

## Aufgabenblatt: Arrays

(1.) Füllen Sie die Lücken aus:

- (a.) Ein eindimensionales Array  $p$  enthält vier Elemente. Die Namen von den Elementen sind: \_\_\_\_\_, \_\_\_\_\_, \_\_\_\_\_ und \_\_\_\_\_!
- (b.) Die Angabe von einem Array, seinem Typ und die Anzahl seiner Dimension nennt man die \_\_\_\_\_ des Arrays
- (c.) In einem zweidimensionalen Array gibt der erste Index die \_\_\_\_\_ von einem Element und der zweite Index die \_\_\_\_\_ von einem Element.
- (d.) Ein  $n \times m$  Array enthält \_\_\_\_\_ Zeilen, \_\_\_\_\_ Spalten und \_\_\_\_\_ Elemente!
- (e.) Der Name von dem Element in Zeile 3 und Spalte 5 im Array  $d$  ist \_\_\_\_\_.

(2.) Definiert wurde ein  $2 \times 3$  Array  $t$  vom Typ `integer`!

- (a.) Schreiben Sie eine Anweisung, welche das Arrays  $t$  deklariert und erzeugt!
- (b.) Wie viele Spalten hat  $t$ ?
- (c.) Wie viele Zeilen hat  $t$ ?
- (d.) Wie viele Elemente hat  $t$ ?
- (e.) Schreiben Sie die Namen aller Elemente in der zweiten Spalte von  $t$  auf!
- (f.) Schreiben Sie die Namen aller Elemente in der dritte Zeile von  $t$  auf!
- (g.) Schreiben eine einzige Anweisung, welches das Element in der Zeile 1 und der Spalte 2 auf Null setzt.
- (h.) Schreiben Sie Anweisungen, welche alle Elemente von  $t$  auf 0 setzt! Verwenden Sie dabei keine Wiederholungsanweisung!
- (i.) Schreiben Sie eine verschachtelte Wiederholungsanweisung, die alle Elemente auf 0 setzt!
- (j.) Schreiben Sie eine verschachtelte Wiederholungsanweisung, die alle Elemente durch den Benutzer einliest!
- (k.) Schreiben Sie eine Serie von Anweisungen, welche den kleinsten Wert in  $t$  bestimmt!
- (l.) Schreiben Sie eine Serie von Anweisungen, welche alle Elemente der ersten Zeile ausgibt!
- (m.) Schreiben Sie eine Serie von Anweisungen, welche die Summe der Elemente der 3. Spalte ausgibt!

(3.) Ein zweidimensionales Zahlenfeld mit  $n$  Zeilen und  $m$  Spalten wird von Ihnen mit Werten gefüllt.

- (a.) Bestimmen Sie anschließend die Position (Zeile, Spalte) des größten und kleinsten Wertes im Zahlenfeld.
- (b.) Erweitern Sie das Programm so, dass die Position (Zeile, Spalte) des größten und kleinsten Wertes im Zahlenfeld berechnet wird.
- (c.) Bestimmen Sie jeweils das kleinste Element in jeder Zeile. Berechnen Sie den Mittelwert dieser kleinsten Werte.
- (I.) Erstellen Sie ein Struktogramm.
- (II.) Schreiben Sie ein Programm!
- (III.) Schreiben Sie eine Methode, die vom Hauptprogramm aufgerufen wird!

(4.) Aufgaben zu Matrizen:

- (a.) Erstellen Sie eine Methode `void MultiplyMatrixWithSkalar(int a, int A[][], int aA[][])`, welche die Matrix  $A$  mit dem Skalar  $a$  multipliziert und geben Sie das Ergebnis mit einer Methode `void OutputMaxtrix(int A[][])` aus!
- (b.) Erstellen Sie eine Methode `void AddMatrixWithMatrix(int A[][], int B[][], int C[][])`, welche die Matrix  $A$  zu der Matrix  $B$  addiert und geben Sie das Ergebnis mit einer Methode `void OutputMaxtrix(int A[][])` aus!
- (c.) Erstellen Sie eine Methode `void MultiplyMatrixWithMatrix(int A[][], int B[][], int C[][])`, welche die Matrix  $A$  zu der Matrix  $B$  addiert und geben Sie das Ergebnis mit einer Methode `void OutputMaxtrix(int A[][])` aus!

## Aufgabenblatt: Arrays

(1.) Füllen Sie die Lücken aus:

(a.) Ein eindimensionales Array  $p$  enthält vier Elemente. Die Namen von den Elementen sind:

$p[0]$ ,  $p[1]$ ,  $p[2]$  und  $p[3]$ !

(b.) Die Angabe von einem Array, seinem Typ und die Anzahl seiner Dimension nennt man die **Deklaration** des Arrays!

(c.) In einem zweidimensionalen Array gibt der erste Index die **Zeile** von einem Element und der zweite Index die **Spalte** von einem Element.

(d.) Ein  $n \times m$  Array enthält  $m$  Zeilen,  $n$  Spalten und  $n*m$  Elemente!

(e.) Der Name von dem Element in Zeile 3 und Spalte 5 im Array  $d$  ist  $d[3][5]$ .

(2.) (a.) `int t[][] = new int[2][3];`

(b.) 2 Spalten!

(c.) 3 Zeilen

(d.)  $2*3=6$  Elemente

(e.) `t[1][0]`, `t[1][1]`, `t[1][2]`

(f.) `t[0][2]`, `t[1][2]`

(g.) `t[1][0]=0;`

(h.) `int t[][] = { {0, 0}, {0, 0}, {0, 0} };`

(i.) `int t[][] = new int[2][3];`

`for (int i=0;i<t.length;i++)`

`for (int j=0;j<t[i].length;j++)`

`t[i][j]=0;`

(j.)

`String tmp;`

`int t[][] = new int[2][3];`

`for (int i=0;i<t.length;i++)`

`for (int j=0;j<t[i].length;j++){`

`tmp=JOptionPane.showInputDialog (null, "Geben Sie die "+i+" Spalte und die "+j+" te Zeile ein:");`

`t[i][j]=Integer.parseInt(tmp);`

`}`

Allererste Zeile über `public class...` ist: `„import javax.swing.JOptionPane;“`

(k) `int t[][] = new int[2][3];`

`// Mit Daten füllen`

`for (int i=0;i<t.length;i++)`

`for (int j=0;j<t[i].length;j++)`

`t[i][j]=(i+1)*(j+1)+10;`

`// Maximum ermitteln`

`int Min;`

`// Ersten Wert ermitteln`

`// Max=-10000 reicht nicht, was wenn alle Werte darunter liegen?`

`Min=t[0][0];`

`// Durchlauf durch alle Elemente`

`for (int i=0;i<t.length;i++)`

`for (int j=0;j<t[i].length;j++)`

`// Gemerktes Maximum kleiner als aktueller Wert?`

`if (Min>t[i][j]){`

`Min=t[i][j];`

`System.out.println("Klaus");`

`}`

`System.out.println("Minimum ist:"+Min);`

(l.)

`int t[][]={{11,12,13},{21,22,23}};`

`for (int i=0;i<t[0].length;i++)`

`System.out.println(i+"-tes Element der 1. Zeile:"+t[0][i]);`

(m.)

```

int t[][]={{11,12,13},{21,22,23}};
int Summe=0;
for (int i=0;i<t.length;i++)
    Summe+=t[1][i];
System.out.println("Summe:"+Summe);

```

(3.) (b.)

```

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

        int a[][]={{4,2,-1},{6,8,2}};
        int min=a[0][0];
        int max=a[0][0];
        int minSpIndex=0,minZlIndex=0;
        int maxSpIndex=0,maxZlIndex=0;
        for (int i=0;i<a.length;i++)
            for (int j=0;j<a[0].length;j++){
                if (min>a[i][j]) {
                    min=a[i][j];
                    minSpIndex=j;
                    minZlIndex=i;
                }
                if (max<a[i][j]) {
                    max=a[i][j];
                    maxSpIndex=j;
                    maxZlIndex=i;
                }
            }
        System.out.println("Das kleinste Element ist:"+a[minZlIndex][minSpIndex]+
            " und hat den Spaltenindex:"+(int)(minSpIndex+1)
            +" und Zeilenindex:"+(int)(minZlIndex+1));
        System.out.println("Das größte Element ist:"+a[maxZlIndex][maxSpIndex]+
            " und hat den Spaltenindex:"+(int)(maxSpIndex+1)
            +" und Zeilenindex:"+(int)(maxZlIndex+1));
    } // end of main
}

```

```
(c.)
public class Test
{
    public static void main(String[] args) {

        double a[][]={{4,1,3},{6,8,2}};
        double min[] =new double[a.length];

        for (int i=0;i<a.length;i++) {
            min[i]=a[i][0];
            for (int j=0;j<a[0].length;j++){
                if (min[i]>a[i][j])
                    min[i]=a[i][j];
            }
        }
        double mittelwert=0;
        for (int i=0;i<a.length;i++) {
            mittelwert=mittelwert+min[i];
        }
        mittelwert=mittelwert/a.length;
        System.out.println("Der Mittelwert ist:"+mittelwert+"!");
    } // end of main
}
```

(4)(a.)

```
public class Test {

    static void MultiplyMatrixWithSkalar(double a, double A[][], double aA[][]) {
        for (int i=0;i<A.length;i++) {
            for (int j=0;j<A[0].length;j++){
                aA[i][j]=a*A[i][j];
            }
        }
    }

    static void OutputMatrix(double A[][]) {
        for (int i=0;i<A.length;i++) {
            for (int j=0;j<A[0].length;j++){
                System.out.print(A[i][j]+" ");
            }
            System.out.println("");
        }
    }

    public static void main(String[] args) {
        double a=2.5;
        double A[][]={{4,1,3},{6,8,2}};
        //double aA[][]={{4,1,3},{6,8,2}};
        double aA[][] = new double[A.length][A[0].length];

        MultiplyMatrixWithSkalar(a,A,aA);
        OutputMatrix(aA);
    } // end of main
}
```

(b.)

```
public class Test {

    static void MultiplyMatrixWithSkalar(double A[][], double B[][], double C[][]) {
        for (int i=0;i<A.length;i++) {
            for (int j=0;j<A[0].length;j++){
                C[i][j]=A[i][j]+B[i][j];
            }
        }
    }

    static void OutputMatrix(double A[][]) {
        for (int i=0;i<A.length;i++) {
            for (int j=0;j<A[0].length;j++){
                System.out.print(A[i][j]+" ");
            }
            System.out.println("");
        }
    }

    public static void main(String[] args) {
        double A[][]={{4,1,3},{6,8,2}};
        double B[][]={{2,3,5},{1,9,4}};
        double C[][] = new double[A.length][A[0].length];

        MultiplyMatrixWithSkalar(A,B,C);
        OutputMatrix(C);
    } // end of main
}
```

(c.)

```

public class Test {

    static void MultiplyMatrixWithMatrix(double A[][], double B[][], double C[][]) {
        double sum=0;
        for (int i=0;i<A.length;i++) {
            for (int j=0;j<B[0].length;j++){
                sum=0;
                for (int k=0;k<A[0].length;k++){
                    sum=sum+A[i][k]*B[k][j];
                }
                C[i][j]=sum;
            }
        }
    }

    static void OutputMatrix(double A[][]) {
        for (int i=0;i<A.length;i++) {
            for (int j=0;j<A[0].length;j++){
                System.out.print(A[i][j]+" ");
            }
            System.out.println("");
        }
    }

    public static void main(String[] args) {
        double A[][]={{4,1,3,2},{6,8,2,1}};
        double B[][]={{3,5},{9,4},{3,2},{2,7}};
        double C[][] = new double[A.length][B[0].length];

        MultiplyMatrixWithMatrix(A,B,C);
        OutputMatrix(C);
    } // end of main
}

```