

实验内容	第 10 周 GUI 设计 (2)			成绩	
姓 名	王秋锋	学号	2015111948	班 级	计算机 2015-03 班
专 业	计算机科学与技术			日 期	2017 年 11 月 10 日

**【实验目的】--GUI 设计**

- ◆ 掌握 JAVA 语言 GUI 设计的概念；掌握 AWT 常用组件类；
- ◆ 掌握 JAVA 语言系统常用组件类的功能和使用；
- ◆ 掌握 JAVA GUI 委托事件处理机制。

**【实验内容】****1、编辑、编译、运行下面 java 程序**

```
import java.awt.*;
public class MenuTestEx01{
    public static void main(String args[]){
        Frame fr=new Frame("Menu");
        MenuBar mb=new MenuBar();
        fr.setMenuBar(mb);
        Menu m1=new Menu("File");
        Menu m2=new Menu("Format");
        Menu m3=new Menu("Help");
        Menu m4=new Menu("进制");
        mb.add(m1);      mb.add(m2);
        mb.setHelpMenu(m3);
        MenuItem m21=new MenuItem("中文");
        MenuItem m41=new MenuItem("二进制");
        MenuItem m42=new MenuItem("八进制");
        MenuItem m43=new MenuItem("十进制");
        m2.add(m21);      m2.add(m4);
        m4.add(m41);      m4.add(m42);      m4.add(m43);
        fr.pack();
        fr.setSize(300,100);
        fr.setLocation(100,100);
        fr.setVisible(true);
    }
}
```

**要求：**(1) 分析、运行该程序，写出运行结果；

(2) 在 File 菜单中，添加“Exit”菜单项，选择 Exit 菜单项，程序退出

(3) 在 Help 菜单中，添加“About”菜单项，选择 About 菜单项，弹出对话框

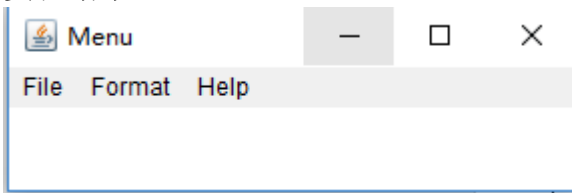
**【实验结果与分析】**

(1)

实验分析：

首先创建 frame 窗体，然后定义组件 File、Format、Help，然后在 Format 中添加子组件，将组件通过 add 方法添加到窗体中，接着对窗体的大小和位置进行设置，通过调用 setVisible(true) 让窗体显示

实验结果:



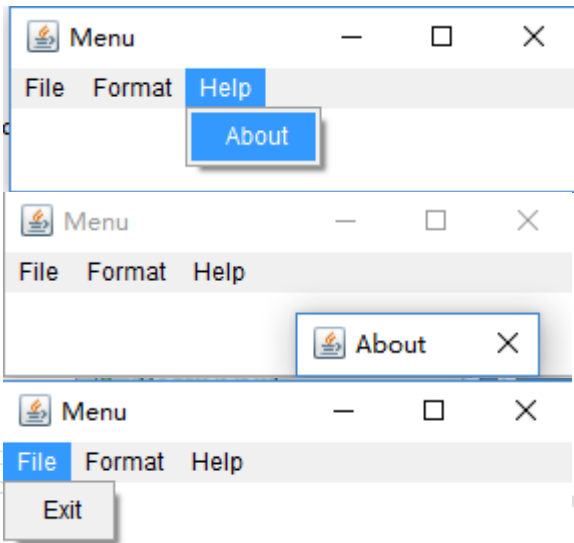
(2)(3)的实验代码:

```
import java.awt.*;
import java.awt.event.*;
import java.awt.Dialog;

public class MenuTest extends WindowAdapter implements ActionListener{
    Frame fr;
    Dialog AboutDia;
    public void display() {
        fr=new Frame("Menu");
        MenuBar mb=new MenuBar();
        fr.setMenuBar(mb);
        fr.addWindowListener(this);
        Menu m1=new Menu("File");
        Menu m2=new Menu("Format");
        Menu m3=new Menu("Help");
        Menu m4=new Menu("进制");
        mb.add(m1);      mb.add(m2);
        mb.setHelpMenu(m3);
        MenuItem m11 = new MenuItem("Exit");
        MenuItem m21=new MenuItem("中文");
        MenuItem m31 = new MenuItem("About");
        MenuItem m41=new MenuItem("二进制");
        MenuItem m42=new MenuItem("八进制");
        MenuItem m43=new MenuItem("十进制");
        m1.add(m11); m1.addActionListener(this);
        m2.add(m21); m2.add(m4);
        m3.add(m31); m3.addActionListener(this);
        m4.add(m41); m4.add(m42); m4.add(m43);
        fr.pack();
        fr.setSize(300,100);
        fr.setLocation(100,100);
        fr.setVisible(true);
        AboutDia = new Dialog(fr, "About", true);
    }
    public static void main(String args[]) {
        (new MenuTest()).display();
    }
}
```

```
public void windowClosing(WindowEvent e){
    System.exit(0);
}
public void actionPerformed(ActionEvent e) {
    if (e.getActionCommand() == "Exit") System.exit(0);
    else if (e.getActionCommand() == "About") {
        AboutDia.setVisible(true);
    }
}
}
```

实验结果:



## 2、编辑、编译、运行下面 java 程序

```
import java.io.*;
import java.awt.*;
import java.awt.event.*;
public class TestGUIDemo01 extends Frame implements ItemListener
{
    Panel lpanel1,lpanel2;
    CheckboxGroup lcheckboxgroup;
    Checkbox plain,bold,italic;
    Button lbutton;
    public TestGUIDemo01(String title)
    {
        super(title);
    }
    public void creat()
    {
        lpanel1=new Panel();
        lpanel2=new Panel();
```

```
lpanel1.setLayout(new FlowLayout());
lpanel2.setLayout(new FlowLayout());
lcheckboxgroup=new CheckboxGroup();
plain=new Checkbox("普通",lcheckboxgroup,true);
plain.addItemListener(this);
bold=new Checkbox("黑体",lcheckboxgroup,false);
bold.addItemListener(this);
italic=new Checkbox("斜体",lcheckboxgroup,false);
italic.addItemListener(this);
lbutton=new Button("西南交通大学");
lpanel2.add(lbutton);
lpanel1.add(plain);
lpanel1.add(bold);
lpanel1.add(italic);
setLayout(new GridLayout(2,1));
add(lpanel1);
add(lpanel2);
setSize(300,400);
setVisible(true);
addWindowListener(new WindowAdapter()           //匿名内部类完成事件处理
{
    public void windowClosing(WindowEvent e)
    {
        System.exit(0);
    }
});
}
public void itemStateChanged(ItemEvent e)
{
    if(e.getSource()==plain)
    {
        lbutton.setFont(new Font("SansSerif",Font.PLAIN,14));
    }
    else if(e.getSource()==bold)
    {
        lbutton.setFont(new Font("SansSerif",Font.BOLD,14));
    }
    else
        lbutton.setFont(new Font("SansSerif",Font.ITALIC,14));
}

public static void main(String[] args)
{
    TestGUIDemo01 fr=new TestGUIDemo01("GUI 测试实例");
    fr.creat();
}
}
```

**要求:**

- (1) 运行、分析该程序
- (2) 修改该程序,把按钮放置在窗体底部,在窗体中央部位,添加文本区组件 (TextArea)  
通过单选按钮,改变文本区组件中的文本颜色  
修改程序,给出程序运行结果

**【实验结果与分析】**

(1)

## 实验分析:

本程序在面板上加上 3 个复选框 `CheckBox` 类, 然后用 `ItemListener` 接口监听事件, 调整文字的字体。

## 实验结果:



## (2) 实验代码:

```
import java.awt.*;
import java.awt.event.*;

public class TestGUIDemo extends Frame implements ItemListener {
    Panel lpanel1, lpanel2;
    CheckboxGroup lcheckboxgroup;
    Checkbox plain, bold, italic;
    Button lbutton;
    TextArea ta;

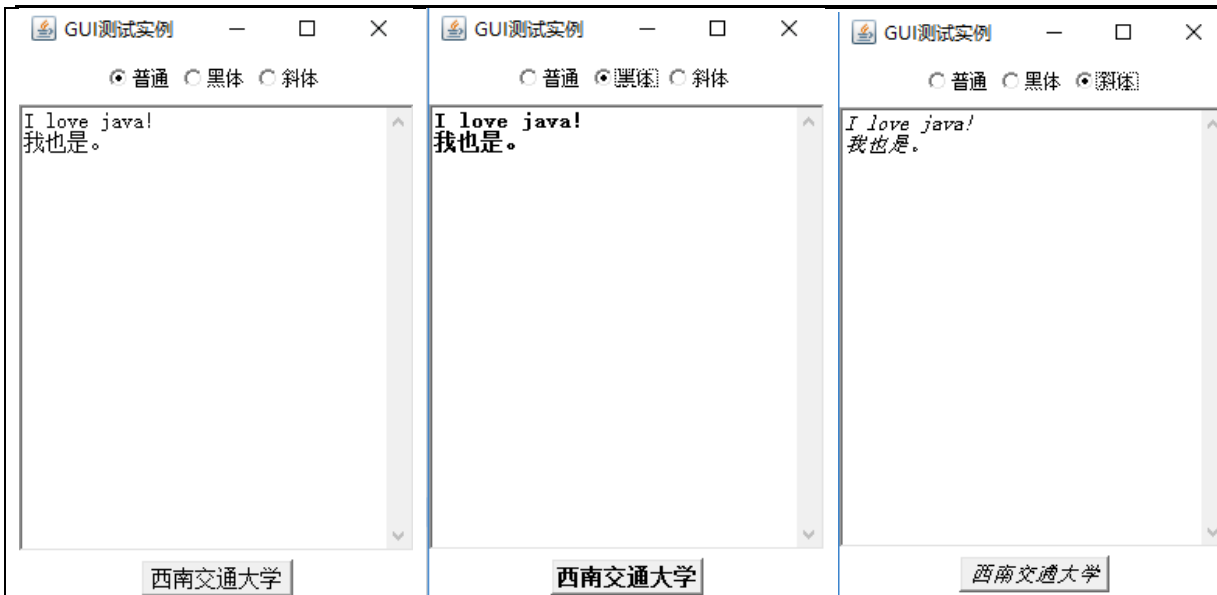
    public TestGUIDemo(String title){
        super(title);
    }

    public void creat() {
        lpanel1 = new Panel();
        lpanel2 = new Panel();

        lpanel1.setLayout(new FlowLayout());
        lpanel2.setLayout(new FlowLayout());
        lcheckboxgroup = new CheckboxGroup();
        plain = new Checkbox("普通", lcheckboxgroup, true);
        plain.addItemListener(this);
        bold = new Checkbox("黑体", lcheckboxgroup, false);
        bold.addItemListener(this);
        italic = new Checkbox("斜体", lcheckboxgroup, false);
        italic.addItemListener(this);
```

```
lbutton = new Button("西南交通大学");
lpanel2.add(lbutton);
lpanel1.add(plain);
lpanel1.add(bold);
lpanel1.add(italic);
ta = new TextArea();
setLayout(new BorderLayout());
add("North", lpanel1);
add("Center", ta);
add("South", lpanel2);
setSize(300, 400);
setVisible(true);
addWindowListener(new WindowAdapter() {
    public void windowClosing(WindowEvent e) {System.exit(0);}
});
}
public void itemStateChanged(ItemEvent e) {
    if (e.getSource() == plain) {
        ta.setFont(new Font("SansSerif", Font.PLAIN, 14));
        lbutton.setFont(new Font("SansSerif", Font.PLAIN, 14));
    }
    else if (e.getSource() == bold) {
        ta.setFont(new Font("SansSerif", Font.BOLD, 14));
        lbutton.setFont(new Font("SansSerif", Font.BOLD, 14));
    }
    else if (e.getSource() == italic) {
        ta.setFont(new Font("SansSerif", Font.ITALIC, 14));
        lbutton.setFont(new Font("SansSerif", Font.ITALIC, 14));
    }
}
public static void main(String []args) {
    TestGUIDemo fr = new TestGUIDemo("GUI测试实例");
    fr.creat();
}
}
```

实验结果:



### 3、编辑并运行下面程序，

```
import java.awt.*;
import java.awt.event.*;
public class TestGUI02 extends Frame implements ActionListener
{
    Button btn1, btn2;
    TextField f, tf1, tf2;
    TextArea Area;
    TestGUI02()
    {
        super("添加组件的窗口");
        addWindowListener(new WindowAdapter()
        {
            public void windowClosing(WindowEvent e)
            {
                System.exit(0);
            }
        });
        setSize(350,250); //设置窗口大小
        setLocation(200,200); //设置窗口显示位置
        setFont(new Font("Arial",Font.PLAIN,12)); //设置字体
        setLayout(new FlowLayout());
        Area=new TextArea (6,40);
        tf1=new TextField(10); tf2=new TextField(10);
        btn1=new Button("显示"); btn2=new Button("退出");
        f=new TextField(20);
        add(Area); add(new Label("用户名"));
        add(tf1); add(new Label("电话"));
        add(tf2); add(f); add(btn1); add(btn2);
        tf1.addActionListener(this); tf2.addActionListener(this);
        btn1.addActionListener(this); btn2.addActionListener(this);
        show();
    }
    public static void main(String args[])
    {
        new TestGUI02();
    }
}
```

```
}  
public void actionPerformed(ActionEvent e)  
{  
    if (e.getSource()==btn1)  
        f.setText("你按下:"+ e.getActionCommand() + "按钮");  
    if (e.getSource()==tf1)  
        Area.append("用户名: "+tf1.getText()+"\n");  
    if (e.getSource()==tf2)  
        Area.append("电 话: "+tf2.getText()+"\n");  
    if (e.getSource()==btn2) {  
        exit(0);  
    }  
}
```

**要求：**理解掌握 ActionEvent 事件的处理，给出程序运行结果

- (1) 输入用户名，并回车；
- (2) 输入电话，并回车
- (3) 点击“显示按钮”

### 【实验结果与分析】

实验结果：



### 4、编译运行下面程序，理解 Java 委托事件处理机制和鼠标事件的处理

```
import java.awt.*;  
import java.awt.event.*;  
public class TestMultiListener implements MouseMotionListener, MouseListener {  
    Frame f = new Frame("多重监听器测试"); //创建窗口框架容器组件对象  
    TextField tf = new TextField(30); //创建文本框对象，长度为 30 个字符  
    public TestMultiListener(){ //定义多重监听构造方法  
        f.add(new Label("请按下鼠标左键并拖动"), "North"); //在窗口容器中添加标签  
        f.add(tf, "South"); //在窗口容器中添加文本框（底部）  
        f.setBackground(new Color(120,175,175)); //设置窗口背景色  
        //在窗口容器中注册(添加)鼠标移动监听器（this 代表窗口实例），实现多重监听  
        f.addMouseMotionListener(this);  
        //在窗口容器中注册(添加)鼠标监听器（this 代表窗口），实现多重监听  
        f.addMouseListener(this);  
        f.setSize(300, 200); //设置窗口容器大小
```

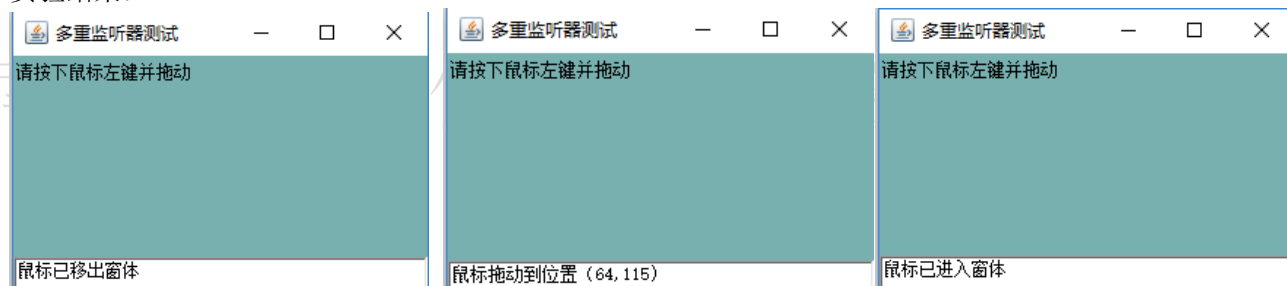


```
f.setVisible(true);           //设为可视
}
public static void main(String args[]) {
    TestMultiListener t = new TestMultiListener(); //创建 TestMultiListener 类的对象实例
}
public void mouseDragged(MouseEvent e) { //定义鼠标拖动方法
    String s = "鼠标拖动到位置 (" + e.getX() + "," + e.getY() + ")"; //取得鼠标坐标
    tf.setText(s);           //设置文本框显示鼠标拖动坐标
}
public void mouseEntered(MouseEvent e) { //定义鼠标进入窗体方法
    String s = "鼠标已进入窗体";
    tf.setText(s);           //设置文本框显示鼠标已进入窗体字符串
}
public void mouseExited(MouseEvent e) { //定义鼠标退出窗体方法
    String s = "鼠标已移出窗体";
    tf.setText(s);
}
public void mouseMoved(MouseEvent e) {} //鼠标移动方法
public void mousePressed(MouseEvent e) {} //鼠标按下方法
public void mouseClicked(MouseEvent e) {} //鼠标点击方法
public void mouseReleased(MouseEvent e) {} //鼠标释放组件方法
}
```

**要求：**编辑上面的程序，给出运行结果，并对程序进行分析，理解委托事件处理模型，理解掌握鼠标事件的处理。

### 【实验结果与分析】

实验结果：



实验分析：

首先我们定义了一个类 `TestMultiListener`，它同时继承了两个接口，为复合监听类。它的成员变量为一个窗口和一块文本框，在该类的构造函数中为窗口添加标签，文本框等设置，最重要的是它为这个窗口注册了两个监听器，可以监听鼠标产生的多组信息。接着我们定义了监听器的响应函数，去处理鼠标事件，并将处理结果在文本框中设置输出。

**5、设计一个简单计算器，**如下图所示。在“操作数”标签右侧的两个文本框输入操作数，当单击操作符+，-，×，÷按钮时，对两个操作数进行运算并将结果填入到“结果”标签右侧的文本框中。

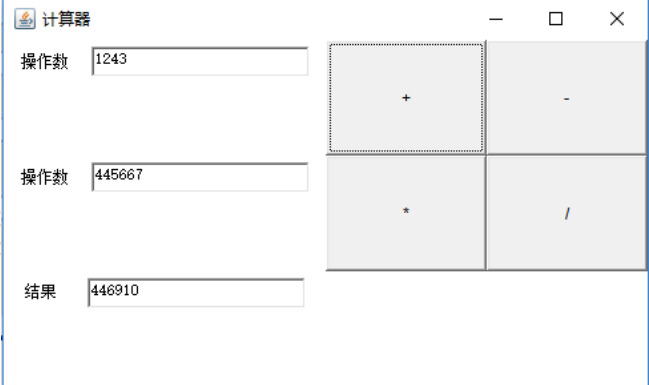
**要求：**编写程序，并给出程序运行结果



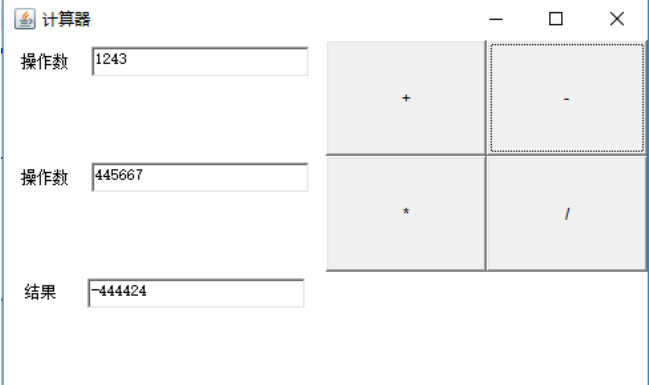
【实验结果与分析】

A、程序运行结果

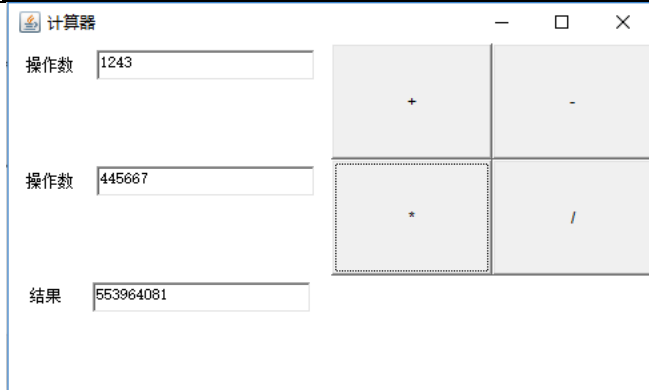
加法：



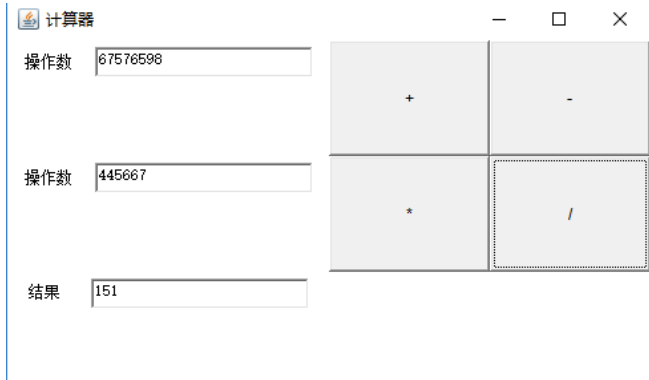
减法：



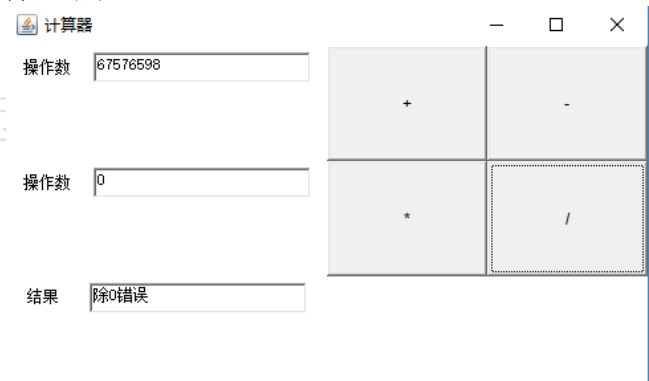
乘法：



除法:



除 0 错误:



B、程序源代码

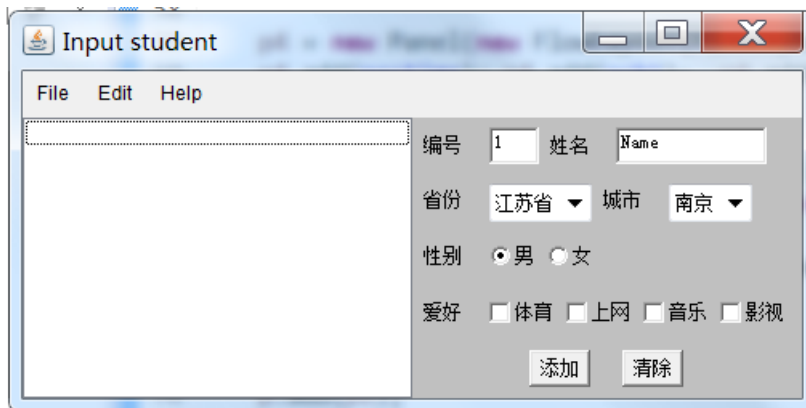
```
import java.awt.*;
import java.awt.event.*;

public class calculator extends WindowAdapter implements ActionListener
{
    Frame fr = new Frame("计算器");
    Button add, sub, mul, div;
    TextField tex1, tex2, tex3;
    public calculator() {
        Panel panel1 = new Panel();
        Panel panel2 = new Panel();
        panel1.setLayout(new GridLayout(3, 1));
        panel2.setLayout(new GridLayout(3, 1));
        Panel panel11 = new Panel();
```

```
Panel panel12 = new Panel();
Panel panel13 = new Panel();
panel11.setLayout(new FlowLayout());
panel12.setLayout(new FlowLayout());
panel13.setLayout(new FlowLayout());
tex1 = new TextField(20); tex2 = new TextField(20);
tex3 = new TextField(20);
panel11.add(new Label("操作数")); panel11.add(tex1);
panel12.add(new Label("操作数")); panel12.add(tex2);
panel13.add(new Label("结果 ")); panel13.add(tex3);
panel1.add(panel11); panel1.add(panel12); panel1.add(panel13);
add = new Button("+"); sub = new Button("-");
mul = new Button("*"); div = new Button("/");
add.addActionListener(this);
sub.addActionListener(this);
mul.addActionListener(this);
div.addActionListener(this);
panel2.add(add); panel2.add(sub);
panel2.add(mul); panel2.add(div);
fr.addWindowListener(this);
fr.setLayout(new GridLayout(1, 2));
fr.add(panel1); fr.add(panel2);
fr.setSize(500, 300);
fr.show();
}
public static void main(String []args) {
    calculator t = new calculator();
}
public void windowClosing(WindowEvent e) {
    System.exit(0);
}
public void actionPerformed(ActionEvent e) {
    try {
        int a = Integer.parseInt(tex1.getText());
        int b = Integer.parseInt(tex2.getText());
        if (e.getSource() == add) {
            int ans = a + b;
            tex3.setText(String.valueOf(ans));
        } else if (e.getSource() == sub) {
            int ans = a - b;
            tex3.setText(String.valueOf(ans));
        } else if (e.getSource() == mul) {
            int ans = a * b;
            tex3.setText(String.valueOf(ans));
        } else if (e.getSource() == div) {
            int ans = a / b;
```

```
        tex3.setText(String.valueOf(ans));  
    }  
} catch (ArithmeticException e1) {  
    tex3.setText("除0错误");  
}  
}  
}
```

6、设计一个 JAVA 程序模仿记事本（Notepad）的部分功能，界面参考如下图。



要求:

1. 学会建立菜单和弹出式菜单（对于菜单的使用，可自行设计）；
2. 点击“help→about”菜单，弹出一个对话框；
3. 可自行设计多个省份，以及每个省份的多个城市；
3. 编写程序

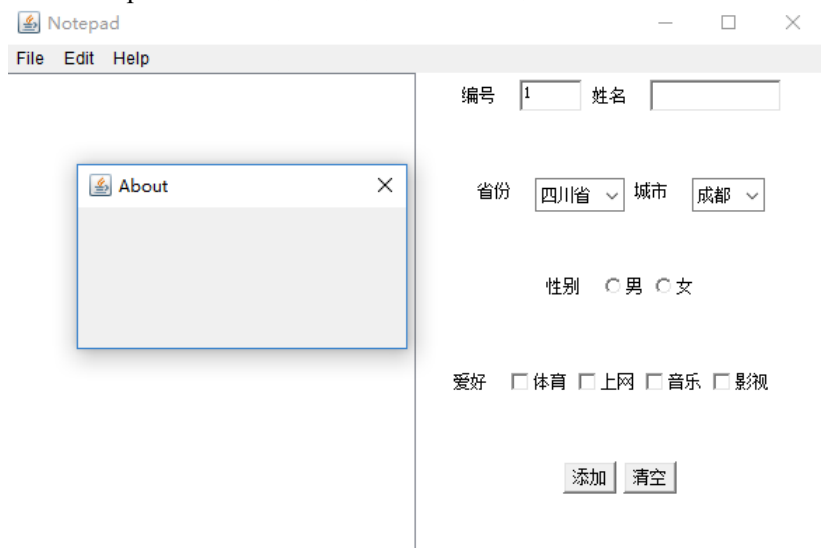
### 【实验结果与分析】

#### A、程序运行结果

初始界面:



点击“help-->about”菜单，弹出一个对话框



选择不同的省份之后会出现不同的下拉菜单



添加信息之后的结果：

The screenshot shows a Notepad window with a list of records on the left and a form on the right. The list contains three entries: 1 王秋锋 四川省 成都 男 体育 上网 音乐 影视, 2 aaa 江苏省 南京 女 体育 影视, and 3 Bob 江苏省 南通 男 音乐 影视. The form on the right has fields for 编号 (4), 姓名 (Bob), 省份 (江苏省), 城市 (南通), 性别 (男), and 爱好 (音乐, 影视). There are buttons for 添加 and 清空.

最后清空键清空所有:

The screenshot shows the same Notepad window after clicking the '清空' button. The list of records is now empty, and the form on the right remains the same.

#### B、程序源代码

```
import java.awt.*;
import java.awt.event.*;

public class Notepad extends WindowAdapter implements ActionListener,
ItemListener
{
    Frame fr;
    MenuBar mb;
    Choice c1, c2;
    CheckboxGroup cg;
    TextField tex1, tex2;
    Panel panel1, panel2;
    Checkbox cb1, cb2; //sex
    Panel panel21, panel22, panel23, panel24, panel25;
    Menu m1, m2, m3; // File, Edit, Help
    MenuItem m11, m31; // Exit, About
```

```
List list; Dialog AboutDia;
Button button1, button2; //add,clear
Checkbox chk41, chk42, chk43, chk44; //hobby
public Notepad() {
    fr = new Frame("Notepad");
    mb = new MenuBar();
    m1 = new Menu("File");
    m2 = new Menu("Edit");
    m3 = new Menu("Help");
    m11 = new MenuItem("Exit");
    m11.addActionListener(this);
    m1.add(m11);
    m31 = new MenuItem("About");
    m31.addActionListener(this);
    m3.add(m31);
    AboutDia = new Dialog(fr, "About", true);
    AboutDia.setSize(250, 150);
    AboutDia.addWindowListener(this);
    mb.add(m1); mb.add(m2); mb.add(m3);

    panel1 = new Panel();
    panel1.setLayout(new GridLayout(1, 1));
    list = new List();
    panel1.add(list);

    panel2 = new Panel();
    panel2.setLayout(new GridLayout(5, 1));
    panel21 = new Panel();
    panel21.setLayout(new FlowLayout());
    panel21.add(new Label("编号"));
    tex1 = new TextField(3);
    tex1.setText("1");
    tex2 = new TextField(10);
    panel21.add(tex1);
    panel21.add(new Label("姓名"));
    panel21.add(tex2);
    panel22 = new Panel();
    panel22.setLayout(new FlowLayout());
    c1 = new Choice();
    c1.addItem("江苏省"); c1.addItem("四川省");
    c1.addItemListener(this);
    c2 = new Choice();
    c2.addItem(" ");
    panel22.add(new Label("省份"));
    panel22.add(c1);
    panel22.add(new Label("城市"));
```



```
panel122.add(c2);
panel123 = new Panel();
panel123.setLayout(new FlowLayout());
cg = new CheckboxGroup();
cb1 = new Checkbox("男", cg, false);
cb2 = new Checkbox("女", cg, false);
panel123.add(new Label("性别"));
panel123.add(cb1); panel123.add(cb2);
panel124 = new Panel();
panel124.setLayout(new FlowLayout());
panel124.add(new Label("爱好"));
chk41 = new Checkbox("体育", false);
chk42 = new Checkbox("上网", false);
chk43 = new Checkbox("音乐", false);
chk44 = new Checkbox("影视 ", false);
panel124.add(chk41); panel124.add(chk42);
panel124.add(chk43); panel124.add(chk44);
panel125 = new Panel();
panel125.setLayout(new FlowLayout());
button1 = new Button("添加");
button1.addActionListener(this);
button2 = new Button("清空");
button2.addActionListener(this);
panel125.add(button1); panel125.add(button2);
panel12.add(panel121); panel12.add(panel122);
panel12.add(panel123); panel12.add(panel124);
panel12.add(panel125);

fr.addWindowListener(this);
fr.setLayout(new GridLayout(1, 2));
fr.add(panel11); fr.add(panel12);
fr.setMenuBar(mb);
fr.setSize(600, 400);
fr.setVisible(true);
}
public static void main(String []args) {
    Notepad t = new Notepad();
}
public void windowClosing(WindowEvent e) {
    if (e.getSource() == AboutDia)
        AboutDia.setVisible(false);
    else
        System.exit(0);
}

public void itemStateChanged(ItemEvent e)
```

```
{
    String a = c1.getSelectedItem();
    c2.removeAll();
    if(a == "江苏省"){
        c2.addItem("南京");
        c2.addItem("苏州");
        c2.addItem("南通");
    }
    else{
        c2.addItem("成都");
        c2.addItem("绵阳");
        c2.addItem("乐山");
    }
}

public void actionPerformed(ActionEvent e) {
    if (e.getSource() == m11)
        System.exit(0);
    else if (e.getSource() == m31)
        AboutDia.setVisible(true);
    else if(e.getSource() == c1){

    }
    else if (e.getSource() == button1) {
        String s;
        s = tex1.getText() + " " + tex2.getText();
        s += " " + c1.getSelectedItem() + " " + c2.getSelectedItem();
        if (cb1.getState()) s += " " + cb1.getLabel();
        else s += " " + cb2.getLabel();
        if (chk41.getState()) s += " " + chk41.getLabel();
        if (chk42.getState()) s += " " + chk42.getLabel();
        if (chk43.getState()) s += " " + chk43.getLabel();
        if (chk44.getState()) s += " " + chk44.getLabel();
        list.add(s);
        tex1.setText(""+(Integer.parseInt(tex1.getText()) + 1));
    }
    else if (e.getSource() == button2)
        list.removeAll();
    }
}
```

## 7、利用 Swing 设计 GUI：编译运行下面程序，理解基于 Swing 组件的 GUI 设计

```
import javax.swing.*;
import java.awt.*;
```

```
import java.awt.event.*;
public class TestGUI03 extends JFrame implements ActionListener {
    JButton button1,button2,button3;
    JToolBar toolBar;
    JTextArea textArea;
    JScrollPane scrollPane;
    JPanel panel;
    public static void main(String[] args) {
        new TestGUI03();
    }
    public TestGUI03() {
        super("带有工具栏按钮的窗口");
        addWindowListener(new WindowAdapter() {
            public void windowClosing(WindowEvent e) {
                System.exit(0);
            }
        });
        button1=new JButton(new ImageIcon("image/01.gif"));
        button2=new JButton(new ImageIcon("image/02.gif"));
        button3=new JButton(new ImageIcon("image/03.gif"));
        button1.addActionListener(this);
        button2.addActionListener(this);
        button3.addActionListener(this);
        toolBar=new JToolBar();
        toolBar.add(button1);
        toolBar.add(button2);
        toolBar.add(button3);
        textArea=new JTextArea(6,30);
        scrollPane=new JScrollPane(textArea);
        panel=new JPanel();
        setContentPane(panel); //JFrame 不能直接添加组件，需要设置内容窗格(内容面板)
        panel.setLayout(new BorderLayout());
        panel.setPreferredSize(new Dimension(300,150));
        panel.add(toolBar,BorderLayout.NORTH);
        panel.add(scrollPane,BorderLayout.CENTER);
        pack(); //调整窗口大小，以适合其子组件的首选大小和布局
        setVisible(true);
    }
    public void actionPerformed(ActionEvent e) {
        String s="";
        if (e.getSource()==button1)
            s="左按钮被单击\n";
        else if (e.getSource()==button2)
            s="中按钮被单击\n";
        else if (e.getSource()==button3)
            s="右按钮被单击\n";
        textArea.append(s);
    }
}
```

**要求:**

- (1) 准备图片文件：在当前目录下建立一个 image 文件夹，存放 3 个 gif 图片文件（文件名：01.gif、02.gif、03.gif）
- (2) 理解掌握在 JFrame 框架中添加组件的方法
- (3) 给出程序运行结果

**【实验结果与分析】**

实验分析：

JFrame 添加组件的两种方式:

对 JFrame 添加组件有两种方式:

1) 用 `getContentPane()` 方法获得 JFrame 的内容面板, 再对其加入组件: `frame.getContentPane().add(childComponent)` 常分开来写

`Container container=getContentPane();`(隐式的 `this.getContentPane()`); 得到 jframe 的内容面板  
以后只要把容器加到 container 就可以了。

2) 建立一个 `JPanel` 或 `JDesktopPane` 之类的中间容器, 把组件添加到容器中, 用 `setContentPane()` 方法把该容器置为 JFrame 的内容面板:

```
JPanel contentPane = new JPanel();
```

```
//把其他组件添加到 JPanel 中
```

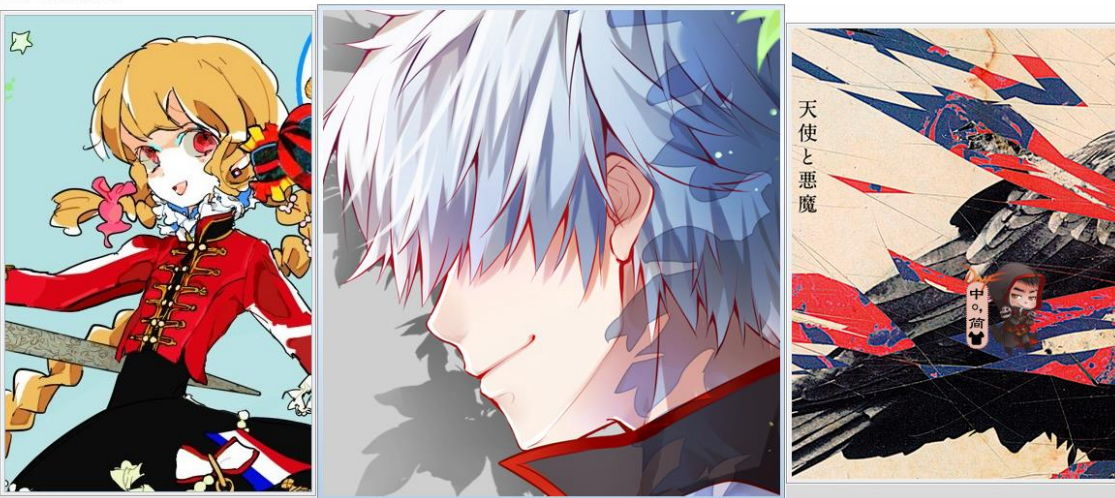
```
frame.setContentPane(contentPane);
```

```
//把 contentPane 对象设置成为 frame 的内容面板
```

实验结果:

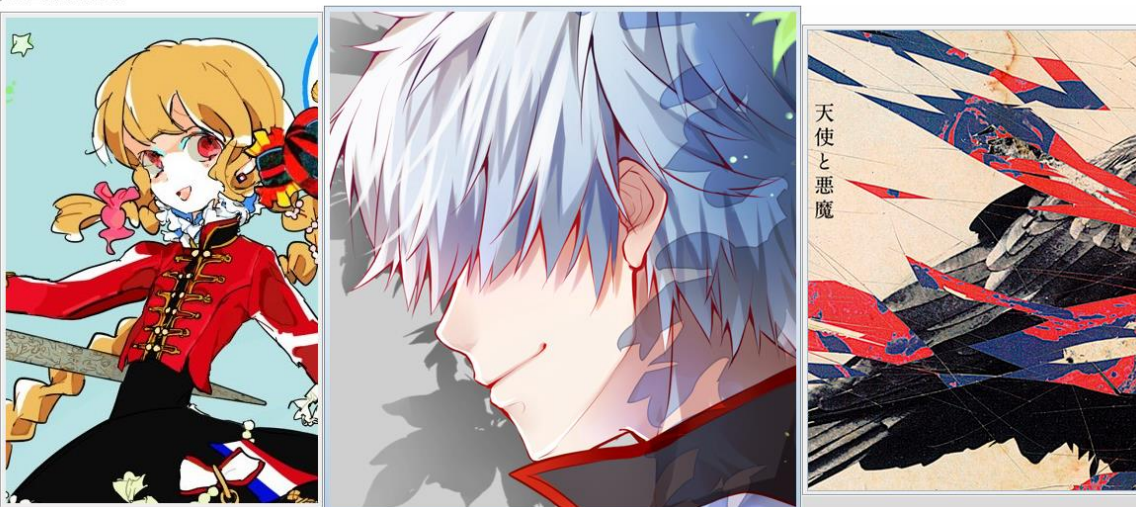
点击前:

带有工具栏按钮的窗口



点击后:

带有工具栏按钮的窗口



中按钮被单击  
左按钮被单击  
右按钮被单击

西南交通大学 信息学院--陈帆 仅供内部使用