温州大学计算机与人工智能学院

Java程序设计（17网工） 实验报告

实验名称	展示图片				
班　　级	18电科2	姓　　名	方涛涛	学　　号	18211110208
实验地点		实验时间	2020-12-27,16:36:03	指导老师	

一、问题编号：

2706

地址：http://10.132.254.54/problem/2706/

二、问题描述：

中间用Label来展示图片。
上面是工具栏，有如下按钮：
（1）打开图片；（2）显示上一张图片；（3）显示下一张图片；（4）放大图片；（5）缩小图片。

如下图所示：

三、输入说明：

四、输出说明：

五、输入样列：

六、输出样列：

七、解答内容：

所用语言：

源代码：

```java
import java.awt.Color;
import java.awt.Image;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.MouseEvent;
import java.awt.event.MouseListener;
import java.awt.event.MouseWheelEvent;
import java.awt.event.MouseWheelListener;
import java.awt.image.BufferedImage;
import java.io.File;
import java.io.FilenameFilter;
import java.io.IOException;
import java.util.ArrayList;
import java.util.List;

import javax.imageio.ImageIO;
import javax.swing.ImageIcon;
import javax.swing.JFileChooser;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JMenu;
import javax.swing.JMenuBar;
import javax.swing.JMenuItem;
import javax.swing.JPanel;
import javax.swing.JScrollPane;


public class Main extends JFrame implements ActionListener, MouseListener, MouseWheelListener {
    JMenuBar jmb;
    JMenu jm1, jm2, jm3;
    JMenuItem[] jmi = new JMenuItem[7];
    JLabel jl, jl1;
    JPanel jp;
    JScrollPane jsp;
    ImageIcon imgIco, showii;
    File f;
    JFileChooser jfc;
    float width;
    float height;

    public static void main(String[] args) {
        Main mf = new Main();
        mf.init();
    }

    public void init() {
        jfc = new JFileChooser();
        jmb = new JMenuBar();
        jm1 = new JMenu("文件");
        jm2 = new JMenu("编辑");
        jmi[0] = new JMenuItem("打开");
        jmi[0].addActionListener(this);
        jmi[1] = new JMenuItem("另存为");
        jmi[1].addActionListener(this);
        jmi[5] = new JMenuItem("上一张");
        jmi[5].addActionListener(this);
        jmi[6] = new JMenuItem("下一张");
        jmi[6].addActionListener(this);
        jmi[2] = new JMenuItem("放大");
        jmi[2].addActionListener(this);
        jmi[3] = new JMenuItem("缩小");
        jmi[3].addActionListener(this);
        jmi[4] = new JMenuItem("原图");
        jmi[4].addActionListener(this);
        jm1.add(jmi[0]);
        jm1.add(jmi[1]);
        jm1.add(jmi[5]);
        jm1.add(jmi[6]);
        jm2.add(jmi[2]);
        jm2.add(jmi[3]);
        jm2.add(jmi[4]);
        jmb.add(jm1);
        jmb.add(jm2);

        jl = new JLabel("请选择图片", JLabel.CENTER);
        jl.setForeground(Color.gray);
        jl.addMouseWheelListener(this);
        jl.addMouseListener(this);
        jsp = new JScrollPane(jl);

        this.setJMenuBar(jmb);
        this.add(jsp);
        this.setTitle("Photo");
        this.setSize(700, 500);
        this.setLocation(300, 200);
        this.setVisible(true);
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }

    public void ShowImg() {
        width = (float) (imgIco.getIconWidth() * 0.09);
        height = (float) (imgIco.getIconHeight() * 0.09);
        showii = new ImageIcon(
                imgIco.getImage().getScaledInstance((int) width, (int) height, 0));
        this.jl.setText("");
        this.jl.setIcon(showii);
    }

    public void Zoom(int flag) {
        if ((width < 5000 && height < 4000) && (width > 5 && height > 4)) {
            if (flag == 0) {
                width *= 1.3;
                height *= 1.3;
            } else if (flag == 1) {
                width *= 0.7;
                height *= 0.7;
            } else if (flag == 2) {
                width = (float) (imgIco.getIconWidth() * 0.09);
                height = (float) (imgIco.getIconHeight() * 0.09);
            }
            showii = new ImageIcon(imgIco.getImage().getScaledInstance((int) width, (int) height, 0));
            this.jl.setIcon(showii);
        } else if (width > 5000 || height > 4000) {
            width *= 0.8;
            height *= 0.8;
        } else if (width < 5 || height < 3) {
            width *= 1.2;
            height *= 1.2;
        }
    }

    public void Save() {
        Image img = showii.getImage();
        try {

            BufferedImage choosed = ImageIO.read(f);
            BufferedImage bi = new BufferedImage((int) width, (int) height, choosed.getType());
            bi.getGraphics().drawImage(img, 0, 0, null);
            int result = jfc.showSaveDialog(this);
            File save = jfc.getSelectedFile();
            if (result == 0 && save != null) {
                String[] n = save.getName().split("\\.");
                ImageIO.write(bi, (n.length == 1 ? "gif" : n[1]), save);
            }
        } catch (IOException e) {
            e.printStackTrace();
        }
    }


    @Override
    public void actionPerformed(ActionEvent e) {
        if (e.getSource() == jmi[0] || e.getSource() == jl) {
            jfc.setCurrentDirectory(jfc.getCurrentDirectory());
            int result = jfc.showOpenDialog(this);
            f = jfc.getSelectedFile();
            if (result == 0 && f != null) {
                mytools.addFile(f);
                new getFileList(jfc.getCurrentDirectory()).start();
                imgIco = new ImageIcon(f.getPath());
                ShowImg();
            }
        } else if (f != null && e.getSource() == jmi[2]) {
            Zoom(0);
        } else if (f != null && e.getSource() == jmi[3]) {
            Zoom(1);
        } else if (f != null && e.getSource() == jmi[4]) {
            Zoom(2);
        } else if (f != null && e.getSource() == jmi[5]) {
            f = mytools.getLast(f);
            imgIco = new ImageIcon(f.getPath());
            ShowImg();
        } else if (f != null && e.getSource() == jmi[6]) {
            f = mytools.getNext(f);
            imgIco = new ImageIcon(f.getPath());
            ShowImg();
        } else if (f != null && e.getSource() == jmi[1]) {
            Save();
        }
    }

    @Override
    public void mouseWheelMoved(MouseWheelEvent e) {
        if (f != null) {
            if (e.getWheelRotation() == 1) {
                Zoom(0);
            } else if (e.getWheelRotation() == -1) {
                Zoom(1);
            }
        }
    }

    @Override
    public void mouseClicked(MouseEvent e) {
        if (f != null) {
                if (e.getClickCount() == 1 && e.getX() < 200) {
                    f = mytools.getLast(f);
                    imgIco = new ImageIcon(f.getPath());
                    ShowImg();
                } else if (e.getClickCount() == 1 && e.getX() > 500) {
                    f = mytools.getNext(f);
                    imgIco = new ImageIcon(f.getPath());
                    ShowImg();
                }
            }
        }

        @Override
        public void mousePressed(MouseEvent e) {

        }

        @Override
        public void mouseReleased(MouseEvent e) {

        }

        @Override
        public void mouseEntered(MouseEvent e) {

        }

        @Override
        public void mouseExited(MouseEvent e) {

        }
    }

class getFileList extends Thread {
    File CurrentDirectory;

    public getFileList(File f) {
        this.CurrentDirectory = f;
    }

    public void run() {
        File[] cd = CurrentDirectory.listFiles(mytools.myFilenameFilter());
        for (int i = 0; i < cd.length; i++) {
            mytools.addFile(cd[i]);
        }
    }
}

class mytools {
    public static List<File> fileBuffer = new ArrayList<File>();

    public static void addFile(File f) {
        if (!fileBuffer.contains(f))
            fileBuffer.add(f);
    }

    public static File getNext(File f) {
        if (fileBuffer.indexOf(f) + 1 < fileBuffer.size())
            return fileBuffer.get(fileBuffer.indexOf(f) + 1);
        else
            return fileBuffer.get(0);
    }

    public static File getLast(File f) {
        if (fileBuffer.indexOf(f) - 1 >= 0)
            return fileBuffer.get(fileBuffer.indexOf(f) - 1);
        else
            return fileBuffer.get(fileBuffer.size() - 1);
    }


    public static FilenameFilter myFilenameFilter() {
        FilenameFilter ff = new FilenameFilter() {
            @Override
            public boolean accept(File dir, String name) {
                if (name.endsWith(".jpg") || name.endsWith(".jpeg")
                        || name.endsWith(".gif") || name.endsWith(".png")
                        || name.endsWith(".bmp"))
                    return true;
                return false;
            }
        };
```

```
            return ff;
        }
}
```

八、判题结果

RE - 运行错误

判题结果补充说明：

test id:6223,result:RE, usedtime:152MS, usedmem:3364KB,score:100 Exception in thread "main" java.awt.HeadlessException: No X11 DISPLAY variable was set, but this program performed an operation which requires it. at java.awt.GraphicsEnvironment.checkHeadless(GraphicsEnvironment.java:173) at java.awt.Window.<init>(Window.java:547) at java.awt.Frame.<init>(Frame.java:419) at java.awt.Frame.<init>(Frame.java:384) at javax.swing.JFrame.<init>(JFrame.java:174) at Main.<init>(Main.java:28) at Main.main(Main.java:42)