Software Development I – Exercises
Übungen zu Softwareentwicklung 1

Winter Term 2017/2018
Assignment 3

Name: _____   Teaching Assistant: _____

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

Group: ☐ G1 ☐ G2 ☐ G3 ☐ G4 ☐ G5 ☐ G6   Deadline: <mark>Tue., November 14, 2017 22:00</mark>

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

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

Problem 1: Range   12 points

Develop a Java program that calculates the range (largest number minus smallest number) as a simple measure of dispersion from an unsorted list of non-negative integer numbers based on user-input. The input should terminate if any negative number is entered. If no number or only one number is entered, an appropriate error message should be prompted.

Examples
```
1 2 3 -1
Smallest Number: 1
Largest Number: 3
Range: 2


2 -1
Only one number entered!

4 2 3 3 1 -3
Smallest Number: 1
Largest Number: 4
Range: 3


-1
No number entered!
```

Problem 2: NumberInformation   12 points

Write a Java program that gives the user some information about integer-numbers. The user can choose between rounding numbers to the nearest multiple of 10, counting the digits or calculating the sum of the digits. The user inputs only one number at time and if the user wants to terminate the program, this can be done by the input of any negative integer-number.

Examples

```
Enter number (an negative number for exit): 16
Type of operation (r ...round, c ... count, s ... sum): r
Nearest multiple of 10: 20


Enter number (an negative number for exit):  2
Type of operation (r ...round, c ... count, s ... sum): r
Nearest multiple of 10: 0

Enter number (an negative number for exit): 11
Type of operation (r ...round, c ... count, s ... sum): r
Nearest multiple of 10: 10

Enter number (an negative number for exit): 135
Type of operation (r ...round, c ... count, s ... sum): r
Nearest multiple of 10: 140

Enter number (an negative number for exit): 9135
Type of operation (r ...round, c ... count, s ... sum): c
Number of digits in this number: 4

Enter number (an negative number for exit): 9135
Type of operation (r ...round, c ... count, s ... sum): s
Sum of digits of this number: 18

Enter number (an negative number for exit): 9
Type of operation (r ...round, c ... count, s ... sum): s
Sum of digits of this number: 9

Enter number (an negative number for exit): -9
Good Bye!
```

For each exercise, hand in the following:

a) Approach to solving the problem (textual representation)

b) Source code (Java classes including 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.