

Übungsblatt 2 – Datentypen und Variablen

Aufgabe 1 – Geeignete Datentypen und Bezeichner finden

Sie sollen verschiedene Variablen in einem Programm definieren. Finden Sie den passenden (möglichst speicherplatzsparenden) Datentyp für eine Variable, die angibt,

- (1) wie viele Menschen in Deutschland leben,
- (2) wie viele Menschen auf der Erde leben,
- (3) ob es gerade Tag ist,
- (4) wie hoch die Trefferquote eines Stürmers bei einem Fußballspiel ist,
- (5) wie viele Semester Sie studieren werden,
- (6) wie viele Studierende sich für einen Studiengang angemeldet haben,
- (7) mit welchem Buchstaben Ihr Nachname beginnt.

Definieren Sie die Variablen und verwenden Sie sinnvolle Bezeichner.

Aufgabe 2 - Ein neues Projekt erstellen, Programm zum Testen der Verwendung von Variablen eingeben und dieses starten und testen

Erstellen Sie ein neues Projekt mit dem Namen `Uebung02-Aufgabe02`.

Fügen Sie anschließend dem Projekt eine neue Klasse `Variablen` hinzu. Dazu erstellen Sie eine neue Quelltextdatei mit dem Namen `Variablen`, in der Sie die gleichnamige Klasse mit Ihrer `main`-Methode anlegen:

```
public class Variablen {  
    public static void main(String[] args) {
```

In der `main`-Methode dieser Klasse sollen die folgenden Variablen definiert werden:

- `bVar1 als byte`
- `sVar1 als short`
- `iVar1 als integer`
- `lVar1 als long`
- `fVar1 als float`
- `dVar1 als double`

- `istPrim1` als `boolean`
- `cVar1` als `char`

Nach der Definition sollen den Variablen folgende Werte zugewiesen werden:

Variablenname	Wert
<code>bVar1</code>	<code>5</code>
<code>sVar1</code>	<code>400</code>
<code>iVar1</code>	<code>-356576</code>
<code>lVar1</code>	<code>1000000000000L</code>
<code>fVar1</code>	<code>0.123f</code>
<code>dVar1</code>	<code>0.123</code>
<code>istPrim1</code>	<code>false</code>
<code>cVar1</code>	<code>x</code>

Nach den erfolgten Wertzuweisungen soll das Programm die Variablen mit der Anweisung `System.out.println` in der Konsole ausgeben.

Wenn das Programm erfolgreich erstellt und abgelaufen ist und die oben genannten Werte ausgegeben wurden, ergänzen Sie das Programm um weitere Variablen. Definieren Sie folgenden Variablen und initialisieren Sie diese mit den Werten in der Tabelle. Beim Programmablauf sollen auch diese auf der Konsole ausgegeben werden.

Variablenname	Wert
<code>bVar2</code>	<code>128</code>
<code>sVar2</code>	<code>-18453</code>
<code>iVar2</code>	<code>2147483648</code>
<code>lVar2</code>	<code>345236577970</code>
<code>fVar2</code>	<code>4.37456678</code>
<code>dVar2</code>	<code>3645.987654321</code>
<code>istPrim2</code>	<code>true</code>

Welche wichtigen Erkenntnisse im Umgang mit Variablen und Datentypen konnten Sie aus dem erstellten Programm gewinnen?

Aufgabe 3 – Verdienst-Rechner

Schreiben Sie ein Java-Programm `Verdienstrechner`, welches ausrechnet wie viel Geld sie mit Ihrem Nebenjob verdient haben. Der Stundenlohn beträgt z.B. in diesem Job 12 EUR und gestern haben Sie insgesamt 7 Stunden gearbeitet. Definieren Sie in dem Programm geeignete Variablen, da ja sowohl Stundenlohn als auch geleistete Arbeitsstunden variieren können.

Das Programm soll folgende Ausgabe erzeugen:



```
Problems @ Javadoc JUnit Declaration Search Console ✕
<terminated> VerdienstBasis [Java Application] /Library/Java/JavaVirtualMachines/jdk
Stundenlohn: 12 EUR
Geleistete Arbeitsstunden: 7
Ich habe 84 EUR verdient.
```

Aufgabe 4 (optional) – Erweiterter Verdienstrechner

Erweitern sie den `Verdienstrechner` so, dass er sowohl Überstunden als auch Feiertagszuschlag berücksichtigt. Ihr Arbeitgeber zahlt Ihnen für jede Überstunde (d.h. jede Arbeitszeit über 8 Stunden) den 1,25-fachen Satz, sowie feiertags einen höheren Stundenlohn von 17,50 EUR. Verwenden Sie den geeigneten Datentyp für den Überstundenzuschlag, sowie die Indikation ob die geleisteten Stunden feiertags oder werktags geleistet wurden.