Hamza Atif

215642861

EECS 3311 – Software Design

Section A

First Software Project

Naeiji Alireza and Kazi Mridul

**Introduction**

This project is intended to introduce students of EECS 3311 about the process of creating a project from start to finish. The goal of the project is to create a software that uses a user interface to load six shapes composed of squares, rectangles, and circles when the “load shapes” button is selected by the user. Then when the shapes are visible to the user, the user has the option to select the “sort shapes” button in order to view the same loaded shapes but in the sequence of smallest surface area to largest surface area. There are a couple of challenges that come to mind when thinking of this goal to achieve like, how to make sure the six shapes fit into the window, how to automatically generate different shapes with different dimensions with colours each time the user clicks upon load shapes, what would be the best sorting algorithm and probably more challenges when actually implementing. I will be using object-oriented design, this will allow me to use the shapes as objects and have methods to create, sort, display the shapes.

I do not see how this can be achieved with the use of object-oriented design as we have been instructed to use a number of classes which all have specific roles that come together to reach the goal output. This will combat code repetition and a more readable software. It is mentioned in the lab guide that we will need a “Shape” class (will be abstract or an interface) which will hold all the general attributes of a shape like, the length, the width and so on. Also, will have class “ShapeFactory” which is used for instantiation of different shapes (circle, square, and rectangle). Will also have a different class which is used to sort the shapes based on surface areas. Since there is the use of inheritance, this means this project will be using the factory pattern.

This project will be structured based on the lab guidance provided by the professor.

**Design of the solution**

First UML class diagram

In this diagram it shows that the ShapeFactory is dependant on the Shape class to make circles, rectangles, and squares. It shows that MyPanel then uses what ShapeFacoty has created to display after it has used the SortTechnique class to short the shapes created.

Diagram

Description automatically generated

Second UML design

Diagram, schematic

Description automatically generated This diagram is based of the factory pattern.

**Implementation of the solution**

The algorithm I have decided to use in the SortTechnique class to sort the shapes based on surface area is bubble sort. Bubble sort uses two nested for loops so that it takes the first element and compares it to the immediate next element. This will iterate and swap when necessary, resulting to a sorted output.

I implemented the first UML diagram as it was more detailed and is mostly confirmed correct as it resembles an in-class example. I started with the shapes class because it was intended to be an abstract class which was then extended by the circle, square and rectangle classes. I then implemented the ShapeFactory class as I now was able to initialize the shapes from the data made by the Shape class. Since I had six shapes being made, I was able to store them in MyPanel as a list and this list would be displayed upon clicking “load shapes” button”. The same list is then sorted into the desired order using the SortTechnique class when the “sort shapes” button is clicked. This is all visible to the user because MyPanel is linked with Frame.

Chart

Description automatically generatedChart

Description automatically generatedGraphical user interface

Description automatically generated

After “sort shapes” is clicked

After “load shapes” is clicked

Initial display

**Conclusion**

Overall, this software project was a great learning experience as I learned many new things like JFrame and creating an interactable window. The part that went really well was having a basic idea of how to attack the problem with the UML diagrams. Preplanning really helps with getting started and makes the problem at hand much easier to start working on. However, there were areas where I did not come to mind while planning out the solution. For example, when I was implementing the SortTechnique class I was unable to sort at all. The shapes were still in the same order when they were first loaded but that was because I was swapping the shapes into order, but I was not swapping the coordinates. This just tells me to think deeper for my next projects and that the more time I put into UML and the logistics, the easier and more efficient coding will be. I have truly learned the value of writing a report about the project as it leads me into deeper thinking and better understanding of how things are supposed to run. My top three recommendations to ease the completion of the software project would be; avail the TA’s help more, pay more attention/ review the course materials beforehand and to practice writing reports and getting better at them.