



PSE Vorkurs Tag 1

Feedback

Bitte füllt den Feedbackbogen aus, damit wir den Vorkurs für die nächsten Jahre verbessern können. Der Bogen ist anonym und dauert nur 2-3 Minuten.

Ihr findet ihn unter: <https://forms.gle/VuAPkwtCdcjug1DW6>

Hilfe nichts funktioniert :(

Manche von euch haben sicher schon Erfahrung mit Programmieren, wenn ihr auf ein Problem stößt, fragt einfach mal eure Sitznachbarn und helft euch gegenseitig. Ihr könnt natürlich auch die freundlichen Tutoren ranholen und um Hilfe bitten.

Highperformer-Aufgaben

Wir haben für jeden Tag am Ende des Blattes eine Highperformer-Aufgabe entworfen. Sie soll die schnelleren von euch zum Knobeln und Zusammenarbeiten anregen. Es ist überhaupt kein Problem wenn ihr nicht mit den Highperformer-Aufgaben fertig werdet. Deren Stoff geht weit über die essenziellen Inhalte des Vorkurses (und zum Teil sogar über die Inhalte der PSE) hinaus.

Wichtige Shortcuts: (Windows/Linux)

- Code automatisch formatieren (z.B. richtig Einrücken): **Strg+Alt+L**
- Code kopieren: **Strg+C**
- Den mit **Strg+C** kopierten Code woanders einfügen: **Strg+V**
- Letzte Aktion rückgängig machen: **Strg+Z**
- IntelliJ Autovervollständigungsvorschläge annehmen: **Tab**

Wichtige Shortcuts: (macOS)

- Code automatisch formatieren (z.B. richtig Einrücken): **cmd+option+L**
- Code kopieren: **cmd+C**
- Den mit **cmd+C** kopierten Code woanders einfügen: **cmd+V**
- Letzte Aktion rückgängig machen: **cmd+Z**
- IntelliJ Autovervollständigungsvorschläge annehmen: **Tab**

Präsentations-Cheatsheet

Variablen:

```
int alter = 20;    // Deklaration + Initialisierung
int zahl;         // Deklaration
zahl = 42;        // Initialisierung
```

Operatoren:

Operator	Bedeutung	Beispiele
+	Addition / String-Verkettung	3+2, "b"+"c"
-	Subtraktion	7-5
*	Multiplikation	3*4
/	Division	10/2
%	Modulo	5%2

String-Addition:

```
System.out.print("Hallo");
System.out.println("Welt");
System.out.println("Zahl: " + 42);
System.out.println("\nLeerzeile");
```

```
HalloWelt
Zahl: 42

Leerzeile
```

Scanner:

```
import java.util.Scanner;

Scanner sc = new Scanner(System.in);
int x = sc.nextInt();
int y = sc.nextInt();
System.out.println(x + y);
```

Aufgabe 1: Variablen und Ausgaben

Mit `System.out.println(einString)` kann der Wert von `einString` in der Konsole (unten in der Mitte bei der Ausführung des Codes) ausgegeben werden. Folgender Code:

```
System.out.println("PSE-Zeit ist Grillzeit")
```

Gibt folgendes aus:

```
PSE-Zeit ist Grillzeit
```

1. Kopiert folgenden Grill und nutzt `print(s)` um ihn in der Konsole auszugeben.

Hinweis: Kopiert den Grill direkt zwischen die Gänsefüßchen in euer

```
System.out.println("grill hier"),
```

```
.....
.....@.....
.....@.....
.....@..@.....
.....@..@+.....
.....@..@.....
.....*.....
.....
..CCCCCCCCCCCCCCCC..
..CCCCCCCCCCCCCCCC++..
..CCCCCCCCCCCCCCCC..
..CCCCCCCCCCCCCCCC-..
.....CCCCC.....
.....-@.....:@.....
.....:@.....%@.....
..CCCC.....@.....
..CC..CCCCCCCCCCCC..
..CC..CC.....@.....
..CCCC.....@.....
```

2. Erstelle eine Variable mit einem beliebigen Wert und gib sie aus, ändere im selben Programm durchlauf den Wert der Variable und gib sie nochmal aus.
3. Erstelle zwei `int`-Variablen und gib deren Summe und deren Produkt aus.
4. Erstelle eine Variable mit deinem Namen und schreibe genau ein Print (nicht mehr), das die Person in der Variable begrüßt.

Aufgabe 2: Scanner

Während das Programm ausgeführt wird können auch Werte aus der Konsole eingelesen werden. Das geht mit einem Scanner-Objekt. Mit `scannerObjekt.next()` wird die nächste eingegebene Zeile eingelesen. Mit **Enter** wird die Eingabe beendet.

Hinweis: Kommentiert am besten die Aufgaben aus, die ihr schon erledigt habt, sonst werden irgendwann bei jedem durchlauf all eure Lösungen ausgeführt werden.

```
Scanner derScanner = new Scanner(System.in);
System.out.println("Gib deinen Namen ein: ");
String deinName = derScanner.next();
System.out.println("Du heißt " + deinName);
```

```
Gib deinen Namen ein: Melanie
Du heißt Melanie
```

Wobei die obere Melanie vom Nutzer eingegeben wird.

1. Programmiere einen Addierer, der zwei Zahlen einliest und die Summe ausgibt. Mit `Scanner.next()` funktioniert etwas nicht, überlegt mal warum (Tipp im Codeanhang)?

```
Gib eine Zahl ein: 12
Gib noch eine Zahl ein: 3
Die Summe ist: 15
```

2. Programmiere ein Modulo-Übungsprogramm, bei dem erst zwei Zahlen mit dem Scanner eingegeben werden. Anschließend soll der Nutzer versuchen die Lösung von `zahl1 % zahl2` selbst zu berechnen und einzugeben. Am Schluss soll dann das korrekte Ergebnis ausgegeben werden.

```
Gib eine Zahl ein: 14
Gib noch eine Zahl ein: 3
Gib dein Ergebnis ein: 6
Das richtige Ergebnis von 14 % 3 ist: 2
```

3. Schreibe eine kurze Geschichte (oder suche dir einen kurzen Text im Internet). Suche dann ein paar Wörter mit ihren Wortarten aus und lass den Nutzer sie mit dem Scanner durch andere ersetzen. Gib dann die Geschichte mit den vom Nutzer eingegebenen Wörtern aus. Zum Beispiel wird: *"Die PSE-Vorkurs Hauptorgas lieben grillen."* zu:

```
Gib ein Thema ein: backen
Gib ein Verb ein: Ofen sprengen
Die backen-Vorkurs Orgas lieben Ofen sprengen.
```

Lass deine Sitznachbarn und/oder die Helfer dein Programm ausführen ohne, dass sie wissen welcher Text dahintersteckt.

HIGHPERFORMER-Aufgabe: Zufallszahlen

Java stellt einen Zufallszahlengenerator zur Verfügung. Er wird ähnlich wie ein Scanner initialisiert und er besitzt unter anderem folgende Funktionen:

```
Random rndGenerator = new Random();

//gibt ein zufälligen float zwischen 0.0 und 1.0 zurück:
float zufallsFloat = rndGenerator.nextFloat();

//gibt ein zufälligen int zwischen 0 und 9 zurück (10 wird als Obergrenze
→ ausgeschlossen)
int zufallsInt = rndGenerator.nextInt(10);
```

Die Offizielle Dokumentation für Random findet ihr hier: <https://docs.oracle.com/javase/8/docs/api/java/util/Random.html>

1. Modifiziere dein Moduloübungsprogramm aus Aufgabe 2.2 so, dass die beiden Anfangszahlen nicht mehr selbst eingegeben werden sondern so, dass diese zufällig generiert werden. Der Nutzer soll immer noch selber versuchen das Ergebnis zu berechnen.
2. Ladet euch das Rahmenprogramm hier herunter. Dort bekommt ihr einen sogenannten Seed (eine 5-Stellige Zahl) vom eingebauten Java Zufallsgenerator. Generiert mit den mathematischen Operatoren die ihr jetzt kennt anhand dieses Seeds eine neue 10-Stellige Zufallszahl (ihr dürft natürlich nicht die Random Klasse verwenden). Wir (die PSE-Vorkurs Orgas) haben ein Testerprogramm geschrieben, dass versucht eure Lösung zu bewerten und eventuelle Schwächen eures Ansatzes aufzuzeigen, doch wenn ihr Lust habt könnt ihr auch versuchen den Test auszutricksen. Wenn ihr einen funktionierenden Ansatz habt, könnt ihr versuchen die Aufgabe noch auf eine andere Art zu lösen.

Anhang

Aufgabe 2.1

`derScanner.next()` gibt einen String zurück. Wenn die beiden addiert werden, dann werden die beiden Zahlen stumpf hintereinandergeschrieben, statt als Zahlen addiert zu werden. Um stattdessen einen `int` einzulesen besitzt Scanner die `nextInt()` Funktion.