

Java Übungsblatt 2

1 Arrays

Aufgabe 1 (M7.1 = CCC4.24)

Array invertieren. Schreibe eine Methode `invert(a)`, die die Reihenfolge der Elemente eines Arrays `a` umkehrt.

Definiere in der `main`-Methode ein Array `a`, invertiere es mit `invert(a)` und gebe das invertierte Array auf der Konsole aus.

Aufgabe 2 (CCC4.3.1): Berge zeichnen

Für die nächste Schatzsuche müssen Bonny Brain und die Crew über Berge und Hügel gehen. Sie bekommt vorher die Höhenmeter mitgeteilt und möchte sich vorher einen Eindruck vom Profil machen.

Aufgabe:

- Schreibe ein Programm mit einer Methode `printMountain(int[] altitudes)`, die ein Array mit Höhenmetern in eine ASCII-Darstellung umsetzt.
- Die Höhe soll dargestellt werden über ein Multiplikationszeichen `*` in genau dieser Höhe von einer Grundlinie. Die Höhen können beliebig sein, aber nicht negativ.

Beispiel:

Das Array `{ 0, 1, 1, 2, 2, 3, 3, 3, 4, 5, 4, 3, 2, 2, 1, 0 }` soll dargestellt werden als:

```
5          *
4         * *
3        *** *
2       **   **
1      **     *
0 *           *
```

Die erste Spalte dient der Verdeutlichung und muss nicht umgesetzt werden.

Optionale Erweiterung:

Deute statt mit `*` mit den Symbolen `/`, `\`, `-` und `^` an, ob wir aufsteigen, absteigen, auf einem Plateau sind oder an der Spitze stehen.

```
5          ^
4         / \
3        --/ \
2       -/   \
1      -/     \
0 /           \
```

Aufgabe 3 (CCC4.4.1) Mini-Sudoku auf gültige Lösung prüfen

Da Überfälle ziemlich anstrengend sind, braucht Bonny Brain einen Ausgleich und beschäftigt sich mit Sudoku. Ein Sudoku-Spiel besteht aus 81 Feldern in einem 9- \times -9-Gitter. Das Gitter lässt sich in neun Blöcke zerlegen, jeder Block ist ein zweidimensionales Array der Größe 3 \times 3. In jedem dieser Blöcke muss jede Zahl von 1 bis 9 genau einmal vorkommen – keine darf fehlen.

Aufgabe:

Schreibe ein Programm, das ein zweidimensionales Array mit neun Elementen daraufhin testet, ob alle Zahlen von 1 bis 9 vorkommen.

Beispiel:

- Das folgende Array ist eine gültige Sudoku-Belegung:

```
int[][] array = {  
    { 1, 2, 3 },  
    { 4, 5, 6 },  
    { 7, 8, 9 }  
};
```

- Das folgende Array ist keine gültige Sudoku-Belegung:

```
int[][] array = { { 1, 2, 3 }, { 4, 5, 6 }, { 7, 8, 8 } };
```

Der Fehler könnte etwa gemeldet werden mit "missing: 9".

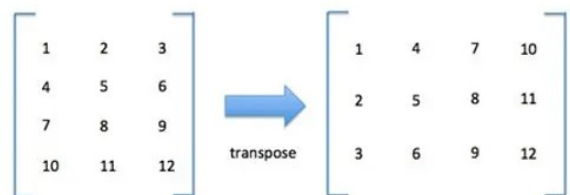
Aufgabe 4

Matrixtransposition. Schreiben Sie eine Methode `transpose(a)`, die eine

Matrix `a` transponiert, d.h. x- und y-Positionen aller Einträge vertauscht (s. rechts).

Geben Sie das originale und das transponierte Array mit einer (selbstgeschriebenen) Methode `printArray(a)` auf der Konsole aus, so dass Sie ihre Inhalte sehen können.

Sie können den Typ der Matrix (int, double, char, boolean, String, java.awt.Point etc.) selbst festlegen, aber es muss sinnvoll darstellbar sein. :-)



* Aufgabe 5 (M7.15)

Variable Anzahl von Parametern. Schreiben Sie eine Methode `poly`, die ein Polynom beliebigen Grades berechnet (z.B. soll `poly(x, c2, c1, c0)` das Polynom $c2 \cdot x^2 + c1 \cdot x + c0 = 0$ berechnen). Die Anzahl der Koeffizienten c_i soll variabel sein und den Grad des Polynoms bestimmen. *

* Aufgabe 6 (M7.9)

Lösung eines linearen Gleichungssystems. Schreiben Sie ein Programm zum Lösen eines linearen Gleichungssystems mit n Unbekannten:

$$\begin{array}{cccccc} a_{0,0} x_0 & + a_{0,1} x_1 & + \dots & + a_{0,n-1} x_{n-1} & = & a_{0,n} \\ a_{1,0} x_0 & + a_{1,1} x_1 & + \dots & + a_{1,n-1} x_{n-1} & = & a_{1,n} \\ \dots & & & & & \\ a_{n-1,0} x_0 & + a_{n-1,1} x_1 & + \dots & + a_{n-1,n-1} x_{n-1} & = & a_{n-1,n} \end{array}$$

Lesen Sie die Zahl n sowie die Koeffizienten $a_{0,0} \dots a_{n-1,n}$ von einer Datei ein und berechnen Sie den Lösungsvektor $x_0 \dots x_{n-1}$ nach dem *gaußschen Eliminationsverfahren*. Wählen Sie die Koeffizienten so, dass eine Lösung möglich ist. Versuchen Sie, Ihr Programm in sinnvolle Methoden zu zerlegen.

Tip: Schreiben Sie das Programm zuerst ohne Einlesen aus einer Datei, sondern geben Sie die Zahlen in der `main`-Methode vor. Falls Sie Zeit und Musse haben, versuchen Sie das Einlesen zu implementieren.

2 Quellen der Aufgaben:

M: Hanspeter Mössenböck, Sprechen Sie Java? Eine Einführung in das systematische Programmieren, 5. Auflage Heidelberg 2014.

U: Christian Ullenboom, Java ist auch eine Insel. Einführung, Ausbildung, Praxis, 16. Ausgabe Bonn 2022.

CCC: Christian Ullenboom, Captain CiaoCiao erobert Java. Ein Trainingsbuch für besseres Java, Bonn 2021.