



Kontrollstrukturen: Schleifen

openHPI-Java-Team
Hasso-Plattner-Institut



- Englisch: loop
- Gleichen Code mehrmals ausführen, aber nur einmal schreiben
- Besteht aus **Schleifenkopf** mit Schleifenbedingung und **Schleifenrumpf**
- Arten:
 - Zählschleife **for** (z.B. Dreimal ausführen)
 - Kopfgesteuerte Schleife **while** (solange..., tue...)
 - Fußgesteuerte Schleife **do while** (wie **while** und immer mindestens einmal ausgeführt)
 - ...

Zählschleife (1/3) for



```
1 for (int i = 0; i < 3; i++) {  
2  
3     //dies ist eine Zählschleife  
4     //sie wird 3 mal ausgeführt  
5  
6  
7 }
```

Schleifenbedingung

Schleifenkopf

Schleifenrumpf

Syntax des Schleifenkopfes:

for (<Zählerinitialisierung>; <Abbruchbedingung>; <Schrittweite>)

for (int i = 0; i < 3; i++)



```
1 for (int i = 0; i < 3; i++) {  
2  
3     //dies ist eine Zählschleife  
4     //sie wird 3 mal ausgeführt  
5  
6  
7 }
```

Schleifenbedingung

Schleifenkopf

Schleifenrumpf

Erklärung Schleifenkopf

`for (int i = 0; i < 3; i++)`

- Ganzzahl `i` wird zu Beginn auf 0 gesetzt (`int i = 0`)
- Vor jedem Schleifendurchlauf wird geschaut, wie groß `i` ist (`i < 3`)
- Nach jedem Schleifendurchlauf wird `i` um eins erhöht (`i++`)

Zählschleife (3/3)

for



```
1 for (int i = 0; i < 3; i++) {  
2     //dies ist ein Beispiel  
3     System.out.println("Zeile "+ i);  
4 }
```

Schleifenbedingung

Schleifenkopf

Schleifenrumpf

Ausgabe:

Zeile 0

Zeile 1

Zeile 2

kopfgesteuerte Schleife (1/3)

while



```
1 while ( i < j ){  
2  
3     //Dies ist eine  
4     //kopfgesteuerte Schleife  
5     //i oder j sollte verändert  
6     //werden  
7 }
```

Schleifenbedingung

Schleifenkopf

Schleifenrumpf

- Bei jedem Schleifendurchlauf Bedingung geprüft
- Bedingung kann beliebig sein (muss nicht `i < j` sein)
- Solange Bedingung erfüllt ist, wird Schleifenrumpf ausgeführt
- **Achtung!** Endlosschleifen möglich

kopfgesteuerte Schleife (2/3)

while



```
1 int i = 0;
2 while ( i < 3 ){
3     //So könnte man die Funktion
4     //einer for-Schleife mit einer
5     //while-Schleife darstellen
6     i++;
7 }
```

Schleifenbedingung

Schleifenkopf

Schleifenrumpf

- Auch mit **while**-Schleife gleiche Funktion wie mit einer **for**-Schleife möglich
- Initialisierung vor der Schleife, Veränderung der Variable in der Schleife

kopfgesteuerte Schleife (3/3)

while



```
1 int i = 0;
2 while ( i < 3 ){
3     //Dies ist das gleiche Beispiel
4     //wie bei der for-Schleife
5     System.out.println("Zeile "+ i);
6     i++;
7 }
```

Schleifenbedingung

Schleifenkopf

Schleifenrumpf

Ausgabe:

Zeile 0

Zeile 1

Zeile 2



```
1  do {  
2  
3      //Dies ist eine  
4      //fußgesteuerte Schleife  
5  
6  } while ( i < j );
```

Schleifenkopf

Schleifenrumpf

Schleifenbedingung

- Nach jedem Schleifendurchlauf Bedingung geprüft
- Solange Bedingung erfüllt wird Schleifenrumpf nochmal ausgeführt
- Im Endeffekt wie **while** nur mindestens einmal ausgeführt
- **Achtung!** Endlosschleifen möglich
- Hinweis: in Praxis nur selten verwendet