Software Development I – Exercises **Winter Term 2012/2013**

Übungen zu Softwareentwicklung 1 **Assignment 2**

Name: _____ Teaching Assistant: _____

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

Group: _____ Deadline: Wed, Oct. 31st, 2012 12:00 noon

Instructor: _____ Editing time: _____

Question 1: Stock Value **5 + 5 = 10 points**

Write a Java program that calculates the value of a stock sale. The user will be prompted to enter the stock price, the number of shares to be sold, and the commission rate.

The program will calculate the value of the shares by multiplying the stock price by the number of shares. It will also calculate the commission (the value of the shares multiplied by the commission rate) and the net proceeds (the value of the shares minus the commission).

a) First start by implementing the ideal case (no input errors). In this case, this is what the user should see on the screen. Provide all the material requested below at the very bottom ("Requested material…")

```
Enter stock price: € 10.125
Enter number of shares: 11
Enter commission rate (as a percentage): 1.5

SUMMARY
Stock price: € 10.13 //rounded (2 places after the decimal point)
Number of shares: 11
Commission rate: 1.50

Value of share: € 111.38
Commission: € 1.67
Net proceeds: € 109.70
```

b) Next, modify your code to handle incorrect input. Provide all the material requested below at the very bottom ("Requested material…").

Here is an example of how input errors can be handled:

```
Enter stock price: € -10.1
Not a valid stock price, please re-enter.
Enter stock price: € 10.111111
Enter number of shares: .1
Not a valid int, please try again: -1
Not a valid entry for number of shares, please re-enter.
Enter number of shares: xyz
Not a valid int, please try again: 11
Enter commission rate (as a percentage): 101
Not a valid commission rate, please re-enter.
Enter commission rate (as a percentage): 33

SUMMARY
Stock price: € 10.11
Number of shares: 11
Commission rate: 33.00

Value of share: € 111.22
```

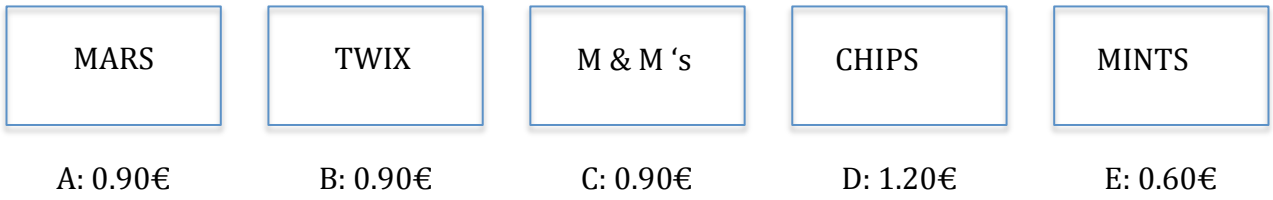
```
Commission: € 36.70
Net proceeds: € 74.52
```

Hints:

- **To format the output data correctly, you can learn how to display a float with decimal places here:**
  http://docs.oracle.com/javase/tutorial/java/data/numberformat.html
- **The Input.java class will handle some errors, but not all.**

Question 2: Vending Machine **14 points**

Write a vending machine program. The vending machine contains the 5 items below. The user will choose the item by selecting A, B, C, D, or E. The price of the item is displayed. The machine will first ask the user which item they want. It will then let them know the price of this item and expect them to input money. The user will input the money. The program will ask the user to input the remaining amount due in case not enough money was entered. Once there is enough money entered, the program will compute the change and output the candy. The change returned should be calculated as the number of each type of coins to return (i.e. 2€, 1€, 50c, 20c, 10c). Assume the vending machine only returns up to 10c coins, and not 2c or 1c. Do not worry about the incoming coins.



The user should see the following on the screen:

Example 1:

```
Please select an item (A, B, C; D, E):
A
Amount due: 0.90€
2
Total Change due: 1.10€
Change: 0 2€, 1 1€, 0 50c, 0 20c, 1 10c
Output: MARS
```

Example 2:

```
Please select an item (A, B, C; D, E):
E
Amount due: 0.60€
.5
Input:0.5
Enter another: 0.10
.1
Total Change due: 0.00€
Change: 0 2€, 0 1€, 0 50c, 0 20c, 0 10c
Output: MINTS
```

c) Write the program in Java. Provide all the material requested below at the very bottom ("Requested material…")

Advice for user input:

- **Usage of the class `Input.java` for user input (read operations) is mandatory for assignments 1-5.** Please find the class (source code) in the exercise directory on the webpage: `(http://www.pervasive.jku.at/Teaching/lvaInfo.php?key=352&do=uebungen)`
  This class provides various methods to read values of different data types. In order to read digits and numbers from the console, you may for example, use the following functions:
  - `Input.readInt()` – reads an integer from the console (data type: `int`)
  - `Input.readFloat()` – reads a floating point number (data type: `float`)

  Example:

```
- int x;
- float y;
- System.out.println("x = ");
- x = Input.readInt ();  // read integer x
- System.out.println("y = ");
- y = Input.readFloat();  // read floating point number y
```

Requested material for all programming problems:

- **For each exercise, hand in the following:**
  a) **The idea for your solution written in text form**
  b) **Source code (Java classes including English(!) comments)**
  c) **Test plan for analyzing boundary values (e.g., minimal temperature 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.