

```

/*****
ALP2 - Uebung8 - Jan Kellermann / Astrid Bertram / Gruppe 15          Tutorin: Susanne Richter
*****/

```

```

public class Zettel8 {

    static void showMatrix(float a[][]) {          // Matrix 1:1 ausgeben.
        System.out.println(); System.out.println("Matrix (" + a.length + "x" + a[0].length + ")");
        for (int i=0; i<a.length; i++) {
            for (int k=0; k<a[i].length; k++) System.out.print(a[i][k] + " ");
            System.out.println();
        }
        System.out.println();
    }

    public static int zeroLine1(float a[][]) throws Exception {
        boolean killflag=false;          // Das "Ein Element war !=0"-Flag mit false initialisieren;
        for (int i=1; i<a.length; i++) { // Zeile fr Zeile
            killflag=false;              // Am Anfang jeder Zeile das Flag auf false setzen;
            for(int k=1; k<a[i].length; k++) // Spalte fr Spalte
                if (a[i][k] !=0.0f)      // berprfen ob aktueller Feldinhalt !=0
                    killflag=true;      // Flag setzen.
            if (killflag==false)         // Wenn Flag in dieser Zeile nicht gesetzt wurde
                return i;                // return ZeilenNr.
        }
        throw new Exception("Keine 0-Zeile gefunden !!!");
    }

    public static int zeroLine2(float a[][]) throws Exception {
        int k=0;                          // Hier brauchen wir die Variable k auch ausserhalb ihrer Schleife
        for (int i=1; i<a.length; i++) {   // Zeile fr Zeile
            for (k=(a[i].length)-1; k>0; k--) // Spalte fr Spalte, ABER rckwts,
                if (a[i][k] !=0.0f) break; // und sofort abbrechen wenn Feldinhalt!=0
            if (k==0)                       // Wenn aus vorheriger Schleife mit index 0
                return i;                   // return ZeilenNr.
        }
    }
}

```

ausgestiegen,

```
    }
    throw new Exception("Keine 0-Zeile gefunden !!!");
}

public static int zeroLine3(float a[][]) throws Exception {
    try {
        int k=0;
        for(int i=1; i<=a.length; i++) { // derung: i<a.length gedert in i<=a.length
            for(k=(a[ i ].length)-1; k>0; k--) if (a[ i ][ k ]!=0.0f) break;
            if (k==0) return i;
        }
        throw new Exception("Keine 0-Zeile gefunden !!!");
    } catch (ArrayIndexOutOfBoundsException e) {
        throw new Exception("Ungueltige Zeile/Spalte im Array angesprungen!!!");
    }
}
}
```

/ Aufgabe 2.d)*

Die erste Variante bricht nicht ab, selbst wenn der erste Fall eines nicht-0-Feldes in einer Zeile Auftritt. Demnach wird immer die gesamte Zeile berprft, auch wenn schon ersichtlich ist, das mindestens 1 Element nicht 0 ist.

Die 2. Variante springt sofort nach erkennen eines nicht-0 Feldes in die nhste Zeile und spart somit viel Zeit und Rechen-Ressourcen. Zudem wird eine boolean-Variable nicht benigt. Einziger Nachteil ist, das das Feld an der Stelle 0 nicht belegt werden darf. D.h. das Feld muss eine Spalte mehr haben als fr die Nutzwerte benigt wird.

Die 2. Variante mit dem Sentinel we die ressourcenschonendste, schnellste und bersichtlichste, daher we diese zu bevorzugen. Gerade bei greren Arrays und komplizierteren Abfragen kann die eine Menge Zeit und Rechenleistung einsparen.

*/

```
public static void main(String[] args) throws Exception{

    // Dummy-Array initialisieren
    float a[][] = new float[ 5][ 7]; // 5 und 7 sind natrlich Beispielwerte !
    for(int i=0; i<a.length; i++) for(int k=0; k<a[ i].length; k++) a[ i][ k]=1.284f; // Fllwert 1.284f

    // Die folgende Zeile einfach auskommentieren oder nicht zum 0-zeile einbauen
    for(int k=0; k<a[ 2].length; k++) a[ 2][ k]=0.0f; // 0-Zeile ist hier z.B. die Zeile mit index 2.

    // Die Matrix ausgeben
    showMatrix(a);

    // Aufgaben nacheinander durchlaufen lassen
    System.out.println(zeroline1(a)); // AUFGABE 2. a
    System.out.println(zeroline2(a)); // AUFGABE 2. b
    System.out.println(zeroline3(a)); // AUFGABE 2. c
}
}
```