【方法教程】Swing 自定义 JScrollPane 的滚动条设置, 重写 BasicScrollBarUI 方法

作者：moonce

原文链接：https://ld246.com/article/1539616424993

来源网站：链滴

1.自定义BasicScrollBarUI类

```
package com.melec.vo.view;
import java.awt.AlphaComposite;
import java.awt.Color;
import java.awt.Dimension;
import java.awt.GradientPaint;
import java.awt.Graphics;
import java.awt.Graphics2D;
import java.awt.Rectangle;
import java.awt.RenderingHints;


import javax.swing.ImageIcon;
import javax.swing.JButton;
import javax.swing.JComponent;
import javax.swing.JScrollBar;
import javax.swing.plaf.basic.BasicScrollBarUI;



/**
 * 自定义滚动条UI
 *
 * @author zsg
 */
public class DemoScrollBarUI extends BasicScrollBarUI {

    @Override
    protected void configureScrollBarColors() {

        // 把手

        // thumbColor = Color.GRAY;

        // thumbHighlightColor = Color.BLUE;

        // thumbDarkShadowColor = Color.BLACK;

        // thumbLightShadowColor = Color.YELLOW;

        // 滑道

        trackColor = Color.black;

        setThumbBounds(0, 0, 3, 10);

        // trackHighlightColor = Color.GREEN;
```

```
}
/**
 * 设置滚动条的宽度
 */
@Override
public Dimension getPreferredSize(JComponent c) {
    // TODO Auto-generated method stub
    c.setPreferredSize(new Dimension(40, 0));
    return super.getPreferredSize(c);
}

// 重绘滑块的滑动区域背景
public void paintTrack(Graphics g, JComponent c, Rectangle trackBounds) {
    Graphics2D g2 = (Graphics2D) g;
    GradientPaint gp = null;
    //判断滚动条是垂直的 还是水平的
    if (this.scrollbar.getOrientation() == JScrollBar.VERTICAL) {
        //设置画笔
        gp = new GradientPaint(0, 0, new Color(80, 80, 80),
                trackBounds.width, 0, new Color(80, 80, 80));
    }
    if (this.scrollbar.getOrientation() == JScrollBar.HORIZONTAL) {
        gp = new GradientPaint(0, 0, new Color(80, 80, 80),
                trackBounds.height, 0, new Color(80, 80, 80));
    }

    g2.setPaint(gp);
    //填充Track
```

```
    g2.fillRect(trackBounds.x, trackBounds.y, trackBounds.width,

            trackBounds.height);

    //绘制Track的边框
    /*       g2.setColor(new Color(175, 155, 95));
     g2.drawRect(trackBounds.x, trackBounds.y, trackBounds.width - 1,
            trackBounds.height - 1);
            */

    if (trackHighlight == BasicScrollBarUI.DECREASE_HIGHLIGHT)

        this.paintDecreaseHighlight(g);

    if (trackHighlight == BasicScrollBarUI.INCREASE_HIGHLIGHT)

        this.paintIncreaseHighlight(g);

}


@Override
protected void paintThumb(Graphics g, JComponent c, Rectangle thumbBounds) {

    // 把绘制区的x，y点坐标定义为坐标系的原点

    // 这句一定一定要加上啊，不然拖动就失效了

    g.translate(thumbBounds.x, thumbBounds.y);

    // 设置把手颜色

    g.setColor(new Color(    230,230,250));

    // 画一个圆角矩形

    // 这里面前四个参数就不多讲了，坐标和宽高

    // 后两个参数需要注意一下，是用来控制角落的圆角弧度

    // g.drawRoundRect(0, 0, 5, thumbBounds.height - 1, 5, 5);

    // 消除锯齿

    Graphics2D g2 = (Graphics2D) g;

    RenderingHints rh = new RenderingHints(RenderingHints.KEY_ANTIALIASING,

            RenderingHints.VALUE_ANTIALIAS_ON);

    g2.addRenderingHints(rh);

    // 半透明
```

```
        g2.setComposite(AlphaComposite.getInstance(AlphaComposite.SRC_OVER,
                0.5f));
        // 设置填充颜色，这里设置了渐变，由下往上
        // g2.setPaint(new GradientPaint(c.getWidth() / 2, 1, Color.GRAY,
        // c.getWidth() / 2, c.getHeight(), Color.GRAY));
        // 填充圆角矩形
        g2.fillRoundRect(0, 0, 40, thumbBounds.height - 1, 5, 5);
    }

    /**
     * 创建滚动条上方的按钮
     */
    @Override
    protected JButton createIncreaseButton(int orientation) {
        JButton button = new JButton();
        button.setBorderPainted(false);
        button.setContentAreaFilled(false);
        button.setBorder(null);
        return button;
    }
    /**
     * 创建滚动条下方的按钮
     */
    @Override
    protected JButton createDecreaseButton(int orientation) {
        JButton button = new JButton();
        button.setBorderPainted(false);
        button.setContentAreaFilled(false);
```

```
        button.setFocusable(false);
        button.setBorder(null);
        return button;
    }
}
```

2.使用自定义类,覆盖JScrollPane滚动条样式

```
JScrollPane scrollPane = new JScrollPane();
scrollPane.getVerticalScrollBar().setUI(new DemoScrollBarUI());
```