Oreoluwa Ogunlude

216594285

EECS 3311 Section A Lab 02: Software Design

TA: Kazi Mridul

Lab 3 Software Project: Displaying Shapes

Part I: INTRODUCTION

The software project’s goal is to create an application that shows an interface with two buttons. Upon clicking the first button the application loads 6 shapes (circles, squares or rectangles) and the second button sorts these shapes based on their surfaces.

I faced several challenges while completing this project. The first problem I got stuck on was figuring out how to run the sample project. I was able to import the project into Eclipse, but it turned out my Java Runtime Environment was outdated so I had to install and change it to the newer version.

This software project will be completed using OO Analysis and OO Design concepts learned during the class lectures. First, I will design a use case and a domain model then I will assign responsibilities to objects and use this to finally design the class diagram. The design patterns used to complete the software project are factory design pattern used to instantiate the Shape class and the singleton design pattern to make an instance of the FactoryPatternDesign class.

This report consists of 4 parts starting from part I through IV.

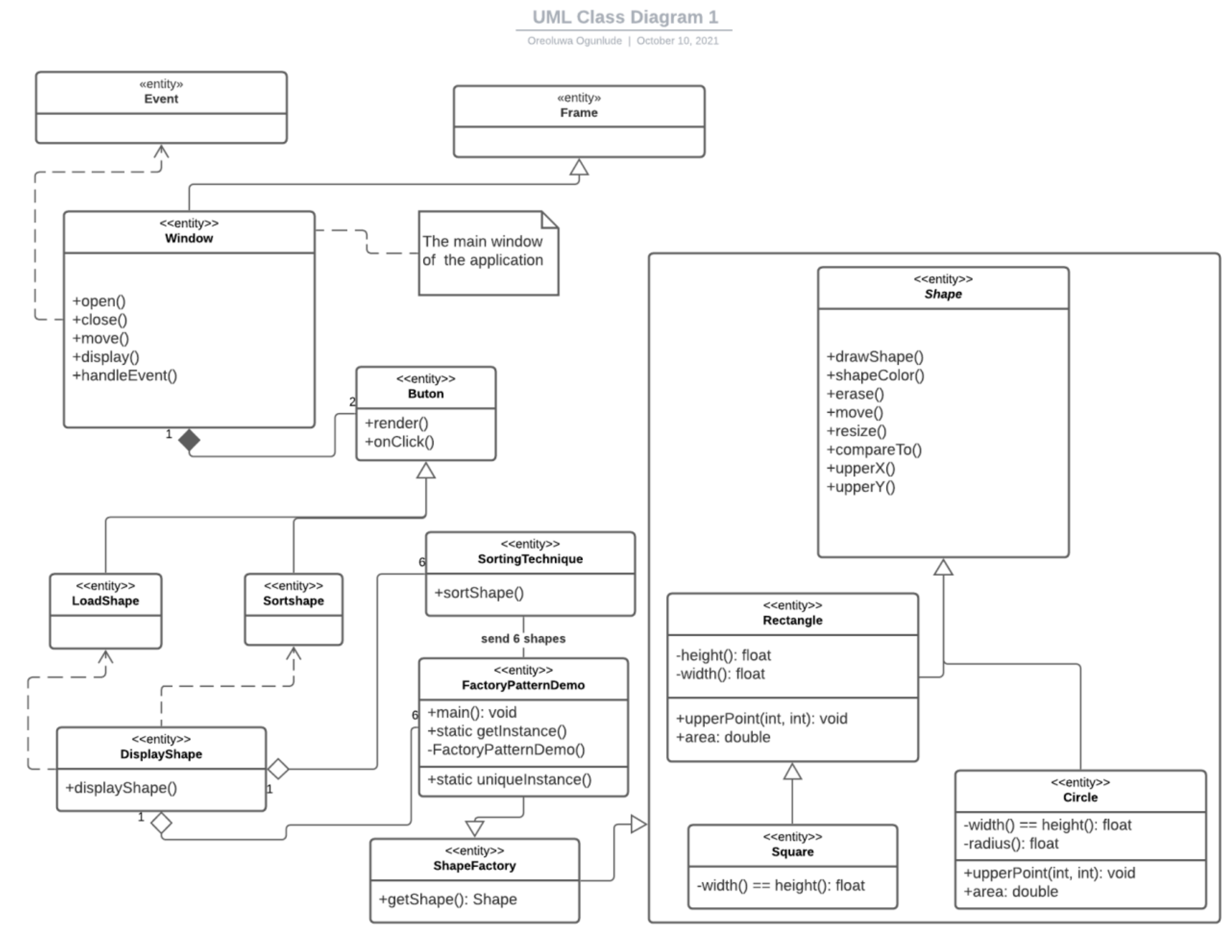
Part I is the introduction into the project explaining the goals and concepts of the project.

Part II consists of 2 UML class design patterns with different design patterns and its commented elements.

Part III describes the sorting technique’s algorithm and the implementation of classes from the UML class diagram in Java

Part IV concludes the project.

Part II: SOLUTION DESIGN


UML Class Diagram 1

Oreoluwa Ogunlude | October 10, 2021

UML Class Diagram 1 Elements

The Shape, Circle, Rectangle and Square class

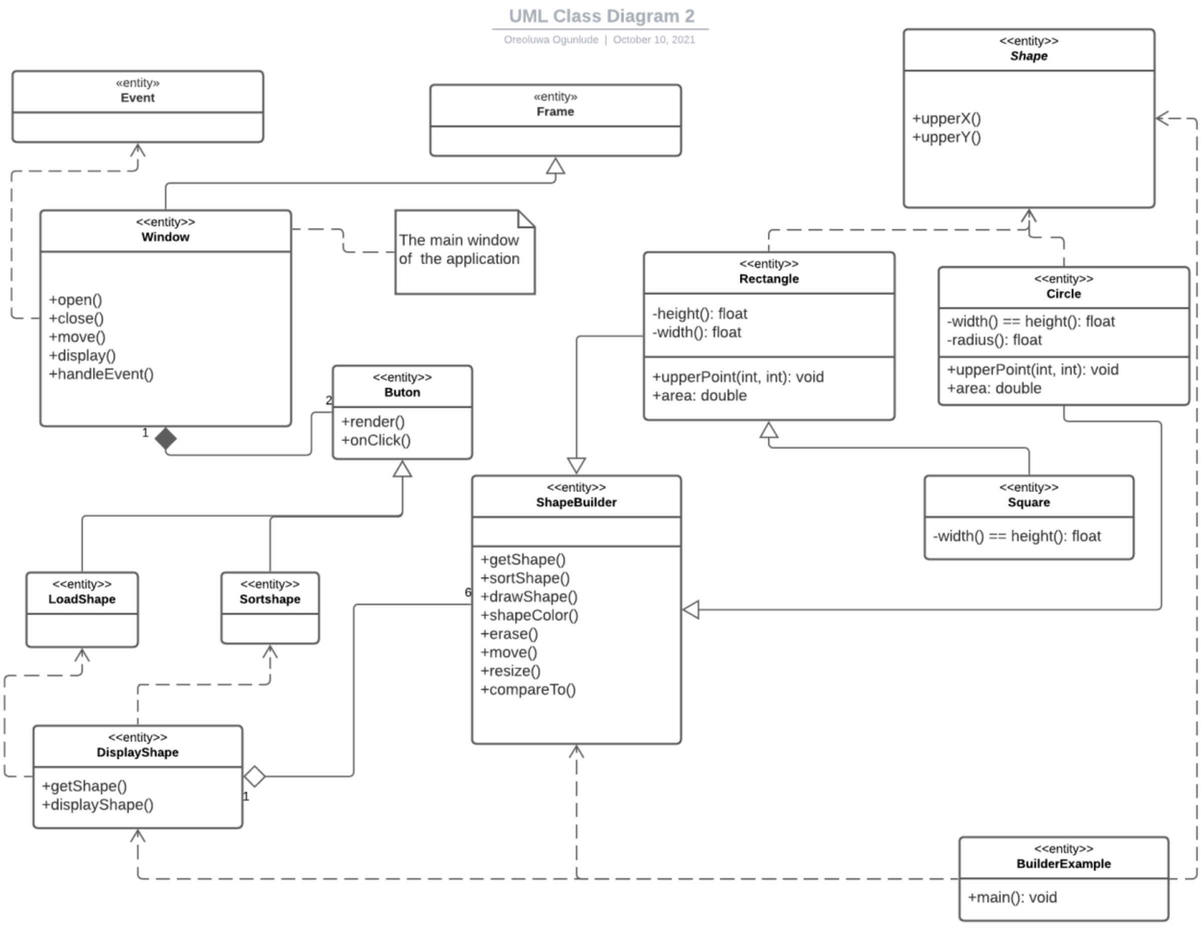
The ShapeFactory, SortingTechnique, DisplayShape and FactoryPatternDemo class

The Button class, LoadShape and SortShape button

The Window, Frame and Event class

The UML class diagram above uses OO Analysis and OO Design principles. The class diagram shows the set of systems principles needed for an application to display shapes and the relationship between these classes. For example, the *Shape* class is a parent class and has the subclasses Rectangle, Circle and

Square. These subclasses inherit attributes from the parent class and this is shown by connecting an arrow with a whitehead from the sub class to the parent class. The factory design pattern is used to instantiate the Shape object by using the ShapeFactory and FactoryPatternDemo classes. The singleton design pattern is used to make instances of the FactoryPatternDemo class so that a unique instance can be used by other classes namely the DisplayShape and SortingTechnique class.



UML Class Diagram 2 Elements

The Shape, Circle, Rectangle and Square class

The ShapeBuilder, BuilderExample and DisplayShape class

The Button class, LoadShape and SortShape button

The Window, Frame and Event class

Part III: SOLUTION IMPLEMENTATION

Sorting technique – Bubble Sort

I have implemented UML Class Diagram 1 in my Java Code.

I used Eclipse IDE 2020-09, JDK-17_Windows-x64

Part IV: CONCLUSION

The lecture videos and slides provided on eclass helped in creating the UML class diagrams for this software project. I used the principles and concept from OO Analysis and OO Design to create different classes and relate them in my class diagrams.

Unfortunately, I was not able to set up my eclipse to run my code properly. I could not resolve the errors in the Sample Project’s code and why any changes in my code did not reflect in the program output. I tried emailing a TA but I could not get any response before the deadline and the student discord server wasn’t of much help either.

This software project has made me better understand the concept of different design patterns and their use. Creational design patterns focus on the instantiation of objects, structural design patterns focus on the composition of classes and behavioral design patterns focus on the communication between objects.

My top three recommendations to ease the completion of the software project are as follows:

1.) TAs should respond to students’ questions about tasks punctually.

2.) Split software projects into smaller portions so that student can learn step by step rather than a bulky submission.

3.) A recorded video explaining in full detail what is expected from the software project, and it should show an example of what the correct program should like.