

# Programmierkurs

## Einführung in Java Tag 3



Erik Dyka

Sommersemester 2020

### Ablauf

#### Arrays

- Arrays erstellen
- Arrayzugriff
- Exceptions

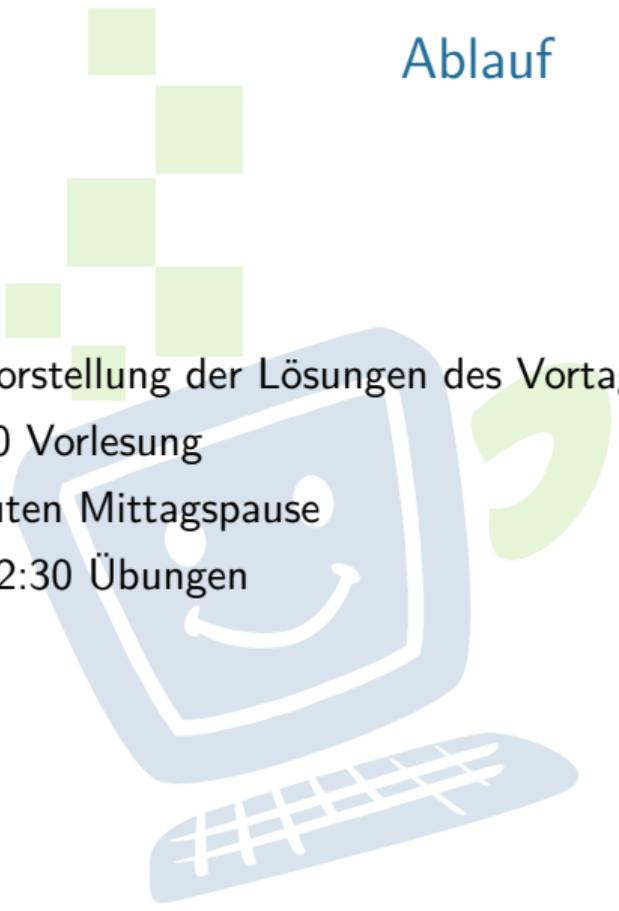
#### Schleifen

- While-Schleifen
- Do-While-Schleifen
- Endlosschleifen
- For-Schleifen
- Welche Schleife?

#### Debugging

#### Quellen & Lizenz

# Ablauf

- 
- ▶ 09:30 Vorstellung der Lösungen des Vortages
  - ▶ ab 10:00 Vorlesung
  - ▶ 60 Minuten Mittagspause
  - ▶ gegen 12:30 Übungen

## Ablauf

### Arrays

Arrays erstellen  
Arrayzugriff  
Exceptions

### Schleifen

While-Schleifen  
Do-While-Schleifen  
Endlosschleifen  
For-Schleifen  
Welche Schleife?

### Debugging

### Quellen & Lizenz

## Inhaltsübersicht Vorkurs

- ▶ Tag 1: Zustände, Variablen, Datentypen, Konvertierungen, Arithmetik, Eclipse Live-Demo
- ▶ Tag 2: Kommentare, Boolesche Ausdrücke, If-Abfragen, Switch-Case
- ▶ Tag 3: Arrays, Exceptions, (Do-)While-Schleife, For-Schleifen, Debugging
- ▶ Tag 4: (statische) Methoden, Klassenvariablen

### Ablauf

#### Arrays

Arrays erstellen  
Arrayzugriff  
Exceptions

#### Schleifen

While-Schleifen  
Do-While-Schleifen  
Endlosschleifen  
For-Schleifen  
Welche Schleife?

#### Debugging

#### Quellen & Lizenz

# Arrays

- ▶ Ein Array fasst mehrere Variablen des gleichen Typs zusammen.

Beispiele: Ein Array von Integern enthält Ganzzahlen:

```
{ 4, 8, 15, 16, 23, 42 }
```

Ein Array von Strings enthält Wörter:

```
{ "Hallo", "Array", "Strings", "Peter" }
```

- ▶ Alle Werte müssen vom gleichen Typ sein.

**Falsch:** { 3, 18, 3.14, 'r', "Hallo" }

## Ablauf

### Arrays

- Arrays erstellen
- Arrayzugriff
- Exceptions

### Schleifen

- While-Schleifen
- Do-While-Schleifen
- Endlosschleifen
- For-Schleifen
- Welche Schleife?

### Debugging

### Quellen & Lizenz

# Arrays erstellen

- ▶ Aufbau eines Array, reines deklarieren:

```
datentyp [] arrayName;
```

- ▶ Aufbau eines Array mit vordefinierter Größe  $n$ , deklarieren und initialisieren:

```
datentyp [] arrayName = new datentyp [n];
```

Das Array wird dann mit Standardwerten gefüllt (bei Zahlen mit 0).

## Ablauf

### Arrays

#### Arrays erstellen

Arrayzugriff

Exceptions

### Schleifen

While-Schleifen

Do-While-Schleifen

Endlosschleifen

For-Schleifen

Welche Schleife?

### Debugging

### Quellen & Lizenz

## Arrays erstellen

- ▶ Um ein Array mit Werten zu initialisieren:

```
datentyp [] arrayName = newigentyp [] { w1, w2 };
```

```
// geht auch kürzer:
```

```
datentyp [] arrayName = { w1, w2 };
```

Die **Größe** eines Arrays kann nachträglich nicht mehr geändert werden. Zum Vergrößern oder Verkleinern muss ein neues Array angelegt werden. Alternativen zu Arrays kommen in der Vorlesung.

### Ablauf

### Arrays

#### Arrays erstellen

Arrayzugriff

Exceptions

### Schleifen

While-Schleifen

Do-While-Schleifen

Endlosschleifen

For-Schleifen

Welche Schleife?

### Debugging

### Quellen & Lizenz

# Arrays erstellen

► Beispiel für Arrays:

```
int [] arrayWithInteger = new int [] { 1, 2, 3 };  
// geht auch kürzer:
```

```
int [] arrayWithInteger = { 1, 2, 3 };
```

```
double [] arrayWithDouble = {1.4, 4.3};
```

```
String [] arrayWithStrings = { "wert1", "wert2" };
```

## Ablauf

### Arrays

#### Arrays erstellen

Arrayzugriff

Exceptions

### Schleifen

While-Schleifen

Do-While-Schleifen

Endlosschleifen

For-Schleifen

Welche Schleife?

### Debugging

### Quellen & Lizenz

# Arrayzugriff

- ▶ Zugriff auf das  $i$ -te Arrayelement:

```
arrayName[i]
```

**Achtung:** Der Index geht bei einem Array mit der Größe  $n$  von 0 bis  $n - 1$ !

- ▶ Die Größe des Arrays ( $n$ ) kann mit

```
arrayName.length
```

bestimmt werden.

## Ablauf

### Arrays

Arrays erstellen

Arrayzugriff

Exceptions

### Schleifen

While-Schleifen

Do-While-Schleifen

Endlosschleifen

For-Schleifen

Welche Schleife?

### Debugging

### Quellen & Lizenz

# Arrayzugriff

```
int[] arrayWithInt = {1,2,3,4}
System.out.println(arrayWithInt [2])
```



Ablauf

Arrays

Arrays erstellen

**Arrayzugriff**

Exceptions

Schleifen

While-Schleifen

Do-While-Schleifen

Endlosschleifen

For-Schleifen

Welche Schleife?

Debugging

Quellen & Lizenz

# Arrayzugriff

```
int [] arrayWithInt = {1,2,3,4}
System.out.println(arrayWithInt [2])
```

Ausgabe: 3

```
arrayWithInt.length
```



Ablauf

Arrays

Arrays erstellen

Arrayzugriff

Exceptions

Schleifen

While-Schleifen

Do-While-Schleifen

Endlosschleifen

For-Schleifen

Welche Schleife?

Debugging

Quellen & Lizenz

# Arrayzugriff

```
int [] arrayWithInt = {1,2,3,4}
System.out.println(arrayWithInt [2])
```

Ausgabe: 3

```
arrayWithInt.length
```

Ausgabe: 4



## Ablauf

### Arrays

Arrays erstellen

**Arrayzugriff**

Exceptions

### Schleifen

While-Schleifen

Do-While-Schleifen

Endlosschleifen

For-Schleifen

Welche Schleife?

### Debugging

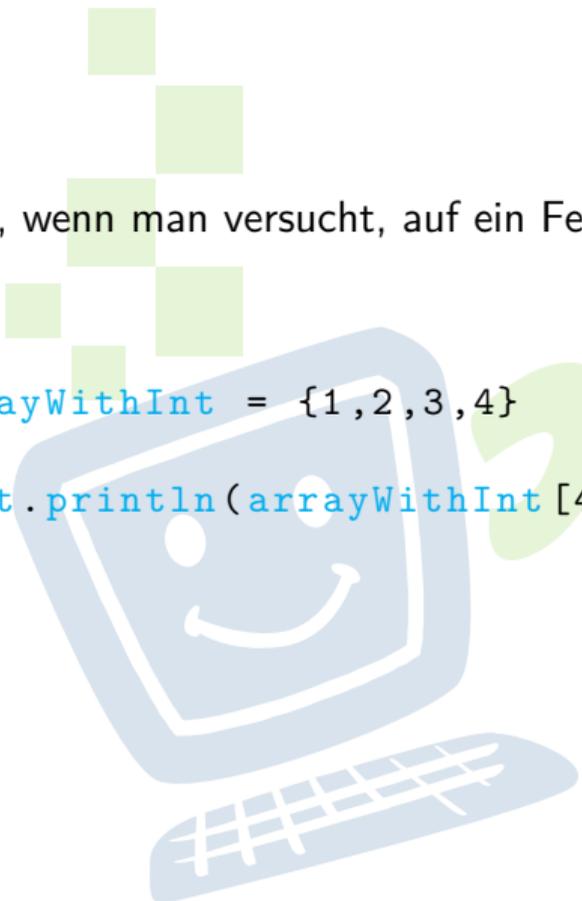
### Quellen & Lizenz

Was passiert, wenn man versucht, auf ein Feld außerhalb des Arrays zuzugreifen?

Beispiel:

```
int [] arrayWithInt = {1,2,3,4}
```

```
System.out.println(arrayWithInt[4]);
```



## Ablauf

### Arrays

Arrays erstellen

Arrayzugriff

Exceptions

### Schleifen

While-Schleifen

Do-While-Schleifen

Endlosschleifen

For-Schleifen

Welche Schleife?

### Debugging

### Quellen & Lizenz

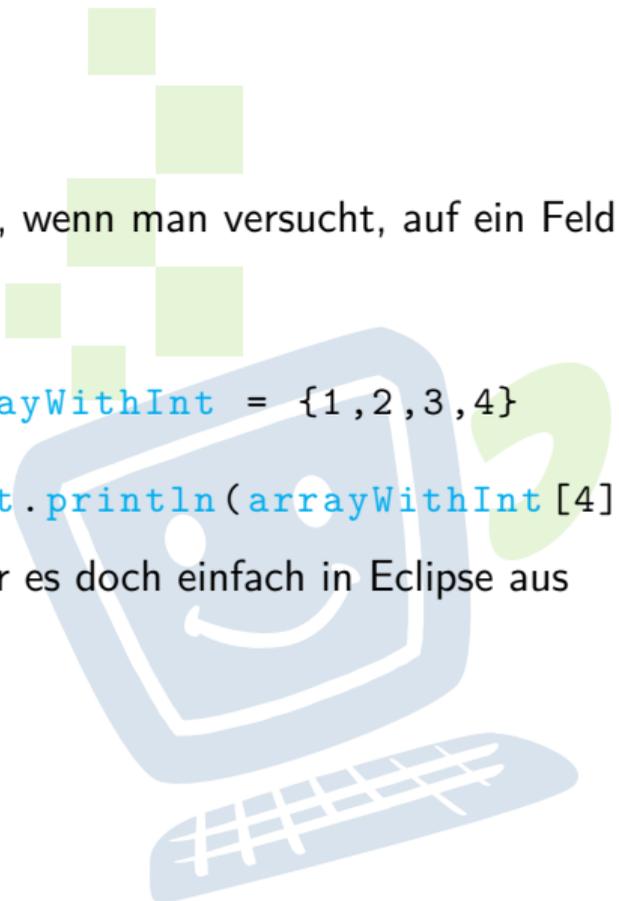
Was passiert, wenn man versucht, auf ein Feld außerhalb des Arrays zuzugreifen?

Beispiel:

```
int [] arrayWithInt = {1,2,3,4}
```

```
System.out.println(arrayWithInt[4]);
```

Probieren wir es doch einfach in Eclipse aus



## Ablauf

### Arrays

Arrays erstellen

Arrayzugriff

Exceptions

### Schleifen

While-Schleifen

Do-While-Schleifen

Endlosschleifen

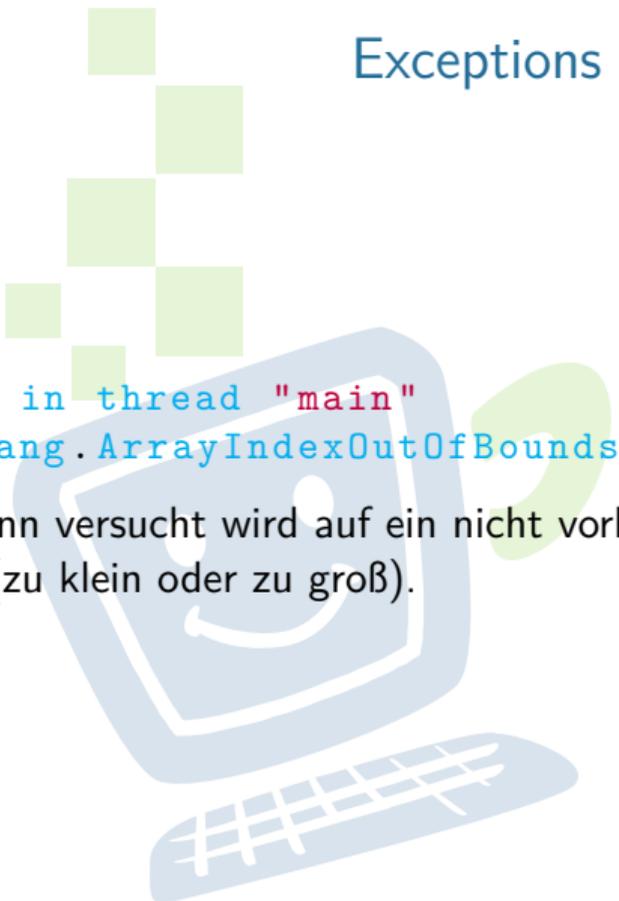
For-Schleifen

Welche Schleife?

### Debugging

### Quellen & Lizenz

# Exceptions



```
Exception in thread "main"  
    java.lang.ArrayIndexOutOfBoundsException
```

Tritt auf, wenn versucht wird auf ein nicht vorhandenes Feld eines Arrays zuzugreifen (zu klein oder zu groß).

## Ablauf

### Arrays

Arrays erstellen

Arrayzugriff

Exceptions

### Schleifen

While-Schleifen

Do-While-Schleifen

Endlosschleifen

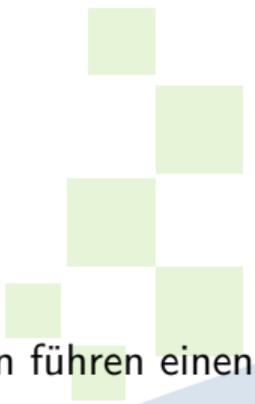
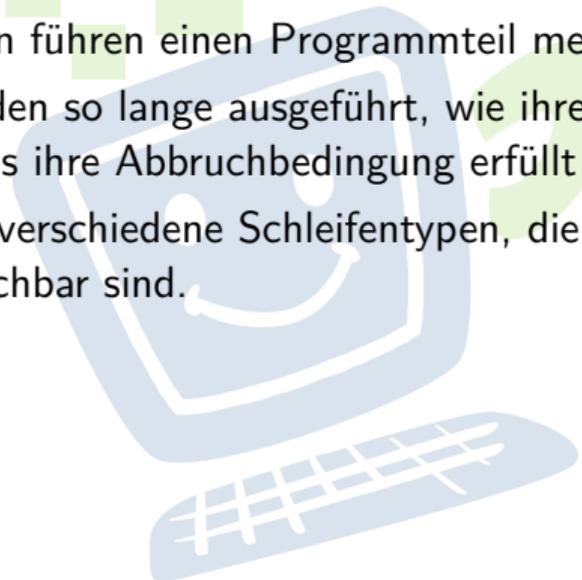
For-Schleifen

Welche Schleife?

### Debugging

### Quellen & Lizenz

# Schleifen

- 
- 
- ▶ Schleifen führen einen Programmteil mehrfach aus.
  - ▶ Sie werden so lange ausgeführt, wie ihre Schleifenbedingung wahr ist (bzw. bis ihre Abbruchbedingung erfüllt ist).
  - ▶ Es gibt verschiedene Schleifentypen, die aber alle untereinander austauschbar sind.

## Ablauf

### Arrays

Arrays erstellen  
Arrayzugriff  
Exceptions

### Schleifen

While-Schleifen  
Do-While-Schleifen  
Endlosschleifen  
For-Schleifen  
Welche Schleife?

### Debugging

### Quellen & Lizenz

# While-Schleifen

Syntax:

```
while (Bedingung) {  
    Anweisung1;  
    Anweisung2;  
    // ...  
}
```



Ablauf

Arrays

Arrays erstellen  
Arrayzugriff  
Exceptions

Schleifen

While-Schleifen

Do-While-Schleifen  
Endlosschleifen  
For-Schleifen  
Welche Schleife?

Debugging

Quellen & Lizenz

# While-Schleifen

Beispiel:

```
int zaehler = 0;
while (zaehler < 10) {
    System.out.println("Hallo Welt");
    zaehler++;
}
```



Ablauf

Arrays

Arrays erstellen  
Arrayzugriff  
Exceptions

Schleifen

While-Schleifen

Do-While-Schleifen  
Endlosschleifen  
For-Schleifen  
Welche Schleife?

Debugging

Quellen & Lizenz

# While-Schleifen

Beispiel:

```
int zaehler = 0;
while (zaehler < 10) {
    System.out.println("Hallo Welt");
    zaehler++;
}
```

Was gibt die Schleife aus?

Schauen wir uns auch das einfach in Eclipse an.

Ablauf

Arrays

Arrays erstellen  
Arrayzugriff  
Exceptions

Schleifen

While-Schleifen

Do-While-Schleifen  
Endlosschleifen  
For-Schleifen  
Welche Schleife?

Debugging

Quellen & Lizenz

# While-Schleifen

Beispiel:

```
int zaehler = 0;
while (zaehler < 10) {
    System.out.println("Hallo Welt");
    zaehler++;
}
```

Was macht die Schleife genau?

- ▶ zaehler ist am Anfang 0.
- ▶ Das Programm überprüft: Ist zaehler kleiner als 10?
- ▶ Ja, deshalb wird 'Hallo Welt' ausgegeben und zaehler um 1 erhöht.
- ▶ Dies wird so oft wiederholt bis zaehler nicht mehr kleiner als 10 ist.

Ablauf

Arrays

Arrays erstellen  
Arrayzugriff  
Exceptions

Schleifen

While-Schleifen

Do-While-Schleifen  
Endlosschleifen  
For-Schleifen  
Welche Schleife?

Debugging

Quellen & Lizenz

# Do-While-Schleifen

Syntax:

```
do {  
    Anweisung1;  
    Anweisung2;  
    // ...  
} while (Bedingung);
```

Anders als While-Schleifen wird eine Do-While-Schleife immer *mindestens einmal* durchlaufen.

Ablauf

Arrays

Arrays erstellen  
Arrayzugriff  
Exceptions

Schleifen

While-Schleifen

**Do-While-Schleifen**

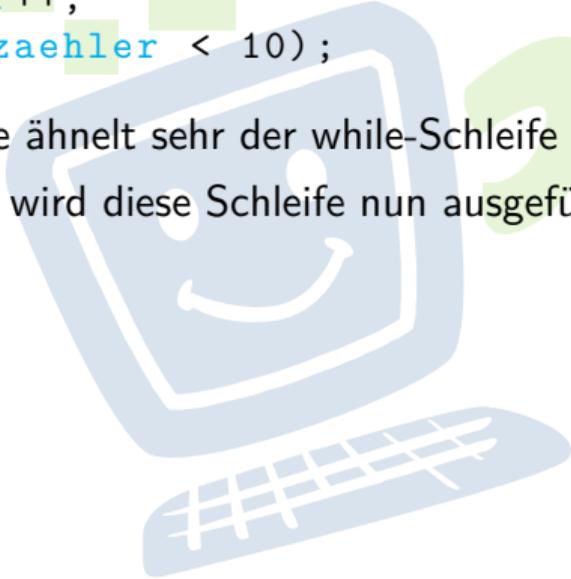
Endlosschleifen  
For-Schleifen  
Welche Schleife?

Debugging

Quellen & Lizenz

```
do {  
    System.out.println("Hallo_Welt");  
    zaehler++;  
} while (zaehler < 10);
```

Diese Schleife ähnelt sehr der while-Schleife von vorher.  
Doch wie oft wird diese Schleife nun ausgeführt?



## Ablauf

### Arrays

- Arrays erstellen
- Arrayzugriff
- Exceptions

### Schleifen

- While-Schleifen

- Do-While-Schleifen**

- Endlosschleifen
- For-Schleifen
- Welche Schleife?

### Debugging

### Quellen & Lizenz

```
do {  
    System.out.println("Hallo_Welt");  
    zaehler++;  
} while (zaehler < 10);
```

Diese Schleife ähnelt sehr der while-Schleife von vorher.

Doch wie oft wird diese Schleife nun ausgeführt?

Die Schleife wird ebenfalls 10mal ausgeführt.

## Ablauf

### Arrays

- Arrays erstellen
- Arrayzugriff
- Exceptions

### Schleifen

- While-Schleifen

- Do-While-Schleifen**

- Endlosschleifen
- For-Schleifen
- Welche Schleife?

### Debugging

### Quellen & Lizenz

## Do-While-Schleifen

Beispiel:

```
int zaehler = 10;
while (zaehler < 10) {
    System.out.println("Hallo_Welt");
    zaehler++;
}
```

und

```
int zaehler = 10;
do {
    System.out.println("Hallo_Welt");
    zaehler++;
} while (zaehler < 10);
```

Wie oft werden hier die Schleifen ausgeführt?

Ablauf

Arrays

Arrays erstellen  
Arrayzugriff  
Exceptions

Schleifen

While-Schleifen  
Do-While-Schleifen

Endlosschleifen  
For-Schleifen  
Welche Schleife?

Debugging

Quellen & Lizenz

## Do-While-Schleifen

Beispiel:

```
int zaehler = 10;
do {
    System.out.println("Hallo_Welt");
    zaehler++;
} while (zaehler < 10);
```

und

```
int zaehler = 10;
while (zaehler < 10) {
    System.out.println("Hallo_Welt");
    zaehler++;
}
```

### Achtung!

In diesem Beispiel werden die Anweisungen in der Do-While-Schleife *einmal* ausgeführt, die in der While-Schleife *überhaupt nicht*.

### Ablauf

#### Arrays

- Arrays erstellen
- Arrayzugriff
- Exceptions

#### Schleifen

- While-Schleifen
- Do-While-Schleifen**
- Endlosschleifen
- For-Schleifen
- Welche Schleife?

#### Debugging

#### Quellen & Lizenz

# Endlosschleifen

Bei Schleifen sollte man immer aufpassen: Wie oft werden hier die Anweisungen ausgeführt?

```
int zaehler = 0;
while (zaehler < 10) {
    System.out.println("Hilfe!");
}
```

```
int i = 10;
while (i > 0) {
    System.out.println("Hilfe!");
    i = i/2 + 1;
}
```

## Ablauf

### Arrays

- Arrays erstellen
- Arrayzugriff
- Exceptions

### Schleifen

- While-Schleifen
- Do-While-Schleifen

### Endlosschleifen

- For-Schleifen
- Welche Schleife?

### Debugging

### Quellen & Lizenz

# For-Schleifen

Syntax:

```
for (Initialisierung; Bedingung; Schritt) {  
    Anweisung1;  
    Anweisung2;  
    // ...  
}
```

- ▶ Zuerst wird die Initialisierungs-Anweisung ausgeführt.
- ▶ Dann wird die Bedingung geprüft.
  - ▶ Ist die Bedingung falsch, wird die Schleife verlassen.
  - ▶ Ist die Bedingung wahr, werden die Anweisungen im Schleifenkörper ausgeführt.
- ▶ Anschließend wird die Schritt-Anweisung ausgeführt. Meistens wird die Laufvariable inkrementiert.
- ▶ Danach wird wieder die Bedingung geprüft.

Ablauf

Arrays

Arrays erstellen  
Arrayzugriff  
Exceptions

Schleifen

While-Schleifen  
Do-While-Schleifen  
Endlosschleifen

For-Schleifen

Welche Schleife?

Debugging

Quellen & Lizenz

# For-Schleifen

Beispiel:

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

Entspricht dieser While-Schleife:

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

Ablauf

Arrays

Arrays erstellen  
Arrayzugriff  
Exceptions

Schleifen

While-Schleifen  
Do-While-Schleifen  
Endlosschleifen

For-Schleifen

Welche Schleife?

Debugging

Quellen & Lizenz

# For-Schleifen

For-Schleifen werden häufig im Zusammenhang mit Arrays eingesetzt.

Beispiel:

```
char [] abc = new char[] { 'a', 'b', 'c' };  
  
for (int i = 0; i < abc.length; i++) {  
    System.out.println(abc[i]);  
}
```

## Ablauf

### Arrays

- Arrays erstellen
- Arrayzugriff
- Exceptions

### Schleifen

- While-Schleifen
- Do-While-Schleifen
- Endlosschleifen

### For-Schleifen

- Welche Schleife?

### Debugging

### Quellen & Lizenz

## Wann welche Schleife?

- ▶ For-Schleifen wenn die *Anzahl der Durchläufe* im voraus *bekannt* ist
  - ▶ (Wenn man einen Zähler braucht)
- ▶ While/Do-While schleifen in anderen Fällen
  - ▶ (Wenn man keinen Zähler braucht)

### Ablauf

### Arrays

Arrays erstellen  
Arrayzugriff  
Exceptions

### Schleifen

While-Schleifen  
Do-While-Schleifen  
Endlosschleifen  
For-Schleifen

### Welche Schleife?

### Debugging

### Quellen & Lizenz

# Debugging

- ▶ Durch Doppelklick in einer Zeile setzt man einen Breakpoint.



```
System.out.println(i);
```

- ▶ Dann durch Klick auf den Käfer den Debuggingmodus starten.



- ▶ Über die Taskleiste kann man den Debuggingmodus steuern.



## Ablauf

### Arrays

Arrays erstellen  
Arrayzugriff  
Exceptions

### Schleifen

While-Schleifen  
Do-While-Schleifen  
Endlosschleifen  
For-Schleifen  
Welche Schleife?

## Debugging

### Quellen & Lizenz

# Debugging

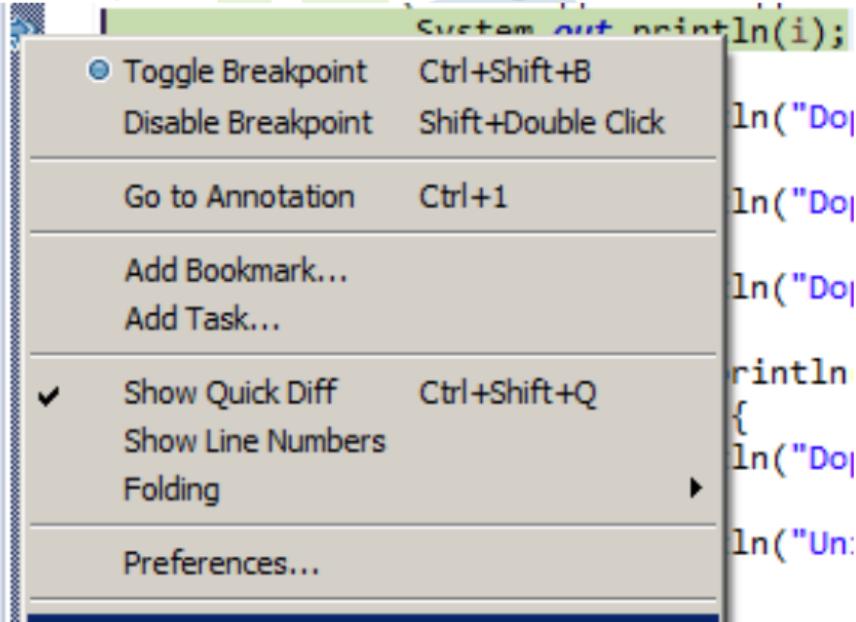
Beim Debugging von Schleifen sind Conditional Breakpoints nützlich.

- ▶ Dazu erst wie gewohnt einen Breakpoint setzen.



```
System.out.println(i);
```

- ▶ Dann per Rechtsklick die Eigenschaften des Breakpoints öffnen.



## Ablauf

### Arrays

Arrays erstellen  
Arrayzugriff  
Exceptions

### Schleifen

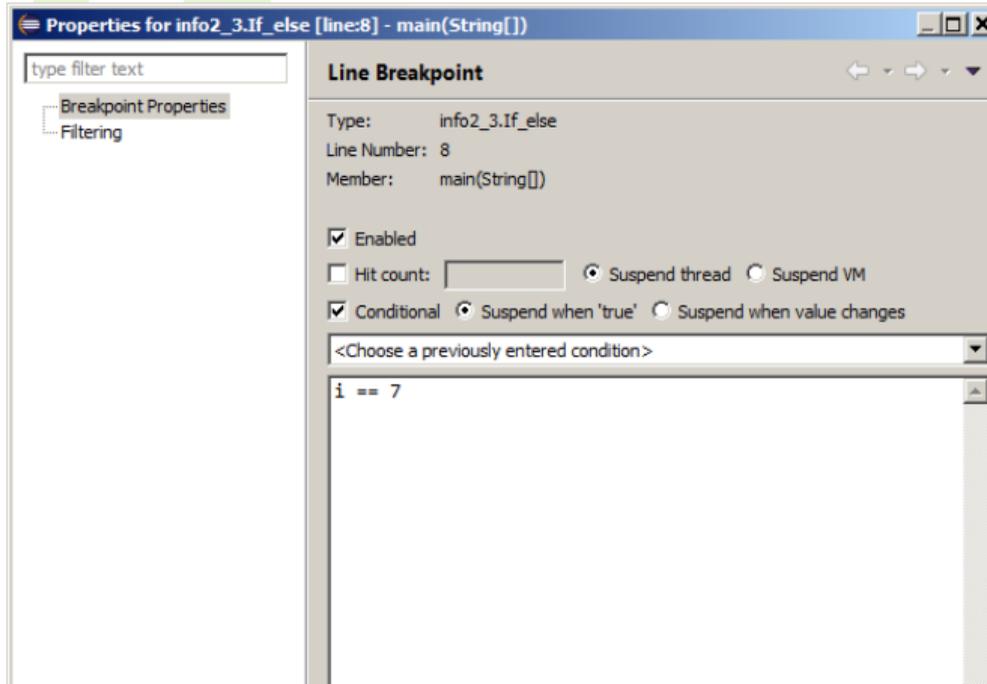
While-Schleifen  
Do-While-Schleifen  
Endlosschleifen  
For-Schleifen  
Welche Schleife?

## Debugging

### Quellen & Lizenz

# Debugging

Über *Condition* kann z.B. die Laufvariable auf einen bestimmten Wert überprüft werden. Nur wenn die Bedingung wahr ist, wird am Breakpoint angehalten.



Ablauf

Arrays

Arrays erstellen  
Arrayzugriff  
Exceptions

Schleifen

While-Schleifen  
Do-While-Schleifen  
Endlosschleifen  
For-Schleifen  
Welche Schleife?

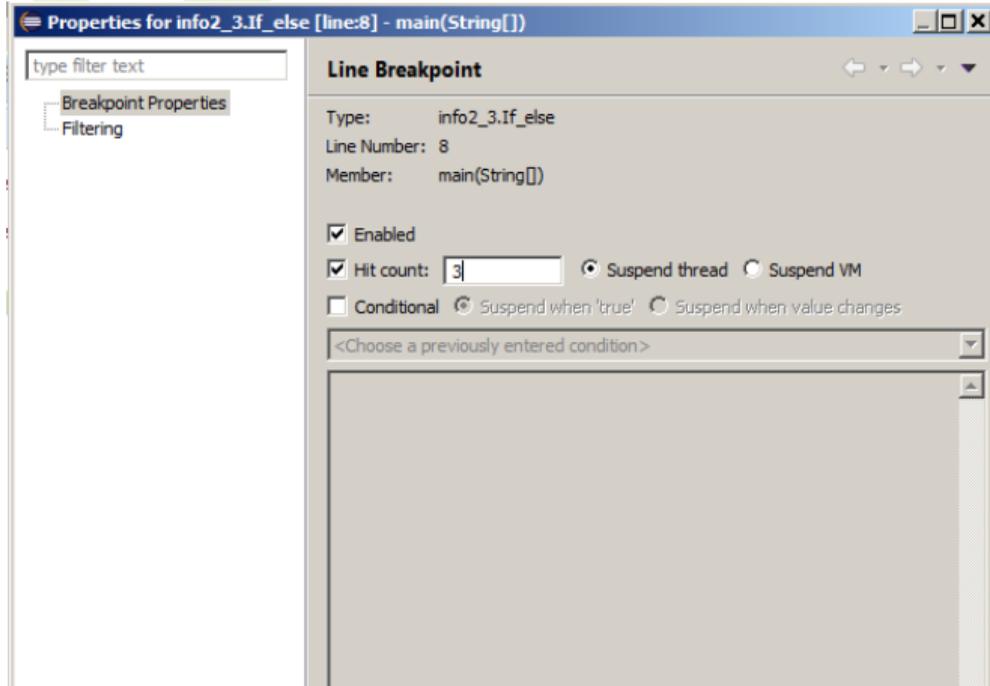
Debugging

Quellen & Lizenz

# Debugging

Mit dem *HitCount* kann man z.B. einstellen, dass erst ab dem 3. Durchlauf am Breakpoint gestoppt werden soll.

Der *HitCount* ist hilfreich, wenn die Schleife keine Laufvariable hat.



## Ablauf

## Arrays

- Arrays erstellen
- Arrayzugriff
- Exceptions

## Schleifen

- While-Schleifen
- Do-While-Schleifen
- Endlosschleifen
- For-Schleifen
- Welche Schleife?

## Debugging

## Quellen & Lizenz

## Quellen und Lizenz

- ▶ Original von ?
- ▶ 2013 von Felix Bürkle
- ▶ Überarbeitet 2015 von Eva Wolkwitz
- ▶ Überarbeitet 2016 von Dominik Hartfelder
- ▶ Überarbeitet 2016 von Giulia Maier
- ▶ Überarbeitet 2017 von Johannes Beierle
- ▶ Überarbeitet 2018 von Giulia Maier
- ▶ Überarbeitet 2020 von Erik Dyka

### Ablauf

#### Arrays

Arrays erstellen  
Arrayzugriff  
Exceptions

#### Schleifen

While-Schleifen  
Do-While-Schleifen  
Endlosschleifen  
For-Schleifen  
Welche Schleife?

#### Debugging

### Quellen & Lizenz