

2022

Programming 2



Abdelrahman Raslan
20221460102

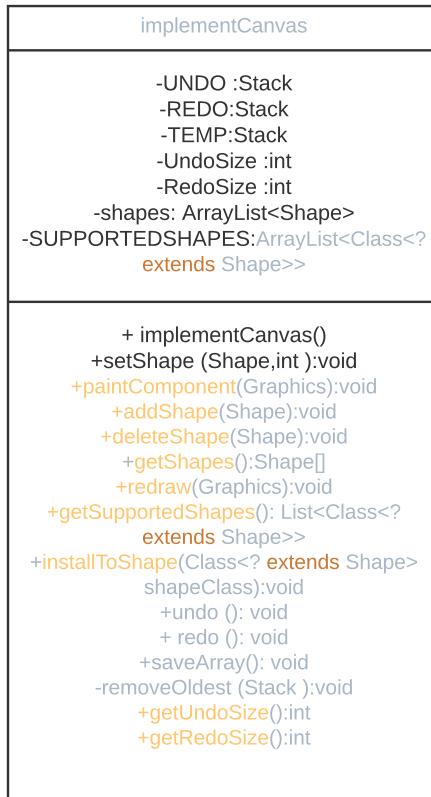
Rewan Salah
20221447143

Aml Ibrahim
20221443988

Suhaila Adel
20221376543

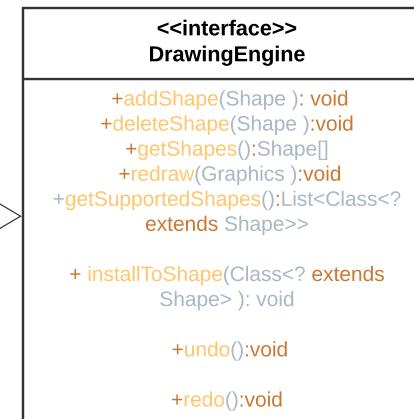
REPORT

Vector Based Drawing Application

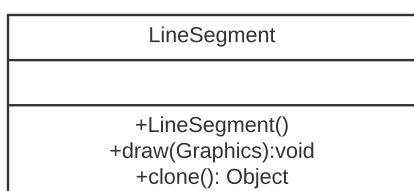
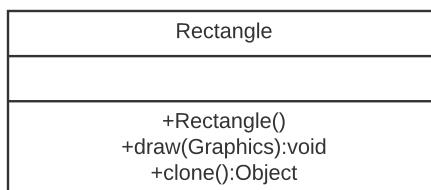
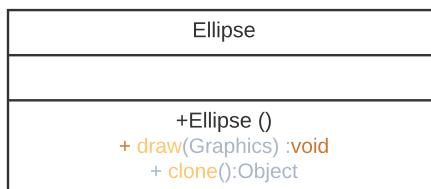
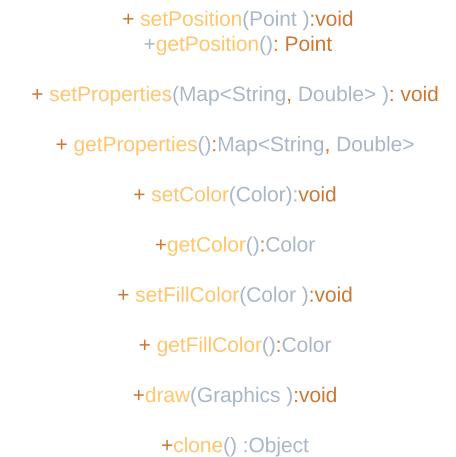


UML class

..... | May 23, 2022



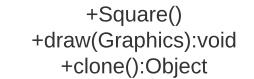
<<interface>> Shape



Shape2



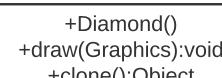
Square



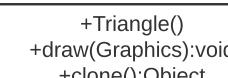
Circle



Diamond



Triangle



Vector Based Drawing Application

libraries

```
import java.awt.Point;
import java.io.File;
import java.lang.reflect.Constructor;
import java.lang.reflect.InvocationTargetException;
import java.net.MalformedURLException;
import java.net.URL;
import java.net.URLClassLoader;
import java.util.Iterator;
import java.util.Map;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JTextField;
import javax.swing.filechooser.FileNameExtensionFilter;
import java.awt.Graphics;
import javax.swing.*;
import java.awt.*;
import java.util.ArrayList;
import java.util.List;
import java.util.Stack;
import java.util.LinkedHashMap;
```

Classes

TASK1: Geometric Shapes Data Model

Interfaces:

- DrawingEngine
- Shape

Classes of Shapes:

- implementCanvas

This Class inherit from JPanel

& Has 14 methods Doing the Following:

1. Make Array list for save every action in running
2. Constructor for implement stake to action undo and redo
3. Set area for drawing shapes
4. Add, Delete and Redraw Shapes
5. Install Shapes to drawing area
6. Undo and Redo from Stack by push & pop at size 100

- Shape2

This Super Class that all shapes inherit from it

& Has 10 methods Doing the Following:

1. Constructor to Create position and color to fill and outline of shapes
2. Setter and getter for position to return position from user
3. Setter and getter for Properties with Mapping on interface
4. Setter and getter for Color Outline and fill Color
5. Take object From Clone and throw to copy in GUI

- Circle
- Square
- Triangle
- Rectangle
- LineSegment
- Ellipse

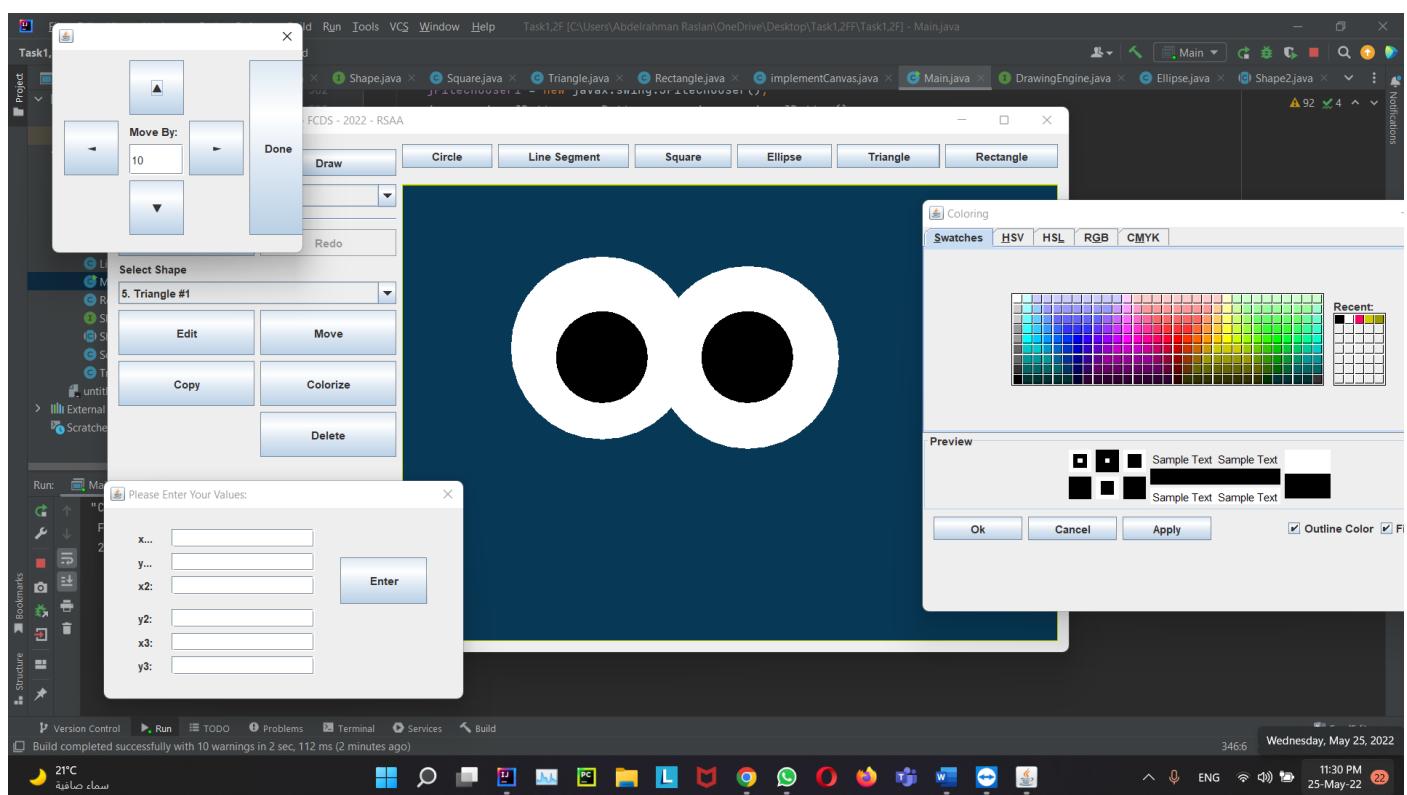
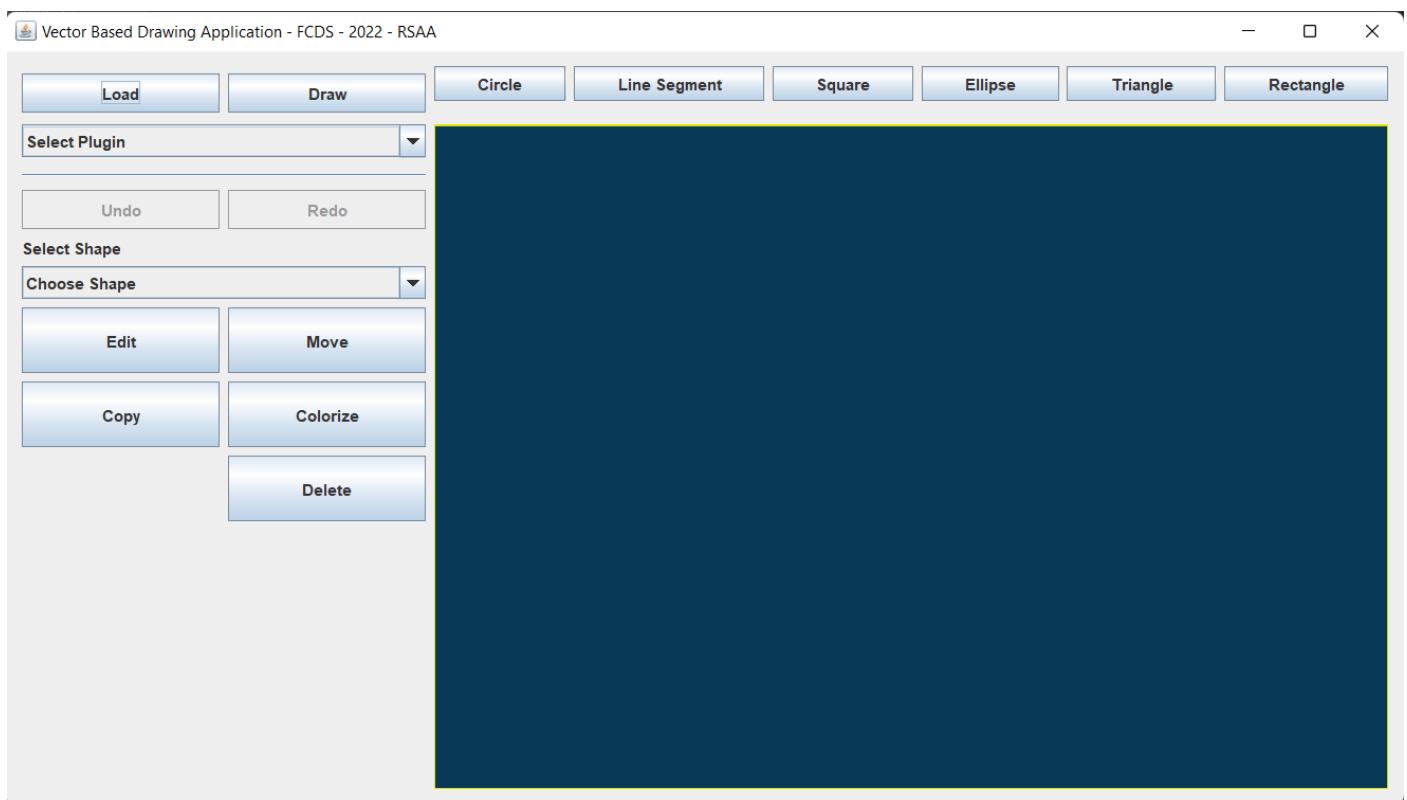
Every class has 3 methods

- constructor: Make position on Point of origin
- draw: Take dimensions of shape as (radius – Height – Width – side)
- clone: duplicate shape with same dimensions

TASK2: Drawing and Painting Application

Class of GU:

- Main



Program Contains 15 Button that do the following actions:

1. Load: This button takes jar file from user

```
507     //Load
508     1 usage
509     private void LoadButtonActionPerformed(java.awt.event.ActionEvent evt) {
510         try {
511             FileNameExtensionFilter jarFilter = new FileNameExtensionFilter( description: "JAR files", ...extensions: "jar");
512             jFileChooser1.setFileFilter(jarFilter);
513             jFileChooser1.showOpenDialog( parent: null);
514             File pluginFile = jFileChooser1.getSelectedFile();
515             String qualifiedClassName = JOptionPane.showInputDialog( message: "Enter Qualified Class Name: ",
516                 initialValue: "Final.prog2/src");
517             URL url = pluginFile.toURL();
518             URL[] URLs = new URL[] {url};
519             ClassLoader loader = new URLClassLoader(URLs);
520             Class myClass = Class.forName(qualifiedClassName, initialize: true, loader);
521
522             Canvas.installToShape(myClass);
523             PluginsList.removeAllItems();
524             Canvas.getSupportedShapes().forEach((supportedShape) -> {
525                 PluginsList.addItem(supportedShape.getSimpleName());
526             });
527
528         } catch (MalformedURLException | ClassNotFoundException ex) {
529             Logger.getLogger(Main.class.getName()).log(Level.SEVERE, msg: null, ex);
530         }
531     }
```

2. Draw: This button draw plugins

```
486     //Draw
487     1 usage
488     private void DrawButtonActionPerformed(java.awt.event.ActionEvent evt) {
489         if(!Canvas.getSupportedShapes().isEmpty()){
490             try {
491                 Constructor<? extends Shape> constructor = Canvas.getSupportedShapes()
492                     .get(PluginsList.getSelectedIndex()).getDeclaredConstructor();
493                 constructor.setAccessible(true);
494                 CurrentShape = constructor.newInstance();
495                 CurrentShapeProperties = CurrentShape.getProperties();
496                 prepareAddDialogBox();
497             } catch (NoSuchMethodException | SecurityException | InstantiationException | IllegalAccessException
498                 | IllegalArgumentException | InvocationTargetException ex) {
499                 Logger.getLogger(Main.class.getName()).log(Level.SEVERE, msg: null, ex);
500             }
501         } else
502             JOptionPane.showConfirmDialog( parentComponent: null, message: "Load A Plugin First !! ", title: "No plugins Found!",
503             JOptionPane.OK_CANCEL_OPTION, JOptionPane.ERROR_MESSAGE);
504     }
505 }
```

3. Undo: pop from stack and undo action

```
465     //Undo
466     1 usage
467     private void UndoButtonActionPerformed(java.awt.event.ActionEvent evt) {
468         Canvas.undo();
469         Canvas.redraw(PaintingCanvas.getGraphics());
470         UpdateList();
471         RedoButton.setEnabled(true);
472         if (Canvas.getUndoSize() == 0)
473             UndoButton.setEnabled(false);
474     }
```

4. Redo: pop from stack and redo action

```
475     //Redo
476     1 usage
477     private void RedoButtonActionPerformed(java.awt.event.ActionEvent evt) {
478         System.out.println(Canvas.getRedoSize());
479         Canvas.redo();
480         Canvas.redraw(PaintingCanvas.getGraphics());
481         UpdateList();
482         UndoButton.setEnabled(true);
483         if (Canvas.getRedoSize() == 0)
484             RedoButton.setEnabled(false);
485 }
```

5. Edit: doesn't change position but change only dimensions

```
333     //Edit
334     1 usage
335     private void EditButtonActionPerformed(java.awt.event.ActionEvent evt) {
336         if (Canvas.getShapes().length != 0){
337             Choice = 'E';
338             CurrentShape = Canvas.getShapes()[ShapesList.getSelectedIndex()];
339             prepareEditDialogBox();
340         }
341         else
342             JOptionPane.showConfirmDialog( parentComponent: null, message: "Please Create A New Shape First !!  ",
343                                         title: "No Shapes Found!", JOptionPane.OK_CANCEL_OPTION, JOptionPane.ERROR_MESSAGE);
343 }
```

6. Move: Take Direction and moving distance

```
344     //Move
345     1 usage
346     private void MoveButtonActionPerformed(java.awt.event.ActionEvent evt) {
347         if (Canvas.getShapes().length != 0){
348             CurrentShape = Canvas.getShapes()[ShapesList.getSelectedIndex()];
349             MoveDialog.setVisible(true);
350             MoveDialog.setSize( width: 290, height: 260);
351         }
352         else
353             JOptionPane.showConfirmDialog( parentComponent: null, message: "Create a New Shape First !!  ",
354                                         title: "No Shapes Found!", JOptionPane.OK_CANCEL_OPTION, JOptionPane.ERROR_MESSAGE);
354 }
```

7. Delete: pop Shape from stack and remove from paint area

```
451     private void DeleteButtonActionPerformed(java.awt.event.ActionEvent evt) {
452         if (Canvas.getShapes().length != 0){
453             Canvas.saveArray();
454             Canvas.deleteShape(Canvas.getShapes()[ShapesList.getSelectedIndex()]);
455             Canvas.redraw(PaintingCanvas.getGraphics());
456             UpdateList();
457             RedoButton.setEnabled(false);
458             UndoButton.setEnabled(true);
459         }
460         else
461             JOptionPane.showConfirmDialog( parentComponent: null, message: "Create A New Shape First !!  ",
462                                         title: "No Shapes Found!", JOptionPane.OK_CANCEL_OPTION, JOptionPane.ERROR_MESSAGE);
463 }
```

8. Copy: Make Clone From shapes & add new far away 50 from x and y axis

```
355     //Copy
356     1 usage
357     private void CopyButtonActionPerformed(java.awt.event.ActionEvent evt) {
358         if (Canvas.getShapes().length != 0){
359             CurrentShape = Canvas.getShapes()[ShapesList.getSelectedIndex()];
360             CurrentShapeProperties = CurrentShape.getProperties();
361             try {
362                 CurrentShape = (Shape) CurrentShape.clone();
363             } catch (CloneNotSupportedException ex) {
364                 Logger.getLogger(Main.class.getName()).log(Level.SEVERE, null, ex);
365             }
366             Canvas.saveArray();
367             Canvas.addShape(CurrentShape);
368             CurrentShape = Canvas.getShapes()[Canvas.getShapes().length-1];
369             MoveDownBy( x: 50);
370             MoveRightBy( x: 50);
371             UpdateList();
372             Canvas.redraw(PaintingCanvas.getGraphics());
373         }
374     else
375         JOptionPane.showConfirmDialog( parentComponent: null, message: "Create a New Shape First !!  "
376             , title: "No Shapes Found!", JOptionPane.OK_CANCEL_OPTION, JOptionPane.ERROR_MESSAGE);
377 }
```

9. Colorize: Change color of fill and outline by 5 types of color palette

```
381     //Colorize
382     1 usage
383     private void ColorizeButtonActionPerformed(java.awt.event.ActionEvent evt) {
384         if (Canvas.getShapes().length != 0){
385             CurrentShape = Canvas.getShapes()[ShapesList.getSelectedIndex()];
386             ColorsFrame.setVisible(true);
387             ColorsFrame.setSize( width: 663, height: 460);
388         }
389     else
390         JOptionPane.showConfirmDialog( parentComponent: null, message: "Please Create A New Shape First !!  "
391             , title: "No Shapes Found!", JOptionPane.OK_CANCEL_OPTION, JOptionPane.ERROR_MESSAGE);
391 }
```

10. Circle: take x, y position and radius

11. Square: take x, y position and side

12. Triangle: take position of 3 vertexes

13. Rectangle: take x, y position and width and height

14. Line Segment: take x, y for start and end point

15. Ellipse: take x, y position and width and height

Methods:

1. Main (Constructor)

```
53     // shape's labels and text
54     1 usage
55     public Main()
56     {
57         initComponents();
58         Canvas = (implementCanvas) PaintingCanvas;
59         TextFields[0] = XTextField;
60         TextFields[1] = YTextField;
61         TextFields[2] = PropertiesTextField1;
62         TextFields[3] = PropertiesTextField2;
63         TextFields[4] = PropertiesTextField3;
64         TextFields[5] = PropertiesTextField4;
65         Labels[0] = xLabel;
66         Labels[1] = yLabel;
67         Labels[2] = PropertiesLabel1;
68         Labels[3] = PropertiesLabel2;
69         Labels[4] = PropertiesLabel3;
70         Labels[5] = PropertiesLabel4;
```

2. CleanDialogBox : this methods unable the useless fields

```
72     //unable the useless fields
73     2 usages
74     private void CleanDialogBox()
75     {
76         Labels[0].setText("x:");
77         Labels[1].setText("y:");
78         for (int i = 0; i<2; i++)
79             TextFields[i].setText("");
80         for (int i = 2; i<6; i++)
81         {
82             TextFields[i].setText("");
83             TextFields[i].setEnabled(false);
84             Labels[i].setEnabled(false);
85             Labels[i].setText("N/A");
86         }
87     }
```

3. prepareAddDialogBox

```
88     //position (x,y)
89     7 usages
90     private void prepareAddDialogBox()
91     {
92         Choice = 'A';
93         CurrentShapeProperties = CurrentShape.getProperties();
94         CleanDialogBox();
95         for (int i = 0; i<2; i++)
96         {
97             TextFields[i].setEnabled(true);
98             Labels[i].setEnabled(true);
99         }
100        setPropertiesFields();
101        AddDialog.setVisible(true);
102    }
```

4. prepareEditDialogBox

```
103     //edit by size ( don't edit x,y)
104     1 usage
105     private void prepareEditDialogBox()
106     {
107         CurrentShapeProperties = CurrentShape.getProperties();
108         CleanDialogBox();
109         for (int i = 0; i<2; i++)
110         {
111             TextFields[i].setEnabled(false);
112             Labels[i].setEnabled(false);
113         }
114         setPropertiesFields();
115         AddDialog.setVisible(true);
116     }
```

5. setPropertiesFields

```
117     //set size of the tab of shapes
118     2 usages
119     private void setPropertiesFields()
120     {
121         Iterator<Map.Entry<String, Double>> Iterator = CurrentShapeProperties.entrySet().iterator();
122         for (int i = 2; Iterator.hasNext();i++)
123         {
124             Map.Entry<String, Double> Entry = Iterator.next();
125             if (Entry.getKey().equals("x2"))
126             {
127                 for (int j = 0; j<2; j++)
128                 {
129                     TextFields[j].setEnabled(true);
130                     Labels[j].setEnabled(true);
131                 }
132                 Labels[0].setText("x1: ");
133                 Labels[1].setText("y1: ");
134             }
135             Labels[i].setText(Entry.getKey() + ": ");
136             Labels[i].setEnabled(true); TextFields[i].setEnabled(true);
137             AddDialog.setSize( width: 410, (142+35*(i-2)));
138         }
139     }
```

6. readProperties

```
140     //check error
141     2 usages
142     private void readProperties()
143     {
144         Iterator<Map.Entry<String, Double>> Iterator = CurrentShapeProperties.entrySet().iterator();
145         for (int i = 2; Iterator.hasNext();i++) {
146             Map.Entry<String, Double> Entry = Iterator.next();
147             if (TextFields[i].getText().isEmpty() && Choice == 'A')
148                 JOptionPane.showConfirmDialog( parentComponent: null, message: "Fill All Fields" ,
149                                         title: "There's no Input ! ", JOptionPane.OK_CANCEL_OPTION, JOptionPane.WARNING_MESSAGE);
150             else if (!TextFields[i].getText().matches( regex: "\\\d*"))
151                 JOptionPane.showConfirmDialog( parentComponent: null, message: "\"" + TextFields[i].getText() + "\" Is Not A Number "
152                                         , title: "Invalid Input !!", JOptionPane.OK_CANCEL_OPTION, JOptionPane.QUESTION_MESSAGE);
153             else
154                 CurrentShapeProperties.put(Entry.getKey(),Double.parseDouble(TextFields[i].getText()));
155         }
156     }
```

7. addNewShape

```
221     // add new shape
222     1 usage
223     private void addNewShape(){
224         CurrentShape.setPosition(getNewPosition());
225         readProperties();
226         CurrentShape.setProperties(CurrentShapeProperties);
227         Canvas.addShape(CurrentShape);
228         UndoButton.setEnabled(true);
229         RedoButton.setEnabled(false);
230         AddDialog.setVisible(false);
231     }
```

8. getNewProperties

```
232     // Edit properties
233     1 usage
234     private Map<String, Double> getNewProperties(){
235         readProperties();
236         AddDialog.setVisible(false);
237         return CurrentShapeProperties;
238     }
```

9. UpdateList

```
239     //list Contains all Shapes
240     5 usages
241     private void UpdateList(){
242         ShapesList.removeAllItems();
243         Shape[] Shapes = Canvas.getShapes();
244         int counter = 0;
245         for (int i = 0; i<Shapes.length; i++) {
246             int currentShapeCounter = 1;
247             for (int j = 0; j<i; j++){
248                 if (Shapes[i].getClass().getSimpleName().equals(Shapes[j].getClass().getSimpleName()))
249                     currentShapeCounter++;
250             }
251             ShapesList.addItem(++counter + ". " + Shapes[i].getClass().getSimpleName() + " #" + currentShapeCounter);
252         }
253         ShapesList.setSelectedIndex(ShapesList.getItemCount()-1);
254         if (ShapesList.getItemCount() == 0)
255             ShapesList.addItem("Choose Shape");
256     }
```

10. EnterButtonActionPerformed

```
310     //Enter in properties
311     1 usage
312     private void EnterButtonActionPerformed(java.awt.event.ActionEvent evt) {
313         boolean flag = false;
314         for (JTextField TextField : TextFields)
315             if (!TextField.getText().matches( regex: "\\d*")){
316                 flag = true;
317                 JOptionPane.showConfirmDialog( parentComponent: null, message: "\\" + TextField.getText() + "\\" Is Not A Number "
318                                         , title: "Invalid Input !!", JOptionPane.OK_CANCEL_OPTION, JOptionPane.QUESTION_MESSAGE);
319                 break;
320             }
321             if (!flag){
322                 switch (Choice) {
323                     case 'A' -> {
324                         Canvas.saveArray();
325                         addNewShape();
326                         UpdateList();
327                     }
328                     case 'E' -> {
329                         CurrentShape.setPosition(getNewPosition());
330                         CurrentShape.setProperties(getNewProperties());
331                     }
332                 }
333             }
334     }
```

GUI Setup:

1. Properties panel

```
594     //Properties panel
595     AddDialog.setTitle("Please Enter Your Values:");
596     propertiesPanel.setPreferredSize(new java.awt.Dimension( width: 400, height: 150));
597     xLabel.setText("x: ");
598     yLabel.setText("y: ");
599     PropertiesLabel1.setText("jLabel4");
600     PropertiesLabel2.setText("jLabel4");
601     PropertiesLabel3.setText("jLabel4");
602     PropertiesLabel4.setText("jLabel4");
603
```

2 Draw, Load and plugin Panel

```
640     //Draw
641     drawButton.setText("Draw");
642     drawButton.setToolTipText("Draw Selected Shape");
643     drawButton.addActionListener(new java.awt.event.ActionListener() {
644         public void actionPerformed(java.awt.event.ActionEvent evt) { DrawButtonActionPerformed(evt); }
645     });
646
647     //Load
648     loadButton.setText("Load");
649     loadButton.setToolTipText("Load A New Shape");
650     loadButton.addActionListener(new java.awt.event.ActionListener() {
651         public void actionPerformed(java.awt.event.ActionEvent evt) { LoadButtonActionPerformed(evt); }
652     });
653
654     //Plugins
655     PluginsList.setModel(new javax.swing.DefaultComboBoxModel<>(new String[] { "Select Plugin" }));
656     PluginsList.setToolTipText("Choose A Loaded Plugin");
657     enterButton.setText("Enter");
658     enterButton.setToolTipText("Confirm Your Values");
659     enterButton.setNextFocusableComponent(XTextField);
660     enterButton.addActionListener(new java.awt.event.ActionListener() {
661         public void actionPerformed(java.awt.event.ActionEvent evt) { EnterButtonActionPerformed(evt); }
662     });
663     enterButton.addKeyListener(new java.awt.event.KeyAdapter() {
664         public void keyPressed(java.awt.event.KeyEvent evt) {
665             }
666     });
667     enterButton.addKeyListener(new java.awt.event.KeyAdapter() {
668         public void keyReleased(java.awt.event.KeyEvent evt) {
669             }
670     });
671     enterButton.addKeyListener(new java.awt.event.KeyAdapter() {
672         public void keyTyped(java.awt.event.KeyEvent evt) {
673             }
674     });
675
```

3 Add Dialog Layout Control

```
719     //Add Dialog Layout Control
720     javax.swing.GroupLayout AddDialogLayout = new javax.swing.GroupLayout(AddDialog.getContentPane());
721     AddDialog.getContentPane().setLayout(AddDialogLayout);
722     AddDialogLayout.setHorizontalGroup(AddDialogLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
723         .addGroup(AddDialogLayout.createSequentialGroup()
724             .addComponent(propertiesPanel, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
725             .addGap(10, 10, 10)
726             .addComponent(PluginsList, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
727             .addGap(10, 10, 10)
728             .addComponent(drawButton, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
729             .addGap(10, 10, 10)
730             .addComponent(loadButton, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
731             .addGap(10, 10, 10)
732             .addComponent(enterButton, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
733         )
734     );
735     AddDialogLayout.setVerticalGroup(AddDialogLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
736         .addGroup(AddDialogLayout.createSequentialGroup()
737             .addGroup(AddDialogLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
738                 .addComponent(propertiesPanel, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
739                 .addComponent(PluginsList, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
740                 .addComponent(drawButton, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
741                 .addComponent(loadButton, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
742                 .addComponent(enterButton, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
743             )
744             .addContainerGap())
745     );
746
```

4 Properties Panel Layout Control

```
675 //Properties Panel Layout Control
676 javax.swing.GroupLayout PropertiesPanelLayout = new javax.swing.GroupLayout(propertiesPanel);
677 propertiesPanel.setLayout(PropertiesPanelLayout);
678 PropertiesPanelLayout.setHorizontalGroup(PropertiesPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
679 .addGroup(PropertiesPanelLayout.createSequentialGroup())
680 .addGroup(min: 26, pref: 26).addGroup(PropertiesPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
681 .addComponent(PropertiesLabel1).addComponent(PropertiesLabel2).addComponent(PropertiesLabel3).addComponent(PropertiesLabel4)
682 .addComponent(yLabel, javax.swing.GroupLayout.PREFERRED_SIZE, pref: 17, javax.swing.GroupLayout.PREFERRED_SIZE)
683 .addComponent(xLabel, javax.swing.GroupLayout.PREFERRED_SIZE, pref: 17, javax.swing.GroupLayout.PREFERRED_SIZE))
684 .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
685 .addGroup(PropertiesPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING, resizable: false)
686 .addComponent(XTextField, javax.swing.GroupLayout.Alignment.LEADING, javax.swing.GroupLayout.DEFAULT_SIZE, pref: 157, Short.MAX_VALUE)
687 .addComponent(YTextField, javax.swing.GroupLayout.Alignment.LEADING)
688 .addComponent(PropertiesTextField1, javax.swing.GroupLayout.Alignment.LEADING)
689 .addComponent(PropertiesTextField2, javax.swing.GroupLayout.Alignment.LEADING)
690 .addComponent(PropertiesTextField3, javax.swing.GroupLayout.Alignment.LEADING)
691 .addComponent(PropertiesTextField4, javax.swing.GroupLayout.Alignment.LEADING)
692 .addGap(min: 28, pref: 28, max: 28).addComponent(enterButton, javax.swing.GroupLayout.PREFERRED_SIZE, pref: 96, javax.swing.GroupLayout.PREFERRED_SIZE)
693 .addContainerGap(pref: 40, Short.MAX_VALUE)));
694 PropertiesPanelLayout.setVerticalGroup(PropertiesPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
695 .addGroup(PropertiesPanelLayout.createSequentialGroup().addContainerGap())
696 .addGroup(PropertiesPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
697 .addComponent(XTextField, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
698 .addComponent(xLabel)).addGroup(PropertiesPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
699 .addGroup(PropertiesPanelLayout.createSequentialGroup().addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
700 .addGroup(PropertiesPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
701 .addComponent(YTextField, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
702 .addComponent(yLabel)).addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
703 .addGroup(PropertiesPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
704 .addComponent(PropertiesTextField1, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
705 .addComponent(PropertiesPanelLayout.createSequentialGroup()
706 .addGap(min: 11, pref: 11, max: 11).addComponent(enterButton, javax.swing.GroupLayout.PREFERRED_SIZE, pref: 51, javax.swing.GroupLayout.PREFERRED_SIZE)
707 .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
708 .addGroup(PropertiesPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
709 .addComponent(PropertiesTextField2, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
710 .addComponent(PropertiesLabel2))
711 .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
712 .addGroup(PropertiesPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
713 .addComponent(PropertiesTextField3, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
714 .addComponent(PropertiesLabel3)).addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
715 .addGroup(PropertiesPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
716 .addComponent(PropertiesTextField4, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
717 .addComponent(PropertiesLabel4)).addGap(min: 0, pref: 0, Short.MAX_VALUE)));
718
```

5 Color Panel

```
730 //Color Panel
731 ColorsFrame.setTitle("Coloring");
732 ColorsFrame.setResizable(false);
733 OKButton.setText("Ok");
734 OKButton.addActionListener(new java.awt.event.ActionListener() {
735     public void actionPerformed(java.awt.event.ActionEvent evt) {
736         OKButtonActionPerformed(evt);
737     }
738 });
739 cancelButton.setText("Cancel");
740 cancelButton.addActionListener(new java.awt.event.ActionListener() {
741     public void actionPerformed(java.awt.event.ActionEvent evt) {
742         CancelButtonActionPerformed(evt);
743     }
744 });
745 OutlineCheckBox.setText("Outline Color");
746 FillCheckBox.setText("Fill Color");
747 applyButton.setText("Apply");
748 applyButton.addActionListener(new java.awt.event.ActionListener() {
749     public void actionPerformed(java.awt.event.ActionEvent evt) {
750         ApplyButtonActionPerformed(evt);
751     }
752 });
```

6. Colors Frame Layout Control

```
754 //Colors Frame Layout Control
755 javax.swing.GroupLayout ColorsFrameLayout = new javax.swing.GroupLayout(ColorsFrame.getContentPane());
756 ColorsFrame.getContentPane().setLayout(ColorsFrameLayout);
757 ColorsFrameLayout.setHorizontalGroup(ColorsFrameLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
758 .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, ColorsFrameLayout.createSequentialGroup()
759 .addComponent(ColorChooser, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
760 .addGroupGap( min: 0, pref: 0, Short.MAX_VALUE)).addGroup(ColorsFrameLayout.createSequentialGroup()
761 .addComponent(OKButton, javax.swing.GroupLayout.PREFERRED_SIZE, pref: 98, javax.swing.GroupLayout.PREFERRED_SIZE)
762 .addComponentPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
763 .addComponent(cancelButton, javax.swing.GroupLayout.PREFERRED_SIZE, pref: 98, javax.swing.GroupLayout.PREFERRED_SIZE)
764 .addComponentPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
765 .addComponent(applyButton, javax.swing.GroupLayout.PREFERRED_SIZE, pref: 98, javax.swing.GroupLayout.PREFERRED_SIZE)
766 .addComponentPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
767 .addComponent(OutlineCheckBox).addComponentPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
768 .addComponent(FillCheckBox).addGroupGap( min: 73, pref: 73, max: 73)));
769 ColorsFrameLayout.setVerticalGroup(ColorsFrameLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
770 .addGroup(ColorsFrameLayout.createSequentialGroup()
771 .addComponent(ColorChooser, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
772 .addComponentPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
773 .addComponent(ColorsFrameLayout.createSequentialGroup()
774 .addComponent(OKButton)
775 .addComponent(ColorsFrameLayout.createSequentialGroup()
776 .addComponent(cancelButton)
777 .addComponent(applyButton)).addComponent(ColorsFrameLayout.createSequentialGroup()
778 .addComponent(FillCheckBox))).addComponentGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)));
```

7. Move Panel

```
778 //Move Panel
779 moveRightButton.setText("→");
780 moveRightButton.addActionListener(new java.awt.event.ActionListener() {
781     public void actionPerformed(java.awt.event.ActionEvent evt) { MoveRightButtonActionPerformed(evt); }
782 });
783 MoveByTextField.setFocusTraversalPolicyProvider(true);
784 MoveByTextField.setNextFocusableComponent(doneButton);
785 moveUpButton.setText("▲");
786 moveUpButton.addActionListener(new java.awt.event.ActionListener() {
787     public void actionPerformed(java.awt.event.ActionEvent evt) { MoveUpButtonActionPerformed(evt); }
788 });
789 moveUpButton.setToolTipText("Move Up");
790 moveUpButton.addActionListener(new java.awt.event.ActionListener() {
791     public void actionPerformed(java.awt.event.ActionEvent evt) { MoveUpButtonActionPerformed(evt); }
792 });
793 moveLeftButton.setText("◀");
794 moveLeftButton.addActionListener(new java.awt.event.ActionListener() {
795     public void actionPerformed(java.awt.event.ActionEvent evt) { MoveLeftButtonActionPerformed(evt); }
796 });
797 moveDownButton.setText("▼");
798 moveDownButton.addActionListener(new java.awt.event.ActionListener() {
799     public void actionPerformed(java.awt.event.ActionEvent evt) { MoveDownButtonActionPerformed(evt); }
800 });
801 moveByLabel.setText("Move By:");
802 doneButton.setText("Done");
803 doneButton.setFocusTraversalPolicyProvider(true);
804 doneButton.setNextFocusableComponent(MoveByTextField);
805 doneButton.addActionListener(new java.awt.event.ActionListener() {
806     public void actionPerformed(java.awt.event.ActionEvent evt) { DoneButtonActionPerformed(evt); }
807 });
808 doneButton.setToolTipText("Move By");
```

8. Move Dialog Layout Control

```
815 //Move Dialog Layout Control
816 javax.swing.GroupLayout MoveDialogLayout = new javax.swing.GroupLayout(MoveDialog.getContentPane());
817 MoveDialog.getContentPane().setLayout(MoveDialogLayout);
818 MoveDialogLayout.setHorizontalGroup(MoveDialogLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
819 .addGroup(MoveDialogLayout.createSequentialGroup().addGap(13, 13, 13)
820 .addComponent(moveLeftButton, javax.swing.GroupLayout.PREFERRED_SIZE, pref: 60, javax.swing.GroupLayout.PREFERRED_SIZE)
821 .addComponent(preferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
822 .addGroup(MoveDialogLayout.createSequentialGroup().addGap(13, 13, 13)
823 .addComponent(moveUpButton, javax.swing.GroupLayout.PREFERRED_SIZE, pref: 60, javax.swing.GroupLayout.PREFERRED_SIZE)
824 .addComponent(moveDownButton, javax.swing.GroupLayout.PREFERRED_SIZE, pref: 60, javax.swing.GroupLayout.PREFERRED_SIZE)
825 .addGroup(MoveDialogLayout.createSequentialGroup().addGap(13, 13, 13)
826 .addGroup(MoveDialogLayout.createSequentialGroup().addGap(13, 13, 13)
827 .addComponent(moveByTextField, javax.swing.GroupLayout.PREFERRED_SIZE, pref: 60, javax.swing.GroupLayout.PREFERRED_SIZE)
828 .addComponent(moveByLabel).addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
829 .addComponent(moveRightButton, javax.swing.GroupLayout.PREFERRED_SIZE, pref: 60, javax.swing.GroupLayout.PREFERRED_SIZE)))
830 .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
831 .addComponent(doneButton).addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE));
832 MoveDialogLayout.setVerticalGroup(MoveDialogLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
833 .addGroup(MoveDialogLayout.createSequentialGroup().addGap(13, 13, 13)
834 .addGroup(MoveDialogLayout.createSequentialGroup().addGap(13, 13, 13)
835 .addComponent(doneButton, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
836 .addGroup(MoveDialogLayout.createSequentialGroup().addGap(13, 13, 13)
837 .addComponent(moveUpButton, javax.swing.GroupLayout.PREFERRED_SIZE, pref: 60, javax.swing.GroupLayout.PREFERRED_SIZE)
838 .addComponent(preferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
839 .addGroup(MoveDialogLayout.createSequentialGroup().addGap(13, 13, 13)
840 .addGroup(MoveDialogLayout.createSequentialGroup().addGap(13, 13, 13)
841 .addComponent(moveRightButton, javax.swing.GroupLayout.PREFERRED_SIZE, pref: 60, javax.swing.GroupLayout.PREFERRED_SIZE)
842 .addComponent(moveLeftButton, javax.swing.GroupLayout.PREFERRED_SIZE, pref: 60, javax.swing.GroupLayout.PREFERRED_SIZE)
843 .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, MoveDialogLayout.createSequentialGroup().addGap(13, 13, 13)
844 .addComponent(moveByTextField, javax.swing.GroupLayout.PREFERRED_SIZE, pref: 34, javax.swing.GroupLayout.PREFERRED_SIZE))
845 .addComponent(preferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
846 .addComponent(moveDownButton, javax.swing.GroupLayout.PREFERRED_SIZE, pref: 60, javax.swing.GroupLayout.PREFERRED_SIZE)))
847 .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)));
848
849
```

9. Main Panel

```
850 //Main Panel
851 setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
852 setTitle("Vector Based Drawing Application - FCDS - 2022 - RSAA");
853 setLocationByPlatform(true);
854 setMinimumSize(new java.awt.Dimension( width: 800, height: 400));
855 setSize(new java.awt.Dimension( width: 800, height: 400));
856 PaintingCanvas.setBackground(new java.awt.Color( r: 7, g: 56, b: 86));
857 PaintingCanvas.setBorder(javax.swing.BorderFactory.createLineBorder(new java.awt.Color( r: 219, g: 227, b: 7)));
858 PaintingCanvas.setForeground(new java.awt.Color( r: 121, g: 9, b: 84));
859 javax.swing.GroupLayout PaintingCanvasLayout = new javax.swing.GroupLayout(PaintingCanvas);
860 PaintingCanvas.setLayout(PaintingCanvasLayout);
861 //Weight
862 PaintingCanvasLayout.setHorizontalGroup(PaintingCanvasLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING).addGap(
863 //Height
864 PaintingCanvasLayout.setVerticalGroup(PaintingCanvasLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING).addGap(
```

10. Buttons Panel

```
866 //Buttons Panel
867 moveButton.setText("Move");
868 moveButton.setToolTipText("Adjust Selected Shape's Position");
869 moveButton.addActionListener(new java.awt.event.ActionListener() {
870     public void actionPerformed(java.awt.event.ActionEvent evt) { MoveButtonActionPerformed(evt); }
871 });
872
873 UndoButton.setText("Undo");
874 UndoButton.setToolTipText("Undo Last Action");
875 UndoButton.setEnabled(false);
876 UndoButton.addActionListener(new java.awt.event.ActionListener() {
877     public void actionPerformed(java.awt.event.ActionEvent evt) { UndoButtonActionPerformed(evt); }
878 });
879
880 RedoButton.setText("Redo");
881 RedoButton.setToolTipText("Redo Last Undo");
882 RedoButton.setEnabled(false);
883 RedoButton.addActionListener(new java.awt.event.ActionListener() {
884     public void actionPerformed(java.awt.event.ActionEvent evt) { RedoButtonActionPerformed(evt); }
885 });
886
887 copyButton.setText("Copy");
888 copyButton.setToolTipText("Clone Selected Shape");
889 copyButton.addActionListener(new java.awt.event.ActionListener() {
890     public void actionPerformed(java.awt.event.ActionEvent evt) { CopyButtonActionPerformed(evt); }
891 });
892
893 colorizeButton.setText("Colorize");
894 colorizeButton.setToolTipText("Colorize selected Shape");
895 colorizeButton.addActionListener(new java.awt.event.ActionListener() {
896     public void actionPerformed(java.awt.event.ActionEvent evt) { ColorizeButtonActionPerformed(evt); }
897 });
898
899 deleteButton.setText("Delete");
900 deleteButton.setToolTipText("Remove Selected Shape");
901 deleteButton.addActionListener(new java.awt.event.ActionListener() {
902     public void actionPerformed(java.awt.event.ActionEvent evt) { DeleteButtonActionPerformed(evt); }
903 });
904
905 //Shapes button
906 ShapesList.setModel(new javax.swing.DefaultComboBoxModel<>(new String[] { "Choose Shape" }));
907 ShapesList.setToolTipText("Choose A Drawn shape");
908 editButton.setText("Edit");
909 editButton.setToolTipText("Adjust Selected Shape's Properties");
910 editButton.addActionListener(new java.awt.event.ActionListener() {
911     public void actionPerformed(java.awt.event.ActionEvent evt) { EditButtonActionPerformed(evt); }
912 });
913
914 circleButton.setText(" Circle ");
915 circleButton.setToolTipText("Create A New Circle");
916 circleButton.addActionListener(new java.awt.event.ActionListener() {
917     public void actionPerformed(java.awt.event.ActionEvent evt) { CircleButtonActionPerformed(evt); }
918 });
919
920 lineSegmentButton.setText(" Line Segment ");
921 lineSegmentButton.setToolTipText("Create A New Line Segment");
922 lineSegmentButton.addActionListener(new java.awt.event.ActionListener() {
923     public void actionPerformed(java.awt.event.ActionEvent evt) { LineSegmentButtonActionPerformed(evt); }
924 });
925
926 squareButton.setText(" Square ");
927 squareButton.setToolTipText("Create A New Square");
928 squareButton.addActionListener(new java.awt.event.ActionListener() {
929     public void actionPerformed(java.awt.event.ActionEvent evt) { SquareButtonActionPerformed(evt); }
930 });
931
932 ellipseButton.setText(" Ellipse ");
933 ellipseButton.setToolTipText("Create A New Ellipse");
934 ellipseButton.addActionListener(new java.awt.event.ActionListener() {
935     public void actionPerformed(java.awt.event.ActionEvent evt) { EllipseButtonActionPerformed(evt); }
936 });
937
938 triangleButton.setText(" Triangle ");
939 triangleButton.setToolTipText("Create A New Triangle");
940 triangleButton.addActionListener(new java.awt.event.ActionListener() {
941     public void actionPerformed(java.awt.event.ActionEvent evt) { TriangleButtonActionPerformed(evt); }
942 });
943
944 rectangleButton.setText(" Rectangle ");
945 rectangleButton.setToolTipText("Create A New Rectangle");
946 rectangleButton.addActionListener(new java.awt.event.ActionListener() {
947     public void actionPerformed(java.awt.event.ActionEvent evt) { RectangleButtonActionPerformed(evt); }
948 });
949
950 selectShapeLabel.setText("Select Shape");
951 selectShapeLabel.setToolTipText("Why Are You Pointing Here :)");
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
```

11. Buttons Layout Control

```
979 //Buttons Layout Control
980 javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
981 getContentPane().setLayout(layout);
982 layout.setHorizontalGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
983 .addGroup(layout.createSequentialGroup()
984 .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
985 .addComponent(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
986 .addComponent(ShapesList, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
987 .addGroup(layout.createSequentialGroup()
988 .addComponent(editButton, javax.swing.GroupLayout.PREFERRED_SIZE, pref: 150, javax.swing.GroupLayout.PREFERRED_SIZE)
989 .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
990 .addComponent(moveButton, javax.swing.GroupLayout.PREFERRED_SIZE, pref: 150, javax.swing.GroupLayout.PREFERRED_SIZE))
991 .addGroup(layout.createSequentialGroup()
992 .addComponent(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
993 .addComponent(copyButton, javax.swing.GroupLayout.PREFERRED_SIZE, pref: 150, javax.swing.GroupLayout.PREFERRED_SIZE)
994 .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
995 .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
996 .addComponent(deleteButton, javax.swing.GroupLayout.PREFERRED_SIZE, pref: 150, javax.swing.GroupLayout.PREFERRED_SIZE)
997 .addComponent(colorizeButton, javax.swing.GroupLayout.PREFERRED_SIZE, pref: 150, javax.swing.GroupLayout.PREFERRED_SIZE)))
998 .addComponent(selectShapeLabel).addGroup(layout.createSequentialGroup()
999 .addComponent(UndoButton, javax.swing.GroupLayout.PREFERRED_SIZE, pref: 150, javax.swing.GroupLayout.PREFERRED_SIZE)
1000 .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
1001 .addComponent(RedoButton, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
1002 .addComponent(PluginsList, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
1003 .addComponent(jSeparator1)
1004 .addGroup(layout.createSequentialGroup()
1005 .addComponent(loadButton, javax.swing.GroupLayout.PREFERRED_SIZE, pref: 150, javax.swing.GroupLayout.PREFERRED_SIZE)
1006 .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
1007 .addComponent(drawButton, javax.swing.GroupLayout.PREFERRED_SIZE, pref: 150, javax.swing.GroupLayout.PREFERRED_SIZE))
1008 .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
1009 .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
1010 .addComponent(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
1011 .addComponent(circleButton)
1012 .addComponent(preferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED))
1013 .addComponent(squareButton).addComponent(preferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED))
1014 .addComponent(ellipseButton).addComponent(preferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED))
1015 .addComponent(triangleButton).addComponent(preferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED).addComponent(rectangleButton))
1016 .addComponent(PaintingCanvas, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
1017 .addComponent(containerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)));
1018 layout.setVerticalGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
1019 .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()
1020 .addComponent(containerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
1021 .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
1022 .addComponent(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
1023 .addComponent(squareButton).addComponent(ellipseButton).addComponent(triangleButton)
1024 .addComponent(rectangleButton)).addComponent(gap(min: 18, pref: 18, max: 18))
1025 .addGroup(layout.createSequentialGroup()
1026 .addComponent(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
1027 .addComponent(loadButton, javax.swing.GroupLayout.PREFERRED_SIZE, pref: 30, javax.swing.GroupLayout.PREFERRED_SIZE)
1028 .addComponent(drawButton, javax.swing.GroupLayout.PREFERRED_SIZE, pref: 30, javax.swing.GroupLayout.PREFERRED_SIZE))
1029 .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
1030 .addComponent(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
1031 .addComponent(PluginsList, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
1032 .addComponent(preferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED))
1033 .addComponent(jSeparator1, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
1034 .addComponent(preferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED))
1035 .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
1036 .addComponent(UndoButton, javax.swing.GroupLayout.PREFERRED_SIZE, pref: 30, javax.swing.GroupLayout.PREFERRED_SIZE)
1037 .addComponent(RedoButton, javax.swing.GroupLayout.PREFERRED_SIZE, pref: 30, javax.swing.GroupLayout.PREFERRED_SIZE))
1038 .addComponent(preferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED).addComponent(selectShapeLabel))
1039 .addComponent(preferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED))
1040 .addComponent(ShapesList, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
1041 .addComponent(preferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED))
1042 .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
1043 .addComponent(editButton, javax.swing.GroupLayout.PREFERRED_SIZE, pref: 50, javax.swing.GroupLayout.PREFERRED_SIZE)
1044 .addComponent(moveButton, javax.swing.GroupLayout.PREFERRED_SIZE, pref: 50, javax.swing.GroupLayout.PREFERRED_SIZE))
1045 .addComponent(preferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED))
1046 .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
1047 .addComponent(copyButton, javax.swing.GroupLayout.PREFERRED_SIZE, pref: 50, javax.swing.GroupLayout.PREFERRED_SIZE)
1048 .addComponent(colorizeButton, javax.swing.GroupLayout.PREFERRED_SIZE, pref: 50, javax.swing.GroupLayout.PREFERRED_SIZE))
1049 .addComponent(preferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED))
1050 .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
1051 .addComponent(deleteButton, javax.swing.GroupLayout.PREFERRED_SIZE, pref: 50, javax.swing.GroupLayout.PREFERRED_SIZE)
1052 .addComponent(PaintingCanvas, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
1053 .addComponent(pack()));
1054 }
```

TASK3: Dynamic Application Extensions and plug-ins

Class of Plugins:

- Diamond

```
import java.awt.*;
import java.util.Iterator;
import java.util.Map;

public class Diamond extends Shape2 implements Shape{

    public Diamond() {
        super();
        this.properties.put("h", 0.0);
        this.properties.put("w", 0.0);
    }

    @Override
    public void draw(Graphics canvas) {
        int x=(int)position.getX();
        int y=(int)position.getY();
        int w=properties.get("w").intValue();
        int h=properties.get("h").intValue();
        int [] xPoints={x+(w/2),x+w,x+(w/2),x};
        int [] yPoints={y,y+(h/2),y+h,y+(h/2)};
        canvas.setColor(this.color);
        canvas.drawPolygon(xPoints, yPoints, 4);
        canvas.setColor(this.fillColor);
        canvas.fillPolygon(xPoints, yPoints, 4);
    }

    @Override
    public Object clone() throws CloneNotSupportedException{
        Shape2 Diamond = new Diamond ();
        Iterator<Map.Entry<String, Double>> Iterator =
properties.entrySet().iterator();
        while(Iterator.hasNext()) {
            Map.Entry<String, Double> Entry = Iterator.next();
            Diamond .properties.put(Entry.getKey(),Entry.getValue());
        }
        Diamond .setPosition(this.getPosition());
        Diamond .setColor(this.getColor());
        Diamond .setFillColor(this.getFillColor());
        return Diamond ;
    }

}
```

**THANK
YOU**

