Software Development I – Exercises **Winter Term 2017/2018**

Übungen zu Softwareentwicklung 1 **Assignment 4**

Name: _____ Teaching Assistant: _____

Student ID (Matr.Nr.): _____ Points (max. 24): _____

Group: ☐ G1 ☐ G2 ☐ G3 ☐ G4 ☐ G5 ☐ G6 Deadline: **Tue., November 21, 2017 22:00**

Instructor: ☐ M. Haslgrübler ☐ C. Wirth ☐ T. Forstner Editing time (hours): _____

Preferred language for comments, proposals for improvements from TA's: ☐ DE ☐ EN

Problem 1: Methods **12 points**

Create a program **Ticketing** based on TicketingOriginal, which for each switch-case then only calls a method that executes the code, which was written before within the switch case.

E.g. the Code in **TicketingOriginal** Program:

```
…
        default:
          System.out.println("Unknown Menu Function");
          break;
        }
…
```

is changed to the following in your new **Ticketing** Program:

```
…
       default:
          printUnknownFunctionHelpText();
          break;
        }
…
  private static void printUnknownFunctionHelpText() {
    System.out.println("Unknown Menu Function");
  }
…
```

Hints:

- You might need to adapt the program code to be able to define methods.
- Reuse already defined methods where it makes sense.
- You are not required to write a solution statement ("Lösungsidee") or a testplan for this exercise part.

Problem 2: Arrays **12 points**

Write a program, which initially fills an array of size specified by the user with negative values and then lets the user sparsely fill an array with positive integer values. Once the user stops the filling processes the sum and average of the array is calculated (negative values are not considered).

Example

```
Enter Size of the Array:
5
Array: [-1, -1, -1, -1, -1]
Enter Position (=<0 to exit):
1
Enter Value (>=0):
5
Array: [5, -1, -1, -1, -1]
Enter Position (=<0 to exit):
3
Enter Value (>=0):
3
Array: [5, -1, 3, -1, -1]
Enter Position (=<0 to exit):
9
Enter Position (=<0 to exit):
5
Enter Value (>=0):
7
Array: [5, -1, 3, -1, 7]
Enter Position (=<0 to exit):
-1
Sum: 15
Avg: 5.0
```

Hints:

- Take care of correctly addressing the array (position!=index) both for lower and upper bound of the array.
- Make sure that no wrong user input is inserted.
- You are allowed to use the `java.util.Arrays.toString(myArray)` function to print the array as shown in the example.:
  `System.out.println("Array: " + Arrays.toString(myArray));`

Requested material for all programming problems:

For each exercise, hand in the following:

a) Approach to solving the problem (textual representation)

b) Source code (Java classes ideally with English(!) comments); in addition to the source code Java source files have to be converted to PDF and are to be included in the ZIP file,

c) Test plan for analyzing boundary values (e.g., minimal value allowed, maximum number of input, etc.) and exceptional cases (e.g., textual input when a number is required, etc.). State the expected behavior of the program for each input and make sure there is no "undefined" behavior leading to runtime exceptions. List all your test cases in a table (test case #, description, user input, expected (return) values).

d) The output of your java program for all test cases in your test plan

Pay attention to using adequate and reasonable data types and meaningful English variable names for your implementation, check the user input carefully and print out meaningful error messages.