

CS5001-p4-vector-drawing Report

Student ID- 190026870

Assignment Number- 04

Assignment topic- Vector Drawings

Date of submission- 22nd November 2019.

This is a short report on CS5001 assignment-4 (Vector drawing). This assignment uses Model Delegate pattern to draw shapes. Model delegate implementation of this assignment includes implementation of PropertyChangeListeners and observers. All the shapes have their own classes and implement a abstract method draw(-) of Shape class (User defined class) of delegate.shape package. In this model the shapes are kept in a separate package called shapes which is a sub package of delegate.

GUIDelegate is the class in delegate package which responds to user events and triggers the appropriate methods of model based on the events. This class extends JFrame and other components such as JPanel, JMenuBar and JToolBar are added to the frame. JButtons are added to the toolbar. ApplicationPanel class of delegate package extends JPanel and overrides paintComponent(Graphics g) method. These classes represent the state of the model if there is any change in the state. All the user inputs are handled in these classes. As PropertyChangeListener is implemented by the GUIDelegate class it calls propertyChange(-) Methods when the model calls notifier.

The main operations involved in the GUIDelegate are creating buttons adding them to toolbar and creating menus adding them to menubar, inturn these toolbar and menubar are added to the frame using BorderLayout. ApplicationPanel(JPanel) is added to the frame in GUIDelegate class but it is a separate class in the delegate package. In this design applicationpanel is designed in a separate class rather than in GUIDelegate to minimise the complexity of the code. Listeners for toolbar and menubar are implemented as classes inside the GUIDelegate which uses the concept of inner classes. ApplicationPanel class also uses the concept of innerclass by implementing the mouse listeners.

Model class in the model package initialises which shape is to be drawn based on the event generated by the users (button pressed by the user). All the changes will be notified by the change listeners so that the delegate will be updated. The concept of polymorphism is used to draw the shapes by iterating the selected shapes in the for loop of GUIDelegate class. Junit testing is done for the methods in model class in the tests package. TestRunner class in tests.model class gives the count of passed and failed tests. Tests for the methods resides in ModelTest class

Two advanced requirements were implemented in this assignment which are drawing other shapes such triangle, parallelogram, hexagon and saving the file and loading a file. For saving and loading a file a separate class is created and these operations will be initialised in the model class using save() and load() methods.

Features implemented-

- Line
- Rectangle
- Ellipse
- Parallelogram
- Triangle
- Hexagon
- Undo
- Redo
- Clear
- Different colors
- Solid fill
- Save
- Load