Software Architectures

Assignment 1: Design Patterns

Assistants: Kennedy Kambona, Humberto Rodriguez Avila
Email: {kkambona, rhumbert}@vub.ac.be
Office: {10F732, 10F730}

Deadline: 3^{rd} March 2016, 23:59

In this assignment, you will explore the source code of **assignment1-DP-Demos** project to examine its use of design patterns and to study its coupling and cohesion. This first assignment consists of three exercises, in which you will need to examine the source code of mentioned project to find instances of a number of given design patterns, to recognize design patterns in the code, and lastly to study the coupling and cohesion of some components of the code.

Assignment

For this assignment you will write a report, answering the three exercises described in this document.

Deadline 3^{rd} March 2016 at 23:59. The deadline is fixed and will not be extended.

Deliverables Write a report (in English) answering the questions given in this document. The report should be handed in as a single PDF file.
The file should follow the naming schema `(Firstname-Lastname_)*1.pdf`, for example: `Kennedy-Kambona_Humberto-Rodriguez_1.pdf`.
Submit it on the Software Architectures course page[1] in PointCarré, by clicking on *Assignments (Opdrachten) > Assignment 1.*

Team work You are allowed to work alone or in a team of two. Only one of you should submit the report on PointCarré, but be sure to mention both names in the report! Note that copying – whether from previous years, from other teams, or from the internet – will not be tolerated, and can lead to a zero for the complete course. If you use any other resources besides those provided in the lectures and in this document, remember to cite them in your report.

[1] Use the English variant "Software Architecture**s**", rather than the Dutch one "Software Architecture**n**".

Grading The exercises will be graded and can become subject of an additional defense.

Exercise 1: Find Instances of Design Patterns

First, set up Eclipse to view the *assignment1-DP-Demos* source code, using the instructions in the last section of this document: *Preparing Eclipse*.

In the first exercise, you should find instances of a number of design patterns used in the *assignment1-DP-Demos* source code. For this exercise you will analyse all the demos of package *exercise1* (exercise1.demo1, ..., exercise1.demo15), and you have to identify any **3** of the following design patterns:

1. Adapter
2. Factory Method
3. Observer
4. Visitor
5. Facade

For each instance:

- Say whether this is a creational, structural, or behavioral pattern.
- Describe the participants: give their class or method name.
- Create a class diagram showing the involved participants. Only include elements necessary for the design pattern.
- Describe the motivation and application of the pattern in its concrete situation in 50-100 words. Do not give a general description of the design pattern.

Note: In practice, instances of these patterns might slightly deviate from the situation described in the class. In that case, describe where the instance deviates from the original pattern, and (if possible) why. Be aware that real instances of these patterns do not need to follow the exact naming conventions of the patterns, and a demo can contain multiple patterns.

Exercise 2: Recognize Design Patterns

For this second exercise you will analyse the package *exercise2.demo16*. This demo implements one design pattern. For this pattern:

- Say which design pattern you found, and whether this is a creational, structural, or behavioral pattern. A list of patterns can be found in the lecture slides.

- Describe the participants.
- Describe the motivation and application of the pattern in its concrete situation in 50-100 words. Do not give a general description of the design pattern.

Exercise 3: Coupling and Cohesion

(a) First, answer the following questions:

- Which is preferable: high or low cohesion? Why?
- Which is preferable: tight or loose coupling? Why?

(b) Analyse the source code of **exercise3.demo17** and answer the following questions:

- Which type of **cohesion** do you find in this class? The types of cohesion (coincidental, logical, etc.) are described in the lecture slides.
- Is this high or low cohesion, and why? If it is a bad type of cohesion, how could it be improved in this case?
- Which type of **coupling** do you find between these classes? The types of coupling (content, common, etc.) are listed in the lecture slides.
- Is this tight or loose coupling? If it is a bad type of coupling, how could it be improved in this case?

Note: If is necessary attach the source code with your improves to this exercise.

Preparing Eclipse

The assignment uses **Eclipse for Java EE Developers**.

What follows is a short step by step description on how to setup Eclipse for your platform and open the given source code of `assignment1-DP-Demos`. You are not required to use Eclipse, you can use any editor (Atom[2], Sublime Text[3], etc.) to find the design patterns in the given code.

1. Download "Eclipse for **Java EE Developers**" from `http://www.eclipse.org/downloads/`.
2. Extract the downloaded file to a path of your choice. We recommend creating an empty workspace for these assignments. Before starting Eclipse, make sure you have installed a Java Development Kit (JDK). See this page[4] for more information.

[2] `https://atom.io/`
[3] `https://www.sublimetext.com/`
[4] `https://docs.oracle.com/cd/E19182-01/820-7851/inst_cli_jdk_javahome_t/`

3. Download `assignment1-DP-Demos.zip` from PointCarré.
4. In Eclipse, choose File > Import...
5. Next, choose General > Existing Projects into Workspace (**not** Archive File).
6. In the next step, choose "Select archive file" and select the downloaded `assignment1-DP-Demos.zip`. Press Finish.