Name:JIANKAI HE

Student ID:215628779

Course:EECS 3311

Section:B

Software project name:LAB 3 first software project

TA name: Naeiji Alireza

Part I.

(1) Explain what the software project about and what are its goals.

This project will create a frame window with two buttons, one will create six random shapes including circle, rectangle and square, another button will sort the shapes by area. The six shapes will be painted on the frame in order.

The goal of the project is to practice the application of JAVA swing, class inheritance, abstract, interface, sort and design pattern.

(2) Explain the challenges associated to the software project.

The challenges include how to implement a JAVA swing application, how to add action listener to buttons, how to draw each shape on the frame, the sort algorithm, and the design pattern for the whole project.

(3) Explain the concepts (e.g., OOD, OOD principles, design patterns) you will use to carry out the software project

I will use inheritance and polymorphism for the super class Shape and derived classes Circle, Rectangle and Square. The Shape class will be defined as abstract, and the derived classes will implement the specific methods.

I will define an Observer class to listen to the sort event, it's Observer Design Pattern.

The shapes will be created from a factory class, it's Factory Design Pattern.

(4) Explain how you are going to structure you report accordingly.

I will show the UML class diagram firstly, then describe the functions of each class, introduce the design pattern and the relations between classes.

Part II.

(1) Create a first UML class diagram of your system (use at least two design patterns), add the corresponding figure in the report and comment its elements.



Factory Pattern and Observer Pattern are used in the design. All the shapes are created from the ShapeFactory class, it's factory pattern. When user clicks on the sort button in frame, the ShapeManager will be notified by the Observer interface, it's Observer Pattern.

(2) Use OO design principles in your class diagram.

Inheritance and polymorphism are used in my class diagram.

The Circle and Rectangle classes derive from Shape class, the Square class derives from Rectangle class. The area() and paint() methods are abstract in Shape class, and implemented in subclasses.

Circle, rectangle and square are instantiated as Shape type in ShapeManager class, and the area() and paint() methods will be called during sort and painting. It's polymorphism.

(3) Propose a design alternative by creating a second UML class diagram.

I can change the design to Factory Pattern and Strategy Pattern.



This design pattern works similar with my first design pattern. The first one is Observer Pattern, the notification can be conveyed to listener easily. The second one is Strategy Patter, the order of shapes are highlighted and operated in separated classes.

Part III.

(1) Describe the algorithm of the sorting technique you have used to sort the shapes.

I used the most traditional bubble sort algorithm to sort the shapes by area.

(2) Describe how you have implemented and compiled all the classes of your class diagram in Java.

I implemented all classes according to first design pattern.

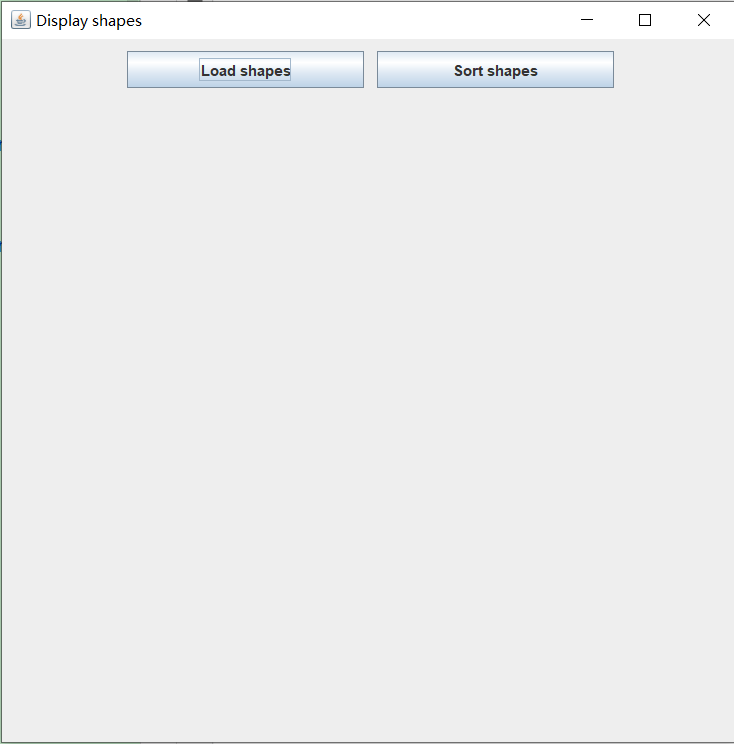
I implemented classed in following order: Shape, Circle, Rectangle, Square, ShapeFactory, ShapeManager, Main.

(3) Specify the tools you have used during the implementation.

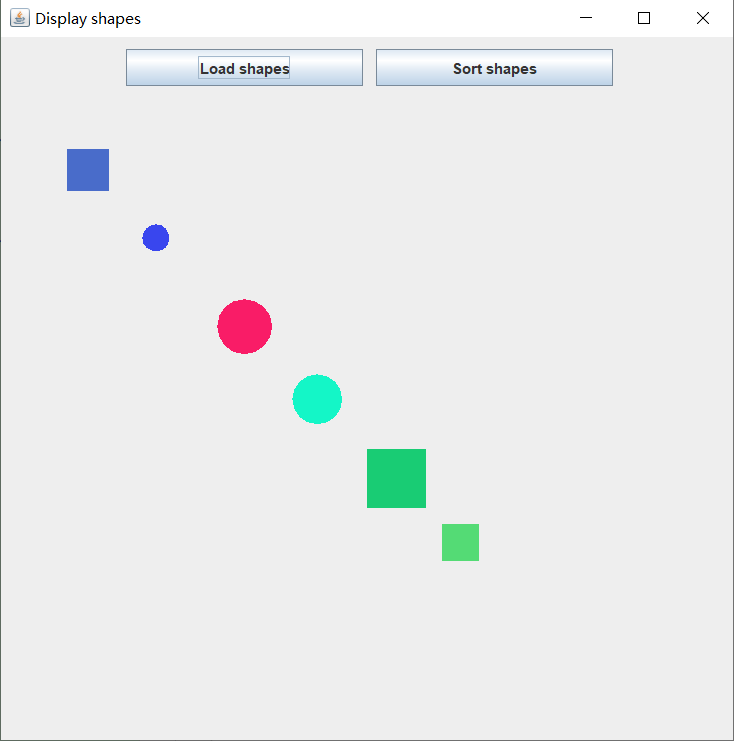
I used Eclipse.

(4) Take a snapshot of the execution of the code (i.e., of the interface) and comment it in the report.

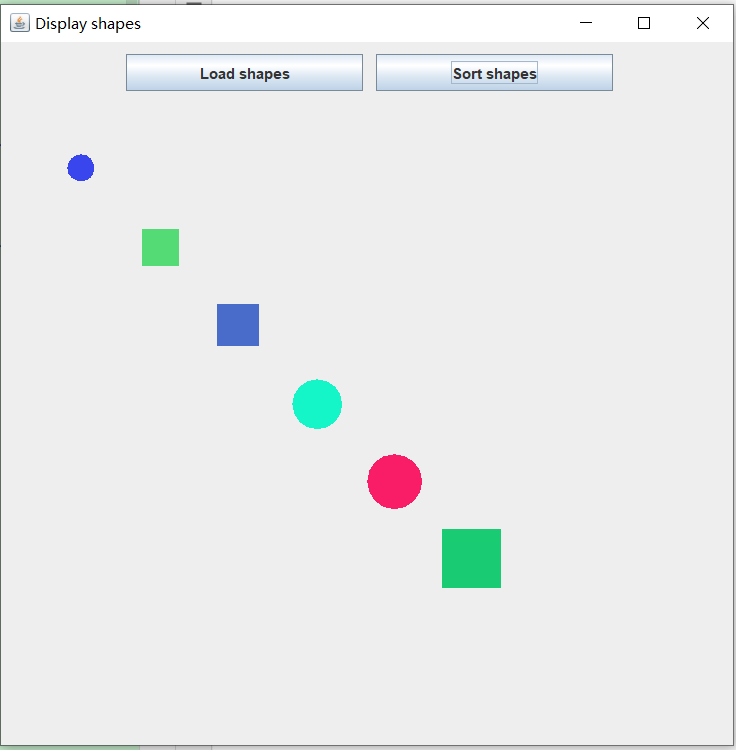
The interface on start:



The interface after click on "Load shapes":



The interface after click on "Sort shapes":



(5) Create a short video (2 to 3 mins) showing how to launch your application and run it.

Refer to the separated video file.

Part IV.

(1) What went well in the software project?

The program can work in accordance with the requirements of the assignment, including:

The frame is empty except two buttons after start;

After the "Load shapes" is clicked, random circle, rectangle and square shapes will be created with random size, and are shown on the frame;

After the "Sort shapes" is clicked, the shapes will be sorted by area and repainted on the frame.

(2) What went wrong in the software project?

The shapes are compared by general getter method in my first version, rather than the required Comparable interface. The code was modified and shapes can be compared by compareTo() method now.

(3) What have you learned from the software project?

I learned the different design patterns from the project.

The knowledge of inheritance and polymorphism was enhanced.

The sort algorithms were learned and implemented by myself.

The knowledge of event drive of buttons, the painting of shapes was used.

(4) What are your top three recommendations to ease the completion of the software project ?

Create the framework of classes and relationship of them first, before filling the detail of each class.

Create the super class firstly, then the derived classes.

Build the frame window firstly, then add each component to the window.