

VARIABLEN

AUFGABE 1

Deklariere zwei Variablen vom Typ `int`.
Weise ihnen die Werte 3 und 4 zu.
Deklariere eine weitere Variable vom Typ `int`.
Weise ihr die Summe von den ersten beiden Variablen zu.
Gib das Resultat auf der Konsole aus.

AUFGABE 2

Deklariere zwei Variablen vom Typ `boolean`,
Weise ihnen die Werte `true` und `false` zu.
Deklariere eine weitere Variable vom Typ `boolean`.
Weise ihr das Resultat der UND Verknüpfung der beiden Variablen zu.
Gib das Resultat auf der Konsole aus.

LOGISCHE OPERATOREN

```
int a = 20;  
int b = 10;  
int c = 5;  
int d = 7;
```

AUFGABE 1

Überlege dir was folgende Ausdrücke ergeben:

<code>a <= d && c > b</code>	
<code>a < b b < c c < d</code>	
<code>!(b < d)</code>	
<code>a > c & !(d > c)</code>	
<code>d == c ^ c < b</code>	

AUFGABE 2

Schreibe einen Ausdruck der prüft ob a kleiner b ist UND ob a kleiner c ist.
Speichere das Resultat dieses Ausdrucks in einer Variable und gib diese aus, um das Resultat zu überprüfen.

AUFGABE 3

Schreibe einen Ausdruck der prüft ob a kleiner b ist ODER ob a kleiner c ist.
Speichere das Resultat dieses Ausdrucks in einer Variable und gib diese aus um das Resultat zu überprüfen.

AUFGABE 4

Schreibe einen Ausdruck der prüft ob b kleiner gleich a ist und ob c nicht kleiner gleich d ist.
Speichere das Resultat dieses Ausdrucks in einer Variable und gib diese aus um das Resultat zu überprüfen.

ARRAYS

AUFGABE 1

Wir haben den array: `int[] someNumbers = {5, 3, 7, 2, 9};`
löse dazu folgende Aufgaben:

--	--	--	--	--

- Fülle die Elemente des Arrays in die grafische Array Repräsentation oben.
- Markiere dieses Element mit rot: `someNumbers[4]`
- Markiere dieses Element mit blau: `someNumbers[2]`
- Wie sieht die Array Repräsentation aus, nachdem diese Zuweisung ausgeführt wurde:
`someNumbers[0] = 1;`

--	--	--	--	--

AUFGABE 2

Wir haben den array: `int[] otherNumbers = new int[6];`
- Zeichne die grafische Array Repräsentation

--	--	--	--	--	--

- Wie sieht die Array Repräsentation aus, nachdem diese Zuweisung ausgeführt wurde:
`otherNumbers[4] = 10;`
- Wie sieht die Array Repräsentation aus, nachdem diese Zuweisung ausgeführt wurde:
`otherNumbers[1] = 9;`
- Wie sieht die Array Repräsentation aus, nachdem diese Zuweisung ausgeführt wurde:
`otherNumbers[6] = 10;`

AUFGABE 3

Erzeuge einen Array für eine Sammlung von 6 Zahlen.
Weise dem 3. Element im Array den Wert 4 zu.
Speichere den Wert des 3. Elements in einer Variable.

Gib die Variable und das 3. Element auf der Konsole aus.

AUFGABE 4

Erzeuge einen Array für eine Sammlung von 3 Wahrheitswerten.
Weise jedem Element einzeln den Wert true zu.
Gib jedes Element auf der Konsole aus.

AUFGABE 5

Erzeuge einen Array für eine Sammlung von 2 Zahlen.
Weise dem 1. Element den Wert 3 zu und dem 2. Element den Wert 4.
Dekлариere eine Variable vom Typ `int`.
Weise ihr die Summe der beiden Array Elementen zu.
Gib das Resultat auf der Konsole aus.

-> Hast du gemerkt, dass diese Aufgabe sehr ähnlich ist wie Aufgabe 1 bei den Variablen? Das ist so, weil man ein Array Element genau gleich behandelt wie eine Variable. Es hat einfach den Namen: `arrayNamen[elementPosition]`. Weshalb man sich den Array als Sammlung von Variablen vorstellen kann.

SCHLEIFEN

AUFGABE 1

Erstelle eine for Schleife bei der die Zähler Variable mit 0 Initialisiert wird und die Schleife abgebrochen wird sobald die Zähler Variable 10 ist. Die Zähler Variable soll für jeden Schleifendurchlauf um 1 erhöht werden.
Gib in jedem Schleifendurchlauf die Zähler Variable auf der Konsole aus.

AUFGABE 2

Erstelle eine for Schleife bei der die Zähler Variable mit 0 Initialisiert wird. Die Schleife soll solange ausgeführt werden wie die Zähler Variable kleiner 10 ist. Die Zähler Variable soll für jeden Schleifendurchlauf um 1 erhöht werden.
Gib in jedem Schleifendurchlauf die Zähler Variable auf der Konsole aus.

AUFGABE 3

Erstelle eine while Schleife die genau das gleiche macht wie die for Schleife aus Aufgabe 1.
Beachte, dass du bei der while Schleife nur die Abbruchsbedingung angeben kannst. Die Zähler Variable musst du selbst handeln.

AUFGABE 4

Erstelle eine for Schleife bei der die Zähler Variable mit 18 Initialisiert wird. Die Schleife soll solange ausgeführt werden wie die Zähler Variable grösser 10 ist. Die Zähler Variable soll für jeden Schleifendurchlauf um 1 verkleinert werden.

Gib in jedem Schleifendurchlauf die Zähler Variable auf der Konsole aus.

AUFGABE 4

Deklariere eine Variable für einen Zahlenwert. Nenne sie `sum` und initialisiere sie mit 0.
Erstelle eine `for` Schleife bei der die Zähler Variable mit 18 Initialisiert wird. Die Schleife soll solange ausgeführt werden wie die Zähler Variable grösser 10 ist. Die Zähler Variable soll für jeden Schleifendurchlauf um 1 verkleinert werden.
Zähle bei jedem Schleifendurchlauf die Zählervariable zu der `sum` dazu. So dass du am Schluss die Summe aller Werte der Zählervariable erhältst.
Gib `sum` auf der Konsole aus, sobald die Schleife fertig ist.

AUFGABE 5

Erstelle eine Schleife, die die zweier Reihe (Multiplikationsreihe) auf der Konsole ausgibt. (also die Zahlen 0, 2, 4, 6, 8, 10)

AUFGABE 6

Erstelle eine Schleife, die die Summe der ersten 10 Natürlichen Zahlen berechnet. Gebe das Resultat dann auf der Konsole aus.

AUFGABE 7

Erstelle eine Schleife, die das Quadrat von allen Natürlichen Zahlen bis 10 angibt. Gebe in jedem Schleifendurchlauf die Zahl und das Quadrat davon aus. Der Output sollte dann so aussehen:

```
Das Quadrat von 1 ist 1
Das Quadrat von 2 ist 4
Das Quadrat von 3 ist 9
Das Quadrat von 4 ist 16
Das Quadrat von 5 ist 25
Das Quadrat von 6 ist 36
Das Quadrat von 7 ist 49
Das Quadrat von 8 ist 64
Das Quadrat von 9 ist 81
Das Quadrat von 10 ist 100
```

AUFGABE 8

Erstelle den array: `int[] theArray = {6,3,8,2,9};`
Erstelle eine Schleife, die jedes Element des Array auf der Konsole ausgibt.

AUFGABE 9

Erstelle den array: `int[] tenNumbers = new int[10];`
Erstelle eine Schleife, die diesen Array mit den Werten 0 bis 9 füllt.

AUFGABE 10

Erstelle eine for-Schleife die einen Array mit Länge 10 mit den Werten 1 bis 10 füllt.

METHODEN

AUFGABE 1

Schreibe die Methode `writeText` die einen String entgegen nimmt und diesen in die Konsole schreibt. Die Methode hat keinen Rückgabewert.

AUFGABE 2

Schreibe die Methode `sum` die zwei Zahlen entgegen nimmt und die Summe der beiden Zahlen zurück gibt.

AUFGABE 3

Schreibe die Methode `concat` die zwei Strings und einen boolean entgegen nimmt. Wenn der boolean `true` ist sollen die Strings in der Reihenfolge zusammengesetzt werden, wie sie der Methode übergeben worden sind. Wenn er `false` ist sollen sie in umgekehrter Reihenfolge zusammengesetzt werden. Der zusammengesetzte String soll zurück gegeben werden. (Achten sie auf die Namensgebung. Der boolean Wert definiert die Reihenfolge. Sollte also etwas mit `order` heissen.)

AUFGABE 4

Schreibe eine Methode die den Radius entgegen nimmt und daraus den Kreisumfang berechnet.

AUFGABE 5

Schreibe eine Methode die eine Zahl von 1 - 12 entgegen nimmt und den entsprechenden Namen des Monats zurück gibt.

AUFGABE 4

Schreibe eine Methode die prüft ob eine Zahl gerade oder ungerade ist. Die Methode gibt `true` zurück, wenn die Zahl gerade ist und `false` wenn sie ungerade ist. Achte darauf der Methode einen sinnvollen Namen zu geben.

Tipp: eine Zahl `% 2` (Modulo oder Rest) ergibt 0, wenn sie gerade ist. (Die Zahl geteilt durch 2 ergibt einen Rest von 0)
`4%2 = 0, 5%2 = 1`

AUFGABE 5

Schreibe eine Methode die alle ungeraden Zahlen für einen gewissen Wertebereich (Startwert bis Endwert) in einem Array zurück gibt. Start- und Endwert sollen Methodenparameter sein. Teste die Methoden Funktionalität indem du die Werte des zurück gegebenen Arrays auf der Konsole ausgibt.

Tipp: eine Zahl % 2 (Modulo oder Rest) ergibt 0, wenn sie gerade ist. (Die Zahl geteilt durch 2 ergibt einen Rest von 0)
 $4\%2 = 0$, $5\%2 = 1$

AUFGABE 6

Schreibe eine Methode die die Seitenlänge eines Quadrats entgegen nimmt und das Quadrat mit Hilfe des Zeichen ,*' auf der Konsole ausgibt.

Beispiel des Outputs für eine Seitenlänge von 5:

Tipp: mit print() kannst du in Processing auf die gleiche Linie schreiben. println() schreibt auf eine neue Zeile. Eine neue Zeile kann auch durch „\n“ erzeugt werden.