cs3307 – Object oriented analysis and design

Assignment 2: Design patterns

Introduction

In this assignment, you will do "drills" on design patterns.

The following material is particularly relevant – please READ it!!!:

- Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides
  Design Patterns: Elements of Reusable Object-Oriented Software, Addison Wesley, 1995.
- Martin Fowler. *UML Distilled*, Third Edition. Addison-Wesley, 2004.
- Other books and material that talk about design patterns.

What you need to do, including important points to note:

1) Write a *business* or *technical* scenario such that its implementation in software, in part, could use a design pattern. For your purpose, you need only be concerned about one design pattern per scenario. An example of a scenario is given below.

   A *business* scenario, for example, would typically include elements, relationships, events, conditions, etc., in the *domain* of the application (e.g., in the context of an accounting system). A *technical* scenario, on the other hand, would typically include such items in the *solution* of the application (e.g., in the design, logic, and behaviour of the accounting system).

   A scenario must have an informative title. Please avoid using acronyms in the title. An acronym in the scenario body must first be defined before use.

   Example business scenario: Rapid spawning of a look-alike system.

- In an attempt to take market lead in the aspect of "hi-tech" cars, a car manufacturer went gung-ho in digitising many functions of the high-end model of their motor car.
  - Example digital functions are: *information provision* functions (e.g., fuel consumption rate, average speed, distance to destination, inside/outside temperature, etc.); *system adjustment* functions (e.g., shock-absorber adjustment from soft to hard, internal light intensity in the front part of the vehicle versus at the rear, headlamp elevation, seat positioning, sound system operations; sunroof open/close and elevation settings); etc.

    - Access to these various functions are through a user interface of an onboard computer.
- Seeing the success of this strategy in the marketplace, management later decided to provide such features, but not all, in a lower-end model of the same make of the motor car.
- Your team is the key software design and development team in the company.
- Your task is to propose to the management a strategy for redesigning the computer system to fit the needs of the lower-end model.
    - Management directives:
        - To consider development time/effort as a prime constraint.
        - To minimise development.
        - To reuse extensively.
    - Your response is to use the "Facade" design pattern as it means you just have to create a new interface layer suited to the lower-end car model and connect the provided access points (menu items, buttons, etc.) to the functions from the previous system without modifying the old system. Users would not be able to access the previous system functions directly.

Please note that you are *not* permitted to use the given example or another similar scenario in this assignment. So as not to enter in any dispute with the instruction team, please create your scenarios to be radically different. The professor's decision is final and may not be appealed.

2) (i) Identify the name of the design pattern that you think fits in with your scenario and (ii) justify its choice.
3) Give a graphical representation of the design pattern in UML, giving domain-specific names to the pattern elements.
4) Explain how the design pattern works and give a convincing argument that this design satisfies the intent described in the scenario.
5) Write "skeleton" C++ code to implement the design pattern (including any supporting code needed to make the skeleton code executable).
6) Explain the code and show the correspondence between the code and the design pattern from (5) and (6) above.
7) Run the code and produce execution traces (e.g., with print statements) to demonstrate that the implemented pattern operates like you claimed in (6).
8) Please write, as numbered list, ***up to three*** critical lessons learnt on design patterns from this scenario.
9) **Number of scenarios to be produced:**
    a. A group of size >= 2 members needs to produce two scenarios; group of size 1 member needs to produce one scenario.
    b. For each scenario, points (1) to (8) apply.

Group Work and policy:

1) This assignment is to be done *collaboratively by the whole group*, not by a subset of members of the group.
2) **Equitable workload within a group**:
    a. It is expected that each member of the group will carry his/her weight and contribute equitably to each scenario and its implementation.
    b. If any group member feels that another group member is not carrying his/her weight in the assigned work, please email immediately to the Prof. so appropriate actions can be taken.
        i. A peer evaluation will take place without any further discussion.
        ii. Marks for each member of the group will be proportionate to the peer evaluation of the group members' contributions. Please read the course description.

Late submission:

- Please note that the policy stated in the course description will be rigorously followed.

Project requirements:

1) Points (1) to (8) each produce an appropriate deliverable. Please label each one clearly.
2) Header page of the document should contain at least the following:
    - Assignment title.
    - Course number and name
    - Group number AND membership with email addresses.
    - Date
3) Please ensure page numbering.
4) The document should be in PDF format.
5) Please note – **only one submission from each group** will be accepted. We will mark only one and is not clear which one so please do not send more than one!
6) **Submission format:** The file name *must* follow the following format:
    - <"Design Patterns" ><underscore "Group" group # underscore><hyphenated last names of group members>
        - E.g., Design Patterns_Group 45_Bloggs-Carpenter-Jones-Whitesides
7) **Deadline:** 10th November, 2014 at 11:59:59
8) Recipient email address: cs3307a@gmail.com.

Have fun!