



Java-Crashkurs

2. Schleifen | Benno Hölz, Matthias Ruf | 06.09.2023

1. Schleifen

while-Schleife

Eigenschaften der while-Schleife

- Verwendung:
 - wenn besonderer Wert auf die Bedingung gelegt wird.
- abweisender Charakter:
 - d.h. es wird bereits beim ersten Schleifendurchlauf die Bedingung überprüft.

while-Schleife

Syntax

```
1 | <Deklaration/Initialisierung>
2 | while(<Bedingung>){
3 |     //Schleifenrumpf
4 |     <Aktion>
5 | }
```

Beispiel

```
1 | int i = 0;
2 | while(i < 10){
3 |     System.out.println(i);
4 |     i++;
5 | }
```

for-Schleife

Eigenschaften der for-Schleife

- Verwendung:
 - meist, wenn etwas anhand eines Index durchlaufen wird. bspw. Strings (dazu später mehr)
- abweisender Charakter:
 - d.h. es wird bereits beim ersten Schleifendurchlauf die Bedingung überprüft.

for-Schleife

Syntax

```
1 | for (<Deklaration/Initialisierung>; <Bedingung>; <Aktion  
   |     >){  
2 |     //Schleifenrumpf  
3 | }
```

Beispiel

```
1 | for (int i = 0; i < 10; i++){  
2 |     System.out.println(i);  
3 | }
```

forEach-Schleife

Eigenschaften der forEach-Schleife

- vereinfachte Form der for-Schleife
- Verwendung:
 - meist, wenn eine iterierbare Datenstruktur (z. B. ein Array) durchlaufen wird und der Index nicht weiter relevant ist

forEach-Schleife

Syntax

```
1 | <Deklaration Datenstruktur>  
2 | for(<Deklaration Element> : <Datenstruktur>){  
3 |     //Schleifenrumpf  
4 | }
```

Beispiel

```
1 | int[] array = {1,2,3,4};  
2 | for(int i : array){  
3 |     System.out.println(i);  
4 | }
```

Übungen zu Schleifen

3 kleine Aufgaben

Schreibe ein Programm, welches die Zahlen von 1 bis 10 rückwärts ausgibt.

- i verwende eine for-Schleife
- ii verwende eine while-Schleife
- iii geht es auch ohne Schleife?

2. Abfragen

if-Abfrage

Es gibt keine if-Schleifen, ...

... sondern nur if-Abfragen! → if-schleife.de

if-Abfrage

Es gibt keine if-Schleifen, ...

... sondern nur if-Abfragen! → if-schleife.de

Syntax

```
1 | if (<Bedingung>) {  
2 |     // Code  
3 | }
```

Code wird **einmal** ausgeführt, wenn Bedingung erfüllt ist.

Für Begeisterte

Weitere interessante Konstrukte

- Switch-Case-Konstrukt
- „Fragezeichenoperator“

3. Abfragen und Schleifen

Abfragen und Schleifen

break & continue

- Schleifen vorzeitig unterbrechen oder einen Durchlauf überspringen
- meist in Verbindung mit Abfragen verwendet

Abfragen und Schleifen

break

Aktuelle Schleifeniteration der innersten Schleife wird ab dieser Zeile abgebrochen.

```
1 | for(int i = 0; i < Integer.MAX_VALUE; i++){
2 |     if(i >= 10){
3 |         break;
4 |     }
5 |     System.out.println(i);
6 | }
```

Abfragen und Schleifen

continue

Aktuelle Schleifeniteration der innersten Schleife wird ab dieser Zeile übersprungen.

```
1 | for(int i = 0; i < Integer.MAX_VALUE; i++){
2 |     if(i >= 10){
3 |         continue;
4 |     }
5 |     System.out.println(i);
6 | }
```

Übungen zu Schleifen und Abfragen

Hierzu wieder 3 kleinere Aufgaben

Schreibe ein Programm, welches alle geraden Zahlen zwischen 0 und 10 ausgibt, indem du folgendes Programm abänderst:

```
1 | for(int i = 0; i <= 10; i++){  
2 |     System.out.println(i);  
3 | }
```

- i verwende eine Abfrage
- ii ändere die Bedingung und die Ausgabe
- iii ändere den Schleifenkopf

Zusammenfassung

- Schleifen
 - `for`-Schleife
 - `while`-Schleife
 - `forEach`-Schleife
- Abfragen
 - `if`-Abfrage
- Abfragen und Schleifen
 - `break` (Schleifendurchlauf abbrechen)
 - `continue` (Durchlauf überspringen)