Inhalt

1. Motivation Schleifen
2. While-Loop
3. Do-While
4. For-Loop

Oft wiederholender Programmcode

Bisher copy-paste

```
System.out.println("Zahl " + 1);
System.out.println("Zahl " + 2);
System.out.println("Zahl " + 3);
```

Laufvariable einführen

```
int zahl = 1;
System.out.println("Zahl " + zahl);
zahl = zahl +1;
System.out.println("Zahl " + zahl);
zahl = zahl +1;
System.out.println("Zahl " + zahl);
```

While Schleife

```
int zahl = 1;

while ( zahl <= 3 ){
  System.out.println("Zahl " + zahl);
  zahl ++;
}
```

Endlose Ausgabe

```
int zahl = 1;
while ( true ){
  System.out.println("Zahl " + zahl);
  zahl ++;
}
```

While Schleife

```
while ( /* BEDINGUNG */ ){
  /*
    Schleifenkörper (loop body)
   */
}
```

- **Schleifenkörper** wird solange ausgeführt, wie **Bedingung** erfüllt ist
- Danach wird aus Schleifenkörper gesprungen
- Bedingung muss boolsches Ergebnis liefern. (zB. `x < 5`, `x %2 == 0`, `true`)
- `while(true)` führt zu **Endlosschleife**

Endlosschleife

```
int x = 1
while ( true ){
  x = x+1;
  System.out.println(x);
}
```

- Führt zu **StackOverflow**, da Integer-Wert nicht groß genug.

Gültigkeitsbereich von Variablen

```
int a = 0;
while ( a < 10 ){
  // hier kann auf a zugegriffen werden
  int b = 10;
}
// hier kann NICHT auf b zugegriffen werden
```

- **Schleifenkörper** wird solange ausgeführt, wie **Bedingung** erfüllt ist
- Danach wird aus Schleifenkörper gesprungen
- Variablen, Konstanten, die in Schleifenkörper deklariert sind, sind auch nur darin gültig

Laufvariablen

Hochzählen einer Laufvariable = increment (+)

```
int i = 0;
while ( i < 10 ){
  System.out.println("Zahl " + i); // 0, 1, 2, 3, 4, 5, 6, 7, 8, 9
}
```

Verringern einer Laufvariable = decrement (-)

```
int i = 10;
while ( i >= 0){
  System.out.println("Zahl " + i); //10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0
}
```

Do While Schleife

While

- **kopfgesteuerte Schleifen**
- Kann völlig übersprungen werden, wenn Bedingung zu Beginn falsch ist.

Do-While

- **fußgesteuerte Schleife**
- Wird **zumindest einmal** ausgeführt
- Dann wird überprüft, ob nochmal ausführen
- => Bedingung wird nach erstem Durchlauf überprüft

Do-While

```
int x = 1;

do{

  // wird einmal ausgeführt

}while(x==10);
```

Beispiel

```
Scanner s = new Scanner(System.in);
int alter;

do {

    System.out.println("Bitte geben sie ihr Alter ein: ");

    alter = s.nextInt();
    s.nextLine(); // Zeilenumbruch

} while(alter <  5 || alter > 100);

System.out.println("Danke");
```

Anwendungsfall:

- Lese solange eine Zahl ein und bilde die Summe, bis der Benutzer -1 eingibt

```
Scanner s = new Scanner(System.in);

int z = s.nextInt();
int summe = 0;

do{

  summe = summe + z;
  z = s.nextInt();
    s.nextLine(); // Zeilenumbruch

}while(z != -1);
```

Break

- Mit `break` kann Schleife zu jedem Zeitpunkt abgebrochen werden.
- Soll eher vermieden werden, wenn nicht zwingend nötig

Beispiel do while

```
int x = 1
while (true){
  if(x== 10){ break;// Schleife wird bei 10 abgebrochen
  }
}

do{
  if(x == 50){ break; // Schleife wird bei 50 abgebrochen
  }
}while(x < 100);
```

Continue

- Mit dem Schlüsselwort `continue` kann aktueller Durchlauf übersprungen werden.
- funktioniert in allen Schleifen

```
int x = 1;
while( x < 10){
  System.out.println(x);
  x = x + 1;
  continue;
  System.out.println("übersprungen"); // wird nie ausgeführt
}
```

Motivation For-Loop

- Zähle von 0 bis 5
- Mit **WHILE**

```
int i = 0;
while( i < 5){
  System.out.println(i);
  i = i + 1;
}
```

- Mit **FOR**

```
for(int i = 0; i < 5; i = i+1){
  System.out.println(i);
}
```

For-Loop

- Laufvariable ist nur in Schleife sichtbar
- Laufvariable kann mit `+,-,/,%,...` geändert werden

```
for(Laufvariable anlegen; Bedingung; Laufvariable ändern){
  // Schleifenkörper
}
```

```
for(int i = 0; i < 5; i = i+1){
  System.out.println(i);
}
```

Beispiele For-Loop

von 10 bis 0 zählen

```
for(int i = 10; i > 0; i = i-1){
  System.out.println(i);
}
```

Alle geraden Zahlen bis 10

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

bis 100 in 5er Schritten

```
for(int i = 0; i <= 100; i = i+5){
  System.out.println(i);
}
```

Zuweisungsoperatoren

```java
int i = 0;

// +=
i = i+5;
i +=5;

// -=
i = i-5;
i -=5;

// /=
i = i/5;
i /=5;

// *=
i = i*5;
i *=5;
```

Präfixnotation/Postfixnotation

Präfixnotation

Wert wird **vor Verwendung** erhöht/verringert

```
++i
--i
```

Postfixnotation

Wert wird **nach Verwendung** erhöht/verringert

```
i++
i--
```

Laufvariable um 1 erhöhen

++

```
int i = 0;
// ++
System.out.println(i);
i = i+1;
```

```
// i hat Wert 0
System.out.println(i++); // 0
// i hat Wert 1
```

```
// i hat Wert 0
System.out.println(++i); // 1
// i hat Wert 1
```

Laufvariable um 1 verringern

--

```
int i = 1;
// --
System.out.println(i);
i = i-1;
```

```
// i hat Wert 1
System.out.println(i--); // 1
// i hat Wert 0
```

```
// i hat Wert 1
System.out.println(--i); // 0
// i hat Wert 0
```

Typische For-Loop

- Folgende Änderungen der Laufvariable liefern gleiches Ergebnis
- Übliche Form zum Hochzählen (**increment**) ist `i++`
- Übliche Form zum Verringern (**decrement**) ist `i--`

```
//for(int i = 0; i <= 100; i = i+1){
//for(int i = 0; i <= 100; i += 1){
for(int i = 0; i <= 100; i = i++){
  System.out.println(i);
}
```