Name: Dhruv Ukani

Student Number: 216821902

**SOFTWARE PROJECT 1 REPORT**

**Part 1: Introduction**

This software project is about creating random shapes and sorting them based on their area. The goals for this software project is to learn OOD principles and their implementation with Graphics2D. To create a frame with two buttons, one to load random shapes and other to sort shapes based on their areas.

OOD (Object-Oriented Design) is a software design approach towards collaborating the process of how objects are defined and their interactions to solve a problem that was identified.

Principles of OOD are:

1. Abstraction: The process of selecting data information from a larger pool of data is called Abstraction. It allows users to only show the relevant information needed about the object.
2. Encapsulation: It encapsulated the data into private mode so no one outside the class can access it. It provides safety to variables and can only be used in other class by invoking it into the method.
3. Inheritance: It is ability of one object to inherit the properties of another object. There are no limitations of reusing the methods. There are various types of inheritance such as single inheritance, multilevel inheritance, hierarchical inheritance, and hybrid inheritance.
4. Polymorphism: It allows to use a class exactly like parent class in our own ways, i.e, each child class can have their own functions and methods.

I will be using all the properties of OOD in my software project to reach the usability and protecting the data. I have used Hierarchical Inheritance to better ease the code and design process.

For structuring the report, I am making quick notes to remind me of what to add when finalizing the report. I have a daily schedule to work on the project.

**Part 2: Design of the solution**

**Diagram

Description automatically generated**

UML CLASS DIAGRAM

**Part 3: Implementation of the solution**

* I used the Selection Sort algorithm of the Sorting class to sort the shapes based on their areas. It is the basic sorting algorithm with the time complexity of 0(n2 ) because of nested for loop inside a main for loop. It compares the first value of array and compares them with the all other afterwards. Whichever value is less than is swaped with the current selected value. After iterating through both the loops we get the first value of an array to be the smallest in area and last one to be the highest in area. And at last I am changing the indexes of the swapped values alongwith changing their frame dimension X and Y.

* Implementation and Compilation: This project is basic understanding of implementing the design from Jframe and JPanel. Such as, frame inherits JFrame and panel inherits JPanel.

MyRectangle, MySquare and MyCircle classes extends Shape class, i.e inherits or acquires all the properties of Shape class, such as all getters and setters method. This defines the use of Hierarchical Inheritance. All the shapes get their dimesionX, dimension, width, height and color herited from the Shape class. Color is randomly selected using Math.random function and this library is also used to give random width and height to all the shapes. Next step is to create a function that allow users to create 6 random shapes form Rectangle, Square and Circle. ShapeFactory class is used to achieve it and store it into List. I also used Singleton property to change all the instants and objects of this method if anything changes.

StartingMain contains the main method to run the java code. I added the frame and panel and two buttons, namely, “Load Shapes” and “Sort Shapes”. Load Shapes button calls the shape generator method from the ShapeFactory and draws them into the panel. Sort button calls the Sorting method to sort the generated shapes based on their area. This sorted list of shapes is again printed into the panel.

* I have used Eclipse Version: 2020-12 (4.18.0). JDK version is 14.0.2
* Execution of code:

1. Graphical user interface, application, Word

   Description automatically generatedScreenshot of ideal Frame
2. Chart

   Description automatically generatedScreenshot of Load Shapes
3. Screenshot of Sort Shapes

Chart, waterfall chart

Description automatically generated

* Video Link: <https://drive.google.com/file/d/13VUjnhFw7uemgSQZkLPbaBZV11K2vIfE/view?usp=sharing>

**Part 4: Conclusion**

* Connecting all the java classes to each other was quite easy and I did it in my first attempt. My code was working the way I wanted it to be until I reached the ShapeFactory. I was successfully able to extend Shape class into rectangle, square and circle class. I was also able to sort the list of shapes using the Comparable<T> on my first attempt. I got correct outputs for Bubble sort and Insertion sort as well, but sticked to Selection Sort.
* The only thing that went wrong was that I was trying to build my project with Shape interface, not the shape as an abstract class. It went pretty good for assigning all the properties of shape to the random shapes generated, but it went quite complicated when creating a list of random shapes generated and storing them so that sort method can sort it. I tried many ways but could not find a way to solve the problem, so I changed the interface into abstract class.
* I learned many different concepts from this project. I am new to Java Swing, but now I think I have good understanding of Java AWT and OOD. I learned how to convert objects into grpahics2D as well. Until now all I coded in Java was backend with no frontend design. So this project helped me a lot to design the software appropriately.
* Recommendations:

1. More smaller labs that can help to understand the concepts before making the projects.
2. More practice by reading books and understanding the core concepts.
3. Collaborate with others to ease out the process of understanding.