

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.) Bestimmen Sie, ob die Aussage wahr oder falsch ist. Ist die Aussage falsch, begründen Sie, warum!

- (a.) Um ein bestimmtes Element eines Arrays zu bestimmen, muss man den Namen des Arrays und den Inhalt des Elementes angeben.
- (b.) Die Deklaration von dem Array reserviert Speicherplatz für das Array
- (c.) Um ein Array mit 100 Elementen festzulegen, schreibt der Programmierer $p[100]$!
- (d.) Eine Anwendung, welche ein 15-elementiges Array auf 0 initialisiert, muss eine for-Wiederholungsanweisung enthalten.
- (e.) Ein Programm, welches die Elemente von einem zweidimensionalen Array aufsummiert, muss eine verschachtelte for-Wiederholungsanweisung enthalten.

(3.) Schreiben Sie Java-Anweisungen(kein volles Programm), welche folgendes umsetzen:

- (a.) Der Wert von dem siebzehnten Element eines String-Arrays f wird ausgegeben.
- (b.) Jedes der fünf Elemente des eindimensionalen Arrays g wird auf 8 initialisiert.
- (c.) Summieren Sie die 100 Elemente des Arrays c vom Typ `double`!
- (d.) Kopieren Sie alle 11 Elemente von dem Array a in den ersten Bereich des Arrays b , welches 34 Elemente enthält! Das Array soll vom Typ `double` sein!
- (e.) Bestimmen Sie den kleinsten und den größten Wert, den das Array w enthält, welches aus 99 Elementen besteht!

(4.) Schreiben Sie Anweisungen, welche auf einem eindimensionalen Array folgendes durchführen:

- (a.) Setzen Sie die 10 Elemente des Arrays $anzahl$ vom Typ `integer` auf den Wert 0.
- (b.) Addieren Sie zu jedem der 15 Elemente des Arrays $bonus$ vom Typ `integer` den Wert 1
- (c.) Gibt die fünf Werte von $bestwerte$ vom Typ `Integer` im Spaltenformat aus!

(5.) Definiert wurde ein 2×3 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!

(6.) Erstellen Sie ein Programm, welches 5 Zahlen einliest, und

- (a.) die Summe
- (b.) den Mittelwert
- (c.) den Maximum
- (d.) das Minimum bestimmt!



(7.) Erstellen Sie eine Funktion `void SchreibeSterne(int AnzSterne)`, welche die Anzahl von Sternen auf dem Bildschirm ausgibt! Verwenden Sie diese dazu, 5 Werte z.B. `int c[] = {5,3,6,2,8}`; visuell am Bildschirm wie folgt auszugeben:

```
*****
***
*****
**
*****
```

Die Funktion hat keinen Rückgabewert, daher wird sie mit `void` deklariert!

(8.) Schreiben Sie eine Funktion, welche den Mittelwert eines Arrays bestimmt!
Warum funktioniert folgende Funktion NICHT immer?

```
double avg( double[] c, int len )
{
    double sum = 0.0;

    for ( int i = 0; i < len; i++ )
        sum += c[ i ];

    return sum / len;
}
```

(9.) Bestimmen Sie die Summe

(a.) aller ungeraden Indizes (1, 3, 5, 7, 9)

(b.) aller ungeraden Elemente in einem Array!

Hinweis: Eine Zahl x ist ungerade, wenn $x \% 2 == 1$ ist!

(10.) Schreiben Sie eine Funktion `Reverse`, welche den Inhalt eines Arrays umkehrt, d.h. aus 2,4,6,8 wird 8,6,4,2!

Verwenden Sie folgende Deklaration:

```
public static void reverse( double[] actArr )
{
    .....
}
```

(11.) Bestimmen Sie die Fehler in folgenden Programmteilen:

(a.)

```
int a[][] = { { 1, 2 }, { 3, 4 } };
```

```
    a[ 1, 1 ] = 5;
```

(b.)

```
int b[] = new int [ 10 ];
```

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

```
    b[ i ] = 1;
```

(12.) Schreibe ein Programm, welches das Produkt von einer Serie von ganzen Zahlen bestimmt. Die Berechnung wird von einer Methode `product` mit variabler Argumentliste vorgenommen. Testen Sie Ihr Programm mit mehreren Aufrufen, insbesondere mit einer unterschiedlichen Anzahl von Argumenten.

(13.) Schreibe ein Programm, welches ein(rudimentäres!!) Buchungssystem einer Fluggesellschaft simuliert! Die Flugzeuge haben jeweils 15 Sitze in der 1. Klasse und 610 Sitze in der Economy Klasse.

Zunächst soll das Programm abfragen, ob der Benutzer in der 1. oder 2. Klasse sitzen möchte: „Bitte geben Sie 1 für 1. Klasse und 2 für 2. Klasse ein.“

Die Speicherung soll mit Hilfe von zwei Arrays geschehen, die nur `true` oder `false` enthalten können, und am Anfang nur Elemente mit `false` enthalten. Bei der Buchung werden dann die gebuchten Plätze auf `true` gesetzt, um anzuzeigen, dass der Platz nicht mehr verfügbar ist.

Wenn eine Klasse ausgebucht ist, soll das Programm fragen, ob eine Buchung in der anderen Klasse ok wäre!

Wenn nicht, dann soll die Meldung ausgegeben werden: „The next flight from NeverComeBack-Airlines leaves in 3 hours!“

(14.) Verkäufe: Verwenden Sie ein zwei-dimensionales Array, im folgendes Problem zu lösen:

Eine Firma hat vier Geschäftsleute welche 5 Produkte verkaufen. Am Ende des Tags wird eine Analyse der verkauften Produkte gemacht. Jede Analyse enthält folgendes:

- Die Nummer des Verkäufers

- Die Produktnummer

- Der Gesamtbetrag, welches dieses Produkt an diesem Tag umgesetzt hat

Jeder Verkäufer verkauft von jedem Produkt 0-5 Einheiten pro Tag. Alle Informationen sollten im Array „Verkauf“ gespeichert sein.

Rest der Beschreibung fehlt!

Original text:

(Total Sales) Use a two-dimensional array to solve the following problem: A company has four salespeople (1 to 4) who sell five different products (1 to 5). Once a day, each salesperson passes in a slip for each type of product sold. Each slip contains the following:

The salesperson number

The product number

The total dollar value of that product sold that day

Thus, each salesperson passes in between 0 and 5 sales slips per day. Assume that the information from all the slips for last month is available. Write an application that will read all this information for last month's sales and summarize the total sales by salesperson and by product. All totals should be stored in the two-dimensional array sales. After processing all the information for last month, display the results in tabular format, with each column representing a particular salesperson and each row representing a particular product. Cross-total each row to get the total sales of each product for last month. Cross-total each column to get the total sales by salesperson for last month. Your tabular output should include these cross-totals to the right of the totaled rows and to the bottom of the totaled columns.

(15.) (Fibonacci-Zahlen) Die Fibonacci-Folge 0, 1, 1, 2, 3, 5, 8, 13, 21, ... beginnt mit 0 und 1 und das nächste Folgenglied berechnet sich aus der Summe der beiden Vorgänger der Folge.

(a.) Schreiben Sie eine Methode `private static int Fibonacci(n)`, welche das n-te Folgenglied dieser Serie berechnet!

(b.) Bestimmen Sie die größte Fibonacci-Zahl, die ihr System berechnen kann!

(c.) Modifizieren Sie die Funktion aus (a.), in dem Sie den Typ des Rückgabewertes von `int` nach `double` ändern und wiederholen Sie die Prüfung aus (b.)!

(16.) (Fibonacci-Zahlen) Die Fibonacci-Folge 0, 1, 1, 2, 3, 5, 8, 13, 21, ... beginnt mit 0 und 1 und das nächste Folgenglied berechnet sich aus der Summe der beiden Vorgänger der Folge.

(a.) Schreiben Sie eine Methode `private static int Fibonacci(n)`, welche das n-te Folgenglied dieser Serie

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.) Nein, den Namen z.B. p und den Index z.B. 2 also $p[2]$!

(b.) Richtig!

(c.) Nein: $p[99]$, da es hier $p[0]$ bis $p[99]$ Elemente gibt!

(d.) Nein: $p[] = \{0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0\}$

(e.) Richtig! Ohne Wiederholungsanweisung geht da nichts!

(3.)(a.) `System.out.println("Ausgabe:"+f[16]);`

(b.) `int g[] = { 8, 8, 8, 8, 8 };`

(c.)

```
double Summe;
double c[] =new double[100];
for (int i=0;i<c.length;i++) {
    c[i]=i;
}
Summe=0;
for (int i=0;i<c.length;i++) {
    Summe+=c[i];
}
System.out.println("Summe"+ Summe);
```

(d.)

```
double a[] =new double[11];
double b[] =new double[34];
for (int i=0;i<10;i++) {
    b[i]=a[i];
}
}
```

(e.)

```
double Min, Max,Summe;
Summe=0;
Random r = new Random();
double values[] =new double[98];
for (int i=0;i<values.length;i++) {
    values[i]=r.nextDouble();
}
Min=values[0];
Max=values[0];
for (int i=1;i<values.length;i++) {
    if (values[i]>Max)
        Max=values[i];
    if (values[i]<Min)
        Min=values[i];
}
System.out.println("Max:"+ Max);
System.out.println("Min:"+ Min);
```

Als allererste Zeile vor `public class` „`import java.util.Random;`“ eingeben!

```
(4.)(a.) int Anzahl[] = { 0,0,0,0,0,0,0,0,0 };
(b.)
    int Bonus [] = { 0,1,2,3,4,5,6,7,8,9,10,11,12,13,14 };
    for (int i=0;i<Bonus.length;i++) {
        Bonus[i]=Bonus[i]+1;
    }
(c.) int BesteWerte [] = { 0,1,2,3,4,5,6,7,8,9,10,11,12,13,14 };
    for (int i=0;i<BesteWerte.length;i++) {
        System.out.println("Werte von Bonus:" + BesteWerte[i]);
    }
```

```
(5.) (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);
```

(6.)(a.)

```
public class Beispiel {

    public static void main(String[] args) {
        int c[]={3,3,6,2,8};

        int summe=0;
        for ( int i = 0; i < c.length; i++ )
            summe+=c[i];

        System.out.println( "Summe ist: "+summe);
    }
}
```

(b.)

```
public class Beispiel {

    public static void main(String[] args) {
        int c[]={3,3,6,2,8};

        int summe=0;
        for ( int i = 0; i < c.length; i++ )
            summe+=c[i];

        System.out.println( "Mittelwert ist: "+(double)summe/c.length);
    }
}
```

(c.)

```
public class Beispiel {

    public static void main(String[] args) {
        int c[]={3,3,6,2,8};

        int min=c[0];
        for ( int i = 1; i < c.length; i++ )
            if (c[i]<min)
                min=c[i];

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

(d.)

```
public class Beispiel {

    public static void main(String[] args) {
        int c[]={3,3,6,2,8};

        int max=c[0];
        for ( int i = 1; i < c.length; i++ )
            if (c[i]>max)
```

```

    max=c[i];

    System.out.println( "Maximum ist: "+max);
}
}
(7.)
public class Beispiel {

    public static void SchreibeSterne(int AnzSterne) {
        for ( int i = 1; i <=AnzSterne; i++ )
            System.out.print( "*" );
        System.out.println();
    }

    public static void main(String[] args) {
        int c[] = {5,3,6,2,8};

        for ( int i = 0; i < c.length; i++ )
            SchreibeSterne(c[i]);

    }
}

(8.)
public class MwArray
{
    public static double mwArray( double[] c )
    {
        double sum = 0.0;

        for ( int i = 0; i < c.length; i++ )
            sum += c[ i ];
        return sum / c.length;
    }

    public static void main( String args[] )
    {
        double c[] = { 2, 7, 5, 4 ,3};
        System.out.print ( "Mittelwert: " + mwArray(c) );
    } // end main

} // ende von MwArray

```

(9.)

(a.)

```

public class SumArray
{
    // main creates c and calls modifyc and modifyElement
    public static void main( String args[] )
    {
        int c[] = { 3, 6, 5, 4 ,11};

        System.out.println( "Die neuen Werte sind:" );
        System.out.print ( "Summe: "+ SumArray(c) );
    } // end main

    public static int SumArray( int[] actArr )
    { int sum;
      int i;
      sum=0;
      for ( i=0;i<actArr.length;i++)
          if (i%2==1)
              sum+=actArr[i];
      return sum;
    }

} // ende von SumArray

```

(b.)

```

public class SumArray
{
    // main creates c and calls modifyc and modifyElement
    public static void main( String args[] )
    {
        int c[] = { 2, 7, 5, 4 ,3};

        System.out.println( "Die neuen Werte sind:" );
        System.out.print ( "Summe: "+ SumArray(c) );
    } // end main

    public static int SumArray( int[] actArr )
    { int sum;
      int i;
      sum=0;
      for ( i=0;i<actArr.length;i++)
          if (actArr[i]%2==1)
              sum+=actArr[i];
      return sum;
    }

} // ende von SumArray

```


(10.)

```

public class ReverseArray
{
    // main creates c and calls modifyc and modifyElement
    public static void main( String args[] )
    {
        int c[] = { 1, 2, 3, 4, 5 };
        // output original c elements
        for (int i=0;i<c.length;i++)
            System.out.print( " "+ c[i]);

        ReverseArray(c); // pass c als Referenz
        System.out.println( "Die neuen Werte sind:" );
        // Ausgabe
        for ( int i=0;i<c.length;i++)
            System.out.print ( " "+ c[i] );
    } // end main

    public static void ReverseArray( int[] actArr )
    { int tmpArr[]=new int[actArr.length];
      int i;
      int numOfEl=actArr.length;
      for ( i=0;i<numOfEl;i++)
          tmpArr[i] =actArr[numOfEl-i-1];
      for ( i=0;i<numOfEl;i++)
          actArr[i] =tmpArr[i];
    }

} // ende von ReverseArray

```

(11.) (a.) Statt `a[1, 1]` muss es `a[1][1]` heißen!(b.) Statt `„i <= b.length“` richtig `“i < b.length“`!

(12.)

```

public class Test
{
    public static double BerechneProdukt(double ... EinigeZahlen){
        double Produkt=1;
        for (double ArrWert:EinigeZahlen)
            Produkt*=ArrWert;
        return Produkt;
    }

    public static void main( String args[] )
    {
        double Zahlen[]={1,2,3,4,5};
        double produkt=BerechneProdukt(Zahlen);
        System.out.println("Produkt:"+produkt);
    }
}

```

```

(13.)
import javax.swing.JOptionPane;
public class Flugreservierung
{
    private static void InitSeats(boolean ... Seats){
        for ( int i=0;i<=Seats.length;i++)
            Seats[i]=false;
    }

    private static int BookOneSeat(boolean Seats[], int NumberofFreeSeats, int Number2Book){
        Seats[Number2Book]=true;
        System.out.println("The next flight from NeverComeBack-Airlines leaves in 3 hours!");
        return NumberofFreeSeats-1;
    }

    public static void main( String args[] )
    {

        int n = JOptionPane.showConfirmDialog(null,"Die erste Klasse ist ausgebucht! Möchten Sie in der Economy
Class buchen?","",JOptionPane.YES_NO_OPTION);
        System.out.println(n);

        // Initialisierung
        boolean FirstClass[] =new boolean[ 14 ];
        boolean EconClass[] =new boolean[ 610 ];
        int FirstClassFree=15;
        int EconClassFree=610;
        InitSeats(FirstClass);
        InitSeats(EconClass);
        //int n=1;
        while (FirstClassFree>0 && EconClassFree>0){
            //String tmp=JOptionPane.showInputDialog (null, "Bitte geben Sie 1 für 1. Klasse und 2 für 2. Klasse ein,
Abbruch beendet das Programm:");
            //if (tmp==null) {
            // System.out.println("Falsche Auswahl! Programm beendet!");
            // return;
            //}
            //int n=Integer.parseInt(tmp);
            //if (n!=1 && n!=2){
            // System.out.println("Falsche Auswahl! Programm beendet!");
            // return;
            //}
            if (n==1) {
                if (FirstClassFree>0)
                    BookOneSeat(FirstClass, FirstClassFree, 2);
                else{
                    String tmp=JOptionPane.showInputDialog (null, "Die erste Klasse ist ausgebucht! Möchten Sie in der
Economy Class buchen(J/N)?");
                    if (tmp=="J")
                        BookOneSeat(FirstClass, FirstClassFree, 2);
                }
            }
            else
                if (EconClassFree>0)
                    BookOneSeat(EconClass, EconClassFree, 2);
                else{
                    String tmp=JOptionPane.showInputDialog (null, "Die Economyklasse ist ausgebucht! Möchten Sie in
der Economy Class buchen(J/N)?");
                    if (tmp=="J")
                        BookOneSeat(FirstClass, FirstClassFree, 2);
                }
            }
    }
}

```

```
}  
}  
}  
}
```

(15.)

```
import javax.swing.JOptionPane;
public class Test
{
    private static long Fibonacci(int n){
        long maxZahlen=3000;
        long FibonacciZahlen[] =new long[ n+1 ];
        FibonacciZahlen[1]=0;
        FibonacciZahlen[2]=1;
        for (int i=3;i<=n;i++)
            FibonacciZahlen[i]=FibonacciZahlen[i-1]+FibonacciZahlen[i-2];
        return FibonacciZahlen[n];
    }
    public static void main( String args[] )
    {
        String tmp=JOptionPane.showInputDialog (null, "Geben Sie den Index der Fibonaccizahl ein.");
        int n=Integer.parseInt(tmp);
        System.out.println("Die "+n+"-te Fibonaccizahl ist:"+Fibonacci(n));
    }
}
```