

Array

Ein **Array** (Feld, Reihung) ist die Zusammenfassung von Daten desselben Typs. Die Daten können Objekte sein oder auch Daten von einfachen Datentypen wie `int` oder `double`.

Deklaration eines Arrays

`Datentyp[] Variablenname`

Erzeugung eines arrays mit n Elementen

`Variablenname = new Datentyp[n]`

`Datentyp[] Variablenname = {erstes Element, zweites Element, ..., n-tes Element}`

Zugriff auf einzelne Elemente eines Arrays

Ein Array hat immer eine bestimmte Anzahl von Elementen, die nachträglich nicht mehr geändert werden kann. Der Zugriff auf die einzelnen Elementen erfolgt über den Index des Elementes in eckigen Klammern.

`Variablenname[0]` ist das erste Element,

`Variablenname[1]` ist das zweite Element,

`Variablenname[n-1]` ist das n-te und damit letzte Element des arrays. Hier steht der Text im Panel-Body

Länge eines Arrays

`Variablenname.length`

Beispiele

Zwei Möglichkeiten ein array mit den Primzahlen kleiner 20 zu erzeugen:

(1)

```
int[] zahlen = {2, 3, 5, 7, 11, 13, 17, 19};
```

(2)

```
int[] zahlen = new int[8];
```

```
zahlen[0]=2; zahlen[1]=3; zahlen[2]=5; zahlen[3]=7; zahlen[4]=11; zahlen[5]=13; zahlen[6]=17; zahlen[8]=19;
```

Nach der Anweisung `int laenge = zahlen.length` hat die Variable `laenge` den Wert 8.

Will man alle Elemente eines arrays verarbeiten, geschieht dies in der Regel mit einer Schleife. Besonders gut dafür geeignet ist eine Zählschleife.

Zählschleife

```
for (int Zählvariable = Anfangswert; Eintrittsbedingung; Änderung der Zählvariablen){  
    Anweisung;  
    Anweisung;  
    ...  
}
```

Beispiel:

(1) Der folgende Programmausschnitt druckt die Zahlen von 0 bis 9 auf der Konsole aus. Bei dieser Zählschleife bekommt die Zählvariable zunächst den Wert 0 zugewiesen. Die Schleife wird solange durchlaufen, wie die

Zählvariable einen Wert kleiner als 10 hat. Nach jedem Schleifendurchlauf wird die Zählvariable durch die Anweisung `i++` um 1 erhöht.

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

(2) Äquivalent zu dieser for-Schleife ist die folgende while-Schleife:

```
int i = 0;  
while (i<10) {  
    System.out.println(""+i);  
    i = i + 1;  
}
```

Verarbeitung eines Arrays mit einer Zählschleife

Beispiel 1 Der folgende Programmausschnitt berechnet die Summe der Quadrate der Zahlen von 1 bis 20:

```
int[] quadratzahlen = new int[20];  
for (int i=0; i<20; i++){  
    quadratzahlen[i]= i*i;  
}  
  
int summe = 0;  
for (int i=0; i<20; i++) {  
    summe = summe + quadratzahlen[i];  
}
```

Beispiel 2

In folgendem Programmausschnitt werden 10 rote Kreise (Circle-Objekte) mit dem Radius 5 an einer zufälligen Position der Zeichenfläche erzeugt.

```
Circle kreise = new Circle[10];  
for (int i = 0; i<10; i++){  
    kreise[i] = new Circle(Tools.randomNumber(5, 595), Tools.randomNumber(5,  
395), 5, Color.RED);  
}
```

1. Array-Variable deklarieren

```
Circle[] lichters;
```

- keine Festlegung der Array-Größe
- häufig als Attribut (Instanzvariable)

2. Array-Objekt erzeugen

```
lichters = new Circle[5];
```

- hier die Größe des Array festlegen.
- häufig erst im Konstruktor.

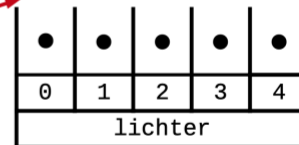
3. Array füllen

z.B. händisch:

```
lichters[0] = new Circle(0,0,25);
```

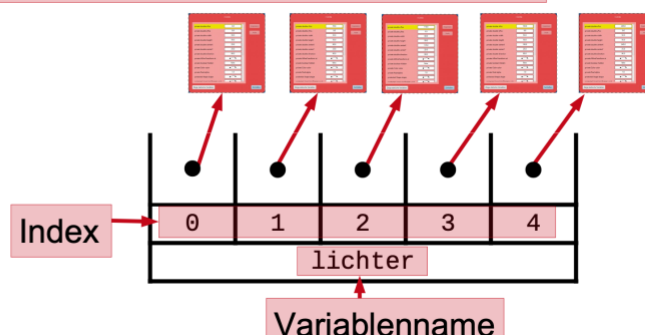
oder mit einer Schleife

Bis hierhin ist ein leeres Array mit 5 Speicherplätzen für Referenzen auf Objekte der Klasse `Circle` erstellt worden.



Abfrage der Länge eines Arrays

```
for(int i = 0; i < lichters.length; i++) {
    lichters[i] = new Circle(i*55, 0, 25);
}
lichters[3].setColor(Color.red);
lichters[0].setColor(Color.blue);
```



```

4 class Lauflichterprogramm
5 {
6     View fenster;
7     Circle[] lichters;
8
9     Lauflichterprogramm()
10    {
11        fenster = new View(550,50,"SaS-Programm");
12        lichters = new Circle[10];
13
14        for(int i = 0; i < lichters.length; i++) {
15            lichters[i] = new Circle(i*55, 0, 25);
16        }
17    }
18 }
  
```

Deklaration Arrayvariable

Erzeugen des Array-Objekts

Erzeugen der Objekte im Array in einer Zählschleife