

Aufgabe 1: (6 + 2)

Entwerfen Sie eine Methode `kopiereListe(...)`, die ein gegebenes und bereits gefülltes Array vom Typ `Ganzzahl` kopiert.

Hier der bereits existierende Quellcode:

```
package ka_arrays_gruppe_a;

import AlgoTools.IO;

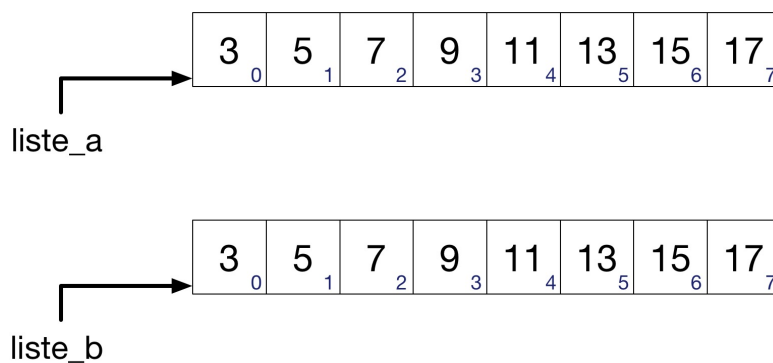
public class ListeKopieren {

    public static void main(String[] args) {

        int[] liste_a = fuelleArray(8);

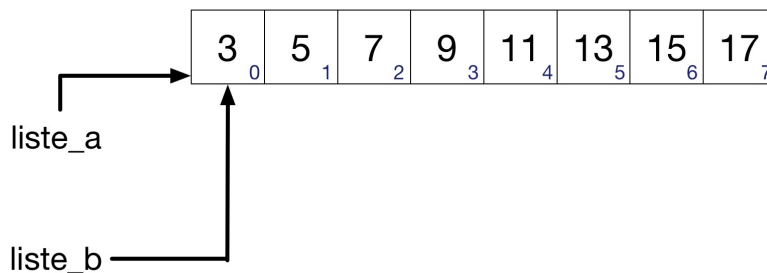
        int[] liste_b = kopiereListe(liste_a);
    }
}
```

Hier das erzeugte Speicherabbild nach Aufruf der gesuchten Methode.



Aufgabenteil b.)

Erklären Sie nun welche Veränderungen durchgeführt werden müssen damit sich das folgend dargestellte Speicherabbild ergibt.



Lösungsvorschlag Aufgabe 1:

```

public static int[] kopiereListe(int[] l) {
    int[] temp = new int[l.length];
    for (int i = 0; i < temp.length; i++) {
        temp[i] = l[i];
    }
    return temp;
}

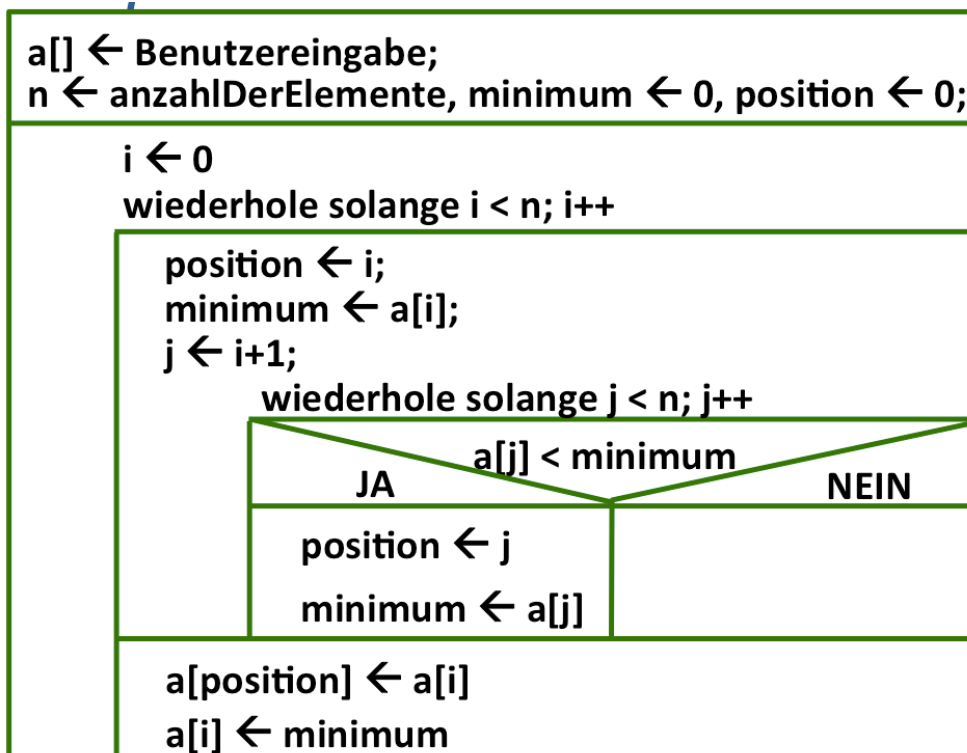
```

Aufgabenteil b.)

```
int[] liste_b = liste_a;
```

Aufgabe 2: (6 + 3)

Es gibt mehrere Möglichkeiten Werte innerhalb eines Arrays zu sortieren. Hier sehen Sie das Struktogramm des Prinzips Selection Sort.



- Implementieren Sie diesen Algorithmus anhand des gegebenen Struktogramms.
- Erklären Sie ob der Algorithmus aufsteigend oder absteigend sortiert.

Lösungsvorschlag: // Die Methoden machen es einfacher - waren aber nicht verlangt

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

    int n = IO.readInt("Wie viele Elemente ?: ");

    int[] a = einlesen(n);

    int minimum = 0, position = 0;

    for(int i=0; i<n; i++)
    {
        position = i;
        minimum = a[i];

        for(int j = i+1; j<n; j++)
        {
            if( a[j] < minimum )
            {
                position = j;
                minimum = a[j];
            }
        }

        a[position] = a[i];
        a[i] = minimum;
    }

    ausgabe(a); // Ausgabe war nicht verlangt
}

public static int[] einlesen(int n) {

    int[] liste = new int[n];

    for (int i = 0; i < liste.length; i++) {

        liste[i] = IO.readInt((i + 1) + ". Wert: ");
    }

    return liste;
}

public static void ausgabe(int[] l) {

    for (int i = 0; i < l.length; i++) {

        IO.print(l[i] + " ");
    }

    IO.println();
}
```

Aufgabe 3: (8)

Begründen Sie zu folgendem Quellcode die Ausgabe:

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

    int anzahl = 16, i = 0, j = 0, k = 0;
    int[] a = new int[anzahl / 2];
    int[] b = new int[anzahl / 2];

    while (k++ < anzahl) {

        if (k % 2 == 0) {

            a[i++] = k;

        } else {

            b[j++] = k;

        }

    }

    for (int m = 0; m < b.length; m++) {
        IO.print((a[m] + b[b.length - 1 - m]) + " ");
    }

}
```

Lösungsvorschlag:

- Die beiden Arrays besitzen jeweils eine Länge von 8 Stellen
- Das Array a erhält alle geraden, das Array b alle ungeraden Werte
- $a \leftarrow \{2, 4, 6, 8, 10, 12, 14, 16\}$
- $b \leftarrow \{1, 3, 5, 7, 9, 11, 13, 15\}$
- Jetzt wird jeweils das erste und das letzte Element aufaddiert, danach das zweite und das vorletzte

$$2 + 15, 4 + 13, 6 + 11, 8 + 9, 10 + 7, 12 + 5, 14 + 3, 16 + 1$$

Die Ausgabe lautet: 17 17 17 17 17 17 17 17

Aufgabe 4: (8)

Schreiben Sie ein Programm, das zwei Arrays vom Typ Ganzzahl jeweils einliest. Die Anzahl wird jeweils auf 4 Elemente festgelegt.

Lassen Sie eine 1 ausgeben wenn die Summe aller Werte des ersten Arrays größer ist als die Summe aller Werte des zweiten Arrays. Ist dies nicht der Fall so soll eine 2 ausgegeben werden.

Lösen Sie die Aufgabe auf dem Blatt:

Lösungsvorschlag: // Die Methoden machen es einfacher - waren aber nicht verlangt

```
package ka_arrays_gruppe_a;

import AlgoTools.IO;

public class A04_Arrayvergleich {

    public static void main(String[] args) {
        // TODO Auto-generated method stub

        int[] liste1 = einlesen(4);
        int[] liste2 = einlesen(4);

        int ausgabe = 2;

        if (getSumme(liste1) > getSumme(liste2)) {

            ausgabe = 1;
        }

        IO.println(ausgabe);
    }

    public static int[] einlesen(int n) {

        int[] liste = new int[4];

        for (int i = 0; i < liste.length; i++) {

            liste[i] = IO.readInt((i + 1) + ". Wert: ");
        }

        return liste;
    }

    public static int getSumme(int[] l) {

        int summe = 0;

        for (int i = 0; i < l.length; i++) {

            summe = summe + l[i];
        }

        return summe;
    }
}
```