# 南开大学

# JAVA 语言与应用

# 画图程序实验报告

姓 名：冯朝芃

学 号：2012039

年 级： 2020 级

学 院： 计算机学院

专 业 ：计算机科学与技术

授课教师：刘嘉欣

完成日期：2021 年 11 月 28 日

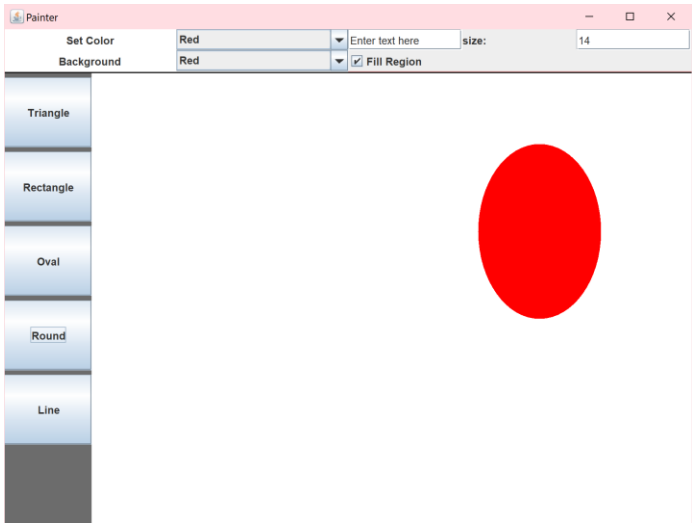
一、概述：

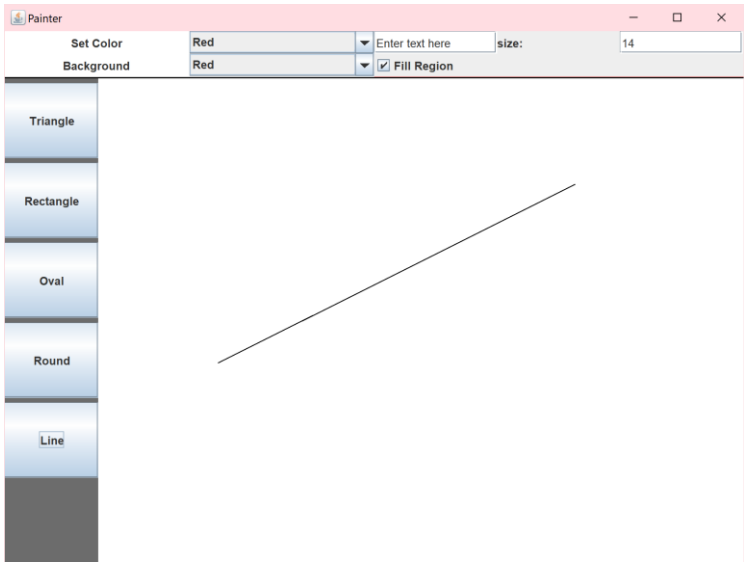
本作业为基于 Swing 的画图程序。本作业实现的功能有：画出矩形、三角形、椭圆形、直线，选择是否填充等功能。本代码使用了 MVC 架构进行开发，充分利用了 Java 的面向对象编程思想，一定程度上具有可扩展性高，代码逻辑框架清晰，复用性强、事件处理流程明确等特点。

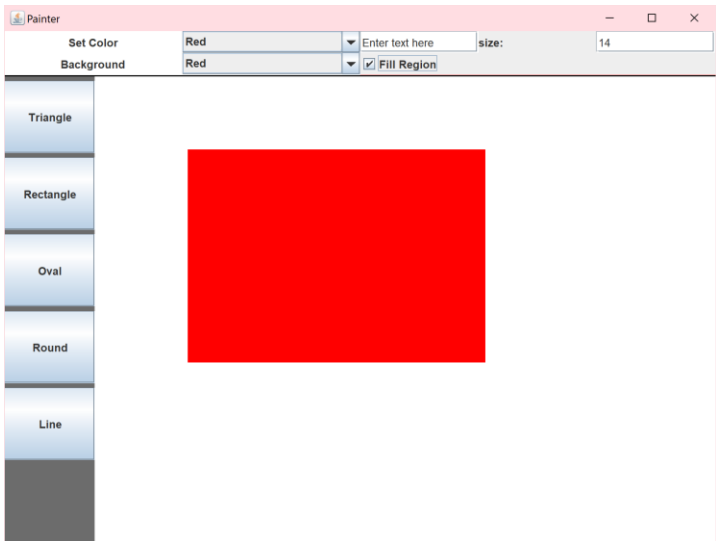
由于作者水平和精力有限，本程序保存作图、撤销操作等功能还在完善中。

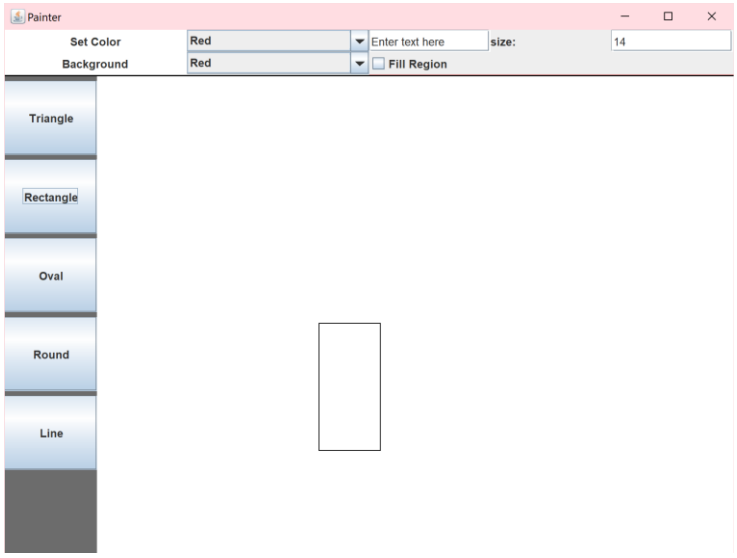
二、运行展示：

运行效果截图：




Painter
Set Color
Red
Enter text here
size:
14
Background
Red
Fill Region
Triangle
Rectangle
Oval
Round
Line


Painter
Set Color
Red
Enter text here
size:
14
Background
Red
Fill Region
Triangle
Rectangle
Oval
Round
Line

附录：完整代码

```
==========================================================
『Manage.java』
==========================================================
package manager;

import viewer.Viewer;

import javax.swing.*;

public class Manage {

    private JFrame mainWindow;
    private Viewer viewer;

    void init(){
        mainWindow = new JFrame();
        viewer = new Viewer(mainWindow);
        viewer.init();
        viewer.show();


        while(true){
            //viewer.drawPanel.drawLine(200,200,100,100);
            viewer.update();
        }
    }

    public static void main(String[] args){
        Manage manage = new Manage();
        manage.init();

    }

}

==========================================================
『store.java』
==========================================================
package storer;

import java.util.Queue;
```

```
public class store {
    public Queue<String> queue;
}

==========================================
 『background.java』
==========================================
package viewer;

import javax.swing.*;
import java.awt.*;

public class background extends JPanel {
    background(){
        setLayout(new GridLayout(1,2));
        setBackground(new java.awt.Color(255, 255, 255));
        setPreferredSize(new java.awt.Dimension(50, 50));

        JLabel label = new JLabel("Background");
        label.setHorizontalAlignment(JLabel.CENTER);
        add(label);

        JComboBox comboBox = new JComboBox();
        comboBox.addItem("Red");
        comboBox.addItem("Green");
        comboBox.addItem("Blue");
        add(comboBox);
    }
}

==========================================
 『buttonSet.java』
==========================================
package viewer;

import javax.swing.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.util.Vector;


class triangularButton extends JButton {
    triangularButton() {
        super("Triangle");
```

```
        setPreferredSize(new java.awt.Dimension(100, 80));
        setContentAreaFilled(true);
        setVisible(true);
        //addActionListener(new Actiona);
    }
}

class rectangularButton extends JButton {
    rectangularButton(){
        super("Rectangle");
        setPreferredSize(new java.awt.Dimension(100, 80));
        setContentAreaFilled(true);
        setVisible(true);
    }
}

class ovalButton extends JButton {
    ovalButton(){
        super("Oval");
        setPreferredSize(new java.awt.Dimension(100, 80));
        setContentAreaFilled(true);
        setVisible(true);
    }
}

class roundButton extends JButton {
    roundButton(){
        super("Round");
        setPreferredSize(new java.awt.Dimension(100, 80));
        setContentAreaFilled(true);
        setVisible(true);
    }
}

class lineButton extends JButton {
    lineButton(){
        super("Line");
        setPreferredSize(new java.awt.Dimension(100, 80));
        setContentAreaFilled(true);
        setVisible(true);
    }
}
```

```java
public class buttonSet extends JButton {
    public static Vector<JButton> buttons;
    public static JButton selected=new lineButton();

    class buttonListener implements ActionListener {
        @Override
        public void actionPerformed(ActionEvent e) {
            selected = (JButton) e.getSource();
            //System.out.println("Selected: " + selected.getText());
        }
    }

    public buttonSet() {
        buttons = new Vector<JButton>();
        buttons.add(new triangularButton());
        buttons.add(new rectangularButton());
        buttons.add(new ovalButton());
        buttons.add(new roundButton());
        buttons.add(new lineButton());

        for (JButton b : buttons) {
            b.addActionListener(new buttonListener());
        }
    }
}
```

==========

『drawPanel.java』

==========

```java
package viewer;

import javax.swing.*;
import java.awt.*;
import java.awt.event.MouseListener;
import java.awt.geom.*;

import static java.lang.Math.abs;


public class drawPanel extends JPanel {
    public int x,x1,x2;
```

```
    public int y,y1,y2;
    private int width = 0;
    private int height = 0;
    private Graphics g;

    // Constructor
    public drawPanel() {
        super();
        setPreferredSize(new java.awt.Dimension(500, 500));
        setBackground(java.awt.Color.white);
        //setVisible(true);

        addMouseListener(new MouseListener() {
            @Override
            public void mouseClicked(java.awt.event.MouseEvent e) {
                x = e.getX();
                y = e.getY();
                //repaint();
            }

            @Override
            public void mousePressed(java.awt.event.MouseEvent e) {
                x1 = e.getX();
                y1 = e.getY();
                repaint();
            }

            @Override
            public void mouseReleased(java.awt.event.MouseEvent e) {
                x2 = e.getX();
                y2 = e.getY();
                repaint();
                //x1 = 0;
                //x2 = 0;
                //y1 = 0;
                //y2 = 0;
            }

            @Override
            public void mouseEntered(java.awt.event.MouseEvent e) {
//                x1 = e.getX();
//                y1 = e.getY();
//                x2 = e.getX();
//                y2 = e.getY();
                repaint();
```

```
            }

            @Override
            public void mouseExited(java.awt.event.MouseEvent e) {
                x = e.getX();
                y = e.getY();
                //repaint();
            }
        }
        );
    }

    @Override
    public void paint(Graphics g) {
        super.paint(g);
        //g.drawLine(200, 200, 400, 400);
    }

    @Override
    public void repaint() {
        super.repaint();
        //drawLine(x1, y1, x2, y2);
    }

    //draw a line on the panel with the given coordinates
    public void drawLine(int x1, int y1, int x2, int y2) {
        Graphics g = getGraphics();
        g.drawLine(x1, y1, x2, y2);
    }


    public void drawRectangle(int x1, int y1, int x2, int y2,boolean fill,
Color color) {
        Graphics g = getGraphics();
        if(fill) {
            g.setColor(color);
            g.fillRect(x1, y1, abs(x1-x2), abs(y1-y2));
        }else {
            g.drawRect(x1, y1, abs(x1 - x2), abs(y1 - y2));
        }

    }

    public void drawCircle(int x1, int y1, int x2, int y2,boolean fill,
Color color) {
```

```
        Graphics g = getGraphics();
        if(fill){
            g.setColor(color);
            g.fillOval(x1, y1, abs(x1-x2), abs(y1-y2));
        }else {
            g.drawOval(x1, y1, abs(x1-x2), abs(y1-y2));
        }
    }

    public void drawTriangle(int x1, int y1, int x2, int y2,boolean fill,
Color color) {
        Graphics g = getGraphics();
        int[] xPoints = {x1, x2, (x1+x2)/2};
        int[] yPoints = {y1, y2, (y1+y2)/2};
        if(fill){
            g.setColor(color);
            g.fillPolygon(xPoints, yPoints, 3);
        }else
        g.drawPolygon(xPoints, yPoints, 3);
    }

    public void drawRound(int x1, int y1, int x2, int y2,boolean fill,
Color color) {
        Graphics g = getGraphics();
        if(fill) {
            g.setColor(color);
            g.fillRoundRect(x1, y1, abs(x1-x2), abs(y1-y2), abs(x1-x2),
abs(y1-y2));
        }else
        g.drawRoundRect(x1, y1, abs(x1-x2), abs(y1-y2), 10, 10);
    }

    public void drawText(int x, int y,String text, Color color) {
        Graphics g = getGraphics();
        g.setColor(color);
        g.drawString(text, x, y);
    }
}

==========
『fillRegion.java』
==========
package viewer;

import javax.swing.*;
```

```java
import javax.swing.event.ChangeListener;
import java.awt.*;

public class fillRegion extends JPanel {
    public JCheckBox checkBox;
    public boolean isChecked;

    class fillRegionListener implements ChangeListener {
        public void stateChanged(javax.swing.event.ChangeEvent e) {
            isChecked = checkBox.isSelected();
        }
    }

    fillRegion(){
        setLayout(new GridLayout(1,1));
        setVisible(true);

        checkBox = new JCheckBox("Fill Region");
        checkBox.addChangeListener(new fillRegionListener());
        add(checkBox);
    }


}
```

==========

『leftControlBar.java』

==========

```java
package viewer;

import javax.swing.*;
import java.util.Vector;

public class leftControlBar extends JPanel {
    public static buttonSet buttonSet;

    public void addAll(Vector buttons){
        for(Object button : buttons){
            this.add((JComponent)button);
        }
    }

    leftControlBar(){
        this.setPreferredSize(new java.awt.Dimension(100, 550));
        this.setBackground(new java.awt.Color(108, 108, 108));
```

```
        //this.setVerticalScrollBarPolicy(JScrollPane.VERTICAL_SCROLLBAR_N
EVER);
        //this.setHorizontalScrollBarPolicy(JScrollPane.HORIZONTAL_SCROLLB
AR_AS_NEEDED);

        buttonSet = new buttonSet();
        this.addAll(viewer.buttonSet.buttons);

    }
}
```

==========

『shapeColor.java』

==========

```
package viewer;

import javax.swing.*;
import javax.swing.event.ChangeListener;
import java.awt.*;
import java.awt.event.ActionListener;

public class shapeColor extends JPanel {
    public Color color=Color.RED;
    public JComboBox comboBox;

    class ComboBoxListener implements ActionListener {
        public void actionPerformed(java.awt.event.ActionEvent e) {
            //convert to Color
            String colorName = (String) comboBox.getSelectedItem();
            color = Color.decode(colorName);

        }
    }

    shapeColor(){
        setLayout(new GridLayout(1,2));
        setBackground(new java.awt.Color(255, 255, 255));
        setPreferredSize(new java.awt.Dimension(50, 50));

        JLabel label = new JLabel("Set Color");
        label.setHorizontalAlignment(JLabel.CENTER);
        add(label);

        comboBox = new JComboBox();
```

```
        comboBox.addItem("Red");
        comboBox.addItem("Green");
        comboBox.addItem("Blue");
        comboBox.addActionListener(new ComboBoxListener());
        add(comboBox);
    }

    public Color getColor() {
        return color;
    }


}
```

==========

『textSetter.java』

==========

```
package viewer;

import javax.swing.*;
import java.awt.*;

public class textSetter extends JPanel {


    textSetter(){
        setLayout(new GridLayout(1,3));

        JTextField textField = new JTextField();
        textField.setText("Enter text here");
        add(textField);

        JLabel label = new JLabel("size:");
        add(label);

        JTextField fontSize = new JTextField("14");
        fontSize.add(new JScrollBar(JScrollBar.HORIZONTAL));

        add(fontSize);
    }
}
```

==========

『TopPanel.java』

```
==========
package viewer;

import javax.swing.*;
import java.awt.*;

public class TopPanel extends JPanel {
    public static shapeColor color;
    public static background background;
    public static textSetter textSetter;
    public static fillRegion fillRegion;

    TopPanel() {
        setLayout(new GridLayout(1,2));
        setPreferredSize(new Dimension(800, 50));
        setBackground(Color.WHITE);
        setBorder(BorderFactory.createMatteBorder(0, 0, 1, 0,
Color.BLACK));

        JPanel leftPanel = new JPanel();
        leftPanel.setLayout(new GridLayout(2,1));
        leftPanel.setPreferredSize(new Dimension(400, 23));
        leftPanel.setBackground(Color.WHITE);
        color = new shapeColor();
        leftPanel.add(color);
        background = new background();
        leftPanel.add(background);

        JPanel rightPanel = new JPanel();
        rightPanel.setLayout(new GridLayout(2,1));
        rightPanel.setPreferredSize(new Dimension(400, 23));
        rightPanel.setBackground(Color.PINK);
        textSetter = new textSetter();
        rightPanel.add(textSetter);
        fillRegion = new fillRegion();
        rightPanel.add(fillRegion);

        this.add(leftPanel);
        this.add(rightPanel);
    }
}

==========
『Viewer.java』
==========
```

```
package viewer;

import javax.swing.*;
import java.awt.*;

public class Viewer {
    private JFrame mainFrame;
    private TopPanel topPanel;
    private leftControlBar leftControlBar;
    //private buttonSet buttonSet;
    private drawPanel drawPanel;

    public Viewer(JFrame main) {
        mainFrame = main;
    }

    public void update() {

        switch(buttonSet.selected.getText()){
            case "Line":
                System.out.println("draw Line");
                drawPanel.drawLine(drawPanel.x1,drawPanel.y1,
drawPanel.x2,drawPanel.y2);
                break;
                case "Rectangle":
                    System.out.println("draw Rectangle");
                drawPanel.drawRectangle(drawPanel.x1,drawPanel.y1,
drawPanel.x2,drawPanel.y2,topPanel.fillRegion.isChecked,topPanel.color.get
Color());
                break;
                case "Oval":
                drawPanel.drawCircle(drawPanel.x1,drawPanel.y1,
drawPanel.x2,drawPanel.y2,topPanel.fillRegion.isChecked,topPanel.color.get
Color());
                break;
                case "Triangle":
                drawPanel.drawTriangle(drawPanel.x1,drawPanel.y1,
drawPanel.x2,drawPanel.y2,topPanel.fillRegion.isChecked,topPanel.color.get
Color());
                break;
            case"Round":
                drawPanel.drawRound(drawPanel.x1,drawPanel.y1,
drawPanel.x2,drawPanel.y2,topPanel.fillRegion.isChecked,topPanel.color.get
Color());
                break;
```

```
            default:
        }

        //mainFrame.repaint();
    }

    public void init() {
        mainFrame.setTitle("Painter");
        mainFrame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        mainFrame.setSize(800, 600);
        mainFrame.setLocationRelativeTo(null);

        mainFrame.setBackground(new java.awt.Color(255, 255, 255));
        mainFrame.setLayout(new BorderLayout());

        topPanel = new TopPanel();
        mainFrame.add(topPanel, BorderLayout.NORTH);

        leftControlBar = new viewer.leftControlBar();
        mainFrame.add(leftControlBar, BorderLayout.WEST);

        drawPanel = new drawPanel();
        mainFrame.add(drawPanel, BorderLayout.CENTER);


    }

    public void show() {
        mainFrame.setVisible(true);
    }
}
```