

## Vorbereitende Aufgabe:

Auf einer Zeichenfläche lege man beliebig vier (oder mehr) Punkte fest. Nun soll von der Turtle jeder Punkt mit jedem anderen Punkt durch eine Strecke verbunden werden. Dazu soll eine Methode *vonPzuP* mit geeigneten Parametern formuliert werden, die zwei Punkte miteinander verbindet.

Zusatzaufgabe: Wie viele Strecken ergeben sich maximal bei  $n$  Punkten?

## Eindimensionale Arrays

In vielen Anwendungen der Informatik arbeitet man nicht mit Einzelwerten, sondern mit Wertemengen, die man in Tabellen oder Listen anordnet. Ein *Array* ist so eine Tabelle von Werten, die einzeln oder gemeinsam verarbeitet werden können.

Ein Array ist eine Tabelle gleichartiger Elemente wie zum Beispiel Zahlen, Zeichen o. A. Die Elemente sind von 0 bis zu einem Endwert nummeriert. Eine Nummer nennt man auch *Index*. Ist z.B.  $a$  ein Array bestehend aus ganzen Zahlen, dann enthält  $a[0]$  die erste Zahl,  $a[1]$  die zweite Zahl,  $a[2]$  die dritte Zahl usw.

### Deklaration

Mit

```
int[] a;
```

wird ein Array mit dem Namen  $a$  *deklariert*. Die Elemente des Arrays sind hier vom Typ `int`. Die Anzahl der Elemente ist damit noch nicht festgelegt und es ist auch noch kein Speicherplatz reserviert. Mit der Deklaration legt man nur den Typ der Elemente und den Sichtbarkeitsbereich (global oder lokal) fest.

### Erzeugung bzw. Initialisierung

Vor der Benutzung muss ein Array erzeugt werden. Die Anweisung

```
a = new int[5];
```

erzeugt ein Array mit 5 Integer-Elementen. Die 5 Zahlen können dann über  $a[0]$ ,  $a[1]$ ,  $a[2]$ ,  $a[3]$  und  $a[4]$  angesprochen werden. Die Indizierung beginnt in Java immer mit 0.

Mit einer weiteren Anweisung

```
a = new int[100];
```

kann später der Variablen  $a$  ein größeres Array zugewiesen werden. Die aktuelle Länge eines Arrays ist über `a.length` erreichbar.

Elemente eines neu erzeugten Arrays werden in Java automatisch *initialisiert*. Numerische Elemente bekommen den Wert 0, `boolean`-Elemente den Wert `false`.

## Benutzung

Mit `a[3]=7;` wird das vierte Element des Arrays zu einer 7. Mit

```
for(int i=0; i<a.length; i++)
{ a[i] = i*i; }
```

werden alle Elemente in einem Durchgang initialisiert.

## Initialisierungs-Tricks

### bei der Erzeugung

Mit

```
a = new int[] {0, 1, 4, 9, 16};
```

können direkt bei der Erzeugung (nach der Deklaration) die einzelnen Elemente angegeben werden.

### bei der Deklaration

Mit

```
int[] a = {0, 1, 4, 9, 16};
```

können direkt die einzelnen Elemente schon bei der Deklaration erzeugt und initialisiert werden. Die angegebene Deklaration ist identisch mit folgendem Programmcode:

```
int[] a;
a = new int[5];
a[0] = 0;
a[1] = 1;
a[2] = 4;
a[3] = 9;
a[4] = 16;
```

## Mehrdimensionale Arrays

Wenn ein Array als Elemente wiederum ein Array enthält, so erhält man eine zweidimensionale Tabelle, also eine *Matrix*. Die Matrix ist ein Array aus Zeilen, jede Zeile ist wiederum ein Array aus Spaltenelementen. Entsprechend können auch höherdimensionale Arrays angelegt werden.

### Deklaration

```
int[] [] a;
```

### Erzeugung bzw. Initialisierung

```
a = new int[4][3];
```

erzeugt ein Array mit 4 Zeilen und 3 Spalten.

## Benutzung

Mit `a[2][1]=7`; wird in der dritten Zeile und der zweiten Spalte eine 7 gespeichert.

Mit `a.length` erhält man die Anzahl der Zeilen, mit `a[1].length` die Anzahl der Spalten in der zweiten Zeile.

## Initialisierungs-Tricks

### bei der Erzeugung

Mit

```
a = new int[][] {{1,2,3}, {4,5,6}, {7,8,9}, {10,11,12}};
```

können direkt bei der Erzeugung (nach der Deklaration) die einzelnen Elemente angegeben werden.

### bei der Deklaration

Mit

```
int[][] a = {{1,2,3}, {4,5,6}, {7,8,9}, {10,11,12}};
```

können direkt die einzelnen Elemente schon bei der Deklaration erzeugt und initialisiert werden. Damit entsteht dann ein Array der folgenden Form:

1	2	3
4	5	6
7	8	9
10	11	12

## Übungsaufgabe 1:

Auf einer Zeichenfläche lege man beliebig eine bestimmte Anzahl Punkte fest, deren Koordinaten in Arrays gespeichert werden sollen. Nun soll von der Turtle jeder Punkt mit jedem anderen Punkt durch eine Strecke verbunden werden.

## Übungsaufgabe 2:

Auf einer Zeichenfläche lege man eine bestimmte Anzahl Punkte fest, die alle mit gleichem Abstand auf einem Kreis liegen und deren Koordinaten in Arrays gespeichert werden sollen. Nun soll von der Turtle wieder jeder Punkt mit jedem anderen Punkt durch eine Strecke verbunden werden.