

Arrays

Eindimensionales Array:

```
public static void main(String[] args) {
    int[] intArray = { 1, 3, 5, 7 };
    for (int i = 0; i < intArray.length; i++) {
        System.out.println(intArray[i]);
    }
}
```

Zweidimensionales Array:

```
float[][] matrix = new float[3][];
matrix[0] = new float[] { 1f };
matrix[1] = new float[] { 2f, 3f };
matrix[2] = new float[] { 4f, 5f, 6f };
```

Bsp 1: Array definieren

Erzeugen Sie ein Integer-Array mit einer von Tastatur eingelesenen Anzahl von Elementen. Befüllen Sie das Array mit absteigenden Werten beginnend bei der eingelesenen Zahl. Z.B. bei einer Eingabe von 5 sollen die Elemente des Array mit den Werten 5, 4, 3, 2 und 1 belegt werden.

Bsp 2: Suchen im Array

Erzeugen Sie ein Integer-Array mit einigebbar vielen Elementen. Befüllen Sie das Array mit Zufallswerten zwischen 0 und 9. Ein Zufallswert in diesem Bereich kann mit `(int) (Math.random()*10)` ermittelt werden. Lesen Sie eine zweite Zahl von der Tastatur und geben Sie aus, wie oft diese Zahl im Array enthalten ist.

Bsp 3: Array invertieren

Schreiben Sie eine Methode, welche die Reihenfolge der Elemente eines Integer-Arrays umkehrt. Das Array soll dabei nicht kopiert werden. Testen Sie Ihre Methode mit sinnvollen Werten.

Bsp 4: Element einfügen

Schreiben Sie eine Methode, um in ein Array von Buchstaben (`char`) an parametrisierbarer Stelle einen Buchstaben einzufügen. Das letzte Element im Array geht dabei verloren. Implementieren Sie auch eine sinnvolle Fehlerbehandlung.

Bsp 5: Nicht enthaltene Elemente

Erzeugen Sie wie in Beispiel 2 beschrieben ein Integer-Array mit Zufallswerten. Geben Sie alle Werte zwischen 0 und 9 aus, die im Array nicht enthalten sind. Diskutieren Sie Vor- und Nachteile Ihrer Lösung.

Bsp 6: Array kopieren

Schreiben Sie eine Methode, welche ein zweidimensionales float-Array kopiert. Verwenden Sie dabei nicht die `clone`-Methode. Überlegen Sie sich eine sinnvolle Signatur für Ihre Methode. Testen Sie Ihre Methode mit repräsentativen Werten.

Bsp 7: Zeilenweise Summen

Schreiben Sie eine Methode, um zeilenweise Summen eines zweidimensionalen float-Arrays zu bilden. Testen Sie Ihre Methode auch mit irregulären Matrizen.

Bsp 8: Summensuche

Erzeugen Sie ein Integer-Array mit einer von Tastatur eingelesenen Größe und initialisieren Sie die Werte des Arrays mit Zufallszahlen (siehe dazu Blatt 4 Beispiel 2). Lesen Sie eine weitere Zahl von Tastatur und ermitteln Sie, ob die Summe zweier unterschiedlicher Zahlen des Arrays die eingegebene Zahl ergibt.

Bsp 9: Spiel des Lebens

Es gibt eine Spielwelt als Matrix von Zellen, die entweder leben können oder tot sind. Jede Zelle hat 8 Nachbarn. Randzellen haben die Zellen des gegenüberliegenden Randes als Nachbarn. Aus der momentanen Zellpopulation kann die Population der nächsten Generation durch folgende Regeln berechnet werden:

- Hat eine tote Zelle genau drei lebende Nachbarn, erwacht sie zum Leben.
- Hat eine lebende Zelle zwei oder drei lebende Nachbarn, bleibt sie am Leben.
- Alle anderen lebenden Zellen sterben.

Schreiben Sie ein Programm, das eine Zahl n sowie eine $n \times n$ -Matrix aus Nullen und Einsen von einer Datei einliest. Die Matrix verkörpert die Initialpopulation: Eine Eins bedeutet lebende Zelle, eine Null eine tote Zelle. Nach jeder berechneten Generation soll die Spielwelt ansprechend ausgegeben und eine Zeile von Tastatur eingelesen werden. Ist die Eingabe `quit`, so soll das Programm beendet werden; ansonsten soll die nächste Generation berechnet werden.