Mehrdimensionale Felder

Wenn ein Feld als Elemente wiederum ein Feld enthält, so entsteht eine zweidimensionale Tabelle oder Matrix.

Die Matrix ist ein Feld aus Zeilen, jede Zeile ist wiederum ein Feld aus Spaltenelementen. Entsprechend können auch höherdimensionale Arrays angelegt werden.

Grundlagen

Deklaration

```
int[][] neueMatrix;
String[][] neueMatrix2;
```

Allgemein:
`meinDatentyp[][] neueMatrix3;`

Initialisierung

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

erzeugt ein Array mit 4 Zeilen und 3 Spalten.

0	0	0
0	0	0
0	0	0
0	0	0

Benutzung

Mit `neueMatrix[2][1]=7;` wird in der dritten Zeile und der zweiten Spalte eine 7 gespeichert. Mit `neueMatrix.length` erhält man die Anzahl der Zeilen; mit `neueMatrix[1].length` die Anzahl der Spalten in der zweiten Zeile.

Initialisierungs-Tricks

Bei der Initialisierung

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

Aufgabe 1
Es soll auf verschiedene Arten ein zweidimensionales 3x5-Feld definiert werden:

1	3	5	7	9
11	13	15	17	19
21	23	25	27	29

a) Trenne Deklaration und Initialisierung.
b) Initialisiere direkt bei der Deklaration.
c) Deklariere zunächst das Feld und füge die Werte mit Hilfe von zwei geschachtelten Zählwiederholungen ein.
d) Schreibe eine Methode, welche die Matrix in obiger Form in der Konsole ausgibt.
Tipp: Benutze „\t" als Tabulator-Einrückung

Aufgabe 2
Implementiere eine Klasse `Sitzplan`! Diese Klasse besitzt ein Attribut `String[][] platz`, welches die Namen der Schüler an der entsprechenden Position speichert.
a) Programmiere auch alle nötige get()/set()-Methoden zum Aus- und Eingeben von einzelnen Schülern
b) Passe die Methode von 1d an diese Aufgabe an.

Aufgabe 3*
Zweidimensionale Felder lassen sich auch mit verschieden langen Zeilen erzeugen. So kann z.B. die 1. Zeile 1 Spalte haben, die 2. Zeile 2 Spalten etc...

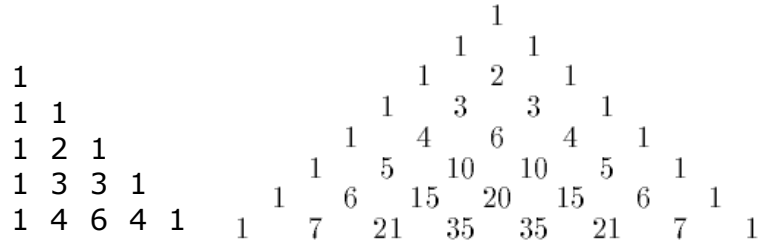
```
int zeilen = 5;
int [][] tabelle;                       //Deklaration der Tabelle
tabelle = new int[zeilen][];            //Festlegen der Zeilenzahl der Tabelle
                                        //Spalten noch nicht festgelegt

                                        //Erzeugen der Zeilen der Tabelle
for (int i = 0; i < zeilen; i++) {      //Durchlauf aller Zeilen
      tabelle[i] = new int[i+1];        //tabelle[i] bekommt i+1 Spalten }
```

Zum Durchlauf wählt man durch `tabelle.length` die Anzahl der Zeilen der Tabelle (zweidim. Array) und durch `tabelle[i].length` die Anzahl der Spalten der Zeile mit Index `i` ermitteln:

```
// Durchlauf aller Zeilen
for (int i = 0; i < tabelle.length; i++)
                                        // Durchlauf aller Spalten der Zeile
                                        // mit Index i
    for (int j = 0; j < tabelle[i].length; j++)
      tabelle[i][j] = i+1;              // Zugriff auf Tabelleneintrag
```

Implementiere eine Klasse `PascalschesDreieck`, welche das Pascalsche Dreieck speichert und verwaltet und ausgibt.

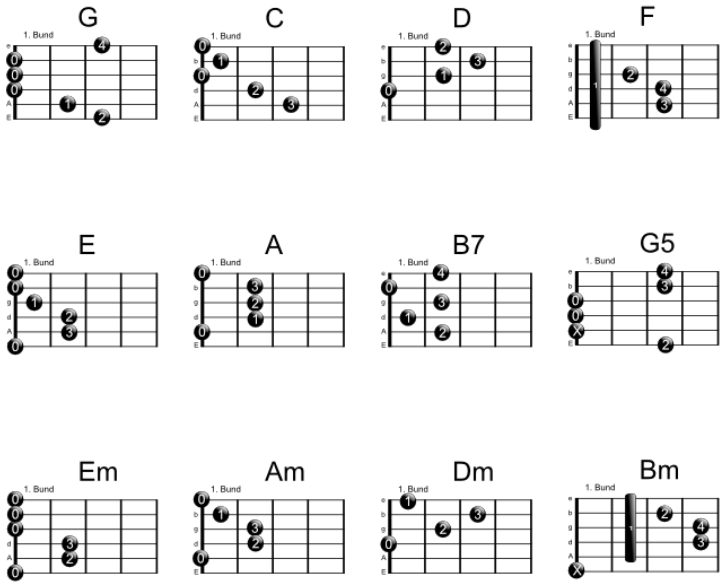


Aufgabe 4*
Implementiere das Spiel TicTacToe für folgende Spiele

a) Spieler-Spieler

b) Spieler-Computer

Aufgabe x*
Griffe auf der Gitarre lassen sich in mehrdimensionalen Matrizen vom Datentyp `int` speichern.



Schreibe eine Klasse Gitarre mit dem Attribut `griffe`, welches in der Form `int[griffNr][saite][Bund]` die Griffe speichert. Der gespeicherte Zahlenwert gibt dabei den Finger (1:Zeigefinger bis 4:kleiner Finger) an.

Folgende Methoden soll es mindestens geben:

public void griffEingeben(`int[][] neuerGriff`);
public void griffAusgeben(int griffNr);
public void griffAusgeben(String ton);
public void alleGriffeAusgeben();