Exercises module 10 – Classes

10.1 – Array helper class

1. Create a new project and create a class named **ArrayHelper** that will have the following methods:

   - **maxNumber**(…)
     A method that takes an integer array as input and returns the biggest value in the array.
   - **minNumber**(…)
     A method that takes an integer array as input and returns the smallest value in the array.
   - **sumArray**(…)
     A method that takes an integer array as input and returns the sum of all the numbers in the array.

     Verify that the method works by creating an **instance** of the **ArrayHelper** class and call the three methods.

10.2 – Lotto generator

1. Write a **LottoGenerator** class that have a method named **generateNumbers** that returns an array containing **7** random numbers from **1-35**

   Make sure the returned array only contains **distinct unique numbers**

2. Call the method from your main application and output the result to the screen, try it several times to verify that you get different numbers each time.

3. Try to refactor your code in your Lotto Generator, perhaps creating private helper methods to simplify the code.

10.3 – Bubble-sort

1. In this exercise, we will implement our own bubble-sort function to sort the following array:

```
int[] nums = new int[] {9,7,5,3,2,7,8,1,5,9 };
```

(enter your own random numbers)

2. Create a new class named **BubbleSort** and in it a method named **sortArray** that takes an array as input and returns an array.

3. In your main method, create an instance of the class and then call the **sortArray** method, passing the **nums** array to it and storing the result in a new array named **result**.

4. **After the sorting**, do print out both the unsorted **nums** array and the sorted **result** array.

   Make sure you create a copy of the array inside the sortArray method, otherwise both arrays will be sorted.

5. Visit https://en.wikipedia.org/wiki/Bubble_sort and read about the **bubble-sort** algorithm and then try to implement it in the method. You can find various ways of implementing this algorithm.

Questions and concepts to study further on your own:

- What does encapsulation mean?
- What does data-hiding mean?
- How many lines per class is too many in Java?
- Using the **this.** Keyword in classes
- Nested classes https://docs.oracle.com/javase/tutorial/java/javaOO/nested.html
- Understand classes vs objects https://softwareengineering.stackexchange.com/questions/66523/
- Sorting algorithms