**NAME :- SHESHAN VERMA STUDENT NO. :- 217631441 SEC :- EECS 3311-A**

**SOFTWARE PROJECT 1 T.A : - NAEIJI ALIREZA**

**Part I – Introduction**

The software project I of EECS 3311 is all about displaying buttons, and some object shapes on an interface. The main goal is to understand the concept of the OOD and software design pattern and implement them to perform the shape appearing that is load the shapes, arranging the shapes according to their sizes that is sort the shapes and all these actions should be taken place upon onclick the buttons present on the interface.

Challenges associate with the project were like creating the button, generating the area of the shapes, calling the shapes with function performed with button click and sorting the shapes according to the sizes.

Some OOD principles which we will be using to perform this project are like **abstractions** class whose relevant method are only share rest keeping hidden, **encapsulation** in a way, **inheritance** used by child class to extend the use of the parent class method also we use the software design pattern **singleton** class where the object is used to coordinate with the system, also **factory pattern** used to instantiate objects of a sub class whose parent class has several sub-classes.

My report will flow through the code that I design for my first software project, with its UML designs and discuss on the solution, it’s implementation and conclusion.

**Part II – Design of the Solution**

Diagram, schematic

Description automatically generated with medium confidence

This is the UML class diagram which flows the trend of OOD principles as we can see all 3 classes Square, Rectangle and Circle are inherit it’s property from the parent class Shape and classes like shapefactory, sortingTechnique, loadButton, sortButton, Main class are followed as singleton classes.

Below is alternative rough UML design for this same task where OOD principles have some form of approach as in we extends the class rectangle into the square to compute their surface area since they both acquires the height and width irrespective to the class circle and another form of looking at the main class where we extend the purpose of main class with JPanel and make the performance of the actionlistener to call the buttons Sort Shape and Load Shape and this buttons do the sorting with the help of sorting technique and load shapes with the shapefactory class(creating this two class singleton to ease the instantiation of the object of the classes)

Diagram

Description automatically generated

**Part III – Implementation of the Solution**

The sorting technique which I use for this project is the insertion sort and also the reference that I took from the website all been shared at the top of the sortingTechnique class.

Text, letter

Description automatically generated

Also attached screenshot is the algorithm which I referred to complete my sorting technique code.

**Shape Class : -** This is my abstract parent class which implements a comparable interface, from which the methods, and field where shared with the child class, this class also compare the computesurface area of the child classes.

**Rectangle Class : -** This class extends the property from its parent class that is the shape class and also has its constructor class, in the follow class we also compute the surface area of the rectangle that is by multiplying height and width, and in the end, there is one more method draw.

**Square Class : -** This class extends the property from its parent class that is the shape class and also has its constructor class, in the follow class we also compute the surface area of the square that is by multiplying width and width, and in the end, there is one more method draw.

**Circle Class : -** This class extends the property from its parent class that is the shape class and also has its constructor class, in the follow class we also compute the surface area of the circle by importing the math function where we divide the with by 2 ,squaring and multiplying the resultant value with the PI value and in the end, there is one more method draw.

**Shape Factory Class :** - This is a singleton class which generate new shapes and form the coloring of shape generate during the execution.

**SortingTechnique Class:** - This is a singleton class which performs the function of sorting the shapes when called, and this sorting is done on the bases of their surface area.

**Panel Class :** - This class extends the JPanel and it’s functions is to get the shapes and paint the shapes formed with the help of the method paintcomponent.

**SortButton Class :** - This class is a singleton which extends the JPanel and implements the actionlistener to form the task a button to do. This class the singleton class sortingTechnique and upon clicking the button in the interface the class should sort shapes which are on the interface which happens according to their surface area. (Sorry I tried lot but don’t know why this is not function the task)

**LoadButton Class :** - This class is a singleton which extends the JPanel and implements the actionlistener to form the task a button to do. This class call the singleton class shapefactory and upon clicking the button in the interface the class should pop up the shapes irrespective to their surface area.(Sorry I tried lot but don’t know why this is not function the task)

**Main Class:** - This class is like the performer class which calls the class and complete the purpose of the interface.

**Version I was using for this project is Eclipse/Java**

Graphical user interface, text, application

Description automatically generated

My code wasn’t running after this, the function of sort button and load button were not performed, please and hopefully this should make out lot of my grading from the projects.Thankyou!

**Part IV – Conclusion**

Creating the shape class and implementing the inheritance into its child class was smooth, running and testing my sorting technique and implementing the code from the algorithm was a good task.

Thing went really wrong was in my project is calling the singleton classes into the sortbutton and loadbutton class and they didn’t perform the functions upon clicking the buttons in the interface

I feel I have learned enough, recapping the java concepts, implementing the OOD principles and design, creating and analysing the UML diagram, this has enhanced my critical software skills, due to lack of time with me I couldn’t finish it.

To ease the project I should have managed my time, follow the understanding concept then doing other things which also took my lot of time.

THANK YOU,

SHESHAN VERMA (217631441)