「オブジェクト指向プログラミング特論」課題5(解答例)

2008/12/22

1 マウスによって矩形を描くプログラム

講義で示したマウスによって線描画を行うプログラム

http://aoba.cc.saga-u.ac.jp/lecture/ObjectOrientedProgramming/javasrc/SampleGUI/simpleDrawer/

を参考にして、マウスを使って矩形を描くプログラムを作成しなさい。

マウスによって矩形を描くとは

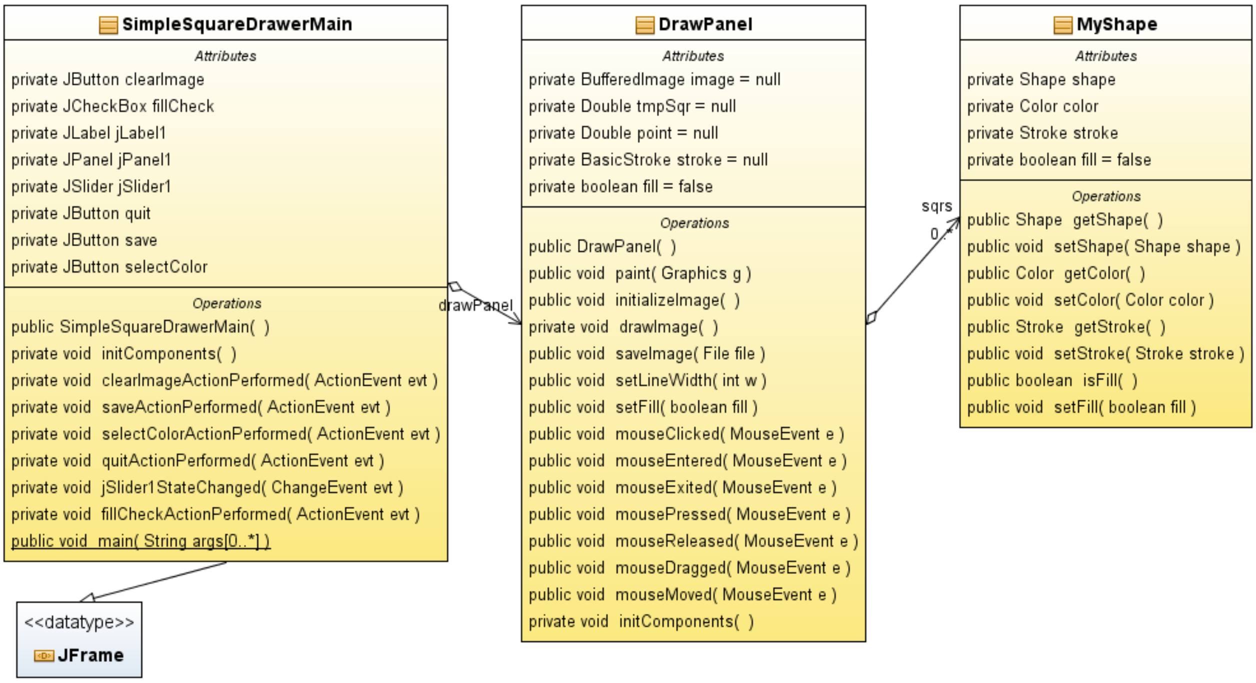
- マウスボタンを最初に押すことで、矩形の左上隅の座標を定める
- マウスをドラッグすることで、矩形の右下隅の座標を定める
- マウスボタンを放すことで、矩形を確定する

ような手順で矩形を描くことである。ただし、一つの矩形だけでなく、複数の矩形を描けること。

プログラムを作るに当たって次の機能についても、追加することを検討すること。

- 矩形ごとに色を設定する。
- 矩形ごとに線の太さを設定する。
- 矩形ごとに塗りつぶしの有無を設定する。

解答例


SimpleSquareDrawerMain
Attributes
private JButton clearImage
private JCheckBox fillCheck
private JLabel jLabel1
private JPanel jPanel1
private JSlider jSlider1
private JButton quit
private JButton save
private JButton selectColor
Operations
public SimpleSquareDrawerMain( )
private void initComponents( )
private void clearImageActionPerformed(ActionEvent evt)
private void saveActionPerformed(ActionEvent evt)
private void selectColorActionPerformed(ActionEvent evt)
private void quitActionPerformed(ActionEvent evt)
private void jSlider1StateChanged(ChangeEvent evt)
private void fillCheckActionPerformed(ActionEvent evt)
public void main(String args[0..*])
drawPanel
DrawPanel
Attributes
private BufferedImage image = null
private Double tmpSqr = null
private Double point = null
private BasicStroke stroke = null
private boolean fill = false
Operations
public DrawPanel( )
public void paint(Graphics g)
public void initializeImage( )
private void drawImage( )
public void saveImage(File file)
public void setLineWidth(int w)
public void setFill(boolean fill)
public void mouseClicked(MouseEvent e)
public void mouseEntered(MouseEvent e)
public void mouseExited(MouseEvent e)
public void mousePressed(MouseEvent e)
public void mouseReleased(MouseEvent e)
public void mouseDragged(MouseEvent e)
public void mouseMoved(MouseEvent e)
private void initComponents( )
sqrs
0..*
MyShape
Attributes
private Shape shape
private Color color
private Stroke stroke
private boolean fill = false
Operations
public Shape getShape( )
public void setShape(Shape shape)
public Color getColor( )
public void setColor(Color color)
public Stroke getStroke( )
public void setStroke(Stroke stroke)
public boolean isFill( )
public void setFill(boolean fill)
<<datatype>>
JFrame

```
/*
 * DrawPanel.java
 *
 * Created on 2008/11/21, 18:28
 */
package simpleSquareDrawer;

import java.awt.Graphics2D;
import java.awt.Color;
import java.awt.geom.Point2D;
import java.awt.image.BufferedImage;
import java.awt.Dimension;
import java.awt.event.MouseEvent;
import java.awt.event.MouseListener;
import java.awt.event.MouseMotionListener;
import java.awt.BasicStroke;
import java.awt.geom.Rectangle2D;
import java.util.Vector;
import java.io.*;

/**
 *
 * @author  tadaki
 */
public class DrawPanel extends javax.swing.JPanel implements
MouseListener, MouseMotionListener {

    private BufferedImage image = null;
    private Vector<MyShape> sqrs = null;
    private Rectangle2D.Double tmpSqr = null;
    private Point2D.Double point = null;
    private BasicStroke stroke = null;
    private boolean fill = false;

    /** Creates new form DrawPanel */
    public DrawPanel() {
        initComponents();
        initializeImage();
        addMouseListener(this);
        addMouseMotionListener(this);
        stroke = new BasicStroke();
        sqrs = new Vector<MyShape>();
    }
```

```
    @Override
    public void paint(java.awt.Graphics g) {
        if (image == null) {
            return;
        }
        //イメージを表示する
        g.drawImage(image, 0, 0, image.getWidth(), image.getHeight(),
this);
    }

    /**
     * 描画イメージを初期化する
     */
    public void initializeImage() {
        Dimension dimension = getPreferredSize();
        image = new BufferedImage(dimension.width, dimension.height,
                BufferedImage.TYPE_INT_RGB);
        Graphics2D g = (Graphics2D) image.getGraphics();
        g.setColor(this.getBackground());
        g.fillRect(0, 0, dimension.width, dimension.height);
    }

    private void drawImage() {
        initializeImage();
        Graphics2D g = (Graphics2D) image.getGraphics();
        for (MyShape r : sqrs) {
            g.setColor(r.getColor());
            g.setStroke(r.getStroke());
            g.draw(r.getShape());
            if (r.isFill()) {
                g.fill(r.getShape());
            } else {
                g.draw(r.getShape());
            }
        }
    }

    /**
     * イメージの保存
     * @param file 保存先ファイル
     */
    public void saveImage(File file) {
        if (!fileChooser.FileUtil.checkWritable(file)) {
            return;
        }
        FileOutputStream out = null;
        try {
            out = new FileOutputStream(file);
        } catch (FileNotFoundException ex) {
            fileChooser.FileUtil.showError(ex.getMessage());
        }
        if (out != null) {
            String ext =
fileChooser.FileUtil.getExtention(file.getName());
            try {
                javax.imageio.ImageIO.write(image, ext, out);
                String message = "イメージを" + file.getName() + "に保存
しました。";
                fileChooser.FileUtil.showMessage(message);
            } catch (IOException ex) {
                fileChooser.FileUtil.showError(ex.getMessage());
            }
        }
    }

    /**
     * 線幅変更
     * @param w 新しい線幅
     */
    public void setLineWidth(int w) {
        if (w < 1) {
            w = 1;
        }
        stroke = new BasicStroke((float) w);
    }

    public void setFill(boolean fill) {
        this.fill = fill;
    }

    /************************************************************/
    /*** マウスイベントの動作 ***************************************/
    public void mouseClicked(MouseEvent e) {
    }

    public void mouseEntered(MouseEvent e) {
    }
```

```
    public void mouseExited(MouseEvent e) {
    }

    public void mousePressed(MouseEvent e) {
        tmpSqr = new Rectangle2D.Double();
        point = new Point2D.Double((double) e.getX(), (double) e.getY());
    }

    public void mouseReleased(MouseEvent e) {
        if (point != null) {
            drawImage();
            Graphics2D g = (Graphics2D) image.getGraphics();
            Point2D.Double current = new Point2D.Double((double)
e.getX(), (double) e.getY());
            tmpSqr.setFrameFromDiagonal(point, current);
            Color c = getForeground();
            g.setColor(c);
            g.setStroke(stroke);
            if (fill) {
                g.fill(tmpSqr);
            } else {
                g.draw(tmpSqr);
            }
            MyShape s = new MyShape();
            s.setColor(c);
            s.setStroke(stroke);
            s.setShape(tmpSqr);
            s.setFill(fill);
            sqrs.add(s);
            point = null;
        }
        repaint();
    }

    public void mouseDragged(MouseEvent e) {
        if (point != null) {
            drawImage();
            Graphics2D g = (Graphics2D) image.getGraphics();
            Point2D.Double current = new Point2D.Double((double)
e.getX(), (double) e.getY());
            tmpSqr.setFrameFromDiagonal(point, current);
            g.setColor(this.getForeground());
            g.setStroke(stroke);
            if (fill) {
                g.fill(tmpSqr);
            } else {
                g.draw(tmpSqr);
            }
        }
        repaint();
    }

    public void mouseMoved(MouseEvent e) {
    }

    /** This method is called from within the constructor to
     * initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is
     * always regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated
Code">//GEN-BEGIN:initComponents
    private void initComponents() {

        setBackground(new java.awt.Color(255, 255, 255));
        setMinimumSize(new java.awt.Dimension(600, 600));
        setVerifyInputWhenFocusTarget(false);

        javax.swing.GroupLayout layout = new
javax.swing.GroupLayout(this);
        this.setLayout(layout);
        layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGap(0, 600, Short.MAX_VALUE)
        );
        layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGap(0, 600, Short.MAX_VALUE)
        );
    }// </editor-fold>//GEN-END:initComponents

    // Variables declaration - do not modify//GEN-BEGIN:variables
    // End of variables declaration//GEN-END:variables
}
```

```java
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

package simpleSquareDrawer;

import java.awt.Shape;
import java.awt.Color;
import java.awt.Stroke;
/**
 *
 * @author tadaki
 */
public class MyShape {
    private Shape shape;
    private Color color;
    private Stroke stroke;
    private boolean fill=false;

    public Shape getShape() {
        return shape;
    }

    public void setShape(Shape shape) {
        this.shape = shape;
    }

    public Color getColor() {
        return color;
    }

    public void setColor(Color color) {
        this.color = color;
    }

    public Stroke getStroke() {
        return stroke;
    }

    public void setStroke(Stroke stroke) {
        this.stroke = stroke;
    }
```

```java
    public boolean isFill() {
        return fill;
    }

    public void setFill(boolean fill) {
        this.fill = fill;
    }
}
```