Prof. Dr.-Ing. Ralf Schenkel
Tobias Zeimetz
Trier University

Wintersemester 21/22

Exercises for the Class

Elements of Computer Science: Programming

Assignment 13

Submission of solutions until 3:00 p.m. at 09.02.2022
at `moodle.uni-trier.de`

- Every task needs to be edited in a meaningful way!
- Please comment your solutions, so that we can easy understand your ideas!
- If you have questions about programming or the homeworks, just ask you teachers!
- **Submission that can't be compiled are rated with 0 points!**

Exercise 1 Reading and processing files

Implement a class `Product`:

1. Read the given file `Productlist.txt` in a static method of your solution (e.g., but not necessarily, in the `main` method). The imported data contains information about products using key-value pairs. Each line identifies its own product. The attributes of a product are `artno`, `name` and `price` The order of the attributes in the file is not fixed, also a `artno` of a product can occur several times.

   The instances of the class `Product` should store the data of a single product in three corresponding instance variables (i.e. in two `String` (`artno`, `name`) and one **`int`** (`price`) variables). Write a constructor for `Product` that appropriately splits a read line (i.e. a string) into these three parts. `artno` and `name` should be stored without the enclosing quotation marks. `StringTokenizers` are not suitable for the decomposition, because the name of a product may contain many different characters (including ',' and ':') (but for simplification it does not contain any further quotation marks).

   Save the read data in a `ArrayList<Product>` file.

2. Write the data into a file `ModProductlist.txt` in a form comparable to the original file, but without additional spaces (except in names and article numbers) and in the fixed order `artno`, `name` and `price` for each product.

3. Write a corresponding file `CheapProducts.txt` with all products that cost less than 20 (Euros).

4. Write a corresponding file `CheapProducts.txt` with all products that cost less than 20 (euros). Calculate the average price of all products and enter it (rounded down to an integer) in the form ‘`Average Price: ... Euro`’ on the console. You can also find the lowest and the highest price and additionally display

   ‘`Price Range: between ... Euro and ... Euro`’

Watch out: Unfortunately, you cannot view the output files (.txt) created with your program. This means that you can only test whether they are correct for reading/writing using the print function. Therefore it is better to compile the tasks on your local system and test it there extensively. The only important thing is that your Java files and the text files are in the same folder.