FACHSCHAFT INFORMATIK
HS Karlsruhe

Programmiervorkurs
Tag 3

Lukas Theurer

Ablauf

- 09:30 Uhr Besprechung der Übungen von Tag 2
- 10:00 Uhr Vorlesung
- 11:30 Uhr Mittagspause
  - 60 Minuten
- Gegen 12:30 Uhr Übungen im LI 136 u. LI 137

Inhaltsübersicht Vorkurs

FACHSCHAFT INFORMATIK
HS Karlsruhe

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

Arrays

- Was ist das überhaupt?
- Wofür nutzt man es?
- Wie kann es genutzt werden?

Was ist ein Array ?

FACHSCHAFT INFORMATIK
HS Karlsruhe

- Eine Art Box
- Fasst mehrere Variablen des gleichen Typs zusammen
- Beispiel: Mehrere Integer Variablen: `{1,3,5,6,18,99,42}`
- Oder Strings: `{"Hund", "Katze", "Maus"}`

Wofür nutzt man ein Array?

- Wird genutzt um mehrere Daten abzuspeichern und zugänglich zu machen

- Vergleichbar mit: „Nem Meter“
  - Hat mehrere gleich große Fächer
  - Es kann nur ein Typ von Schnaps abgestellt werden
    - Z. Bsp. Obstler oder Kräuterschnaps

Wie können sie genutzt werden?

FACHSCHAFT INFORMATIK HS Karlsruhe

- Um ein Array vom Typ **int** zu erstellen schreibt man:

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

- Für unser „Meter“-Beispiel:

```
Obstbrand [] schnapsMeter = new Obstbrand [8];
```

- Arrays werden, falls keine Werte zum füllen angegeben werden immer mit **0** initialisiert.
  - Also: Schnapsglas vorhanden, aber leer ☹

Wie können sie genutzt werden?

FACHSCHAFT INFORMATIK HS Karlsruhe

- Um ein Array mit Werten zu initialisieren kann man schreiben:

```
int [] vieleZahlen = new int [] { 1, 2, 3, 10, 11};
```

- Oder:

```
int [] vieleZahlen = {1, 2, 3, 10, 11};
```

- Das geht natürlich auch mit Strings:

```
String [] Tiere = {"Hund", "Katze", "Maus"};
```

Wissenswertes

FACHSCHAFT INFORMATIK
HS Karlsruhe

- Ist ein Array einmal mit einer bestimmten Größe initialisiert (erstellt) worden, so kann diese **Größe nicht mehr verändert werden**
- Zum vergrößern oder verkleinern muss ein neues Array angelegt und die Werte kopiert werden.
- Die Größe eines Arrays erhält man durch: `Tiere.length();`
- Z. Bsp. Über : `System.out.println(Tiere.length());`

Arrayzugriff

FACHSCHAFT INFORMATIK HS Karlsruhe

- Zugriff erfolgt über einen Index
  - Der Index beginnt bei 0 und endet bei n Elementen bei n-1

```
String [] Tiere = {"Hund", "Katze", "Maus"};

String ausgabe = Tiere[0];
//in ausgabe steht nun "Hund"
```

- In diesem Beispiel ist **n = 3**
- Der kleinste Index ist **0**, wenn man auf das **erste** Element zugreifen möchte
- Der größte Index ist **2**, wenn man auf das **letzte** Element zugreifen möchte

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.

While-Schleifen

```
while (Bedingung) {
    Anweisung;
    /*Code der mehrmals ausgeführt
    werden soll*/
}
```

While Schleifen

FACHSCHAFT INFORMATIK HS Karlsruhe

- Schleife erhöht mit jedem Durchlauf i um den Wert 1
- Schleife wird beendet sobald i größer oder gleich 100 ist
  - →Sie wird nur solange ausgeführt wie i < 100 ist

```
int i = 0;

while (i < 100){
    i++;
}
```

Was passiert hier?

FACHSCHAFT INFORMATIK
HS Karlsruhe

- Vor jedem Schleifendurchlauf wird die Bedingung überprüft.
- Da i am Anfang 0 ist, ist die Bedingung WAHR, also wird die Schleife das erste mal ausgeführt.
- Das letzte Mal, dass die Bedingung WAHR ist, ist bei i = 99

```
int i = 0;

while (i < 100){
    i++;
}
```

VORSICHT !!!

FACHSCHAFT INFORMATIK
HS Karlsruhe

- Verwechselt man den Vergleichsoperator, so bekommt man ungewollt eine Endlosschleife, in der das Pr0gramm dann fest hängt

- Oder wie hier: Die Bedingung ist nie WAHR und deshalb wird die Schleife NICHT ausgeführt

```
int i = 0;

while (i > 100){
    i++;
}
```

For-Schleifen

FACHSCHAFT INFORMATIK
HS Karlsruhe

```
for(Initialisierung; Bedingung; Schritt) {
    Anweisung;
    /*Code der mehrmals ausgeführt
    werden soll*/
}
```

- Zuerst wird die Initialisierungsanweisung ausgeführt
  - z. Bsp. `int zaehler = 0;`
- Dann wird die Bedingung geprüft und ggf. die Anweisung ausgeführt
  - z. Bsp. `zaehler < 10`
- Danach wird der Schritt ausgeführt.
  - z. Bsp. `zaehler++`

Das könnte dann so aussehen…

FACHSCHAFT INFORMATIK
HS Karlsruhe

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

- Ausgabe:

```
zaehler= 0
zaehler= 1
zaehler= 2
zaehler= 3
zaehler= 4
zaehler= 5
zaehler= 6
zaehler= 7
zaehler= 8
zaehler= 9
```

For- und While-Schleife

FACHSCHAFT INFORMATIK
HS Karlsruhe

- Die For-Schleife:

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

- Entspricht der While-Schleife:

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

Schleifen und Arrays

FACHSCHAFT INFORMATIK HS Karlsruhe

- Schleifen, insbesondere For-Schleifen, werden für das Durchlaufen eines Arrays eingesetzt.

```
String [] tiere = {"Hund", "Katze", "Maus"};

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

Ausgabe: Hund
Katze
Maus