

Leonard Berresheim

06 Arrays

M21 - Grundlagen der Programmierung

Arrays

Erklärung

Ein *Array* ist eine geordnete **Sammlung** von Elemente desselben **Datentyps**, die man unter einem **gemeinsamen** Namen ansprechen kann. [1]

Vornamen
Oliver
Mohammad
Arlo
Anjali
Zoe
Elias

```
String vorname1 = "Oliver";  
String vorname2 = "Mohammad";  
String vorname3 = "Arlo";  
String vorname4 = "Anjali";  
String vorname5 = "Zoe";  
String vorname6 = "Elias";
```

Nicht sehr elegant. Und bei größeren Datenmenge nutzlos!

Arrays

Initialisierung

Die **Initialisierung** geschieht in 2 Schritten. Erst wird die Variable als Array **definiert**. Und dann ein Array **erzeugt** und dessen **größe** bestimmt.

```
Datentyp[] Arrayname;
```

Die **[]**-Klammern geben an, dass es sich bei *Arrayname* um ein **Array** handelt.

```
Arrayname = new Typ[Größe];
```

new gibt an, dass ein **neues** Array erzeugt wird und in den **[]**-Klammern wird die **Größe** des Arrays angegeben.

Die Initialisierung kann auch in **einer** Zeile ausgeführt werden.

```
Datentyp[] Arrayname = new Datentyp[Größe];
```

Arrays

Initialisierung

Beispiel:

```
int[] x;           // Definition der Variable x als Array vom Typ int
x = new int[10];   // Erzeugen von x als Array vom Typ int der Größe 10
```

Alternativ geht auch:

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

Dies erzeugt folgendes **leeres** Array:

x

Index	0	1	2	3	4	5	6	7	8	9
Wert	-	-	-	-	-	-	-	-	-	-

Achtung!

In den meisten Programmiersprachen wird beim *Indexing* mit **0** und **nicht** mit **1** angefangen.

Wenn also das Array eine Größe von **N** hat, dann befindet sich der höchste Wert an der Stelle **N-1**.

Arrays

Befüllen des Arrays

x

Index	0	1	2	3	4	5	6	7	8	9
Wert	-	-	-	-	-	-	-	-	-	-

Über den **Index** kann jetzt auf die **Elemente** des Arrays zugegriffen werden.

$x[0] = 5;$

x

Index	0	1	2	3	4	5	6	7	8	9
Wert	5	-	-	-	-	-	-	-	-	-

$x[6] = 3;$

x

Index	0	1	2	3	4	5	6	7	8	9
Wert	5	-	-	-	-	-	3	-	-	-

Arrays

Befüllen des Arrays

x

Index	0	1	2	3	4	5	6	7	8	9
Wert	-	-	-	-	-	-	-	-	-	-

Über den **Index** können jetzt Werte in das Array **geschrieben** werden.

$x[0] = 5;$

x

Index	0	1	2	3	4	5	6	7	8	9
Wert	5	-	-	-	-	-	-	-	-	-

$x[6] = 3;$

x

Index	0	1	2	3	4	5	6	7	8	9
Wert	5	-	-	-	-	-	3	-	-	-

Arrays

Befüllen des Arrays

Das **Array** kann auch direkt bei der **Initialisierung** befüllt werden.

```
int[] x = {5, 3, 10, 24, 1, 0, 3, -1, 78, 2};
```

Hier kann man sich die Angabe der **Größe** sparen.

x

Index	0	1	2	3	4	5	6	7	8	9
Wert	5	3	10	24	1	0	3	-1	78	2

Arrays

Auslesen des Arrays

x

Index	0	1	2	3	4	5	6	7	8	9
Wert	5	3	10	24	1	0	3	-1	78	2

Und über den **Index** können Elemente aus dem Array **ausgelesen** werden

```
System.out.println(x[6]);
```

oder

```
int add = x[0] + int[6];
```

oder

```
int summe = 0;
for(int i = 0; i < array.length; i++){
    summe += x[i];
}
```

array.length ist eine Methode, die die Länge des Arrays wiedergibt. Mehr dazu in einer späteren Vorlesung

Arrays

Aufgabe

Initialisiere folgendes Array:

Wochentage

Index	0	1	2	3	4	5	6
Wert	“Mo”	“Di”	“Mi”	“Do”	“Fr”	“Sa”	“So”

Arrays

Aufgabe

Initialisiere folgendes Array:

wochentage

Index	0	1	2	3	4	5	6
Wert	“Mo”	“Di”	“Mi”	“Do”	“Fr”	“Sa”	“So”

```
String[] wochentage = {“Mo”, “Di”, “Mi”, “Do”, “Fr”, “Sa”, “So”};
```

oder:

```
String[] wochentage = new String[7];  
wochentage[0] = “Mo”;  
wochentage[1] = “Di”;  
...  
wochentage[6] = “So”;
```

Arrays

Aufgabe

Initialisiere ein Array der Größe 100 und befülle es mit den Zahlen -99 bis 0:

zahlen

Index	0	1	2	3	4	5	6	...	98	99
Wert	-99	-98	-97	-96	-95	-94	-93		-1	0

Arrays

Aufgabe

Initialisiere ein Array der Größe 100 und befülle es mit den Zahlen -99 bis 0:

zahlen

Index	0	1	2	3	4	5	6	...	98	99
Wert	-99	-98	-97	-96	-95	-94	-93		-1	0

```
int[] zahlen = new int[100];
int wert = -99;

for(int i = 0; i < 100; i++){
    zahlen[i] = wert;
    wert++;
}
```

Arrays

foreach-Schleifen

Die **foreach**-Schleife wird speziell zum **Durchlaufen** von Arrays benutzt. Sie erlaubt es **successiv** auf **alle** Elemente im Array zuzugreifen.

Beispiel:

```
for (String tage : wochentage){  
    System.out.println(tage);  
}
```

Äquivalent zu:

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

Arrays

Mehrdimensionale Arrays

Mehrdimensionale Arrays lassen sich **ähnlich** erzeugen.

```
Datentyp [ ] [ ] [ ] Arrayname = new Datentyp[Zeilen][Spalten];
```

Die Anzahl der **[]**-Klammern gibt die Anzahl der **Dimensionen** an.

Beispiel:

```
int[ ] [ ] x = new int[2][3];
```

2x3 Matrix - x

Index	0	1	2
0	-	-	-
1	-	-	-

Arrays

Befüllen von Mehrdimensionalen Arrays

Hier benötigt man dieses Mal **zwei** Indexe um auf die Elemente im Array **zuzugreifen**.

Wobei der erste Index die **Zeilennummer** ist und der zweite die **Spalte**.

`x[0][2] = 5;`

2x3 Matrix - x

Index	0	1	2
0	-	-	5
1	-	-	-

`x[?][?] = 2;`

2x3 Matrix - x

Index	0	1	2
0	-	-	5
1	2	-	-

Arrays

Befüllen von Mehrdimensionalen Arrays

Hier benötigt man dieses Mal **zwei** Indexe um auf die Elemente im Array **zuzugreifen**.

Wobei der erste Index die **Zeilennummer** ist und der zweite die **Spalte**.

`x[0][2] = 5;`

2x3 Matrix - x

Index	0	1	2
0	-	-	5
1	-	-	-

`x[1][0] = 2;`

2x3 Matrix - x

Index	0	1	2
0	-	-	5
1	2	-	-

Arrays

Befüllen von Mehrdimensionalen Arrays

Hier benötigt man dieses Mal **zwei** Indexe um auf die Elemente im Array **zuzugreifen**.

Wobei der erste Index die **Zeilennummer** ist und der zweite die **Spalte**.

`x[0][2] = 5;`

2x3 Matrix - x

Index	0	1	2
0	-	-	5
1	-	-	-

`x[1][0] = 2;`

2x3 Matrix - x

Index	0	1	2
0	-	-	5
1	2	-	-

Arrays

Befüllen von Mehrdimensionalen Arrays

Auch hier kann man wieder direkt bei der **Initialisierung** das Array **befüllen**.

```
int[][] x = {{1, 12, 5}, {2, 3, 26}};
```

Wobei die Elemente **Zeilenweise** angegeben werden.

x kann auch betrachtet werden als **1**-Dimensionales Array was in jeder **Zeile** ein 1-Dimensionales Array beinhaltet.

```
int[][] x = new int[2][3];  
x[0] = {1, 12, 5};  
x[1] = {2, 3, 26};
```

2x3 Matrix - x

Index	0	1	2
0	1	12	5
1	2	3	26

Arrays

Aufgabe

Erstelle folgendes Array:

2x3 Matrix - x

Index	0	1	2
0	1	12	5
1	2	3	26
2	2	2	2
3	10	4	2

Arrays

Aufgabe

Erstelle folgendes Array:

4x3 Matrix - x

Index	0	1	2
0	1	12	5
1	2	3	26
2	2	2	2
3	10	4	2

```
int[][] x = {{1, 12, 5}, {2, 3, 26}, {2, 2, 2}, {10, 4, 2}};
```

Arrays

Aufgabe

Gegeben ist:

```
int[][] x = {{1, 12, 5}, {2, 3, 26}};
```

2x3 Matrix - x

Index	0	1	2
0	1	12	5
1	2	3	26

Lese alle Elemente in x aus. In folgender Reihenfolge: 1, 12, 5, 2, 3, 26

Arrays

Aufgabe

Gegeben ist:

```
int[][] x = {{1, 12, 5}, {2, 3, 26}};
```

2x3 Matrix - x

Index	0	1	2
0	1	12	5
1	2	3	26

Lese alle Elemente in x aus. In folgender Reihenfolge: 1, 12, 5, 2, 3, 26

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



www.htw-berlin.de

Quellen

[1] Grundkurs JAVA Von den Grundlagen bis zu Datenbank- und Netzanwendungen by Dietmar Abts