

# Programmierkurs

Java – Programmierkurs  
Philipp Herzig, B.Sc.  
2009

# Datenstrukturen

- Wir beschäftigen uns heute mit Datenstrukturen.
- Wir wissen bereits wie wir Werte speichern können.
- Oftmals ist es nützlich wenn wir viele Werte des gleichen Typs in einer Sammlung ablegen könnte. Man bezeichnet solche Sammlungen als Datenstrukturen

# Beispiel

Beispiel CD und DVD Sammlung

# Anwendung Erzeugung

- Anwendung auf uns bekannte Typen
- Speicherung von vielen Ganzzahlen (int) in einem Array

```
int[] zahlenarray = {1, 5, 4, 3, 2, 10, 3}
```

# Anwendung Erzeugung II

- Sind die Elemente des Arrays bei der Erzeugung nicht bekannt (was meistens der Fall ist) kann ein array auch wie folgt erzeugt werden:

```
int[] zahlenarray = new int[10].
```

- Dabei gibt 10 hier beispielhaft die Anzahl der zur Verfügung stehenden Speicherplätze an

# Anwendung Erzeugung III

- Möglichkeit #3 ist die Deklaration eines Arrays mit späterer Definition

*int[] a;*

*anweisungen....*

*a = new int[10] oder a = {1, 2, 3, 4}*

# Zugriff

*int[] zahlenarray = {1, 5, 4, 3, 2, 10, 3}*

- mit folgender Anweisung können wir darauf zugreifen

*zahlenarray[3]* → liefert die 3 zurück

- **Warum die 3?**

# Zugriff II

- Großer Unterschied zur Mathematik. Dort beginnt man immer alles ab der 1 durchzuzählen in der Informatik (**meistens**) ab der 0
- Ebenfalls kann der an der Stelle gespeicherte Wert verändert werden
- $zahlenarray[3] = 4 \rightarrow$  modifiziert den an Index 3 gespeicherten wert



# Verwendung von Arrays

- Wir können nun mit array genauso arbeiten wie mit normalen Werten:

```
int x = 3;
```

```
int y = 3+x+zahlenarray[3];
```

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

# Länge eines Arrays

- Die Länge von einem Array lässt sich wie folgt ermitteln:

`zahlenarray.length;`

z.B. Ausgabe:

- `System.out.println(zahlenarray.length);`

# N-Dimensionale Arrays

- Bisher haben wir nur 1 dimensionale Arrays betrachtet. Wir können dies aber auf beliebige Dimensionen ausweiten.
- Analogie CD Schrank herstellen

# Definition

- → wohlgeformtes Array

```
int[][] zahlenarray = new int[10][10]
```

- → ausgefranstes Array

```
int[][] zahlenarray = new int[10][];
```

```
zahlenarray[0] = new int[5];
```

```
zahlenarray[1] = new int[10];
```

```
int[][] zahlenarray = {{1, 2, 3}, {2,3}, {1,2,4,5}};
```

*usw....*

# Zusammenfassung

- Arrays dienen als Datenspeicher von vielen Werten gleichen Typs
- Ein Array kann N Dimensionen haben
- Arrays sind semi-dynamisch, d.h. man kann sie während der Laufzeit erzeugen, sie aber nur durch Überschreiben in ihrer Größe verändern.

# Until next week....

Vielen Dank für die Aufmerksamkeit  
Viel Spaß!!!

