

Praktikum C: Arrays ein- und mehrdimensional

Programmierung II

Wintersemester 2021/22

Prof. Dr.-Ing. Marco Block-Berlitz

Prof. Dr.-Ing. Dietrich Kammer

Dipl.-Ing. Jan Roeper

Arrays und Matrizen

C01

Deklariieren Sie eine Liste a vom Typ `int` und füllen Sie die Liste mit 10 positiven Zahlen Ihrer Wahl. Verwenden Sie dabei die Literale Erzeugung.

C02

Deklariieren Sie $n \times m$ -Matrix vom `int`. Weisen Sie jedem Eintrag der Matrix einen zufälligen Wert zwischen 4 und 10 zu. Schreiben Sie ebenfalls eine Funktion, die alle Einträge dieser Matrix in Matrixform ausgibt.

C03

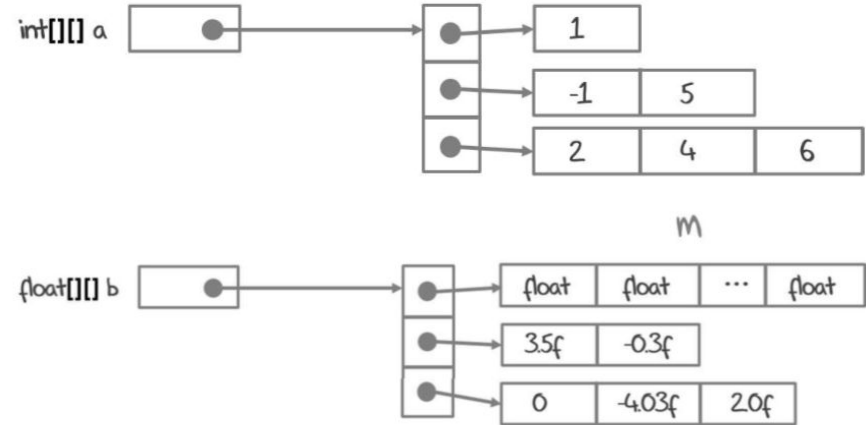
Wie lassen sich die folgenden Zeilen jeweils im Speicher interpretieren. Geben Sie dafür Zeichnungen an:

```
int[][] a = new int[n][m];  
int[][] c = {{1,2},{3}};
```

Speicherbelegungen

C04

Welche Programmzeilen erzeugen die beiden folgenden Speicherkonstellationen:



Vektorarithmetik

C05

Werten Sie das folgende Programm aus und geben Sie die Belegungen für c, d, e und f nach Abarbeitung des Programms an.

```
public static boolean[][] falsiere (boolean[][] l) {
    boolean[][] kopie = l.clone();
    for (boolean[] d1 : kopie)
        for (boolean d2 : d1)
            d2 = false;
    return kopie;
}

public static void main (String[] args) {
    boolean[][] a = {{true}, {true, false}, {false, false}};
    boolean[][] b = falsiere(a);
    boolean c = a == b;
    b[2][1] = true;
    b[1] = a[0];
    boolean d = a[0][0] == b[0][0];
    boolean e = a[2][1] != b[2][1];
    a[0][0] = true;
    boolean f = b[1][0] != b[2][0];
}
```

Vektorarithmetik

C06

Schreiben Sie eine Methode `addVektoren`, die als Eingaben zwei `int`-Arrays `a` und `b` erhält. Die Funktion liefert das Ergebnis der elementweisen Addition beider Vektoren, falls Sie die gleiche Dimension besitzen und ansonsten `null` (damit repräsentieren wir ein leeres Objekt). Geben Sie weiterhin eine Funktion `zeigeVektor` an, die den Inhalt eines Vektors ausgibt. Anschließend sollten beide Funktionen wie folgt ohne Fehlermeldungen getestet werden können:

```
int[] a = {1,2,3};           // funktionierendes Beispiel
int[] b = {4,5,6};
zeigeVektor(a);
zeigeVektor(b);
int[] c = addVektoren(a, b);
zeigeVektor(c);

int[] d = {1,2};             // fehlerhaftes Beispiel
int[] e = {4,5,6};
zeigeVektor(d);
zeigeVektor(e);
int[] f = addVektoren(d, e);
zeigeVektor(f);
```

Vektoren & Ränder

C07

Analog zu Aufgabe C05 sollen Funktionen für Vektorsubtraktion und das Produkt aus Vektor und Skalar (bzw. Skalar und Vektor) implementiert werden. Alle Parameter sind wieder ganzzahlig.

C08

In dem Projekt Conway's Game of Life haben wir die Randproblematik dadurch gelöst, dass wir einen zusätzlichen Rand bereitgestellt haben. Nehmen Sie den Rand wieder heraus und ändern Sie die Version derart, dass die Überprüfung der Nachbarschaft intelligent und fehlerfrei erfolgt.