Aufg5b.java](#)

- (c) Schreiben Sie ein möglichst kurzes Programm, was die einfache Rechnung $3/2$ durchführt und ausgibt. Was stellen Sie fest?

Lösung: `Aufg5c.java`

- (d) Entwickeln Sie ein kleines Programm zur Berechnung der Tage, Stunden, Minuten und Sekunden aus einer vorgegebenen Zahl von Sekunden, hier 3798. Sie dürfen nur durch 24 oder 60 dividieren (nicht z.B. durch 3600). Überlegen Sie, in welcher Reihenfolge Sie die Ausgaben zweckmäßigerweise berechnen können. Sie dürfen Zwischenergebnisse einführen.

Lösung: `Aufg5d.java`

Lösung: `Aufg5dMitModulo.java`