Assignment 5a/b - Design Patterns

Version: April 4, 2023

General

In this assignment, we will introduce Design Patterns while also incorporating everything you've learned so far by creating a little "game".

This document covers everything you need to do, but keep in mind that this assignment is big and is designed for a couple of weeks. THIS ASSIGNMENT WILL TAKE TIME! The A part is very simple, the B part not so much. So plan accordingly.

You should apply all you have learned so far, testing, good GitHub workflow, commits, clean repo etc. It will not be mentioned specifically it is implied.

You will need to create a PDF document (specifics below) for your Deliverable B which you will submit on Canvas.

Make sure you stick to the requirements in this assignment and include all necessary information in your PDF or in your README. We will not reward points later on if you forget to submit something. Make sure you clone your repo at the end and test if it runs to make sure that you have put all necessary files in the repo. This is good practice anyway so you are sure what you submit actually works.

This assignment is split up into 2 deliverables (a/b) to make sure you work continuously and not just in the last couple of days. See below for details on what you need to submit for each deliverable.

The specifics

1. Work on Git and GitHub for this assignment (on a private repo, add ser316asu as collaborator). This includes good commit messages and a clean repo, etc. (7 points)
2. The implementation needs to be in Java
3. Each Design Pattern is worth 15 points (45 points) :
    - You need to create 3 design patterns from the Gang of Four for the requirements mentioned in the separate document.
    - You do need to implement different Design Patterns
    - You have two choices:

    a) The minimal requirement for up to 95% of credit: Have the Design Patterns implemented independently from each other (the patterns should not "work together"). Basically, each one of them can be run separately. In this case, you would implement each pattern in a separate package. In a Main class, first show the functionality of one pattern and then the other etc. This is the easier version, since the DP are not intertwined which many struggle with.

    b) Getting full points plus some extra credit: Implement your code with the Design Patterns being intertwined (you can of course still have packages if you like), working together and forming one good project. The DP still need to be implemented correctly and make sense!

  - Make sure you comment your code well so we can see what you are doing or tried to do.
  - You do NOT need to meet all the functional requirements specified in the other document, but each Design Pattern should at least implement 3 of the requirements (not all might fit into your Design Pattern but might just be other side functionalities in your code). In your PDF, describe which requirements you fulfilled and also make sure you comment on it in your code. If you chose version 2 above, then 9 requirements need to be fulfilled overall in your code.
  - The code needs to be fully functional and needs to "do something"! There needs to be a basic simulation that shows the functionality of the requirements you chose. Just creating the skeleton of the Design Pattern with no data is NOT sufficient. I want to see a running functional implementation.
  - Make sure your Main does have good print outs so we and you can see what is going on.
  - We will grade you based on the code that you've written. This is based on all you have learned so far: Good coding practice, testing, coding standards, clean repo, etc.
  - **Minimal Requirement** for each package: There is some kind of simulation included in Main that includes some data being created and some kind of simulation doing something that makes sense. Just creating the rough outline of the code structure is not enough.
  - **Things in Main need to run fully automated, we will not have time during grading to actually "play"**

4. Create a Gradle file (5 points) - requirements (**All the below Gradle requirements are mandatory, you will not get points for this assignment if this does not work**). We will NOT import anything into an IDE!
  - Your code HAS TO execute via "gradle run" via command line
  - Include Checkstyle and Spotbugs
  - Include JUnit 4 or 5

5. Checkstyle and Spotbugs: make sure you adhere to the coding standards you set in your Checkstyle XML and that Checkstyle and SpotBugs do not show anything. If you cannot get rid of everything, you should mention this in your PDF document and explain why you were not able to fix it.

   In your PDF (or README): include a screenshot of your Spotbugs and Checkstyle report. You will not get points for this if there are no screenshots that show us your reports. (10 points)

6. GitHub Actions: Set up GitHub Actions CI so it builds your project, runs your Unit Tests and also Spotbugs and Checkstyle. (5 points)

7. Include JUnit tests and test your code. You should reach at least 70% code coverage (excluding your Main, getters and setters).

   In your PDF (or README): include a screenshot of your JUnit and Jacoco report, showing your tests pass and your code coverage. You will not get points for this if there are no screenshots that show us your reports.(15 points)

8. Include a short screencast (does not have to have audio but is appreciated) showing you doing a git clone from your repo, gradle build, showing your reports (JUnit, Spotbugs, Checkstyle, GitHub Actions CI), and gradle run. Show your system in action (max. 6min). (8 points)

9. **In your README** on GitHub (well written and easy to find everything)

   - Include a link to your screencast
   - Explain each of your Design Patterns briefly and mention which of the requirements you fulfilled with this Pattern. (5 points)

IMPORTANT: If your Code has compile errors you will receive 0 points, we will not debug anything. Make sure your Main runs without errors. Example you are only able to make 2 of the 3 Design Pattern run in your Main and the 3rd is implemented but does not work. Then you should not include number 3 in your Main. You will get credit for the the two DP and will get partial credit for number 3 (if it has no compile errors).

As stated before, you do not have to implement all the requirements and you can choose any 3 design patterns from the Gang of Four for task 1 that you see fit. You are relatively free on how to do things. Please mark where you see your Design Patterns in your code. Again, this is coding intensive. Start early, as it will take a while to figure things out.

Submission A (15 points)

For submission A I want you to have your basic setup done. Private repo, simple Main (can just be a hello world) that can be run through Gradle, GitHub Actions included (and passing), Checkstyle, Spotbugs, basic test setup with a simple test. So basically everything but the real implementation should be done.

I also want to have a short summary in the Readme.md on GitHub which explains your rough idea for the Design Patterns you plan to implement (you can still change your mind later). This can be very basic since we have not covered a lot about Design Patterns yet.

Submit your link to the private repo on Canvas (ser316asu added as collaborator). No PDF is needed at this point and no screencast.

This submission has 15 of the overall points

Clean Git repo - 3pts
Gradle works correctly - 3pts
Checkstyle and Spotbugs included - 3pts
GitHub Action setup and passes - 2pts
JUNIT included and test runs and passes - 2pts Readme has basic idea explained - 2pts

Submission B (85 point)

You need to submit your code with your Design Patterns and all that is asked above

1. **A link** to your GitHub repository (also include in the PDF)
2. **A link** to your screencast (also include in the PDF)
3. **A PDF document** explaining everything that is asked above in the general assignment. Submit the PDF directly on Canvas.