Object-Oriented Programming

Exercise series 8

Exercise 1

Open the `FormatChecker.java` file provided along with this document and have a look at it. In particular, have a look at the documentation of the classes from the Java library used by this program. Then, answer the following questions.

— Are the exception classes used in this file checked or unchecked ?

— Why is `reader` declared before the first `try` in `extractContent()` ?

— Why doesn't `extractContent()` throw `FileNotFoundException` ?

— Could you use a single `catch` block in the `main()` method ? If yes, which one ?

— What would happen if `extractContent()` had no `throws` in its signature ?

Tip : compile this program and run it with a correct file name or with an incorrect file name. This file name must be given as a command line argument (i.e., you will run `FormatChecker` by running a command such as `java FormatChecker myFile.txt`). If you are using Eclipse IDE, you can provide this argument by going in the *Run* menu and selecting *Run Configurations...*. In the new window, select the *Arguments* tab and write the name of a file in the *Program arguments* text area before running the program.

Exercise 2

Create a `Credits` class that stores a course code as a string, an amount of (ECTS) credits and a grade and which implements the following operations.

— It must be possible to get the amount of credits and the grade. It must also be possible to get the result for the course as the grade weighed by the amount of credits (e.g., if there are 5 credits while the grade is 12/20, the weighed result is 3).

— It must be possible to instantiate a new `Credits` object by providing a string formatted as follows : `course,amount,grade`. `course` is the code of the course. Both `amount` and `grade` must be integers, with `amount` being strictly positive and `grade` ranging from 0 to 20 (included). For instance, `INFO0062,5,14` is a valid string for instantiating a new `Credits` object. An incorrect format must be signaled by throwing an exception.

You are free to create your own exception classes. Consider also the following tips.

— Use the `split()` method from the `String` class to chunk a line on the basis of a separator string. The provided `FormatChecker.java` file shows you how you can use this method.

— Use the class method `Integer.parseInt()` to convert a `String` object into an integer. Check the exception(s) it can throw by consulting the documentation of this method.

Exercise 3

Complete the `main()` method in `FormatChecker.java` to put the `Credits` class to practice. Use the code already provided in this file to extract the content of a file and read it line by line.

Then, check if each line is fit to be turned into a `Credits` object. For incorrectly formatted lines, display an error message with the cause and the number of the line. For correctly formatted lines, compute the sum of the weighed results and the total of ECTS credits and display both.

Test your solution by applying your final program on a file mixing together correctly formatted and incorrectly formatted lines.