

## Programmierung 1 – Übung 11

1. Welchen Wert hat 'i' nachdem der Code ausgeführt wurde?

```
int[] x = {2, 1, 1, 5, 7};
int limit = 3;
int i = 0;
int sum = 0;
while ((sum < limit) && (i < x.length)){
    i++;
    sum += x[i];
}
```

- a) 0
- b) 1
- c) 2
- d) 3**
- e) 4

---

2. Ordnen Sie die Zeilen so, dass getestet wird, ob das Array aufsteigend sortiert ist.

```
return valid;
boolean valid = true;
if (iNumbers[i] > iNumbers[i+1]){
public static boolean aufgabe2(int[] iNumbers){
for (int i = 0; i < iNumbers.length-1; i++) {
valid = false;
}
}
}
```

Lösung:

```
public static boolean aufgabe2(int iNumbers[]){
    boolean valid = true;

    for (int i = 0; i < iNumbers.length-1; i++) {
        if (iNumbers[i] > iNumbers[i+1]){
            valid = false;
        }
    }
    return valid;
}
```

3. Welche Methoden testen, ob alle Elemente im Array gleich sind?

```
public static boolean aufgabe3a(int[] testArray){
    boolean ok = true;
    for (int i = 0; i < testArray.length; i++) {
        if (testArray[i] != testArray[i + 1]) ok = false;
    }
    return ok;
}
```

```
public static boolean aufgabe3b(int[] testArray){
    boolean ok = true;
    for (int i = 0; i < testArray.length - 1; i++) {
        if (testArray[i] != testArray[i + 1]) {
            ok = false;
        }else {
            ok = true;
        }
    }
    return ok;
}
```

```
public static boolean aufgabe3c(int[] testArray){
    boolean ok = true;
    for (int i = 0; i < testArray.length-1; i = i + 2) {
        if (testArray[i] != testArray[i + 1]) ok = false;
    }
    return ok;
}
```

```
public static boolean aufgabe3d(int[] testArray){
    boolean ok = true;
    for (int i = 0; i < testArray.length-1; i = i + 2) {
        if (testArray[i] != testArray[i + 1]) ok = false;
    }
    for (int j = 1; j < testArray.length-1; j = j + 2) {
        if (testArray[j] != testArray[j + 1]) ok = false;
    }
    return ok;
}
```

```

public static boolean aufgabe3e(int[] testArray){
    boolean ok = true;
    for (int i = 0; i < testArray.length-1; i++) {
        if (testArray[i] != testArray[i + 1]) ok = false;
    }
    return ok;
}

```

```

public static boolean aufgabe3f(int[] testArray){
    int count = 0;
    for (int i = 1; i < testArray.length; i++) {
        if (testArray[i] == testArray[i + 1]) count++;
    }
    if (count == testArray.length) {
        return true;
    }else {
        return false;
    }
}

```

- a) aufgabe3a
- b) aufgabe3b
- c) aufgabe3c
- d) aufgabe3d**
- e) aufgabe3e**
- f) aufgabe3f

---

4. Beschreiben Sie in einem Satz, was diese Methode macht.

```

public static int aufgabe4(int[] x){
    int z = 0;
    for (int i = 0; i < x.length; ++i)
        z = z + x[i];
    return z;
}

```

Lösung: Alle Integer im Array werden aufsummiert.

5. Was wird durch folgenden Code auf der Konsole ausgegeben?

```
public static void aufgabe5(){
    int n = 4;
    for (int row=1; row<=n; row++) {
        for (int column=1; column<=n; column++) {
            if (row==1 || row==n || column==1 || column==n)
                System.out.print ("* ");
            else
                System.out.print (" ");
        }
        System.out.println( );
    }
}
```

a)

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

b)

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

c)

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

d)

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

6. Welche Aussagen sind korrekt?

```
public static int aufgabe6(int[] data, int x){
    int z = -1;
    for (int i=0; i < data.length; i++ )
    {
        if(data[i] == x)
            z = i;
    }
    return z;
}
```

- a) Das Array 'data' wird nach dem Wert 'x' durchsucht.
  - b) Wenn 'x' nicht gefunden wurde, wird 0 zurückgegeben.
  - c) Wenn 'x' gefunden wurde, wird 1 zurückgegeben.
  - d) Wenn 'x' gefunden wurde, wird der Index des 1. Vorkommens zurückgegeben.
  - e) Wenn 'x' gefunden wurde, wird der Index des letzten Vorkommens zurückgegeben.
  - f) Nichts davon.
- 

7. Welche Werte sind im Array 'x' nachdem der Code ausgeführt wurde?

```
int[] x = {0, 1, 2, 3};
int temp;
int i = 0;
int j = x.length-1;
while (i < j) {
    temp = x[i];
    x[i] = x[j];
    x[j] = 2*temp;
    i++;
    j--;
}
```

- a) {3, 2, 2, 0}
- b) {0, 1, 2, 3}
- c) {3, 2, 1, 0}
- d) {0, 2, 4, 6}
- e) {6, 4, 2, 0}

8. Welchen Wert hat 'count' nachdem der Code ausgeführt wurde?

```
int[] x1 = {1, 3, 4};
int[] x2 = {2, 4, 5};

int i1 = x1.length-1;
int i2 = x2.length-1;

int count = 0;

while ((i1 >= 0) && (i2 >= 0))
{
    if (x1[i1] == x2[i2])
    {
        ++count;
        --i1;
        --i2;
    }
    else
        if (x1[i1] < x2[i2])
        {
            --i2;
        }
        else
        {
            --i1;
        }
}

System.out.println("count = " + count);
```

- a) 0
- b) 1**
- c) 2
- d) 3
- e) 4

9. Beschreiben Sie in einem Satz, was diese Methode macht.

```
public static int aufgabe9(int[] numbers) {
    int num = 0;
    for (int i=0; i < numbers.length; i++)
    {
        if (numbers[i] > 0)
            num = num + numbers[i];
    }
    return num;
}
```

Lösung: Die positiven Werte des Arrays werden aufsummiert.

---

10. Welchen Wert hat 'count' nachdem der Code ausgeführt wurde?

```
int [] x = {1, 2, 3, 3, 3};
boolean b[] = new boolean[x.length];
for (int i = 0; i < b.length; i++) {
    b[i] = false;
}
for (int i = 0; i < x.length; i++) {
    b[x[i]] = true;
}
int count = 0;
for (int i = 0; i < b.length; i++) {
    if (b[i] == true) count++;
}
}
```

- a) 0
- b) 1
- c) 2
- d) 3**
- e) 4