|  |
| --- |
| import javax.swing.event.\*;  import javax.swing.\*;  import java.awt.\*;  import java.awt.event.\*;  import javax.swing.border.\*;  import java.util.\*;  public class GUI implements ActionListener, MouseListener, MouseMotionListener  {  JFrame f;    public static int x1, y1, x2, y2;    boolean dragged;  String drawing;    JPanel mainPanel, paintPanel, btnPanel;  JButton btnLine, btnRect, btnClear;    PaintClass paint;  Shape shape;    ArrayList<Shape> shapeArray;    public GUI()  {  f = new JFrame("PAINT");    dragged = false;  drawing = "line";    paint = new PaintClass();    btnPanel = new JPanel(new FlowLayout());    btnLine = new JButton("LINE");  btnRect = new JButton("RECTANGLE");  btnClear = new JButton("CLEAR");    shapeArray = new ArrayList<Shape>();    mainPanel = new JPanel(new BorderLayout());  mainPanel.setPreferredSize(new Dimension(400, 400));  mainPanel.setBorder(new EtchedBorder(EtchedBorder.LOWERED));    paint.setBorder(new SoftBevelBorder(SoftBevelBorder.LOWERED));    paint.addMouseListener(this);  paint.addMouseMotionListener(this);    btnLine.addActionListener(this);  btnRect.addActionListener(this);  btnClear.addActionListener(this);    btnPanel.add(btnLine);  btnPanel.add(btnRect);  btnPanel.add(btnClear);    mainPanel.add(paint, BorderLayout.CENTER);  mainPanel.add(btnPanel, BorderLayout.SOUTH);    f.add(mainPanel);    f.pack();  f.setLocationRelativeTo(null);  f.setVisible(true);    }    public void actionPerformed(ActionEvent e)  {    //Draw a line on the screen  if(e.getSource() == btnLine)  drawing = "Line";    //Draw a rectangle on the screen  else if(e.getSource() == btnRect)  drawing = "Rectangle";    //Reset all the previous and current shapes on the paint screen  else if(e.getSource() == btnClear)  {    shapeArray.clear();    x1 = 0; x2 = 0; y1 = 0; y2 = 0;  paint.repaint();  }  }    /\*\*  \* MouseListener and MouseMotionListeners that are not used  \*/  public void mouseClicked(MouseEvent e){}  public void mouseEntered(MouseEvent e){}  public void mouseExited(MouseEvent e){}  public void mouseMoved(MouseEvent e){}    /\*\*  \* Purpose : create a new Shape and store it in the shapeArray  \*/  public void mouseReleased(MouseEvent e)  {  if(dragged)  {    shape = new Shape(x1, x2, y1, y2, drawing);  shapeArray.add(shape);  }  }    /\*\*  \* Purpose : Get the starting and final x and y coordinates of  \* the mouse when the mouse button is pressed  \*/  public void mousePressed(MouseEvent e)  {  x1 = e.getX();  y1 = e.getY();  dragged = false;  }    /\*\*  \* Purpose : Get the final mouse coordinate with the mouse is  \* dragged while pressing  \*/  public void mouseDragged(MouseEvent e)  {  //Store the x and y coordinates of the mouse in the pointB array  x2 = e.getX();  y2 = e.getY();    dragged = true;    paint.repaint();  }    //Main  public static void main(String[] args)  {  new GUI();  }    //PAINT CLASS -- INTERNAL CLASS --> INHERITANCE  class PaintClass extends JPanel  {  int width, height, x, y;    public void paintComponent(Graphics g)  {  super.paintComponent(g);    //Set the background to white color  g.setColor(Color.WHITE);  g.fillRect(0, 0, getWidth(), getHeight());    g.setColor(Color.BLACK);    //Draw all the previous shapes  for(Shape x: shapeArray)  {    //If the Shape is line, draw the line  if(x.get\_shape().equalsIgnoreCase("line"))  {  line(g, x.get\_x1(), x.get\_y1(), x.get\_x2(), x.get\_y2());  }    //If the Shape is Rectangle, draw rectangle  else  {  rectangle(g, x.get\_x1(), x.get\_y1(), x.get\_x2(), x.get\_y2());  }    }    //Draw the current Line  if(drawing.equalsIgnoreCase("Line"))  {  line(g, x1, y1, x2, y2);  }    //Draw the current Rectangle  else if(drawing.equalsIgnoreCase("Rectangle"))  {  rectangle(g, x1, y1, x2, y2);  }    }//end paintComponent(Graphics)    //Draw Lines  public void line(Graphics g, int x1, int y1, int x2, int y2)  {  g.setColor(Color.RED);  g.drawLine(x1, y1, x2, y2);    }//end line(Graphics, int, int, int, int)    //Draw rectangles  public void rectangle(Graphics g, int x1, int y1, int x2, int y2)  {  width = Math.abs(x2 - x1);  height = Math.abs(y2 - y1);    x = Math.min(x2, x1);  y = Math.min(y2, y1);    g.setColor(Color.BLUE);  g.fillRect(x, y, width, height);    }//end rectangle(Graphics, int, int, int, int)    }    }  /\*\*  \* Auto Generated Java Class.  \*/  public class Shape {    private int x1, y1, x2, y2;  private String shape;    public Shape()  {    }    public Shape(int x1, int x2, int y1, int y2, String shape)  {  this.x1 = x1;  this.x2 = x2;  this.y1 = y1;  this.y2 = y2;  this.shape = shape;  }    //Get the x1 coordinate of the shape  public int get\_x1()  {  return this.x1;  }    //Get the x2 coordinate of the shape  public int get\_x2()  {  return this.x2;  }    //Get the y1 coordinate of the shape  public int get\_y1()  {  return this.y1;  }    //Get the y2 coordinate of the shape  public int get\_y2()  {  return this.y2;  }    public String get\_shape()  {  return this.shape;  }    public String toString()  {  String xString = "x2", yString = "y2";    if(!this.shape.equalsIgnoreCase("line"))  {  xString = "width";  yString = "height";  }    return "Dimensions of " + this.shape + " are : x1 = " + this.x1 + " , y1 = " + this.y1  + " , "+ xString + " = " + this.x2 + ", "+ yString + " = " + this.y2;  }  } |
| Screenshots: |