



Abgabe für Rückmeldungen bis zum 16.11.2020 (08:00 Uhr), Besprechung ab dem 16.11.2020.

Von nun an wird jedes Übungsblatt in „Aufgaben“ und „Präsenzaufgaben“ aufgeteilt sein. Erstere sollten Sie vor der Übung bearbeiten; Ihre Fragen und Lösungen können Sie für Rückmeldungen an Ihren Übungsleiter senden. Letztere werden erst in den Übungen selbst behandelt, aber Sie sind eingeladen, sich auch schon vorab damit zu beschäftigen.

Erstellen Sie ein neues Java-Projekt `Blatt_01` und für jede Aufgabe, in der programmiert werden muß, ein Package (z.B. `a_0` für Aufgabe 01-0). In diesem Package erstellen Sie dann die Klassen im Rahmen der zu bearbeitenden Aufgabe. Wenn es sich um eine Aufgabe handelt, bei der nur mit einer Klasse gearbeitet wird und der Klassenname nicht gegeben wird, nennen Sie die Klasse einfach `Main`.

### Aufgabe 01-0 (Addition)

Betrachten Sie folgendes Listing.

```
1 public class Main {
2     public static void main (String[] args){
3         int i=1;
4         int j=2;
5         System.out.println(i+j);
6     }
7 }
```

Was wird bei seiner Ausführung in der Konsole ausgegeben?

### Aufgabe 01-1 (Compiler/Formatierung)

In dieser Aufgabe untersuchen wir, wie sich bestimmte Veränderungen am Code auf das Programm auswirken. Betrachten Sie hierzu ein lauffähiges Programm, z.B. das Listing aus Aufgabe 01-0. Nach jeder jeweiligen Umsetzung folgender Punkte prüfen Sie, ob das Programm noch kompilierbar ist, und wenn ja, ob sich sein Verhalten geändert hat.

- Entfernen Sie alle Einrückungen („Tabulatorzeichen“ und Leerzeichen)
- Entfernen Sie alle Zeilenumbrüche
- Fügen Sie 5 Semikolons vor die letzte, abschließende, geschweifte Klammer ein

Erklären Sie Ihre Beobachtungen.

### Aufgabe 01-2 (Fragen zur Vorlesung)

Betrachten Sie die folgenden Aussagen.

0. Die virtuelle Maschine liest den Bytecode und übersetzt ihn in Anweisungen an den Computer (genauer gesagt an das Betriebssystem).
1. Man kann nur mit einer IDE programmieren.

2. Jede Klasse muß eine `main`-Methode enthalten.
3. Der Java-Compiler wandelt Quellcode in Bytecode um.

Entscheiden und begründen Sie, ob sie jeweils wahr oder falsch sind.

### Aufgabe 01-3 (Addition - Variation 0)

Betrachten Sie erneut das Listing aus Aufgabe 01-0. Modifizieren Sie Zeile 4 so, daß auf der Konsole 10 ausgegeben wird.

### Aufgabe 01-4 (Addition - Variation 1)

Betrachten Sie erneut das Listing aus Aufgabe 01-0. Modifizieren Sie Zeile 5 so, daß auf der Konsole `i+j` (sic!) ausgegeben wird. Warnungen der Form „The value of the local variable ... is not used“ dürfen Sie ignorieren.

### Präsenzaufgabe 01-5 (Zeitrechner)

Schreiben Sie ein Java-Programm, welches eine `int`-Variable `seconds` (Anzahl von Sekunden) in die entsprechende Anzahl von Stunden, Minuten und Sekunden umrechnet. Die Ausgabe soll die Form `h:m:s` haben, wobei `h` der Anzahl von Stunden, `m` der Anzahl von Minuten und `s` der Anzahl von Sekunden entspricht.

Zum Beispiel soll die Eingabe 1234 zur Konsolenausgabe `0:20:34` führen, was wir dann als 0 Stunden, 20 Minuten, 34 Sekunden interpretieren.

### Präsenzaufgabe 01-6 (Zeitrechner mit Eingabe)

Betrachten Sie folgendes Listing.

```

1 import java.util.Scanner;
2
3 public class Scanner_test {
4     static public void main(String[] args) {
5         Scanner s = new Scanner(System.in);
6         String zeile = new String();
7
8         /* Zeile einlesen und Eingabe schliessen */
9         zeile = s.nextLine();
10        s.close();
11
12        /* Zeile verarbeiten */
13        int a = Integer.parseInt(zeile);
14        System.out.println(a + 4);
15
16        /* Zeile ausgeben */
17        System.out.println(zeile);
18    }
19 }
```

Dieses Programm liest eine Zeile als String aus der Konsole ein (vgl. Aufgabe 00-1) und wandelt sie mit `Integer.parseInt` in einen Integer um, der in `a` geschrieben wird, und mit dem normal weitergearbeitet werden kann.

Modifizieren Sie Ihre Lösung aus Aufgabe 01-5, sodaß der zu verarbeitende Integer `seconds` aus der Konsole eingelesen wird.

### Präsenzaufgabe 01-7 (Kleine Übungen)

Schreiben Sie ein Programm, welches für eine vorgegebene ganze Zahl ihren Betrag und ihr Quadrat ausgibt und aussagt, ob es sich um eine gerade oder ungerade Zahl handelt. Versuchen Sie, nur mit einer einzigen Variable zu arbeiten. Der Gebrauch von Kontrollstrukturen könnte hilfreich sein.

### Präsenzaufgabe 01-8 (Rundung von Fließkommazahlen)

In dieser Aufgabe beschäftigen wir uns mit Fließkommazahlen. Obwohl die genauen Details ihrer Umsetzung den Rahmen dieser Vorlesung sprengen würden, ist eine intuitive Herangehensweise an ihre Eigenschaften möglich. Betrachten Sie hierzu folgendes Listing.

```
1 public class Main {
2     public static void main (String[] args){
3         float a = 1;
4         System.out.println(a);
5
6         a = a / 10;
7         System.out.println(a);
8     }
9 }
```

Welche Ausgabe erwarten Sie von dem Programm und welche erhalten Sie?

Kopieren sie die Zeilen 6-7 und fügen Sie sie nach Zeile 7 an, sodaß also die letzten beiden Zeilen des ursprünglichen Programms wiederholt werden (Division durch 10 und Ausgabe von a). Welche Ausgabe erwarten und erhalten Sie? Wiederholen Sie diesen Vorgang erneut. Was fällt Ihnen auf?

Ändern Sie den Typ von a von float auf double und führen Sie das Programm erneut aus. Was fällt Ihnen auf?

### Präsenzaufgabe 01-9 (Aufrunden auf Zehner/Fünfer)

Schreiben Sie ein Programm, welches eine vorgegebene nichtnegative ganze Zahl auf den nächsthöheren durch zehn teilbaren Wert aufrundet. Wenn die Zahl bereits durch zehn teilbar ist, soll sich ihr Wert nicht ändern. Es sollen also zum Beispiel 65 auf 70, 87 auf 90, 11 auf 20 und 100 auf 100 gerundet werden.

Variieren Sie ihr Programm, indem Sie nun die Zahl auf den nächsthöheren durch fünf teilbaren Wert aufrundet. Es sollen nun also zum Beispiel 65 auf 65, 87 auf 90, 11 auf 15, 0 auf 0 und 100 auf 100 gerundet werden.

Erweitern Sie Ihr Programm, sodaß analog zu Präsenzaufgabe 01-6 die zu rundende Zahl über die Konsole eingelesen wird.