Folge 4
einfache Arrays

Einführung in die Objektorientierte Programmierung (OOP)

1a) Deklariere und initialisiere einen Array `zahlen` aus 100 int-Zahlen:

1b) Deklariere und initialisiere einen Array `gewicht` aus 50 double-Zahlen:

2) Schreibe eine Methode, die alle 100 Zahlen des Arrays `zahlen` ausgibt:

```
public
{









}
```

3) Schreibe eine Methode, die alle 100 Zahlen des Arrays `zahlen` summiert und zurückgibt

```
public
{




}
```

4) Schreibe eine Methode, die alle 100 Zahlen des Arrays `zahlen` rückwärts ausgibt:

```
public
{




}
```

Einführung in die Objektorientierte Programmierung (OOP)

5) Schreibe eine Methode, die nur die durch 3 teilbaren Zahlen des Arrays `zahlen` ausgibt:

```
public
{




}
```

Einführung in die Objektorientierte Programmierung (OOP)

6) Schreibe eine Methode, die zurückgibt, wie viele der 100 Zahlen durch 3 teilbar sind:

```
public
{




}
```

Einführung in die Objektorientierte Programmierung (OOP)

7) Schreibe eine Methode, die jede der 100 Zahlen verdoppelt:

```
public
{




}
```

8) Schreibe eine Methode, die die 100 Zahlen des Arrays in 10 Zeilen mit je 10 Zahlen ausgibt:

```
public
{




}
```

9) Schreibe eine Methode, die Zahlen eines double-Arrays in 10 Zeilen mit je 10 Zahlen ausgibt. Die Zahlen sollen mit 6 Stellen ausgegeben werden, davon 2 Nachkommastellen.

```
public
{




}
```

10) Schreibe eine Methode, die den **Wert** des *kleinsten* Elementes des Arrays `zahlen` zurückgibt:

```
public
{




}
```

11) Schreibe eine Methode, die den **Index** des *größten* Elementes des Arrays `zahlen` zurückgibt:

```
public
{




}
```

12) Schreibe eine Methode, die die **Differenz** zwischen den *benachbarten* Zahlen im Array ausgibt.

```
public
{




}
```

Beispiel:

5 - 3 - 6 - 8 - 10 - 15 - 13 - 9 - 15 - 20 - 17 - 12

Ausgabe:

-2, 3, 2, 2, 5, -2, -4, 6, 5, -3, -5

12) Schreibe eine Methode, die die **Differenz** zwischen den *benachbarten* Zahlen im Array ausgibt.

```
public void ausgebenDifferenzen()
{
    for (int i = 0; i < max - 1; i++)
    {
        int diff = zahlen[i + 1] - zahlen[i];
        System.out.println
             ("Differenz an Position "
               + i + " = " + diff);
    }
}
```

Beispiel:

5 - 3 - 6 - 8 - 10 - 15 - 13 - 9 - 15 - 20 - 17 - 12

Ausgabe:

-2, 3, 2, 2, 5, -2, -4, 6, 5, -3, -5

13) Schreibe eine Methode, die das **zweitkleinste** Element des Arrays `zahlen` zurückgibt:

```
public int gibMiniPos2()
{




}
```

13) Schreibe eine Methode, die das **zweitkleinste** Element des Arrays `zahlen` zurückgibt:

```
public int gibMiniPos2()
{
    int mp = gibMiniPos(); int mini2 = zahlen[0]; int miniPos2 = 0;

    for (int i=1; i<zahlen.length; i++)
       if (zahlen[i] < mini2 && i != mp)
       {
          mini2 = zahlen[i];     miniPos2 = i;
       }

    return miniPos2;
}
```

14) Gegeben ist ein Array mit Platz für 100 int-Zahlen, der aber nur mit `anzahl` Zahlen belegt ist.
Schreibe eine Methode, die eine weitere Zahl an Position `index` in den Array einfügt, sofern noch Platz ist.

```
public void insert(int neu, int index)
{




}
```

14) Gegeben ist ein Array mit Platz für 100 int-Zahlen, der aber nur mit `anzahl` Zahlen belegt ist.
Schreibe eine Methode, die eine weitere Zahl an Position `index` in den Array einfügt, sofern noch Platz ist.

```
public void insert(int neu, int index)
{
    if (anzahl >= 100) return;  // kein Platz mehr

    if (index < 0 || index > anzahl) return;  // ungültiger Index

    // alle Elemente nach rechts verschieben
    for (int i = anzahl; i > index; i--)
       zahlen[i] = zahlen[i - 1];

    // neuen Wert einfügen
    zahlen[index] = wert;

    // Anzahl erhöhen
    anzahl++;
}
```

15) In einer Klasse sind drei int-Arrays zahlen1 (100 Elemente), zahlen2 (100) und zahlenE (200) deklariert.
Erstelle eine Methode, die die Elemente aus zahlen1 und zahlen2 *abwechselnd* nach ZahlenE überträgt.

```
public void mischeArrays()
{




}
```

15) In einer Klasse sind drei int-Arrays zahlen1 (100 Elemente), zahlen2 (100) und zahlenE (200) deklariert.
Erstelle eine Methode, die die Elemente aus zahlen1 und zahlen2 *abwechselnd* nach ZahlenE überträgt.

```
public void mischeArrays()
{
    int indexE = 0;

    for (int i = 0; i < 100; i++)
    {
        zahlenE[indexE] = zahlen1[i];
        indexE++;

        zahlenE[indexE] = zahlen2[i];
        indexE++;
    }
}
```