Tutorial problems – Wed 16 Oct and Fri 18 Oct 2019

5min Ask whether the students have any questions. Collect them by writing them on the board and decide with the whole group which of these to address.

10min Explain the difference between `f1` and `f2` by comparing `f1(2,1)` and `f2(2,1)`. Are there input values for which they compute the same return value?

```
public static int f1(int x, int y){
    if (x > y) {
        x = y - 1;
    } else if (x < y) {
        x = y + 3;
    }
    return x;
}
```

```
public static int f2(int x, int y){
    if (x > y) {
        x = y - 1;
    } if (x < y) {
        x = y + 3;
    }
    return x;
}
```

10min Explain how arrays can be initialized. Write a method that checks whether the first and last element in an array are the same (taken from `http://codingbat.com/java/Array-1`). You may see solutions in which students write a piece of code corresponding to

```
if (booleanExpression)
     return true;
else return false;
```

which should of course be writen as `return booleanExpression`. Discuss with the students.

25min Let a non-empty array `double[] a` be given. Write a method `public static double average(double[] a)` that computes the average of the values. (e.g., with `double[] salaries =` {`2000.0, 2000.0, 3000.0, 2000.0`} it should compute `2250.0`).

Explain how the loop (starting with position zero up to position length - 1) is gone through.

Likewise write a method `public static double average(double[][] a)` that computes the average of the values in a non-empty *two dimensional* array.

Explain how the loop (starting with zeroth position up to position length - 1) is gone through.

Careful, the method may need a counter if we want to be able to deal with arrays of non-rectangular shape such as `double[][] a =` {{`1`}, {`2,3`}, {`4,5,6`}}.

Find some code bits on the back.

```java
/**
 * This class contains the tutorial handout exercises of week 4.
 *
 * @authors Alexandros Evangelidis and Manfred Kerber.
 * @version 30-10-2018
 */
public class Wk3 {
    /**
     * This method compares x and y using an else-if statement.
     *
     * @param x an integer to be compared.
     * @param y an integer to be compared.
     * @return the value of x, accordingly.
     */
    public static int f1(int x, int y) {
        if (x > y) {
            x = y - 1;
        } else if (x < y) {
            x = y + 3;
        }
        return x;
    }
    /**
     * This method compares x and y using an if statement.
     *
     * @param x an integer to be compared.
     * @param y an integer to be compared.
     * @return the value of x, accordingly.
     */
    public static int f2(int x, int y) {
        if (x > y) {
            x = y - 1;
        }
        if (x < y) {
            x = y + 3;
        }
        return x;
    }
    /**
     * Method to print the output of f1 and f2, when called with the same integers
     * as input.
     *
     * @param x an integer to be given as input to both f1 and f2.
     * @param y an integer to be given as input to both f1 and f2.
     */
    public static void compare(int x, int y) {
        System.out.println("f1(" + x + "," + y + "): " + f1(x, y));
        System.out.println("f2(" + x + "," + y + "): " + f2(x, y));
    }
    /**
     * Method which computes the average of a one-dimensional array.
     *
     * @param a one-dimensional array of type double.
     * @return the average as a double.
     */
    public static double average(double[] a) {
        double sum = 0;
        for (int i = 0; i < a.length; i++) {
            sum += a[i];
        }
        return sum / a.length;
    }
    /**
     * Method which checks whether the first and last elements are equal.
     *
     * @param a an array of type int.
     * @return true or false depending on whether the first and last elements are
     *         equal.
     */
    public static boolean sameFirstLast(int[] a) {
```

```
        return a[0] == a[a.length - 1];
    }
    /**
     * Method which computes the average of a two-dimensional array.
     *
     * @param a a two-dimensional array of type double.
     * @return the average as a double.
     */
    public static double average(double[][] a) {
        double sum = 0;
        double count = 0;
        for (int i = 0; i < a.length; i++) {
            for (int j = 0; j < a[i].length; j++) {
                sum += a[i][j];
                count++;
            }
        }
        return sum / count;
        // alternatively for a rectangular shape (counter not necessary):
        // return sum/a.length/a[0].length;
    }
    /**
     * main method to test the class.
     */
    public static void main(String[] args) {
        compare(2, 1);
        compare(1, 1);
        compare(1, 2);
        int[] s1 = { 1, 2, 3, 4, 3, 2, 1 };
        System.out.println(sameFirstLast(s1));
        int[] s2 = { 1, 2, 3, 4, 5, 6, 7 };
        System.out.println(sameFirstLast(s2));
        double[] a = { 2000.0, 2000.0, 3000.0, 2000.0 };
        System.out.println(average(a));
        double[][] b = { { 1 }, { 2, 3 }, { 4, 5, 6 } };
        System.out.println(average(b));
        double[][] c = { { 1, 2 }, { 3, 4 }, { 5, 6 } };
        System.out.println(average(c));
    }
}
```