Mini-Project-5
I/O Streams

Rahim Sharifov, Sara Shamilova

November 2018

1 Objective

Objective Mini-Project-5 is to write swing application that enables user to create some figures on the displaying window, it also gives user permission to move, remove or change color of figure. Application should also able to save project or open an existing project.

2 Program

Program consist of 4 classes: Main, MenuBar, Edit, Figure

2.1 class Edit

Drawing on the displaying window is the main job class Edit.

It extends class JPanel because object of Edit class is supposed to be our canvas and implements MouseListener,MouseMotionListener as we will use mouse to draw or change something using mouse and Serializable as we will store object of this class in file.

Atributes isOval of type boolean is used to draw an Oval or Rectangle ,variable 'state' is our choise which has 4 option:"Create","Remove","Move","Change" which help us to choose which action to make when MouseEvent occurs.

Xinit,MouseX,Yinit,MouseY represent coodinates on our canvas.

Color c is color of drawn figure which is set to WHITE by default at first.

As class Edit is subclass of Jpanel, it provides us paintComponent() method to draw on this panel which is called by default when program is run by user.

CreateItem() method is intend to create figures. It initializes new Figure which is whether circle or rectangle based on state of variable isValue with

Color c . Class Figure also extends JPanel, it means object of this class is also a Component.Every time this method is invoked, new panel is created,so every figure that is drawn in our main canvas will be panel.So we add this component to main canvas. 50 is default size of every figure and to make mouse cursor be in center of every figure when they are created we make coordinates of panel 25 less than actual coordinates of mouse cursor.Bounds is set to the same size of figure , no other spaces is needed. SetOpaque() method is set to make background of panel transparent,as when we draw Circle there is free space that normally would be gray,now it's gone. So when we call method with repaint() method a figure with certain color and shape on certain coordinate will be appeared on main panel.

MoveItem() method is intent to move figures on the Panel. It sets the location of panel based on the coordinate of mouse cursor,so location of component changes as we drag mouse. It checks component if it is not main canvas, as we want to move little panels not main canvas.

ChangeColor() method is intent to change color of panel. It assings given component to be Figure, since all of our components on the board is instance of Figure. So method assings color of this figure to be color that is choosen using JColorChooser.

MouseClicked() gets coordinates of mouse where it was clicked and checks if state is "Create", if so, it invokes CreateItem() method which creates figure on that coordinate, and repaint() method illustrates it on the screen.

MousePressed() assing coordinates of mouse to Xinit,Yinit. Difference between Xinit,Yinit and MouseX,MouseY is that first one is initial coordinate when mouse is pressed. When mouse dragged MouseX changes, but Xinit remains the same. So we get Component where mouse is pressed, based on state of the variable "state" it either removes that component or changes its color.

MouseDragged() is mainly intend to move figures, create and remove is additional, so it gets coordinates of mouse cursor when it is dragged , and based state of the variable "state", it either it calls moveItem(), either creates multiple panels in a row, or removes panels whichever is in its way.

2.2 class Figure

Figure class is a class that we use to create figures which are panels.It has two attributes: boolean isOval which is used whether draw oval or rectangle, Color c which is color of drawn figure.

paintComponent() method simply sets the color and draw Oval or Rectangle on the coordinate (0,0) with size 50.

2.3 class MenuBar

Class MenuBar is intent to, as it is obvious from its name, create a menu bar on the top of window, to make application easy to use by user. It has attribute 'ed' of type Edit, as we will need access some attributes or method of that class , JFrame j is intent set contentPane of main frame when user open an existing file. ObjectOutputStream out as we will store a project from this application as object of Edit class.

In abstract, created menubar is consist of 3 section which in java is JMenu : File, Edit,Tools.

File section also consist of 3 JMenuItem, that are "New Project" to start a new project, "Open Project" to open a new project, "Save Project" to save project.

When user clicks the "New Project" using JFileChooser a dialog appears on the screen to allow user to choose where he/she want to create new project. From this dialog program gets path and name of project that user wants to create, and it add .swg extension to its end, as it is our custom format of out application.And it creates file variable that points to this file. And from this file variable ObjectOutputStreamer out is created to that. By using this object we can write any object inside of created file. This is all "New Project" button do , it creates new file and object streamer to that file.

When user clicks the "Save Project" button, using 'out' object we write our main canvas with all its figure on it to the file streamer links to. Our work is saved.

When user clicks the "Open Project", using JFileChooser, program shows openDialog whatever file user picked program gets its path, creates FileInputStreamer from that, and using FileInputStreamer it creates ObjectInputStreamer. From file that InputStreams points to program gets object that is inside that file which should be object Edit class.And Frame J which is main frame of our application sets this panel as contentPane , and drawing on this panel appears on the window.

Other JMenu section is Edit,which has 4 JMenuItems, "Create" to turn state of application to create new figures, "Move" to turn state of application to move selected objects , "Remove" to clear whole window , "Change Color" to change color of drawn figure. Only item that is defined in our application is Remove, JMenuItem itself is AbstractButton, so using ActionPerformed method of inteface AbstractAction we make application do something when that item is selected, so when remove button is selected it changes the state of variable clearAll of class Edit and calls repaint() which clears the whole window.

Last JMenu section is Tools which provides items "Color" to set color , and "Shape" to set the shape of figure to be drawn . When Color button is selected, JColorChooser shows dialog with color options , when user chooses a color , that color is set to Color c of class Edit.After that figures that will be drawn will be in that color. "Shape" item itself has subitems "Oval" and "Rectangle" to set shape of figure. Button Oval turns the variable isOval to be true ,after that figures that will be drawn will be ovals.Button ”Rectangle” is opposite of that.

All that is created inside getMenuBar() method.

2.4 class Main

class Main is intent to implement both Edit and Menubar classes. Inside main method myFrame is created which is our main frame. Close Operation is set to EXIT_ON_CLOSE and size is set to (500,500). (Line 7-9)

instance menuBar of type MenuBar is created and using getMenuBar method of this instance a menu bar is added to main frame (Line 11-12)

contenPane of frame myFrame is set to instance of class Edit which is also Panel. (Line 13)

Background color is set to WHITE , frame is set to visible (Line 15-16)

Inside InvokeLater MotionListener, MouserMotionListener is added to make main canvas to be sensible to MouseEvents.

2.5 Conclusion and Instructions

So, this is a application that allow user Create,Move,Remove figures and change the color of figure, user also can save current project, also start new project or open an existing one

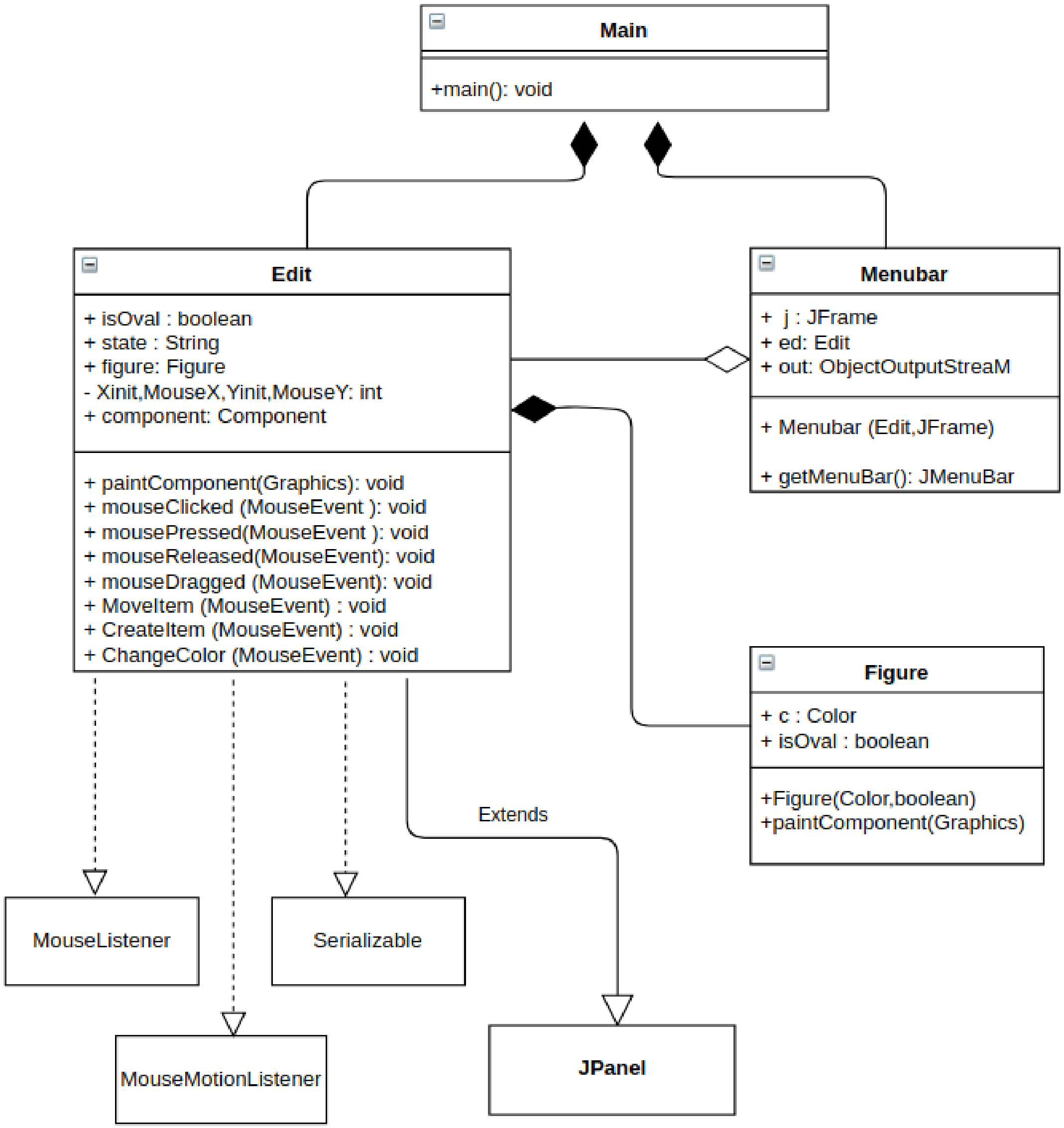
Program is very user-friendly. By default color is set to white, state is Create, and shape is Oval.
To create new figure with different color, choose color from tools section then click on frame. You can change shape to be rectangle from Tools-Shapes-Rectangle.

To move figures on the window ,set Edit-Move, then press the figure that you want to move move it without releasing the mouse.

To remove , set Edit-Remove , press figure that you want to remove, it will disappear.

To change color of figure , set Edit-Change,then set different color from Tools-Color, then press figure you want change the color.
To start new Project, File-New Project.
To open existing Project, File-Open File
To save Project, File-Save Project .

3 UML


Main
+main(): void
Edit
+ isOval : boolean
+ state : String
+ figure: Figure
- Xinit,MouseX,Yinit,MouseY: int
+ component: Component
+ paintComponent(Graphics): void
+ mouseClicked (MouseEvent): void
+ mousePressed(MouseEvent): void
+ mouseReleased(MouseEvent): void
+ mouseDragged (MouseEvent): void
+ MoveItem (MouseEvent) : void
+ CreateItem (MouseEvent) : void
+ ChangeColor (MouseEvent) : void
Menubar
+ j : JFrame
+ ed: Edit
+ out: ObjectOutputStreaM
+ Menubar (Edit,JFrame)
+ getMenuBar(): JMenuBar
Figure
+ c : Color
+ isOval : boolean
+Figure(Color,boolean)
+paintComponent(Graphics)
Extends
MouseListener
Serializable
MouseMotionListener
JPanel