

Java Grundlagen 2

Bioinformatik Bachelor

Propädeutikum WS 2019/2020

29. Oktober 2019

Christian Hoffmann

Outline

- Schleifen / Loops
- Bereiche / Arrays
- Fehler lesen



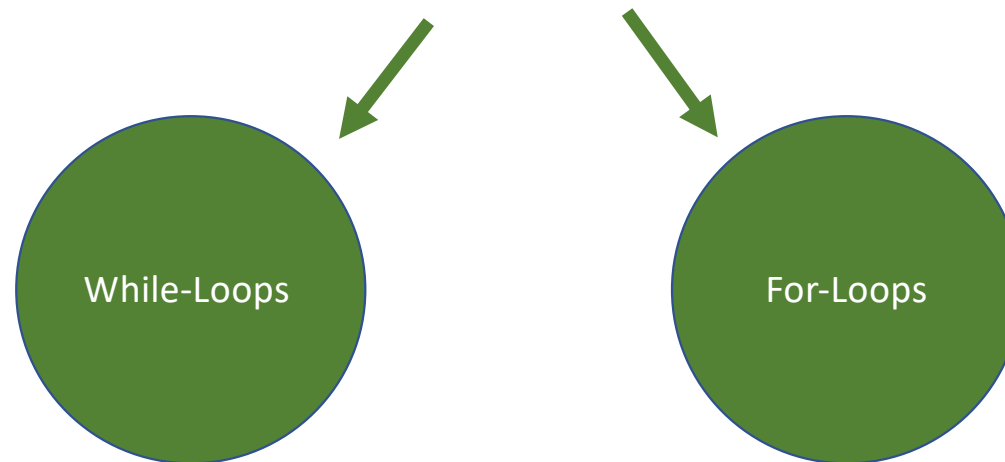
Schleifen / Loops

Schleife = Kontrollstruktur

Wiederholt einen Anweisungs-Block solange die Schleifenbedingung gültig bleibt

Achtung: Endlosschleifen

Es gibt mehrere Arten von Schleifen, zwei der wichtigsten sind:



While - Loop

Java **While-Loop** Syntax:

```
while (condition) {  
    // code block to be executed  
}
```

Die While-Loop durchläuft einen Codeblock so lange, bis eine Abbruchsbedingung erfüllt wurde.

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



```
0  
1  
2  
3  
4
```

Schleifen / Loops



```
System.out.println("codeblock starting");  
int i = 0;  
while (i < 3) {  
    System.out.println(i + ". Durchlauf");  
    i++;  
}
```

Output:

i not defined

Schleifen / Loops



```
System.out.println("codeblock starting");  
int i = 0;  
while (i < 3) {  
    System.out.println(i + ". Durchlauf");  
    i++;  
}
```

Output:
«codeblock starting»

i not defined

Schleifen / Loops



```
System.out.println("codeblock starting");  
int i = 0;  
while (i < 3) {  
    System.out.println(i + ". Durchlauf");  
    i++;  
}
```

Output:
«codeblock starting»

i = 0

Schleifen / Loops

Check: $i < 3$? \rightarrow true




```
System.out.println("codeblock starting");  
int i = 0;  
while (i < 3) {  
    System.out.println(i + ". Durchlauf");  
    i++;  
}
```

Output:
«codeblock starting»

$i = 0$

Schleifen / Loops




```
System.out.println("codeblock starting");  
int i = 0;  
while (i < 3) {  
    System.out.println(i + ". Durchlauf");  
    i++;  
}
```

Output:
«codeblock starting»
«0. Durchlauf»

i = 0

Schleifen / Loops



```
System.out.println("codeblock starting");  
int i = 0;  
while (i < 3) {  
    System.out.println(i + ". Durchlauf");  
    i++;  
}
```

Output:
«codeblock starting»
«0. Durchlauf»

i = 1

Schleifen / Loops

Check: $i < 3$? \rightarrow true




```
System.out.println("codeblock starting");  
int i = 0;  
while (i < 3) {  
    System.out.println(i + ". Durchlauf");  
    i++;  
}
```

Output:

«codeblock starting»
«0. Durchlauf»

$i = 1$

Schleifen / Loops




```
System.out.println("codeblock starting");  
int i = 0;  
while (i < 3) {  
    System.out.println(i + ". Durchlauf");  
    i++;  
}
```

Output:
«codeblock starting»
«0. Durchlauf»
«1. Durchlauf»

i = 1

Schleifen / Loops



```
System.out.println("codeblock starting");  
int i = 0;  
while (i < 3) {  
    System.out.println(i + ". Durchlauf");  
    i++;  
}
```

Output:
«codeblock starting»
«0. Durchlauf»
«1. Durchlauf»

i = 2

Schleifen / Loops

Check: $i < 3?$ → true




```
System.out.println("codeblock starting");  
int i = 0;  
while (i < 3) {  
    System.out.println(i + ". Durchlauf");  
    i++;  
}
```

Output:

```
«codeblock starting»  
«0. Durchlauf»  
«1. Durchlauf»
```

$i = 2$

Schleifen / Loops




```
System.out.println("codeblock starting");  
int i = 0;  
while (i < 3) {  
    System.out.println(i + ". Durchlauf");  
    i++;  
}
```

Output:

```
«codeblock starting»  
«0. Durchlauf»  
«1. Durchlauf»  
«2. Durchlauf»
```

```
i = 2
```

Schleifen / Loops



```
System.out.println("codeblock starting");  
int i = 0;  
while (i < 3) {  
    System.out.println(i + ". Durchlauf");  
    i++;  
}
```

Output:

```
«codeblock starting»  
«0. Durchlauf»  
«1. Durchlauf»  
«2. Durchlauf»
```

```
i = 3
```


Schleifen / Loops

Check: $i < 3$? \rightarrow false!



```
System.out.println("codeblock starting");  
int i = 0;  
while (i < 3) {  
    System.out.println(i + ". Durchlauf");  
    i++;  
}
```

Output:

```
«codeblock starting»  
«0. Durchlauf»  
«1. Durchlauf»  
«2. Durchlauf»
```

$i = 3$

Schleifen / Loops

```
System.out.println("codeblock starting");  
int i = 0;  
while (i < 3) {  
    System.out.println(i + ". Durchlauf");  
    i++;  
}
```

Ende der Loop!



Output:

```
«codeblock starting»  
«0. Durchlauf»  
«1. Durchlauf»  
«2. Durchlauf»
```

`i = 3`

For - Loop

Java **For-Loop** Syntax:

```
for (statement 1; statement 2; statement 3) {  
    // code block to be executed  
}
```

Wenn man genau weiß, wie oft eine Loop durchlaufen werden soll, kann man die **For-Loop** benutzen.

Statement 1: Wird genau einmal ausgeführt, zu Beginn des 1. Durchlaufs

Statement 2: Definiert die Abbruchsbedingung der Schleife

Statement 3: Wird immer am Ende des Codeblocks ausgeführt (also pro Schleifendurchlauf ein mal)

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



```
0  
1  
2  
3  
4
```

for vs while

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

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

Initialisierung der Laufvariable

Abbruchbedingung der Loop

Hochzählen der Laufvariable

Bereiche / Arrays



Arrays: Werden benutzt um mehrere Werte in eine einzelne Variable zu speichern
Um ein Array zu deklarieren, definiere den Variablen-Typ mit **eckigen Klammern**:

```
String[] cars;
```

Damit wurde eine Variable deklariert, welche ein Array aus Strings enthält.

Erste Werte einfügen:

```
String[] cars = {"Volvo", "BMW", "Ford", "Mazda"};
```

Integer-Array:

```
int[] myNum = {10, 20, 30, 40}
```

Bereiche / Arrays

Zugriff auf die Elemente eines Arrays läuft mithilfe einer Index-Nummer ab:

```
String[] cars = {"Volvo", "BMW", "Ford", "Mazda"};  
System.out.println(cars[0]);  
// Outputs Volvo
```

Note: Erstes Element des Arrays hat den Index 0!!



Bereiche / Arrays



Element eines Arrays ändern: Zugriff über Index

```
String[] cars = {"Volvo", "BMW", "Ford", "Mazda"};
cars[0] = "Opel";
System.out.println(cars[0]);
// Now outputs Opel instead of Volvo
```

Länge eines Arrays: „Length“-Attribut eines Arrays

Note: Arrays haben in Java immer eine feste Länge, man kann nicht einfach neue Elemente hinzufügen!

```
String[] cars = {"Volvo", "BMW", "Ford", "Mazda"};
System.out.println(cars.length);
// Outputs 4
```

Loop Through an Array



Man kann mithilfe einer Loop durch ein Array iterieren, indem man eine **For-Loop** und das Length-Attribut eines Arrays benutzt.

Folgendes Beispiel gibt alle Elemente unseres Car-Arrays aus:

```
String[] cars = {"Volvo", "BMW", "Ford", "Mazda"};

for (int i = 0; i < cars.length; i++) {
    System.out.println(cars[i]);
}
```



```
"Volvo"
"BMW"
"Ford"
"Mazda"
```


Break / Continue

Das Break-Statement wird benutzt, um eine Loop vorzeitig zu verlassen:

```
for (int i = 0; i < 5; i++) {  
    if (i == 3) {  
        break;  
    }  
    System.out.println(i);  
}
```



```
0  
1  
2
```

Das Continue-Statement bricht nur den aktuellen Durchlauf ab:

```
for (int i = 0; i < 5; i++) {  
    if (i == 3) {  
        continue;  
    }  
    System.out.println(i);  
}
```



```
0  
1  
2  
4
```

Fehler lesen

IDE's zeigen euch unter anderem **Compiler-Fehler** und **Laufzeit-Fehler** an:

Tipp: Rote Unterringelung **NIE** ignorieren

Compiler-Fehler:

Tritt auf, wenn der Java-Compiler den Code nicht kompilieren kann. Das liegt häufig an Syntax-Fehlern, z.B:

- Fehlende Klammern
- Fehlendes Semikolon

```
public class Main {  
    public static void main(String[] args) {  
        int i = "10";  
    }  
}
```

Incompatible types.
Required: **int**
Found: **java.lang.String**

Fehler lesen

Laufzeit-Fehler: Code konnte kompiliert werden → Syntaktisch also korrekt. Diese Fehler treten auf, während das Programm ausgeführt wird (runtime). Bei einem Runtime-Error wird an einem bestimmten Punkt abgebrochen. Beispiele:

- Division durch 0
- Array-Index liegt außerhalb der Array-Länge („out of bounds“)

```
String[] cars = {"Volvo", "BMW", "Ford", "Mazda"};  
System.out.println(cars[7]);
```



```
Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: Index 7 out of bounds for length 4  
at Chris.Main.main(Main.java:7)
```

Gibt euch somit die Error-Message und die Zeile im Code an. In der Nachricht oben wäre der Fehler in Zeile 7 → Klick auf das Blau-Hinterlegte würde euch zu der gefragte Stelle bringen.