Software Engineering Methods 2014

Instructor: Dr. Alberto Bacchelli — Assignment 2

Teaching Assistants: Friso Abcouwer, Rob van Bekkum, Moritz Beller, Thijs Boumans, Aimee Ferouge, Jan Giesenberg, Michael de Jong — Week 04

The goal of this assignment is to apply Design Patterns. To correctly complete this assignment you **must**:

- Carry out the assignment with your team only–unless otherwise stated. You are free to discuss solutions with other teams, but each team should come up its own personal solution. A strict plagiarism policy is going to be applied on all the artifacts submitted for evaluation.
- Your team has to complete the assignment by following the SCRUM methodology:
  - You have to submit a sprint plan for this assignment, using the template "Sprint Plan Template" available on Blackboard, by **Sep 23, 2014 @ 13:00**.
  - You have to submit a sprint reflection for this assignment, using the template "Sprint Reflection Template" available on Blackboard, by **Sep 30, 2014 @ 13:00**.
- Solutions to this assignment will consist (depending on the exercise) in changes to the source code of your project and in explanations (*e.g.*, of decisions taken):
  - All the explanations must be included in a single PDF file with the name: *Group[devhub id]-[AssignmentNumber].pdf* (a correct name would be: *Group1-2.pdf*).
  - Changes and explanations must be pushed to the master branch of your Devhub repository by **Sep 27, 2014 @ 23:55**.[1]

Exercise 1 - Simple logging (30 pts)

1. Extend your implementation of the game to support logging. The game has to log all the actions happened during the game (*e.g.*, player moved Tetris piece from position X to position Y). Define your requirements and get them approved by your teaching assistant. The implementation and process will be based on the same criteria used for the working version, plus it will take into account whether you use design patterns and advanced object-oriented programming (**20 pts**).
2. During the analysis and design phases of this extension use responsibility driven design and UML (push to the repository a *single* PDF file including all the documents produced) (**10 pts**).

Exercise 2 - Design patterns (45 pts)

Choose **three** design patterns among those that we saw in class.[2] For **each** chosen design pattern, you must have a corresponding implementation in your code (*excluding code written for logging in Exercise 1*). If not, refactor your code to include it. Then, per each chosen design pattern, complete the following points:

1. Write a natural language description of *why* and how the pattern is implemented in your code (**5 pts**).
2. Make a class diagram of how the pattern is structured statically in your code (**5 pts**).
3. Make a sequence diagram of how the pattern works dynamically in your code (**5 pts**).

Exercise 3 - *Optional*: One more design pattern (15 pts)

Repeat exercise 2 for a design pattern (excluding Singleton) that you do *not* already had implemented in your code (*excluding code written for logging in Exercise 1*).

[1] Solutions sent within the first 24 hours after the deadline will be given 50% of the points they would normally get. Solutions sent after 24 hours from the deadline will not be graded.

[2] *i.e.*, strategy, observer, decorator, factory method, abstract factory, singleton, adapter, iterator, composite