CSC 2620 Object Oriented Design – Spring 2024
Unit 4 - Design Patterns

4.1 Design Patterns - Student Presentations

Software Design Patterns:
`https://en.wikipedia.org/wiki/Software_design_pattern`

- Both software and network systems can be modeled after "design patterns"
  - You cannot choose "Model-View Controller", it is its own unit.
  - Sign up in the Project Groups signup sheet available in the class Classroom directory.
  - First come-first served, so get your choice in early. No one can choose the same design pattern as anyone else. *Every design pattern must be chosen.*

4.1.1 Goals and Rubric (Out of 15 Points)

- Introduce the design pattern, and discuss what general family of software it's used for, if any (some of them are just too general). Also, discuss what issue it resolves or problem it solves or what shortcut/design best practice it provides. (2 points)
- Teach the class what the design pattern is. Give an example, either with code itself, or in pictures. Someone should walk away from this part of the presentation understanding how they can implement your design pattern in their own work. You must show an example. (6 points)
- Discuss pros and cons of using the design pattern. Does it have limitations? Where is it used in the real world? (3 points)
- You will also receive points for the general quality of your presentation - Your visuals (figures), code, clarity, presentation style, everyone presented equally, etc. (4 points)

Remember, you are *teaching* the topic to the students of the class. A good presentation would allow someone to hear it and begin to use the design pattern in their own work, and that's how points will be assigned.

4.2 The Model View Controller Design Pattern

This is the classic GUI *design pattern.*

Implementing in Java:
`https://www.oracle.com/technical-resources/articles/java/java-se-app-design-with-mvc.html`

An additional example:
`https://medium.com/@ssaurel/learn-to-make-a-mvc-application-with-swing-and-java-8-3cd24cf7cb10`

Example code on Classroom.

4.2.1 Practice

1. Create a new Java project.
2. Set up a Model-View-Controller pattern in your code by creating new classes for `Model`, `View`, and `Controller` (you can combine view and controller if you'd like, or designate the main class the controller, or simply create a separate controller class).
3. Start with the `Model`
   (a) Grab the file `words.txt` from Classroom. It contains thousands of words in the English language.
   (b) Load it into a data structure/collection of your choosing in your project, making sure all data manipulation is handled by the `Model` class.
4. Build a GUI (a `View`) that allows the user to view the entire list of words (probably should use a text area).
5. Add functionality to your GUI that allows the user to see all of the words that start with a particular letter or set of letters
   (a) Create a series of check boxes, one for each letter of the alphabet
   (b) When a box is checked, all letters that start with those words should be displayed.
   (c) Initialize all the boxes to be "unchecked"

4.3 Rubric

Points	Criteria
Introduction	
2	Introduction, discussion of family, and discussion is problem it solves
Teaching the Pattern	
3	Description of the design pattern (in code or diagrams)
3	Example of the design pattern
Pros and Cons	
1	Advantages of the design pattern
2	Disadvantages of the design pattern
Overall Presentation Quality	
2	Good visuals used
1	Code and text readable
1	Everyone presented equally