实验四 图形界面

实验日期： 2019 年 5 月 24 日 班级： 软件1702

学号（后四位）： 0224 姓名： 刘文博 成绩：

一．实验目的

1. 掌握图形用户界面的设计方法
2. 掌握常用的构建用户界面的组件的用法
3. 掌握事件操作的原理
4. 能够对所设计的用户界面进行事件处理
5. 能够应用运算符解决实际小问题
6. 进一步熟悉Java的面向对象的编程思想

二．支撑的课程目标

“本实验项目可以支撑“课程目标1：能够按照设计方案要求进行计算机软硬件的界面设计”、“课程目标2：能够运用图形界面设计和事件驱动机制进行程序编程，以支持复杂工程问题的求解”、“课程目标3：通过多种途径的学习，掌握面向对象中不同组件的使用方法”。

本实验通过“任务驱动”方法，使学生根据软件工程中项目对图形界面的需求，设计方案，通过Java图形界面的相关知识点，合理设计图形界面的布局、组件和事件驱动，并进行满足任务需要的图形界面开发。

三．实验题目（前2题任选1题）

1.采用图形界面实现两个内容的交换，图形界面如下图1所示所示：



**代码：**

package Entity;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import javax.swing.JButton;

import javax.swing.JFrame;

import javax.swing.JPanel;

import javax.swing.JTextField;

import Util.FrameUtil;

public class Experiment4\_1 {

public static void main(String[] args) {

init();

}

public static void init() {

JFrame frame = new JFrame("实验一");

JPanel panle = new JPanel();

JTextField txt1 = new JTextField("123456", 10);

JTextField txt2 = new JTextField("654321",10);

JButton but = new JButton("exchange");

panle.add(txt1);

panle.add(txt2);

panle.add(but);

but.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

String temp = txt1.getText();

txt1.setText(txt2.getText());

txt2.setText(temp);

}

});

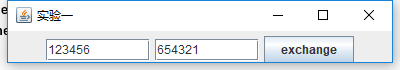
frame.add(panle);

FrameUtil.initFrame(frame, 400, 70);

}

}

**实验截图**



**点击后**

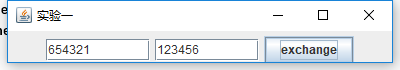


图1 内容交换

2. 采用图形界面设计如下图2所示的界面。并能够实现当点击“+”按钮时，标签给出点击次数，如图3所示

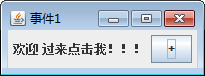


图2 开始界面

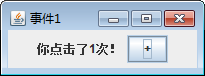


图3 点击“+”按钮后的界面

**代码：**

package Entity;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import javax.swing.JButton;

import javax.swing.JFrame;

import javax.swing.JLabel;

import javax.swing.JPanel;

import javax.swing.JTextField;

import Util.FrameUtil;

public class Experiment4\_2 {

public static void main(String[] args) {

init();

}

public static void init() {

JFrame frame = new JFrame("实验二");

JPanel panle = new JPanel();

JLabel lable = new JLabel("欢迎过来点击我！");

JButton but = new JButton("点击");

panle.add(lable);

panle.add(but);

frame.add(panle);

but.addActionListener(new ActionListener() {

int count = 0;

@Override

public void actionPerformed(ActionEvent e) {

lable.setText("您点击了" + ++count + "次");

}

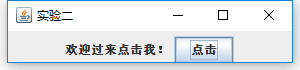
});

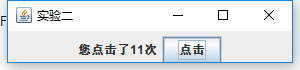
FrameUtil.initFrame(frame, 300, 70);

}

}

**实验截图：**





3．教材P226页，第3题，创建一个会员注册窗体，要求使用上JCheckBox,JRadioButton,JTextFiled,JTextArea等组件，并将用户的注册信息显示到文本区中。

**代码：**

package Entity;

import java.awt.FlowLayout;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.awt.event.ItemEvent;

import java.awt.event.ItemListener;

import java.io.File;

import java.util.ArrayList;

import java.util.Iterator;

import javax.swing.\*;

import javax.swing.event.DocumentEvent;

import javax.swing.event.DocumentListener;

import Util.FrameUtil;

import Util.ObjFileRW;

public class Experiment4\_3 {

static ArrayList<String> logonmessage = new ArrayList<String>() {

@Override

public String toString() {

Iterator<String> it = this.iterator();

StringBuffer str = new StringBuffer();

while(it.hasNext()) {

str.append(it.next()+"\n");

}

return str.toString();

}

};

static String password = "";

public static void main(String[] args) {

init();

}

public static void init() {

logonmessage.add("用户名：");

logonmessage.add("密码：");

logonmessage.add("性别：");

logonmessage.add("所在地：");

logonmessage.add("兴趣爱好：");

logonmessage.add("个人简介：");

JFrame frame = new JFrame("用户注册");

JTextArea info = new JTextArea(logonmessage.toString(),18, 60);

info.setEditable(false);

info.setLineWrap(true);

JPanel panle = new JPanel();

frame.add(panle);

JLabel namelable = new JLabel("用户名:");

JLabel passwordlable = new JLabel("密码:");

JTextField namefile = new JTextField(10);

JPasswordField passwordfile = new JPasswordField(10);

namefile.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

logonmessage.set(0, namelable.getText() + namefile.getText());

info.setText(logonmessage.toString());

}

});

passwordfile.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

String realpassword = new String(passwordfile.getPassword());

password = realpassword;

String showpass = "";

for (int i = 0; i < password.length(); i++) {

showpass += "\*";

}

logonmessage.set(1, passwordlable.getText() + showpass);

info.setText(logonmessage.toString());

}

});

panle.add(namelable);

panle.add(namefile);

panle.add(passwordlable);

panle.add(passwordfile);

JLabel sexlable = new JLabel("性别:");

JRadioButton man = new JRadioButton("男",true);

JRadioButton woman = new JRadioButton("女");

ButtonGroup group = new ButtonGroup();

group.add(man);

group.add(woman);

man.addItemListener(new ItemListener() {

@Override

public void itemStateChanged(ItemEvent e) {

if(man.isSelected()) {

logonmessage.set(2,sexlable.getText() + man.getText());

info.setText(logonmessage.toString());

}

}

});

woman.addItemListener(new ItemListener() {

@Override

public void itemStateChanged(ItemEvent e) {

if(woman.isSelected()) {

logonmessage.set(2,sexlable.getText() + woman.getText());

info.setText(logonmessage.toString());

}

}

});

panle.add(sexlable);

panle.add(man);

panle.add(woman);

/\*\*

\* 北京市、天津市、上海市、重庆市

\* 河北省、山西省、辽宁省、吉林省、黑龙江省、江苏省、浙江省、安徽省、

\* 福建省 、江西省、山东省、河南省、湖北省、湖南省、广东省、海南省、四川省、

\* 贵州省、云南省、陕西省、甘肃省、青海省、台湾省

\* 广西壮族自治区、内蒙古自治区、西藏自治区、宁夏回族自治区、新疆维吾尔自治区

\*/

Object [] city = {"北京市","天津市","上海市","重庆市",

"河北省","山西省","辽宁省","吉林省","黑龙江省","江苏省","浙江省","安徽省",

"福建省 ","江西省","山东省","河南省","湖北省","湖南省","广东省","海南省","四川省",

"贵州省","云南省","陕西省","甘肃省","青海省","台湾省",

"广西壮族自治区","内蒙古自治区","西藏自治区","宁夏回族自治区","新疆维吾尔自治区"};

JLabel citylable = new JLabel("所在城市");

JComboBox<Object> cities = new JComboBox<Object>(city);

panle.add(citylable);

panle.add(cities);

cities.addItemListener(new ItemListener() {

@Override

public void itemStateChanged(ItemEvent e) {

logonmessage.set(3,"所在城市：" + (String)e.getItem());

info.setText(logonmessage.toString());

}

});

JLabel hobitlable = new JLabel("兴趣爱好:");

JCheckBox hobitbox1 = new JCheckBox("学习");

JCheckBox hobitbox2 = new JCheckBox("JAVA");

JCheckBox hobitbox3 = new JCheckBox("PYTHON");

JCheckBox hobitbox4 = new JCheckBox("C/C++");

JCheckBox hobitbox5 = new JCheckBox("唱");

JCheckBox hobitbox6 = new JCheckBox("跳");

JCheckBox hobitbox7 = new JCheckBox("RAP");

JCheckBox hobitbox8 = new JCheckBox("篮球");

panle.add(hobitlable);

panle.add(hobitbox1);

panle.add(hobitbox2);

panle.add(hobitbox3);

panle.add(hobitbox4);

panle.add(hobitbox5);

panle.add(hobitbox6);

panle.add(hobitbox7);

panle.add(hobitbox8);

hobitbox1.addItemListener(new ItemListener() {

@Override

public void itemStateChanged(ItemEvent e) {

if(!hobitbox1.isSelected()) {

if(logonmessage.get(4).contains(hobitbox1.getText())) {

logonmessage.set(4,logonmessage.get(4).replace(hobitbox1.getText()+" ", ""));

info.setText(logonmessage.toString());

}

}

else if(hobitbox1.isSelected()) {

logonmessage.set(4,logonmessage.get(4) + hobitbox1.getText() + " ");

info.setText(logonmessage.toString());

}

}

});

hobitbox2.addItemListener(new ItemListener() {

@Override

public void itemStateChanged(ItemEvent e) {

if(!hobitbox2.isSelected()) {

if(logonmessage.get(4).contains(hobitbox2.getText())) {

logonmessage.set(4,logonmessage.get(4).replace(hobitbox2.getText() + " ", ""));

info.setText(logonmessage.toString());

}

}

else if(hobitbox2.isSelected()) {

logonmessage.set(4,logonmessage.get(4) + hobitbox2.getText() + " ");

info.setText(logonmessage.toString());

}

}

});

hobitbox3.addItemListener(new ItemListener() {

@Override

public void itemStateChanged(ItemEvent e) {

if(!hobitbox3.isSelected()) {

if(logonmessage.get(4).contains(hobitbox3.getText())) {

logonmessage.set(4,logonmessage.get(4).replace(hobitbox3.getText() + " ", ""));

info.setText(logonmessage.toString());

}

}

else if(hobitbox3.isSelected()) {

logonmessage.set(4,logonmessage.get(4) + hobitbox3.getText() + " ");

info.setText(logonmessage.toString());

}

}

});

hobitbox4.addItemListener(new ItemListener() {

@Override

public void itemStateChanged(ItemEvent e) {

if(!hobitbox4.isSelected()) {

if(logonmessage.get(4).contains(hobitbox4.getText())) {

logonmessage.set(4,logonmessage.get(4).replace(hobitbox4.getText() + " ", ""));

info.setText(logonmessage.toString());

}

}

else if(hobitbox4.isSelected()) {

logonmessage.set(4,logonmessage.get(4) + hobitbox4.getText() + " ");

info.setText(logonmessage.toString());

}

}

});

hobitbox5.addItemListener(new ItemListener() {

@Override

public void itemStateChanged(ItemEvent e) {

if(!hobitbox5.isSelected()) {

if(logonmessage.get(4).contains(hobitbox5.getText())) {

logonmessage.set(4,logonmessage.get(4).replace(hobitbox5.getText() + " ", ""));

info.setText(logonmessage.toString());

}

}

else if(hobitbox5.isSelected()) {

logonmessage.set(4,logonmessage.get(4) + hobitbox5.getText() + " ");

info.setText(logonmessage.toString());

}

}

});

hobitbox6.addItemListener(new ItemListener() {

@Override

public void itemStateChanged(ItemEvent e) {

if(!hobitbox6.isSelected()) {

if(logonmessage.get(4).contains(hobitbox6.getText())) {

logonmessage.set(4,logonmessage.get(4).replace(hobitbox6.getText() + " ", ""));

info.setText(logonmessage.toString());

}

}

else if(hobitbox6.isSelected()) {

logonmessage.set(4,logonmessage.get(4) + hobitbox6.getText() + " ");

info.setText(logonmessage.toString());

}

}

});

hobitbox7.addItemListener(new ItemListener() {

@Override

public void itemStateChanged(ItemEvent e) {

if(!hobitbox7.isSelected()) {

if(logonmessage.get(4).contains(hobitbox7.getText())) {

logonmessage.set(4,logonmessage.get(4).replace(hobitbox7.getText() + " ", ""));

info.setText(logonmessage.toString());

}

}

else if(hobitbox7.isSelected()) {

logonmessage.set(4,logonmessage.get(4) + hobitbox7.getText() + " ");

info.setText(logonmessage.toString());

}

}

});

hobitbox8.addItemListener(new ItemListener() {

@Override

public void itemStateChanged(ItemEvent e) {

if(!hobitbox8.isSelected()) {

if(logonmessage.get(4).contains(hobitbox8.getText())) {

logonmessage.set(4,logonmessage.get(4).replace(hobitbox8.getText() + " ", ""));

info.setText(logonmessage.toString());

}

}

else if(hobitbox8.isSelected()) {

logonmessage.set(4,logonmessage.get(4) + hobitbox8.getText() + " ");

info.setText(logonmessage.toString());

}

}

});

JLabel infolabe1 = new JLabel(" ");

JLabel infolabe2 = new JLabel("个人简介：");

JTextArea infoarea = new JTextArea(5,53);

infoarea.setLineWrap(true);

infoarea.getDocument().addDocumentListener(new DocumentListener() {

@Override

public void removeUpdate(DocumentEvent e) {

logonmessage.set(5,infolabe2.getText() + infoarea.getText());

info.setText(logonmessage.toString());

}

@Override

public void insertUpdate(DocumentEvent e) {

logonmessage.set(5,infolabe2.getText() + infoarea.getText());

info.setText(logonmessage.toString());

}

@Override

public void changedUpdate(DocumentEvent e) {

logonmessage.set(5,infolabe2.getText() + infoarea.getText());

info.setText(logonmessage.toString());

}

});

panle.add(infolabe1);

panle.add(infolabe2);

panle.add(infoarea);

JButton okbut = new JButton("提交");

panle.add(okbut);

okbut.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

if(logonmessage.get(0).length() == 4) {

JOptionPane.showMessageDialog(null, "请输入用户名", "警告",JOptionPane.ERROR\_MESSAGE);

}

else if(logonmessage.get(1).length() == 3) {

JOptionPane.showMessageDialog(null, "请输入密码", "警告",JOptionPane.ERROR\_MESSAGE);

}

else if(logonmessage.get(2).length() == 3) {

JOptionPane.showMessageDialog(null, "请选择性别", "警告",JOptionPane.ERROR\_MESSAGE);

}

else if(logonmessage.get(3).length() == 4) {

JOptionPane.showMessageDialog(null, "请选择城市", "警告",JOptionPane.ERROR\_MESSAGE);

}

else if(logonmessage.get(4).length() == 5) {

JOptionPane.showMessageDialog(null, "请选择兴趣爱好", "警告",JOptionPane.ERROR\_MESSAGE);

}

else if(logonmessage.get(5).length() == 5) {

JOptionPane.showMessageDialog(null, "请输入个人简介", "警告",JOptionPane.ERROR\_MESSAGE);

}

if(logonmessage.get(0).length() != 4 && logonmessage.get(1).length() != 3 && logonmessage.get(2).length() != 3

&& logonmessage.get(3).length() != 4 && logonmessage.get(4).length() != 5 && logonmessage.get(5).length() != 5

) {

if(logonmessage.get(4).length() == 16 && logonmessage.get(4).contains("唱")&& logonmessage.get(4).contains("跳")

&& logonmessage.get(4).contains("RAP") && logonmessage.get(4).contains("篮球")) {

JOptionPane.showMessageDialog(null, "鉴定完毕，你是蔡徐坤！！", "恭喜恭喜",JOptionPane.WARNING\_MESSAGE);

}

User user = new User(logonmessage.get(0).substring(4),password ,logonmessage.get(2).substring(3)

,logonmessage.get(3).substring(4) ,logonmessage.get(4).substring(5) ,logonmessage.get(5).substring(5));

/\*\*

\* 进行存档（对象的读写）

\*/

File file = new File("G:/JAVASpace/GUIDemo/src/data/userData.dat");

ObjFileRW of = new ObjFileRW(file);

of.write(user);

User user1 = (User)of.read();

System.out.println(user1.toString());

}

}

});

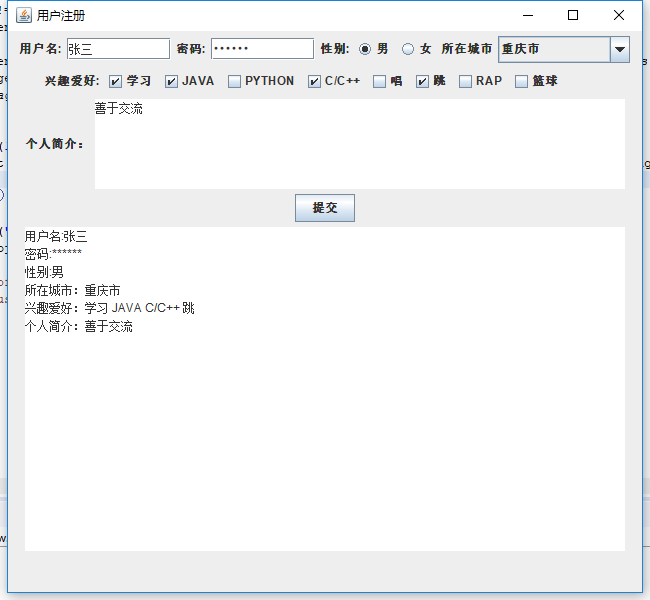
panle.add(info);

FrameUtil.initFrame(frame, 650, 600);

}

}

截图：



4． 教材P260页，以Windows的记事本为参考模型，创建一个简易的文本编辑器，在文本编辑器中可以实现打开文件对字体大小、字形（粗体，斜体）、颜色进行设置，可以对内容进行添加、删除、修改，并保存文本。

说明：（1）可以选用单选框、复选框、列表、下拉框或菜单任意一种或多种组合实现字体的设置

（2）根据自己的编程能力实现功能，可以只实现字体的设置，或者字体和内容的编辑，或者全部功能都实现。

（3）本题的目的是常用组件的使用。

package Entity;

import java.awt.BorderLayout;

import java.awt.FileDialog;

import java.awt.Font;

import java.awt.Frame;

import java.awt.event.\*;

import java.io.BufferedReader;

import java.io.BufferedWriter;

import java.io.File;

import java.io.FileInputStream;

import java.io.FileNotFoundException;

import java.io.FileReader;

import java.io.FileWriter;

import java.io.IOException;

import javax.swing.\*;

import javax.swing.event.AncestorEvent;

import javax.swing.event.AncestorListener;

import javax.swing.event.CaretEvent;

import javax.swing.event.CaretListener;

import Util.FrameUtil;

public class Experiment4\_4{

static int font = 12; //字体大小

static String copystr = null; //复制

public int mark = 0; //粘贴

JFrame frame = new JFrame("记事本");

JMenuBar bar = new JMenuBar();

JMenu filemenu = new JMenu("文件");

JMenu editMenu = new JMenu("编辑");

JMenu formatMenu = new JMenu("格式");

JMenu seeMenu = new JMenu("查看");

JMenu helpMenu = new JMenu("帮助");

JTextArea area = new JTextArea(20,30);

/\*\* 文件栏 \*/

JMenuItem openmenu = new JMenuItem("打开");

JMenuItem savemenu = new JMenuItem("保存");

JMenuItem closemenu = new JMenuItem("关闭");

/\*\* 编辑栏 \*/

JMenuItem copymenu = new JMenuItem("复制");

JMenuItem pastemenu = new JMenuItem("粘贴");

/\*\* 格式栏 \*/

JMenu typefacemenu = new JMenu("字体");

JMenuItem overstrikingmenu = new JMenuItem("加粗");

JMenuItem underlinemenu = new JMenuItem("模糊");

JMenuItem italicmenu = new JMenuItem("斜体");

JMenuItem backmenu = new JMenuItem("还原");

JMenu wordsizemenu = new JMenu("字号");

JMenuItem wordsize11 = new JMenuItem("11");

JMenuItem wordsize12 = new JMenuItem("12");

JMenuItem wordsize13 = new JMenuItem("13");

JMenuItem wordsize14 = new JMenuItem("14");

JMenuItem wordsize15 = new JMenuItem("15");

JMenuItem aboutmenu = new JMenuItem("关于记事本");

public void init() {

/\*\* 文件栏 \*/

filemenu.add(openmenu);

filemenu.add(savemenu);

filemenu.add(closemenu);

/\*\* 编辑栏 \*/

editMenu.add(copymenu);

editMenu.add(pastemenu);

/\*\* 格式栏 \*/

typefacemenu.add(overstrikingmenu);

typefacemenu.add(underlinemenu);

typefacemenu.add(italicmenu);

typefacemenu.add(backmenu);

formatMenu.add(typefacemenu);

wordsizemenu.add(wordsize11);

wordsizemenu.add(wordsize12);

wordsizemenu.add(wordsize13);

wordsizemenu.add(wordsize14);

wordsizemenu.add(wordsize15);

formatMenu.add(wordsizemenu);

/\*\* 帮助栏 \*/

helpMenu.add(aboutmenu);

bar.add(filemenu);

bar.add(editMenu);

bar.add(formatMenu);

bar.add(seeMenu);

bar.add(helpMenu);

frame.add(bar,BorderLayout.NORTH);

frame.add(area);

FrameUtil.initFrame(frame, 650, 600);

/\*\* 打开文件 \*/

openmenu.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

FileDialog d = new FileDialog(new Frame(), "请选择文件", FileDialog.LOAD);

d.setVisible(true);

File file = null;

try {

file = new File(d.getDirectory() + d.getFile());

} catch (Exception e3) {

}

String str = "";

String num = null;

try (FileReader fread = new FileReader(file);

BufferedReader bfreead = new BufferedReader(fread)){

while ((num = bfreead.readLine()) !=null) {

str += num + "\n";

}

area.setText(str);

} catch (FileNotFoundException e1) {

System.err.println("未选择文件！！");

} catch (IOException e2) {

}

}

});

/\*\* 保存文件 \*/

savemenu.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

String path = null;

FileDialog filedialog = new FileDialog(frame, "请选择要保存的文件夹", FileDialog.SAVE);

filedialog.setVisible(true);

path = filedialog.getDirectory();

System.out.println(path);

String name = null;

if(path != null) {

name = filedialog.getFile();

}

try(FileWriter fw = new FileWriter(new File(path + "/" + name));

BufferedWriter bfw = new BufferedWriter(fw)

){

bfw.write(area.getText().replaceAll("\n", "\r\n"));

} catch (Exception e2) {

// TODO: handle exception

}

}

});

/\*\* 关闭 \*/

closemenu.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

System.exit(0);

}

});

/\*\* 字体加粗 \*/

overstrikingmenu.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

area.setFont(new Font("加粗",Font.BOLD, font));

}

});

/\*\* 字体变轻 \*/

underlinemenu.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

area.setFont(new Font("变细", Font.HANGING\_BASELINE, font));

}

});

/\*\* 倾斜 \*/

italicmenu.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

area.setFont(new Font("斜体", Font.ITALIC, font-1));

}

});

/\*\* 还原 \*/

backmenu.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

area.setFont(new Font(null));

}

});

/\*\* 字号 \*/

wordsize11.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

font = 11;

area.setFont(new Font("", 0, font));

}

});

wordsize12.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

font = 12;

area.setFont(new Font("", 0, font));

}

});

wordsize13.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

font = 13;

area.setFont(new Font("", 0, font));

}

});

wordsize14.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

font = 14;

area.setFont(new Font("", 0, font));

}

});

wordsize15.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

font = 15;

area.setFont(new Font("", 0, font));

}

});

/\*\* 复制 \*/

copymenu.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

copystr = area.getSelectedText();

}

});

/\*\* 粘贴 \*/

pastemenu.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

StringBuffer str = new StringBuffer(area.getText());

str.insert(mark, copystr);

area.setText(str.toString());

return ;

}

});

/\*\* 获取光标位置 \*/

area.addCaretListener(new CaretListener() {

@Override

public void caretUpdate(CaretEvent e) {

mark = e.getMark();

}

});

}

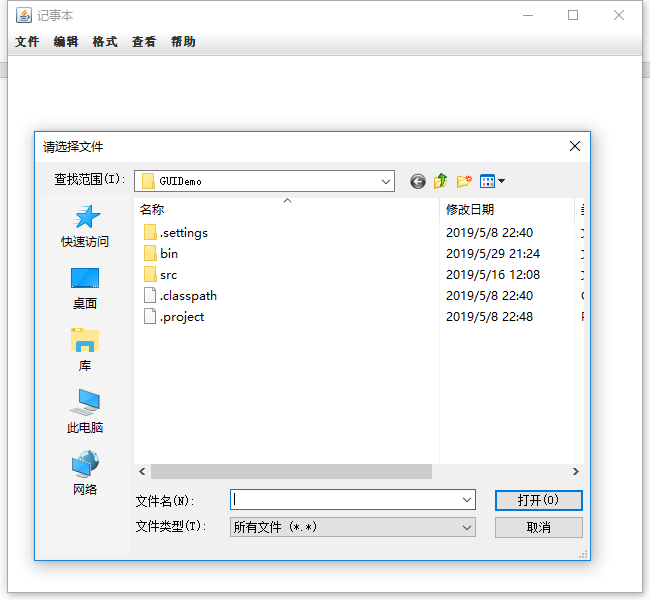
public static void main(String[] args) {

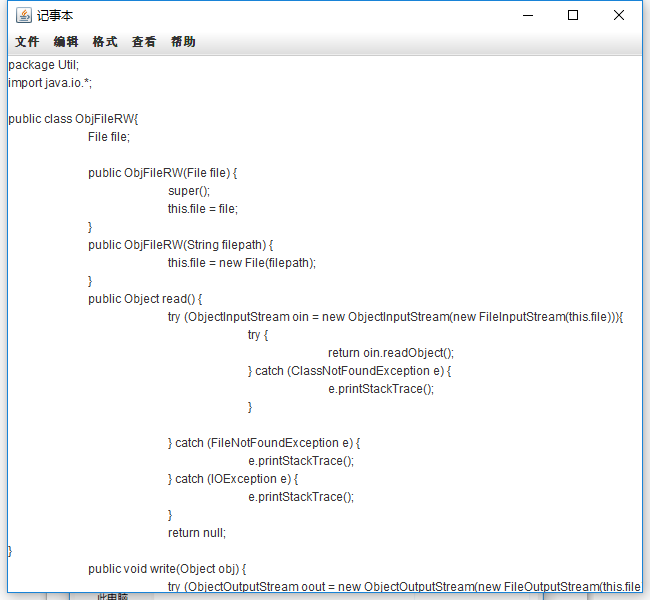
new Experiment4\_4().init();

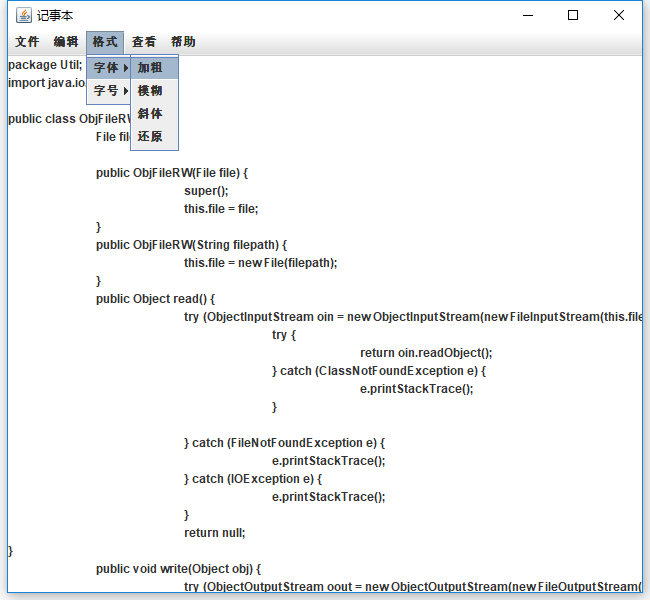
}

}

**实验截图：**







四．在本次实验中的问题和解决方法

无

五．心得体会

不是所有的监听器都叫ActionListener，有不一样的！！！！比如CaretListener，具体用哪个，根据功能而定