1. Das Array

   (a) Schreiben Sie ein Programm in dem Sie ein Array (bis zum Index 10) mit seinem Index füllen. Wie viele Plätze hat das Array?

   Lösung: Aufg1a.java

   (b) Was halten Sie von folgendem Programmcode? Markieren und erklären Sie kurz alle Fehler?

Aufg1b.java

```
int array;
array = new int[5];

int secArray[];
secArray = new int [5];

int[]a1 = new int[10];
[]int a2 = new int[10];
int a3 [] = new int[10];

a3[-1] = 4;
a3[10] = 5;
a3[0] = 2;
a1 = new int[];
a1 = new int [] {1,2,3,4};
a2 = {1,2,3,4};
a2 = new {1,2,3,4};

int [] twoDimArray1 = new int [][{1,2},{3,4},{5,6}};
int [][] twoDimArray2 = new int [][]{{1,2},{3,4},{5,6}};
int [][] twoDimArray3 = new int[4][];
int [][] twoDimArray4 = new int[][];
int [][] twoDimArray5 = new int[][5];

int a [] = new int[2];
int b [] = new int[8];
int c [];
c = a;
a = b;
b = c;

int i [] = new int[55];
double [] d = new double[33];
short s [] = new short[44];
i = d;
d = i;
i = s;
```

   Lösung: Aufg1b.java

   (c) Was bedeutet nun eigentlich dieses verdammte *String args[]*? Was genau ist *args*?

Lösung: Aufg1c.java

(d) Schreiben Sie das Programm aus Aufgabe 1c des Aufgabenblatts 3 so um, dass Sie ohne switches und if/else auskommen. Verwenden Sie stattdessen ein *String[] array*!

Lösung: Aufg1d.java

2. Die for-Schleife

(a) Schreiben Sie das Programm aus Aufgabe 1a dieses Aufgabenblatts so um, dass Sie das Array nicht manuell sondern mit einer for-Schleife befüllen. Füllen Sie dabei das Array einmal von vorne (Start bei 0) und einmal von hinten (Start beim letzten Index). Was glauben Sie, welche Version schneller in der Ausführung ist, die des manuellen oder automatischen Befüllens mit einer Schleife?

Lösung: Aufg2a.java

(b) Analysieren Sie folgenden Code. Welche Schleife läuft aus dem Arraybereich heraus und verursacht somit einen Fehler? Welche Ausgabe wird generiert (sofern das Programm nicht abstürzt).

Aufg2b.java

```
public class Aufg2b {
    public static void main(String[] args){
        int[] array = new int[10];

        System.out.println("Ausgabe 1:");
        for(int i = 0; i <= 10; i++){
            array[i] = i;
            System.out.print(array[i]+ " ");
        }

        System.out.println("\nAusgabe 2:");
        for(int i = 0; i < 10; i = i + 1){
            array[i] = i;
            System.out.print(array[i]+ " ");
        }

        System.out.println("\nAusgabe 3:");
        for(int i = 0; i > array.length; i++){
            array[i] = i + 3;
            System.out.println(array[i]+ " ");
        }

        System.out.println("\nAusgabe 4:");
        for(int i = 0; i <= 10; i++){
            i = i + 3;
            array[i] = i;
            System.out.println(array[i] + " ");
        }
    }
}
```

Lösung: Aufg2b.java

(c) Schreiben Sie ein Programm, das ein zweidimensionales Array (*int[][] array = new int [10][10]*), füllt. Der Inhalt sollte sich aus den Indizes ergeben.

Lösung: Aufg2c.java

(d) Schreiben Sie ein Programm zur Berechnung der Summe der Zahlen von 1 bis n. Dabei wird n als Parameter übergeben und das Ergebnis soll ausgegeben werden. Verwenden Sie dabei eine for-Schleife.

Lösung: Aufg2d.java

(e) Schreiben Sie ein Programm zur Berechnung der Fakultät der Zahl n. Dabei wird n als Parameter übergeben. Verwenden Sie dafür eine for-Schleife. Testen Sie bis zu welcher Zahl das Ergebnis stimmt.

Lösung: Aufg2e.java

(f) Schreiben Sie ein Programm dem eine Zahl n als Parameter übergeben wird und das daraus resultierende Quadrat zeichnet (Hinweis: Sie benötigen zwei in sich geschachtelte Schleifen, sowie mehrere if-Konstrukte):

```
n = 0:
keine Ausgabe...

n = 1:
*

n = 2:
**
**

n = 3:
***
* *
***

n = 4:
****
* *
* *
****

usw.
```

Lösung: Aufg2f.java

(g) Schreiben Sie ein Programm, das mit Hilfe geschachtelter Schleifen folgende Ausgabe erzeugt:

```
1 abcdefg
12 abcdef
123 abcde
1234 abcd
12345 abc
123456 ab
1234567 a
```

Lösung: Aufg2g.java

3. Die while-Schleife

   (a) Schreiben Sie die Schleifen zur Berechnung der Summe aus 3d und der Fakultät aus Aufgabe 3e jeweils in eine while-Schleife um.

   Lösung: Aufg3a.java

   (b) Wo liegt im folgenden Programmcode das Problem und was wird passieren wenn wir das Programm starten?

Aufg3b.java

```
class Aufg3b {
    public static void main(String[] args){
        int n = 2;
        while(n%2==0){
            n = n + 2;
            System.out.println(n);
        }
    }
}
```

   Lösung: Aufg3b.java