

**计算机与信息 学院实验报告**

|  |  |  |
| --- | --- | --- |
| 实验课程： | 面向对象程序设计实验 | |
| 实验编号： | 实验十一 | |
| 实验名称： | 图形用户界面基础编程 | |
| 实验人员： | 学号 | 22111302041 |
| 姓名 | 高蝶 |
| 班级 | 22级计算机科学与技术1班 |
| 指导教师： | 陶涛 | |
| 实验室： | 学院楼2楼202 | |
| 实验日期： | 2023.5.10 | |

# 实验目的

1. 了解使用JFrame类创建窗口的常用方法；

2. 掌握JButton、JTextField等常见组件的常用方法；

3. 掌握FlowLayout、BorderLayou等布局管理器的使用方法；

4. 熟练掌握组件的添加和布局方法。

# 实验要求

1. 所编写的程序应符合面向对象程序编写规范；

2. 对JFrame类的相关方法功能进行测试，了解其应用方法；

3. 对问题进行充分的需求分析，设计正确、全面的测试数据，设计类的定义框架及UML图后，再进行编程实现；

4. 所设计的测试数据全部通过后，再进行程序提交，提高程序的通过率。

# 实验内容

**1.验证型实验**

请检查该程序有无错误、下拉列表是否可以正常显示，若有则修改，使之正确，得到其运行结果。请为该程序添加注释语句。

import javax.swing.\*;

import java.awt.\*;

public class T1 extends JFrame {

public T1(){

setBounds(100,100,400,300);

setVisible(true);

//下拉框：

JComboBox comboBox = new JComboBox();

comboBox.addItem("秒杀特价");

comboBox.addItem("夏季特款");

comboBox.addItem("爆款热款");

add(comboBox);

setDefaultCloseOperation(WindowConstants.EXIT\_ON\_CLOSE);

}

public static void main(String[] args) {

new T1();

}

}

**2.应用型实验**

**2.1 按钮布局**

现在有East、South、West、North、Center五个按钮，为每一个按钮设置自己喜欢的不同背景颜色，利用BorderLayout布局管理器完成五个按钮的布局；每一个按钮点击后变为灰色（java.awt.Color.gray）。

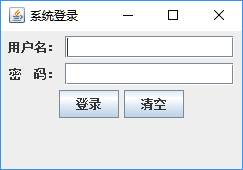
**2.2**  **登录验证界面**

设计并完成一个用户登录界的编写，点击“登录”判断用户名和密码是否正确匹配，如果正确，打开一个新的JFrame窗口，标题为用户名；如果错误，打开一个JFrame窗口，标题为“登录失败”；点击“清空”按钮，清除两个文本框的内容。假设以下是正确的用户名及其对应的密码

用户名：user1 密码：111

用户名：user2 密码：222

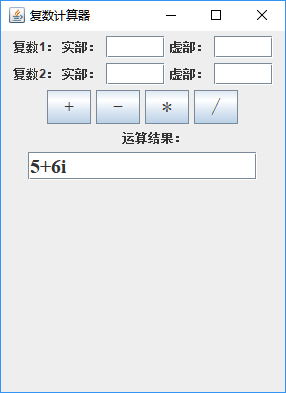
用户名：user3 密码：333



**2.3 复数计算界面**

设计并完成以下图形界面的编写，实现复数的四则运算。输入两个复数实部和虚部后，点击+、-、\*、/按钮，将相应的运算结果填入下方的文本框中。

要求：+、-、\*、/ 具体运算调用以前复数类的相关方法实现，不允许直接在事件除法方法中直接编写。



**3. 设计型实验**

**3.1信息收集**

设计并完成以下图形界面的编写。要求如下：

（1）选中一个省份，自动显示“市”的信息，同样，选中某个市后，自动显示“区县”信息。

（2）至少添加3个省，每个省3个市，每个市3个区县以上的数据；

（3）点击“提交”后，弹出一个对话框，显示填写的全部信息。

对话框弹出方法参考以下图片中的内容，其中第一个参数使用窗口对象，第二个参数是显示的信息，第三个参数是标题，第四个参数使用JOptionPane.INFORMATION\_MESSAGE（其他也行）





# 程序清单

2.应用型实验

（1）

分析：

1. 请描述本实验的解决方案

**定义窗口，再定义按钮设定颜色，再添加按钮，按钮调用实现相应方法的类进行事件处理，设置窗口为可见，关闭窗口。**

1. 请以本题为例，描述Java事件处理方法和步骤。

**创建事件源button->自定义事件监听器Mylistener->为事件源注册监听器。**

**源代码：package prj\_anniubuju;**

**import** java.awt.\*;

**import** javax.swing.\*;

**import** java.awt.event.\*;

**public** **class** Main {

**public** **static** **void** **main**(String[] **args**) {

JFrame **f**=**new** JFrame();

f.setBounds(100,100,400,300);

f.setLayout(**new** BorderLayout());

JButton **btn1**=**new** JButton("East");

JButton **btn2**=**new** JButton("South");

JButton **btn3**=**new** JButton("West");

JButton **btn4**=**new** JButton("North");

JButton **btn5**=**new** JButton("Center");

btn1.setBackground(Color.***blue***);

btn2.setBackground(Color.***cyan***);

btn3.setBackground(Color.***green***);

btn4.setBackground(Color.***PINK***);

btn5.setBackground(Color.***YELLOW***);

f.add(btn1,BorderLayout.***EAST***);

f.add(btn2,BorderLayout.***SOUTH***);

f.add(btn3,BorderLayout.***WEST***);

f.add(btn4,BorderLayout.***NORTH***);

f.add(btn5,BorderLayout.***CENTER***);

Mylistener **listen**=**new** Mylistener();

btn1.addActionListener(listen);

f.setVisible(**true**);

f.setDefaultCloseOperation(JFrame.***EXIT\_ON\_CLOSE***);

}

}

**class** Mylistener **implements** ActionListener

{

**public** **void** **actionPerformed**(ActionEvent **e**) {

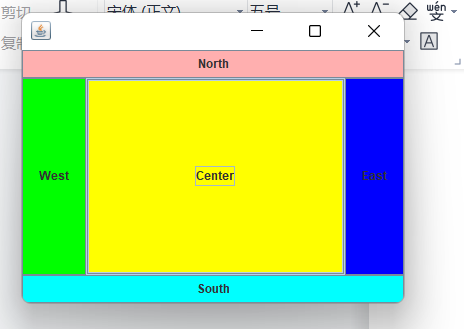
JButton **b**=**new** JButton ();

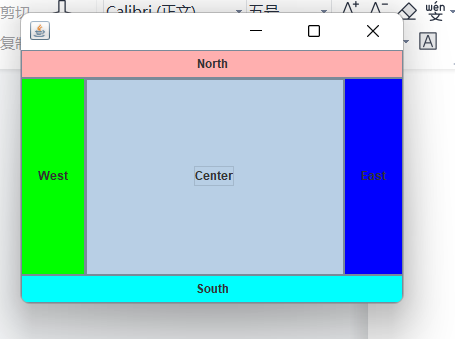
b.setBackground(Color.***gray***);

}

}

运行结果：





发现的问题和解决方法：

**1：在实现接口的类的方法中，不知该怎样设置调用该方法的按钮的颜色为灰，尝试new了一个按钮发现是可以的。**

**2：在界面运行出来后删除不掉，加上setDefaultCloseOperation(JFrame.*EXIT\_ON\_CLOSE*);就可以正常删除了。**

（2）

分析：

1. 请描述本实验的解决方案

**先搭起界面的具体框架，在进行点击按钮的具体事件响应**

1. 请以本题为例，描述Java事件处理方法和步骤。

**创建事件源butclear，butin->自定义事件监听器 W->为事件源注册监听器进行事件响应。**

源代码：

**package** prj\_denluyanzhenjiemian;

**import** java.awt.\*;

**import** javax.swing.\*;

**import** java.awt.event.\*;

**public** **class** Main {

**public** **static** **void** **main**(String[] **args**) {

W **p**=**new** W();

}

}

**class** W **extends** JFrame **implements** ActionListener {

JTextField userName = **new** JTextField();

JPasswordField passWord = **new** JPasswordField();

JButton butin = **new** JButton("登录");

JButton butclear = **new** JButton("清空");

**public** **W**() {

init();

JLabel **label1** = **new** JLabel("用户名：");

JLabel **label2** = **new** JLabel("密 码：");

label1.setBounds(10,5,70,20);

label2.setBounds(10,30,50,20);

userName.setBounds(80,5,150,20);

passWord.setBounds(80,30,150,20);

butin.setBounds(70,60,60,40);

butclear.setBounds(135,60,60,40);

butin.addActionListener(**this**);

butclear.addActionListener(**this**);

**this**.getContentPane().add(label1);

**this**.getContentPane().add(label2);

**this**.getContentPane().add(userName);

**this**.getContentPane().add(passWord);

**this**.getContentPane().add(butin);

**this**.getContentPane().add(butclear);

**this**.setVisible(**true**);

}

**private** **void** **init**() {

**this**.setSize(300,200);

**this**.setTitle("系统登录");

**this**.setLocationRelativeTo(**null**);//居中

**this**.setDefaultCloseOperation(***EXIT\_ON\_CLOSE***);//关闭模式设置

**this**.setLayout(**null**);//清除格式

}

@Override

**public** **void** **actionPerformed**(ActionEvent **e**){

Object **eventsource** = e.getSource();//获得目前这个事件的事件源

**if**(eventsource==butclear){

userName.setText("");

passWord.setText("");

}**else** **if**(eventsource==butin){

String **name** = userName.getText();

String **mima** = **new** String(passWord.getPassword());//passWord.getPassword()返回char[] 数组类型；

**if**(name.equals("user1")&&mima.equals("111")||name.equals("user2")&&mima.equals("222")||name.equals("user3")&&mima.equals("333")){

JFrame **jf** = **new** JFrame(name);

jf.setSize(300,300);

jf.setLocationRelativeTo(**null**);//居中

jf.setAlwaysOnTop(**true**);

jf.setVisible(**true**);

}**else**{

JFrame **jf** = **new** JFrame("登录失败");

jf.setSize(300,300);

jf.setLocationRelativeTo(**null**);//居中

jf.setAlwaysOnTop(**true**);

jf.setVisible(**true**);

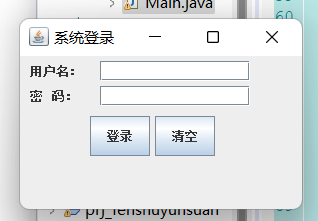
}

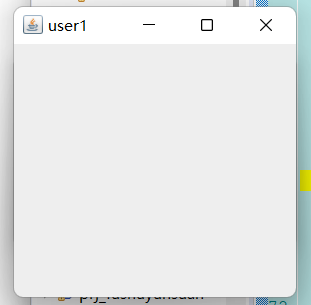
}

}

}

运行结果：





发现的问题和解决方法：

**采用了class W extends JFrame implements ActionListener 这种类的构建，对于具体内部如何实现及很多写法不理解。如add()可实现添加，为什么用getContentPane().add()。上网查没找到想要的结果，问了同学。**

**使用setBounds进行图形界面具体布局不是预期的那样，加上setLayout(null)就正常显示了。**

（3）

分析：

1. 请描述本实验的解决方案

**先搭起界面，进行事件处理，其中涉及到复数的加减乘除运算及输出的格式问题**

1. 请以本题为例，描述Java事件处理方法和步骤。

**创建事件源：所有的按钮及文本框->自定义事件监听器Fushu->为事件源注册监听器进行事件响应。**

源代码：

package prj\_fushujisuan;

import java.awt.\*;

import javax.swing.\*;

import java.awt.event.\*;

public class Main {

public static void main(String[] args) {

Fushu p=new Fushu();

}

}

class Fushu extends JFrame implements ActionListener {

JTextField c1shi = new JTextField("");

JTextField c1xu = new JTextField("");

JTextField c2shi = new JTextField("");

JTextField c2xu = new JTextField("");

JTextField res = new JTextField("");

JButton add = new JButton("+");

JButton sub= new JButton("-");

JButton mult = new JButton("\*");

JButton div = new JButton("/");

public Fushu() {

initJFrame();

JLabel lable1 = new JLabel("复数1：实部：");

JLabel lable3 = new JLabel("复数2：实部：");

JLabel lable2 = new JLabel("虚部：");

JLabel lable4 = new JLabel("虚部：");

JLabel lable5 = new JLabel("运算结果：");

lable1.setBounds(5,5,90,30);

lable2.setBounds(200,5,50,30);

lable3.setBounds(5,40,90,30);

lable4.setBounds(200,40,50,30);

lable5.setBounds(150,140,70,30);

c1shi.setBounds(95,5,95,30);

c1xu.setBounds(250,5,95,30);

c2shi.setBounds(95,40,95,30);

c2xu.setBounds(250,40,95,30);

add.setBounds(60,80,55,50);

sub.setBounds(120,80,55,50);

mult.setBounds(180,80,55,50);

div.setBounds(240,80,55,50);

res.setBounds(30,180,300,30);

add.addActionListener(this);

sub.addActionListener(this);

mult.addActionListener(this);

div.addActionListener(this);

this.getContentPane().add(lable1);

this.getContentPane().add(c1shi);

this.getContentPane().add(lable2);

this.getContentPane().add(c1xu);

this.getContentPane().add(lable3);

this.getContentPane().add(c2shi);

this.getContentPane().add(lable4);

this.getContentPane().add(c2xu);

this.getContentPane().add(add);

this.getContentPane().add(sub);

this.getContentPane().add(mult);

this.getContentPane().add(div);

this.getContentPane().add(lable5);

this.getContentPane().add(res);

this.setVisible(true);

}

private void initJFrame() {

this.setSize(370,370);

this.setTitle("复数计算器");

this.setLocationRelativeTo(null);//居中

this.setDefaultCloseOperation(EXIT\_ON\_CLOSE);//关闭模式设置

this.setLayout(null);//清除格式

}

@Override

public void actionPerformed(ActionEvent e){

Object obj = e.getSource();

Double strshi1=Double.parseDouble(c1shi.getText());

Double strshi2=Double.parseDouble(c2shi.getText());

Double strxu1=Double.parseDouble(c1xu.getText());

Double strxu2=Double.parseDouble(c2xu.getText());

Complex c1=new Complex(strshi1,strxu1);

Complex c2=new Complex(strshi2,strxu2);

Complex ans=null;

String str=null;

if(obj==add) {

ans=c1.add(c1,c2);

}else if(obj==sub) {

ans=c1.subduction(c1,c2);

}else if(obj==mult) {

ans=c1.multi(c1,c2);

}else if (obj==div) {

ans=c1.divide(c1,c2);

}

res.setText(ans.print());

}

}

class Complex {

public double real;//复数实部

public double imaginary;//复数虚部

public Complex()//复数类的无参构造方法

{

this.imaginary=0;

this.real=0;

}

public Complex(double real,double imaginary)//复数类的有参构造方法

{

this.imaginary=imaginary;

this.real=real;

}

//复数加法(a+bi)+(c+di)=(a+c)+(b+d)i

public Complex add(Complex a,Complex b)

{

return new Complex(a.real+b.real,a.imaginary+b.imaginary);

}

//复数减法(a+bi)-(c+di)=(a-c)+(b-d)i

public Complex subduction(Complex a,Complex b)

{

return new Complex(a.real-b.real,a.imaginary-b.imaginary);

}

//复数的乘法

public Complex multi(Complex a,Complex b)

{

return new Complex ((a.real\*b.real-a.imaginary\*b.imaginary),(a.real\*b.imaginary+a.imaginary\*b.real));

}

//复数的除法

public Complex divide (Complex a,Complex b)

{

return new Complex((a.real\*b.real+a.imaginary\*b.imaginary)/(b.real\*b.real+b.imaginary\*b.imaginary),(-a.real\*b.imaginary+a.imaginary\*b.real)/(b.real\*b.real+b.imaginary\*b.imaginary));

}

String print(){

String ans="";

if(real==0&&imaginary==0) ans="0";

else if(real==0&&imaginary!=0) {

if(Math.abs(imaginary-(int)imaginary)<=1e-6) ans=(int)imaginary+"i";

else ans=imaginary+"i";

}else if(real!=0&&imaginary==0) {

if(Math.abs(real-(int)real)<=1e-6) ans=String.valueOf((int)real);

else ans=String.valueOf(real);

}else {

if(imaginary>0) {//ans=real+"+"+imaginary+"i";

if(Math.abs(imaginary-(int)imaginary)<=1e-6&&Math.abs(real-(int)real)<=1e-6) {

ans=(int)real+"+"+(int)imaginary+"i";

}else if(Math.abs(imaginary-(int)imaginary)<=1e-6) {

ans=real+"+"+(int)imaginary+"i";

}else if(Math.abs(real-(int)real)<=1e-6) {

ans=(int)real+"+"+imaginary+"i";

}else {

ans=real+"+"+imaginary+"i";

}

}else {//ans=real+(imaginary+"i");

if(Math.abs(imaginary-(int)imaginary)<=1e-6&&Math.abs(real-(int)real)<=1e-6) {

ans=(int)real+((int)imaginary+"i");

}else if(Math.abs(imaginary-(int)imaginary)<=1e-6) {

ans=real+((int)imaginary+"i");

}else if(Math.abs(real-(int)real)<=1e-6) {

ans=(int)real+(imaginary+"i");

}else {

ans=real+(imaginary+"i");

}

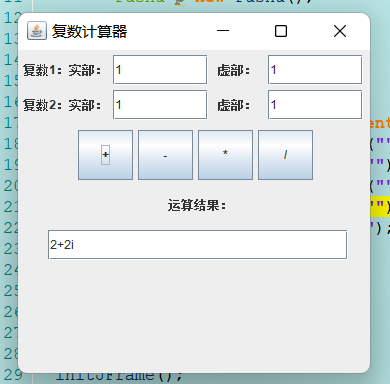
}

}

return ans;

}

}

运行结果：

发现的问题和解决方法：

**1：按钮的布局问题，采用网格布局还添加了空标签来实现题目所要的界面布局。**

**2：复数的输出格式问题，只能多种情况一点点分类讨论。**

3.设计型实验

（1）

分析：

1. 请描述本实验的解决方案

**先添加组件布局搭建好页面，在进行具体的事件响应，主要包括两个，一个获得所填写的内容，一个是点击下拉列表后的响应。**

1. 请以本题为例，描述Java事件处理方法和步骤。

**创建事件源：所有的按钮，文本框，文本区，单选多选的选项->自定义事件监听器Xinxi和Mylistener1 ->为事件源注册监听器进行事件响应**。

源代码：package prj\_xinxishouji;

import java.awt.\*;

import javax.swing.\*;

import java.awt.event.\*;

public class Main {

public static void main(String[] args) {

Xinxi x=new Xinxi();

}

}

class Xinxi extends JFrame implements ItemListener{

JTextField nametext;

JTextField addresstext;

JComboBox choiceprv;

JComboBox choiceshi;

JComboBox choicearea;

JTextArea awardsarea;

JButton btn;

JLabel name=new JLabel("姓名：");

JLabel gender=new JLabel("性别：");

JLabel age=new JLabel("年龄：");

JLabel address=new JLabel("地址：");

JLabel province=new JLabel("省份：");

JLabel shi=new JLabel("市：");

JLabel area=new JLabel("区/县：");

JLabel hobby=new JLabel("爱好：");

JLabel awards=new JLabel("曾获奖项：");

CheckboxGroup gendercg=new CheckboxGroup();

CheckboxGroup agecg=new CheckboxGroup();

ButtonGroup group1 = new ButtonGroup();

ButtonGroup group2 = new ButtonGroup();

JRadioButton man = new JRadioButton("男",false);//创建单选框，true为默认选中，不需要可去掉

JRadioButton woman = new JRadioButton("女",false);

JRadioButton age1 = new JRadioButton("18岁以下",false);

JRadioButton age2 = new JRadioButton("18-35",false);

JRadioButton age3 = new JRadioButton("36-59",false);

JRadioButton age4 = new JRadioButton("60岁以上",false);

JCheckBox c1=new JCheckBox("阅读",null,false);

JCheckBox c2=new JCheckBox("音乐",null,false);

JCheckBox c3=new JCheckBox("玩游戏",null,false);

JCheckBox c4=new JCheckBox("羽毛球",null,false);

JCheckBox c5=new JCheckBox("跳舞",null,false);

JCheckBox c6=new JCheckBox("演讲",null,false);

JPanel pn1=new JPanel();

JPanel pn2=new JPanel();

JPanel pn3=new JPanel(new BorderLayout());

JPanel pn4=new JPanel(new GridLayout(2,3));

JPanel pn5=new JPanel();

JPanel pn6=new JPanel();

public Xinxi(){

initJFrame();

Box boxh1=Box.*createHorizontalBox*();

Box boxh2=Box.*createHorizontalBox*();

Box boxh3=Box.*createHorizontalBox*();

Box boxv1=Box.*createVerticalBox*();

awardsarea=new JTextArea(5,20);

nametext=new JTextField(5);

addresstext=new JTextField(15);

choiceprv=new JComboBox();

choiceshi=new JComboBox();

choicearea=new JComboBox();

btn=new JButton("提交");

choiceprv.addItem("安徽");

choiceprv.addItem("江苏");

choiceprv.addItem("云南");

choiceprv.setSelectedIndex(-1);

pn1.add(name);

pn1.add(nametext);

pn1.add(gender);

group1.add(man);//将单选框组件加入单选框按钮组，否则两个都可以选择

group1.add(woman);

pn1.add(man);

pn1.add(woman);

pn2.add(age);

group2.add(age1);

group2.add(age2);

group2.add(age3);

group2.add(age4);

pn2.add(age1);

pn2.add(age2);

pn2.add(age3);

pn2.add(age4);

choiceprv.setSize(200, 30);

choiceprv.setLocation(130,60);

choiceshi.setSize(200, 30);

choiceshi.setLocation(130,60);

choicearea.setSize(200, 30);

choicearea.setLocation(130,60);

choiceprv.addItemListener((ItemListener)this);

choiceshi.addItemListener((ItemListener)this);

choicearea.addItemListener((ItemListener)this);

boxh1.add(address);

boxh1.add(province);

boxh1.add(choiceprv);

boxh1.add(shi);

boxh1.add(choiceshi);

boxh2.add(area);

boxh2.add(choicearea);

pn3.add(boxh1,BorderLayout.*NORTH*);

pn3.add(boxh2,BorderLayout.*CENTER*);

pn3.add(addresstext,BorderLayout.*SOUTH*);

pn5.add(hobby);

pn4.add(c1);

pn4.add(c2);

pn4.add(c3);

pn4.add(c4);

pn4.add(c5);

pn4.add(c6);

pn5.add(pn4);

pn6.add(awards);

pn6.add(awardsarea);

add(pn1);

add(pn2);

add(pn3);

add(pn5);

add(pn6);

Mylistener1 listen1=new Mylistener1();

listen1.setXinxi(this);

btn.addActionListener(listen1);

add(btn);

}

private void initJFrame() {

this.setSize(500,500);

this.setDefaultCloseOperation(JFrame.*EXIT\_ON\_CLOSE*);//关闭模式设置

this.setTitle("");

this.setLocationRelativeTo(null);//居中

this.setLayout(new FlowLayout());//清除格式

this.setVisible(true);

}

public void itemStateChanged(ItemEvent e){

if(e.getSource()==choiceprv) {

JComboBox com =(JComboBox)e.getSource();

String str = com.getSelectedItem().toString();

System.*out*.println(str);

String[] province = getCityByProvince(str);

choiceshi.removeAllItems();

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

choiceshi.addItem(province[i]);

}

//choiceshi.setSelectedIndex(-1);

}else if(e.getSource()==choiceshi) {

JComboBox com =(JComboBox)e.getSource();

String str = com.getSelectedItem().toString();

String[] shi = getQuNames(str);

choicearea.removeAllItems();

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

choicearea.addItem(shi[i]);

}

choicearea.setSelectedIndex(-1);

}

}

private String[] getCityByProvince(String str) {

String[] city = new String[3];

if( str.equals("安徽")){

city[0] = "芜湖";

city[1] = "滁州";

city[2] = "合肥";

}

if( str.equals("江苏")){

city[0] = "南京";

city[1] = "无锡";

city[2] = "苏州";

}

if( str.equals("云南")){

city[0] = "玉溪";

city[1] = "昆明";

city[2] = "保山";

}

return city;

}

private String[] getQuNames(String str){

String[] qu = new String[3];

if( str.equals("芜湖")){

qu[0]="弋江区";

qu[1]="镜湖区";

qu[2]="鸠江区";

}

if( str.equals("滁州")){

qu[0]="琅琊区";

qu[1]="南谯区";

qu[2]="明光市";

}

if( str.equals("合肥")){

qu[0]="瑶海区";

qu[1]="庐阳区";

qu[2]="蜀山区";

}

if( str.equals("南京")){

qu[0]="浦口区";

qu[1]="鼓楼区";

qu[2]="玄武区";

}

if( str.equals("无锡")){

qu[0]="锡山区";

qu[1]="惠山区";

qu[2]="滨湖区";

}

if( str.equals("苏州")){

qu[0]="姑苏区";

qu[1]="相城区";

qu[2]="虎丘区";

}

if( str.equals("玉溪")){

qu[0]="红塔区";

qu[1]="江川区";

qu[2]="易门县";

}

if( str.equals("昆明")){

qu[0]="五华区";

qu[1]="盘龙区";

qu[2]="西山区";

}

if( str.equals("保山")){

qu[0]="隆阳区";

qu[1]="龙陵县";

qu[2]="昌宁县";

}

return qu;

}

}

class Mylistener1 implements ActionListener{

Xinxi x;

public void setXinxi(Xinxi x) {

this.x=x;

}

public void actionPerformed(ActionEvent e) {

if(e.getSource()==x.btn) {

StringBuffer info=new StringBuffer();

info.append(x.name.getText());

info.append(x.nametext.getText()+"\n");

info.append(x.gender.getText());

if(x.man.isSelected()) {

info.append(x.man.getText()+"\n");

}

if(x.woman.isSelected()) {

info.append(x.woman.getText()+"\n");

}

info.append(x.age.getText());

if(x.age1.isSelected()) {

info.append(x.age1.getText()+" ");

}

if(x.age2.isSelected()) {

info.append(x.age2.getText()+" ");

}

if(x.age3.isSelected()) {

info.append(x.age3.getText()+" ");

}

if(x.age4.isSelected()) {

info.append(x.age4.getText()+" ");

}

info.append("\n");

info.append(x.address.getText());

info.append(x.province.getText());

info.append(x.choiceprv.getSelectedItem()+" ");

info.append(x.shi.getText());

info.append(x.choiceshi.getSelectedItem()+"\n");

info.append(x.area.getText());

info.append(x.choicearea.getSelectedItem()+"\n");

info.append("详细地址：");

info.append(x.addresstext.getText());

info.append("\n");

info.append(x.hobby.getText());

Component[] jcbs = x.pn4.getComponents();//将勾选的内容存入数组

for(Component component : jcbs) {

JCheckBox jcb = (JCheckBox) component;

if(jcb.isSelected()) {

info.append(jcb.getText()+" ");

}

}

info.append("\n");

info.append(x.awards.getText());

info.append(x.awardsarea.getText());

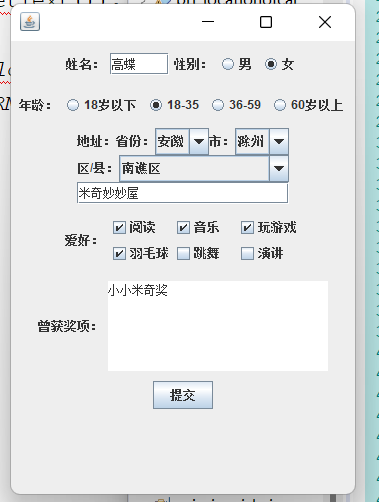
JOptionPane.*showMessageDialog*(new Xinxi(),"请确认信息"+"\n"+info, "信息", JOptionPane.*INFORMATION\_MESSAGE*);

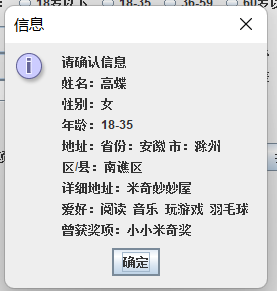
}

}

}

运行结果：





发现的问题和解决方法：

**1：单选和多选的实现和具体填写内容的获取问题，查看老师的ppt课件找到相关案例学习相应方法。**

**2：如何实现点击一个省份自动出现相应的市，一个省份选项被选中时把相应的市给添加进去，这样区县的选项中就可以自动出现相应的市了。在点击不同的省份后，相应的市的选项需要清空。**

# 实验结果及分析

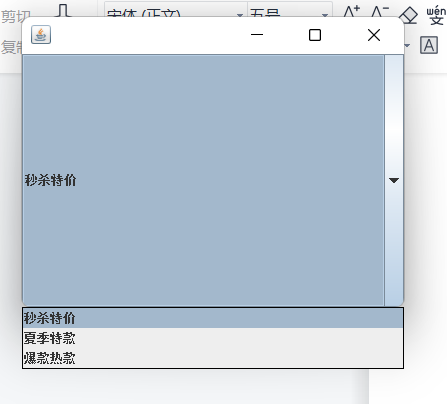
1.验证型实验

（1）分析程序错误原因，分析程序的执行过程，给出其运行结果

**错误原因：setVisible(true)放在程序前面的位置。**

**执行过程：首先在主方法中创建容器对象，设置容器大小，创建下拉框，将选项添加进去，再将下拉框添加到容器中，设置为可见，再关闭。**

**运行结果：**



2.应用型实验

（1）上一部分已具体填写运行结果和分析。

（2）上一部分已具体填写运行结果和分析。

（3）上一部分已具体填写运行结果和分析。

3.设计型实验

（1）上一部分已具体填写运行结果和分析。

# 实验小结

[本次实验的反思、收获和体会]

**反思：**

**1：这一块有很多地方需要自己去了解相应实现方法，因此如何快速准确地查找到相应的信息就显得很重要，看课件，查书，百度，问同学，问老师等等途径需要自行探索什么情况适合什么样的解决办法。**

**2：网上找的也好，问同学的也好，具体做法都应该知道怎么用为什么用，而不是仅仅为了写出这一题。**

**收获：这些题目基本上包括了各组件的常用使用方法，通过这些题目的巩固，基本上了解了大致用法。如：单选先用JRadioButton创建，再将其放入ButtonGroup中设置为只一项可被选上，再添加入相应的容器。多选则直接使用JCheckBox创建对象添加。在获得单选多选的具体勾选内容时，需要先判断是否需按选择上isSelected()再获取。对于下拉列表，要想一开始不显示具体选项需设置setSelectedIndex(-1);在输入密码时要不显示具体内容，需创建JPasswordField对象 ，并用getPassword()获得密码。**

**体会：这一块还有好多东西不了解，需进一步学习，同时也希望自己能用到这些知识自己做一点有意思的东西出来。**