Assignment 7 - Design Patterns

Version: April 19, 2024

NO LATE SUBMISSIONS AT ALL (neither late pass nor with 10% deduction.)

General Info

Since I honestly believe that the best way to learn Design Patterns is to research and practice them, I want to give you exercises here that I believe will help you learn more about Design Patterns and how to apply them.

There is not enough time to learn about all Design Patterns, but I hope this will help you understand them better.

For all tasks, you will need a document in which you can paste images and write your answers. More information follows in the specific tasks.

Objectives

- Understand how and when to apply a Design Pattern
- Apply a Design Pattern in your project (individual task)

Task 1: Research Design Patterns (5 points extra credit)

This task requires you to research design patterns on your own. You can, of course, start with the other tasks and then mention the resources you used. You will basically need this to do the other tasks.

To complete this task, you must:

- Join the public #DesignPattern channel on Slack (1 point)
- Add a new good resource or post a meaningful comment on a resource you found helpful (1 point each, up to 3 points).
- In your submission document, under a Task 1 section, list links for up to 3 of your new resource posts and meaningful comments. Use the 'copy link' feature in Slack to link directly to your Slack message. (1 point).

Task 2: Understand Design Patterns and their Application (30 points)

On Canvas, you find 3 different projects (all very simple projects). All projects have a little Main, hopefully helping you understand what each one does. None of these projects use Design Patterns yet. This task basically requires you to research design patterns and understand them better.

Add a section "Task 2" to your document for this task.

For your convenience I am giving you a list of Design Patterns from the Gang of Four you should look at (you will need some of them, not all)

- Factory
- Builder
- Composite
- Adapter
- Decorator
- Strategy
- Observer
- Facade

Your task is for 2 out of the 3 projects (you can choose which ones):

1. understand what the code does, run it
2. create a UML Class diagram for the original code (level of abstraction is up to you; choose one that makes sense) - export to document - 3 points
3. research for a Design Pattern which you can include into this project to **improve** the project
4. make the necessary changes to the code to incorporate this Design Pattern into the code
5. test that the code works include your tests - running it in Main is fine. You can also include Unit Tests if you are fancy - you will need to submit this code
6. create a UML diagram for the new code - export to document
7. Explain in your own words briefly why you chose that Design Pattern (which one was it), why you believe this improved the code, and what you learned by applying this pattern. - export to document

You must submit two zip files named <DesignPatternName>_asurite.zip (e.g., Singleton_amehlhas.zip - it is not the Singleton pattern, by the way). You will still add more details to the PDF in Task 3.

Task 3: Apply two Design Patterns to a System from the Course (40 points)

In this task, you will create **two diagrams and two implementations** that demonstrates a Go4 Design Pattern of your choice. The diagram will have to match the implementations. The diagram/implementation pairs can be independent of each other, e.g., one implementation shows the Builder pattern with a corresponding diagram, and the other implementation shows the Decorator Pattern with the corresponding diagram.

You may not choose the Singleton Pattern.

The Design Pattern should fit into the content of your Challenge system, the Tutoring System, OR the Shop System from the lectures. Do not use the same Design Pattern twice; you can still reuse the same system, though.

You are allowed to:

- enhance the requirements
- only use a subset of requirements

For each Design Pattern (and thus implementation and diagram):

1. write a **description** in your document which must include:
   - The name of your Design Pattern with a description of the Design Pattern. (in your own words, do not copy/paste from websites)
   - A diagram (drawn in any UML tool you like) showing your Design Pattern for your example (specifically for the system). You choose the kind of diagram, and by now, you should know best what helps you show the Design Pattern.
   - An explanation about your design pattern's application (how it's being used in your example scenario and why it is good to use in this context). How does it improve the general design?
   - If you enhanced the requirements, explain how and why.
2. Create a simple implementation, which matches your diagrams and example. The implementation should have a short Main running your program and showing that it works and does what it is supposed to do (eg. only allow one instance of a class to be created for a Singleton). This can be a simple implementation only containing the parts and functionality necessary to show the advantage of the Design Pattern, but it has to do something that makes sense, and it has to show the Design Pattern! So you might need to build different objects with your builder class, which will then be used for some purpose.

IMPORTANT: Ensure that you include all the necessary attributes and methods required in order to make the design pattern functionally sound based on its **description**. The code should do something that makes sense for the system you chose and should not just be a Design Pattern skeleton without real functionality. Also, make sure that these attributes and methods in your diagrams and code match exactly.

Submission

Submit the following on Canvas - 1 PDF and 3 zip files (please check the individual tasks for details):

- One document containing all your answers and explanations (please see the tasks to know exactly what is needed.)
- Task 1: Up to 3 resources used in your research added to your document.
- Task 2: You must submit two zip files named <DesignPatternName>_asurite.zip (e.g. Singleton_amehlhas.zip - it is not the Singleton pattern, by the way). Explanations about your Design Pattern in your document.
- Task 3: One zip file design_pattern_asurite.zip containing the implementations of your Design Patterns (in two separate folders). Include a section in your document with explanations and your diagrams.