



1.1. Array-Übungen

Aufgabe 1: Zufälliges Element eines Arrays ausgeben
Arbeitsform: Einzelarbeit
Zeit: 25 Minuten

10 Sportarten werden eingelesen und in einem Array gespeichert. Eine Zufallszahl zwischen 0 und 9 wird generiert. Anschliessend muss ein zufälliges Element des Arrays (entsprechend der Zufallszahl) ausgegeben werden.

Sportart eingeben: **Schlitteln**

Sportart eingeben: **Strandschnecken spucken**

Sportart eingeben: **Fussball**

Sportart eingeben: **Frauentragen**

Sportart eingeben: **Luftgitarre**

Sportart eingeben: **Sumpffussball**

Sportart eingeben: **Extrembuegeln**

Sportart eingeben: **Baumstamm Weitwurf**

Sportart eingeben: **Becherstapeln**

Sportart eingeben: **Unterwasser Eishockey**

Zufällig ausgewählt: **Luftgitarre**



Lösung:

```
import java.util.Random;
import java.util.Scanner;
public class Aufgabe1
{
    public static void main(String[] args)
    {
        String sportart[ ]=new String[10];
        int zahl;
        Scanner einlesen = new Scanner(System.in);
        Random r=new Random();
        for (int i=0;i<10;i++){
            System.out.print("Sportart eingeben: ");
            sportart[i]=einlesen.nextLine();
        }
        zahl=r.nextInt(10);
        System.out.println(
            "Zufällig ausgewählt: " + sportart[zahl]);
    }
}
```



Aufgabe 2: 10 Eingegebene Zahlen addieren
Arbeitsform: Teamarbeit
Zeit: 50 Minuten

10 Zahlen müssen eingegeben werden. Diese Zahlen liegen in einem Bereich von 1 bis 100. Falls eine falsche Zahl eingegeben wird, muss die Eingabe wiederholt werden. Diese Zahlen werden in einem Array abgelegt. Anschliessend wird die Summe der Zahlen berechnet und ausgegeben.

Tipps: Schreiben sie zuerst ein Programm, welches 10 Zahlen in einen Array schreibt und überprüfen sie, ob es funktioniert. Dann programmieren sie die Überprüfung der Zahlen 1 bis 100. Wenn alles funktioniert, bauen sie die Berechnung und Ausgabe der Summe ein.

1.Zahl zwischen 1 und 100 eingeben: **22**
2.Zahl zwischen 1 und 100 eingeben: **66**
3.Zahl zwischen 1 und 100 eingeben: **11**
4.Zahl zwischen 1 und 100 eingeben: **13**
5.Zahl zwischen 1 und 100 eingeben: **8**
6.Zahl zwischen 1 und 100 eingeben: **5**
7.Zahl zwischen 1 und 100 eingeben: **88**
8.Zahl zwischen 1 und 100 eingeben: **2**
9.Zahl zwischen 1 und 100 eingeben: **77**
10.Zahl zwischen 1 und 100 eingeben: **3**
Summe :
295



Lösung:

```
import java.util.Scanner;
public class Aufgabe2
{
    public static void main(String[] args)
    {
        Scanner einlesen = new Scanner(System.in);
        int arrZahlen[]=new int[10];
        int zahl;
        int summe=0;
        for (int i=0;i<10;i++){
            do {
                System.out.print((i+1) +
                    ".Zahl zwischen 1 und 100 eingeben: ");
                zahl=einlesen.nextInt();
            } while (zahl<1 || zahl>100);
            arrZahlen[i]=zahl;
        }
        for (int m=0;m<10;m++){
            summe = summe + arrZahlen[m];
        }
        System.out.println("Summe:");
        System.out.print(summe);
    }
}
```



Aufgabe 3: Grösste und Kleinste Zahl eines Arrays
Arbeitsform: Einzelarbeit
Zeit: 60 Minuten

In einer Klasse werden vier Methoden generiert. Eine Methode generiert einen Array mit 10 Zufallszahlen in einem Bereich von 1 – 100 (Methode erhält Grösse des Arrays als Parameter, der Rückgabewert ist ein Array). 2 Methoden ermitteln jeweils die grösste und die kleinste Zahl (Methode erhält Array mit Zahlen und die Grösse des Arrays als Parameter, die grösste oder kleinste Zahl wird zurückgegeben). Eine Methode summiert alle Zahlen des Arrays (Methode erhält Array mit Zahlen und die Grösse des Arrays als Parameter, die Summe wird zurückgegeben).

Tipps: Die Methode zu Erstellung von 10 Zufallszahlen.

```
public static int[] generiereZufallszahlen(int GArray) {  
    int ZArray[] = new int[GArray];  
    .....  
    return ZArray;  
}
```

Die Methode zur Ermittlung der kleinsten Zahl.

```
public static int MinZahl(int[] ZArray, int GArray) {  
    int Zahl = ZahlMax;  
    .....  
    return Zahl;  
}
```

Zahlenbereich zwischen 0 und 1000! 10 Zahlen werden generiert!

Die kleinste generierte Zahl ist: 246

Die grösste generierte Zahl ist: 990

Die Summe aller Zahlen betraegt: 4995



Lösung:

```
import java.util.Random;
public class Aufgabe3
{
    static final int AnzZahlen=10;
    static final int ZahlMax=1000;
    //Methode Main
    public static void main(String[] args)
    {
        int ZufallZ[ ] = new int[AnzZahlen];
        ZufallZ = generiereZufallszahlen(AnzZahlen);
        System.out.print("Zahlenbereich zwischen 0 und " +
            ZahlMax + "! ");
        System.out.println(AnzZahlen + " Zahlen werden generiert!");
        System.out.println("Die kleinste generierte Zahl ist: " +
            MinZahl(ZufallZ, AnzZahlen));
        System.out.println("Die grösste generierte Zahl ist: " +
            MaxZahl(ZufallZ, AnzZahlen));
        System.out.println("Die Summe aller Zahlen betraegt: " +
            Summe(ZufallZ, AnzZahlen));
    }
    //Zufallzahlen generieren
    //Zahlenarray und Groesse des Arrays werden übergeben
    public static int[ ] generiereZufallszahlen(int GArray) {
        Random ZZahl = new Random();
        int ZArray[ ]=new int[GArray];
        for (int i = 0; i < GArray; i++) {
            ZArray[i] = ZZahl.nextInt(100) + 1;
        }
        return ZArray;
    }
    //Groesste Zahl des Array zurückgeben
    public static int MaxZahl(int[ ] ZArray, int GArray) {
        int Zahl = 0;
        for (int i = 0; i < GArray; i++) {
            if (ZArray[i] > Zahl)
                Zahl = ZArray[i];
        }
        return Zahl;
    }
    //Kleinste Zahl des Array zurückgeben
    public static int MinZahl(int[ ] ZArray, int GArray) {
        int Zahl = ZahlMax;
        for (int i = 0; i < GArray; i++) {
            if (ZArray[i] < Zahl)
                Zahl = ZArray[i];
        }
        return Zahl;
    }
    //Summe aller Zahlen berechnen
    public static long Summe(int[ ] ZArray, int GArray) {
```



```
long SummeZ = 0;
for (int i = 0; i < GArray; i++) {
    SummeZ += ZArray[i];
}
return SummeZ;
}
```