OO Final Project

Requirements

For the final project, you are to design and develop a Java application that you choose. The goal of the final project is for you to combine the things that you've learned this semester into a working and useful application. The best project idea is something that interests you! This is also an opportunity to develop something that you can put in your portfolio to show a prospective employer.

The scope of the application you develop is something you can design and develop in the time remaining in the semester. The application should not be trivial, but it should also not be so big and complex that you are unable to finish and test it.

Due date: 11:59pm, Friday, Reading Day

Program name: You may choose any name that is appropriate for your application's main class. But, as has been the case with other programming challenges the name is to begin with your pawprint with the first letter capitalized and the other letters in lower case. The remainder of the name is to be camel-cased. For example, if the application is to be named DocumentEditor and the pawprint is abcxyz9 then the main class name is to be named Abcxyz9DocumentEditor.

Language: Java 8 SE

Tools: NetBeans IDE and JavaFX Scene Builder

UI Development: The application's user interface is to be created using JavaFX. Scene Builder should be used to develop the interface based on FXML. If you choose to develop the interface using code to create the JavaFX objects rather than using FXML created by Scene Builder then you need to present a good reason for doing so in the final project documentation.

Architecture: The application is to be built on the Model View Controller (MVC) architecture as shown in class.

Required Elements: The following are elements that are required to be included in the application:

1. Object oriented elements that you write the code for:
    a. Classes.
    b. Subclasses.
    c. At least one abstract class
    d. At least one Interface
2. Code elements that you utilize:
    a. One or more collection classes.
    b. Exception Handling.
3. The application must have a clearly defined model (as in the M in MVC).

4. The UI must utilize multiple scenes and at least one of the scenes will have the contents of the scene graph changed based on the application state.
5. There must be a way to access “About” information that includes information about you and the application.
6. The application must save data and load data. The target for saving/loading data can be files, a network service, and/or a database.

Expectations:

1. The application is functional for a defined activity, task, and purpose. The goal is to develop a complete application not just a pile of code that doesn’t serve a purpose.
2. The user interface is useable, organized, and understandable.
3. The code is well-structured and logically organized.
4. The application you build is not to be trivial in simply meeting the requirements set forth in this document. Yes, you are to meet the requirements but you are also to build an application that has a purpose and delivers functionality or capability. The requirements are parameters to be used in design and implementation of the application. They are not intended to be the end product.
5. You should design and build an application that you would be happy to show a prospective employer or client.

Documentation: A document is to be written titled “ProjectDocumentation” that is a plain text file (.txt) or a PDF (.pdf) and included in the zip file for the project that describes how you met the requirements provided in this document. You should be able to point to instances in your application where the requirements have been met. This means, you should name which requirement you are meeting, label which file it is in, and which lines of code the requirement is accomplished on. **This document is for your protection!!** If the grader/instructor must search your application’s code to find the places where you met the requirements then they may miss where you met them. This document should be like you are looking over the shoulder of the grader and saying, "Yeah, right there is where I have the code that allows you to save data to a file." By explicitly identifying how you met the criteria the grader/instructor can be sure they are not missing something. It will also save time for the grader that is in short supply during finals week. **Easy to find things --> makes grader happy --> good grade.**

UML: A UML diagram of the application is to also be included in the zip file for the project. The UML diagram can be a PDF, PNG, JPG, or GIF file with a base name of PawprintUML. So, for example if the file is a PDF and your pawprint is Abcxyz9, it is to be called Abcxyz9UML.pdf.

NOTE: If you have any questions about the requirements, what you can do, or if you are unsure about a project idea, then feel free to run it by the TAs to double check. It will be wise to check your project by the TAs during office hours to determine if you have met the requirements before submission. If you have any other questions, then ask the TAs or come by office hours. **TAs will not check whether you have satisfied the requirements via email.** However, you can ask general questions via email.