

Aufgabe 1: "Find Text"**[12 Punkte]**

Schreiben Sie eine Methode, welche es ermöglicht ein als `char[]` gegebenen **Such-Text** (`searchText`) in einem als `char[][]` gegebenen **Text-Feld** (`textField`) zu suchen.

`boolean findText(char[][] textField, char[] searchText)`

Dabei darf der **Such-Text** nur vertikal von link oben nach rechts unten vorkommen (siehe Beispiel).

```
...
char[][] textField = {{'a','b','c','d','H','e','f','g','x'},
                      {'h','i','j','k','l','U','m','n','a','b'},
                      {'o','p','q','r','s','t','G','u'},
                      {'v','w','x','y','z','a','b','O','y'}};

char[] searchText = {'H','U','G','O'};

if (findText(textField, searchText)) {
    Out.print("kommt vor!");
} else {
    Out.print("kommt nicht vor!");
}
...
```

Das zu vervollständigende Programmgerüst können Sie über **MOODLE** herunterladen.

Stellen Sie sicher, dass Ihr Programm mit beliebigen Textfeldern arbeiten kann. Sichern Sie Ihr Methode entsprechend ab.

Vervollständigen Sie dazu folgendes Programm:

```
class FindText {
    public static void main(String [] args) {
        char[][] textField = {{'a','b','c','d','H','e','f','g','x'},
                              {'h','i','j','k','l','U','m','n','a','b'},
                              {'o','p','q','r','s','t','G','u'},
                              {'v','w','x','y','z','a','b','O','y'},
                              {},
                              null};

        char[] searchText = {'H','U','G','O'};

        Out.print("\nText-Feld:\n-----\n");
        print(textField);
        Out.print("\n-----\n");
        print(searchText);

        if (findText(textField, searchText)) {
            Out.print("\nkommt vor!");
        } else {
            Out.print("\nkommt nicht vor!");
        }
    } // end main

    static boolean findText(char[][] texts, char[] searchText) {
        // TODO -----
    } // end findText()

    static void print(char[][] texts) { ... }
    static void print(char [] text) { ... }
} // end class FindText } // end main()
```

Abgabe:

→ Mit Ihrem Code **vervollständigtes JAVA-Programm "FindText"**.

Aufgabe 2: "Cut-Away"**[12 Punkte]**

Gesucht ist eine Methode, die es ermöglicht, den Inhalt einer Matrice (**m**) um einen bestimmte Anzahl von Elementen (**cutAway**) zu "beschneiden" und das Ergebnis als neues zweidimensionales Felder zurück zu geben.

Dabei soll, wie im Beispiel gezeigt ein Matrice so "beschnitten" werden, sodass jede Zeile um den Wert von **cutAway** in ihrer Länge reduziert werden.

Realisieren Sie ein Methode, mit dem aus einer übergebenen Matrix aus Ganzzahlwerten eine neue "Beschnittene" erstellt und diese zurück gibt.

```
int[][] cutAway(int[][] matrix, int cutAway)
```

Achten Sie bei Ihrer Implementierung auf die korrekte Behandlung von Sonder- und Fehlerfällen.

```
...
    int[][] m = {
        {1, 2, 3, 4, 5, 6, 7},
        {8, 9, 0, 1, 2, 3, 4},
        {5, 6, 7, 8, 9, 0, 1, 3, 4},
        {},
        {2, 3, 4, 5, 6, 7, 8},
        null,
        {9, 0, 1, 2, 3, 4, 5}};

...
int cutAway = 2;
int[][] c = cutAway(m, cutAway);
...
```

Liefert als Ergebnis zurück...

```
{ {1, 2, 3, 4, 5},
  {8, 9, 0, 1, 2},
  {5, 6, 7, 8, 9, 0, 1},
  {},
  {2, 3, 4, 5, 6},
  null,
  {9, 0, 1, 2, 3}}
```

Das zu vervollständigende Programmgerüst können Sie über **MOODLE** herunterladen.

Stellen Sie sicher, dass Ihr Programm mit beliebigen Textfeldern arbeiten kann. Sichern Sie Ihr Methode entsprechend ab.

Vervollständigen Sie dazu folgendes Programm:

```
class CutAway {
    public static void main(String [] args) {

        int[][][] m = {
            {1, 2, 3, 4, 5, 6, 7},
            {8, 9, 0, 1, 2, 3, 4},
            {5, 6, 7, 8, 9, 0, 1, 3, 4},
            {},
            {2, 3, 4, 5, 6, 7, 8},
            null,
            {9, 0, 1, 2, 3, 4, 5}};

        Out.print("\nAusgangs-Matrize");
        Out.print("\n-----");
        print(m);

        Out.print("\nWie viel soll abgeschnitten werden: ");
        int cutAway = In.readInt();

        Out.print("\n\n    ... cut away " + cutAway + "\n");
        int[][][] c = cutAway(m, cutAway);

        Out.print("\nBeschnittene-Matrize");
        Out.print("\n-----");
        print(c);

    } // end main()

    static void print(int[][][] m) { ... }

    static int[][][] cutAway(int[][][] m, int cutAway) {

        // TODO -----

    }

} // end CutAway
```

Abgabe:

→ Mit Ihrem Code **vervollständigtes JAVA-Programm "Text-Analyser"**.