

实验内容	第 11 周 Swing GUI 设计			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 17 日

【实验目的】--GUI 设计

- ◆ 掌握 JAVA 语言 GUI 设计的概念;
- ◆ 掌握 JAVA 语言 Swing 报的组件
- ◆ 能设计相对复杂的 GUI

【实验内容】**1、编辑、编译、运行下面 java 程序****//JTable 的练习与使用**

```
import java.awt.*;
import javax.swing.*;
public class Demo
{
    public static void main(String args[])
    {
        JFrame form = new JFrame("Table试验");
        final Object[]row = {"姓名","学号","语文成绩","数学成绩","总分"};
        Object[][] rowData = { {"张三","01","81","80","160"}, {"李四","02","93","72","165"},
                                {"王麻子","03","99","97","196"}
        };
        JTable tb= new JTable(rowData,row);
        JScrollPane pane = new JScrollPane(tb);
        form.getContentPane().add(tb);
        form.setVisible(true);
        form.pack();
    }
}
```

要求：给出程序运行结果

【实验结果与分析】

实验结果：

Table试验				
姓名	学号	语文成绩	数学成绩	总分
张三	01	81	80	160
李四	02	93	72	165
王麻子	03	99	97	196

实验分析:

调用 form.getContentPane().add(tb);实验结果不会显示姓名学号栏, 改成 form.getContentPane().add(pane);就可以了

JTable 是 Swing 中的表格控件:

创建 JTable

JTable 提供了 2 个构造器可以让你用数据和头部直接生成它:

JTable(Object[][] rowData, Object[] columnNames)

JTable(Vector rowData, Vector columnNames)

这两个构造器有一些特性你必须要注意:

1.JTable 所有的单元格都是可编辑的;

2.它将所有数据都当做 string 来处理。

2、编辑、编译、运行下面 java 程序

// JScrollPane 的练习与使用

```
import java.awt.Container;
import javax.swing.JFrame;
import javax.swing.JScrollPane;
import javax.swing.JTextArea;
import javax.swing.WindowConstants;

/*
 * 1:设置界面时,可能会遇到在一个较小的容器窗体中显示一个较大部分的内容的情况,这时
 * 可以使用JScrollPane面板
 *
 * 2:JScrollPane面板是带滚动条的面板,它也是一个面板,但是JScrollPane只能
 * 放置一个组件,并且不可以使用布局管理器
 *
 * 3:如果需要在JScrollPane面板中放置多个组件,需要将多个组件放置在JPanel面板上,
 * 然后将JPanel作为一个整体组件添加到JScrollPane组件上。
 *
 * 4:从本实例可以得到在窗体中创建一个带滚动条的文字编辑器,首先需要初始化编辑器,
 * 并且在初始化时完成编辑器的大小指定,当创建带滚动条的面板时,将编辑器加入面板中
 * ,最后将带滚动条的编译器放置在容器中即可
 */
public class Demo extends JFrame{
//导入包的快捷键 shift+ctrl+o
    public Demo(){//定义一个构造方法
        Container container=getContentPane();//创建一个容器

        //创建文本区域组件
        JTextArea ta=new JTextArea(20,50);

        //创建 JScrollPane()面板对象,并将文本域对象添加到面板中
        JScrollPane sp=new JScrollPane(ta);

        //将该面板添加到该容器中
        container.add(sp);
    }
}
```

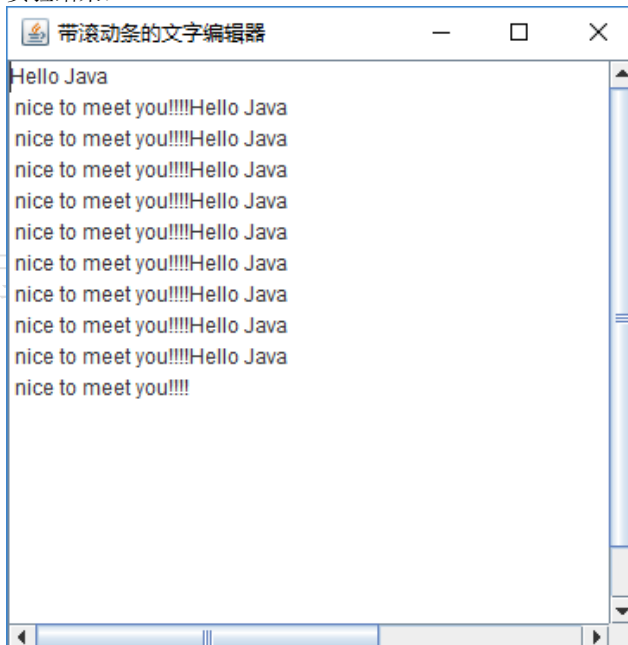
```
        for(int i = 0;i<10;i++)
        ta.append("Hello Java \n nice to meet you!!!!");
        //设置容器的外部特性
        setTitle("带滚动条的文字编辑器");//设置窗口的标题文字
        setSize(200,150);//设置窗口的大小
        setLocation(200,300);
        setVisible(true);//设置可视化
        //设置窗口的关闭方式
        setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
    }
    public static void main(String[] args) {

        Demo jp=new Demo();

    }
}
```

【实验结果与分析】

实验结果:



实验分析:

JScrollPane 的成员函数是 scrollRectToVisible(Rectangle rect); 该成员函数能够使某一块 rect 在当前的滚动面板内可见, 即滚动面板能够通过滚动来使得需要显示的区域(由 rect 界定出的那块区域)被显示。在使用了 JScrollPane 的 JTextArea 中, 当指定 JTextArea 大小后, 如何使当一行的数据过长时, 自动折行, 而不是出现横滚动条。本程序初始化 JTextArea, 再 add 到 JScrollPane, 并且设置可视化。

3、编辑并运行下面程序

// JSlider 的练习与使用

```
import javax.swing.*.*;
import javax.swing.border.*;
import java.awt.*.*;
import java.awt.event.*;
import javax.swing.event.*; //ChangeEvent 属于 Swing 事件
public class Demo implements ChangeListener {
    JFrame f = null;
```

```
JSlider slider1;
JSlider slider2;
JSlider slider3;
JLabel label1;
JLabel label2;
JLabel label3;

public Demo() {
    f = new JFrame("JSlider Example");
    Container contentPane = f.getContentPane();

    JPanel panel1 = new JPanel();
    panel1.setLayout(new GridLayout(2, 1));
    slider1 = new JSlider();// 建立一个默认的 JSlider 组件.
    label1 = new JLabel("目前刻度: " + slider1.getValue());
    panel1.add(label1);
    panel1.add(slider1);
    panel1.setBorder(BorderFactory.createTitledBorder(BorderFactory
        .createEtchedBorder(), "Slider 1", TitledBorder.LEFT,
        TitledBorder.TOP));

    JPanel panel2 = new JPanel();
    panel2.setLayout(new GridLayout(2, 1));
    /*
     * 下面五行程序建立一个水平方向 JSlider 组件,并设置其取大值,最小值,初始值与延伸区值,所谓的延伸区值
     * JScrollBar 限制 JSlider 刻度可变动的范围,也就是说延伸区就像是一个障碍区,是无法通行的.延伸区
     * 设得越大,刻度可变动的范围就越小
     * 例如若 minimum 值设为 0,maximan 值设为 100,而 extent 值设为 0,则 JSlider 刻度可变动
     * 的区域大小为 100-50-0=50 刻度(从 0-50).
     */
    slider2 = new JSlider(JSlider.HORIZONTAL);
    slider2.setMinimum(0);
    slider2.setMaximum(100);
    slider2.setValue(30);
    slider2.setExtent(50);
    label2 = new JLabel("目前刻度: " + slider2.getValue());
    panel2.add(label2);
    panel2.add(slider2);
    panel2.setBorder(BorderFactory.createTitledBorder(BorderFactory
        .createEtchedBorder(), "Slider 2", TitledBorder.LEFT,
        TitledBorder.TOP));

    JPanel panel3 = new JPanel();
    panel3.setLayout(new GridLayout(2, 1));
    // 下面两行建立一个具有最大最小值的 JSlider 组件,并设置此 JSlider 组件为垂直方向.
    slider3 = new JSlider(20, 80);
    slider3.setOrientation(JSlider.VERTICAL);
    label3 = new JLabel("目前刻度: " + slider3.getValue());
    panel3.add(label3);
    panel3.add(slider3);
    panel3.setBorder(BorderFactory.createTitledBorder(BorderFactory
        .createEtchedBorder(), "Slider 3", TitledBorder.LEFT,
        TitledBorder.TOP));

    slider1.addChangeListener(this);
    slider2.addChangeListener(this);
    slider3.addChangeListener(this);

    panel1.setPreferredSize(new Dimension(300, 100));
```

```
panel2.setPreferredSize(new Dimension(300, 100));
panel3.setPreferredSize(new Dimension(150, 200));

GridBagConstraints c;
int gridx, gridy, gridwidth, gridheight, anchor, fill, ipadx, ipady;
double weightx, weighty;
Insets inset;

GridBagLayout gridbag = new GridBagLayout();
contentPane.setLayout(gridbag);

gridx = 0; // 第 0 行
gridy = 0; // 第 0 列
gridwidth = 2; // 占两单位宽度
gridheight = 1; // 占一单位高度
weightx = 0; // 窗口增大时组件宽度增大比率 0
weighty = 0; // 窗口增大时组件高度增大比率 0
anchor = GridBagConstraints.CENTER; // 容器大于组件 size 时将组件
// 置于容器中央
fill = GridBagConstraints.BOTH; // 窗口拉大时会填满水平与垂
// 直空间
inset = new Insets(0, 0, 0, 0); // 组件间间距
ipadx = 0; // 组件内水平宽度
ipady = 0; // 组件内垂直高度
c = new GridBagConstraints(gridx, gridy, gridwidth, gridheight,
    weightx, weighty, anchor, fill, inset, ipadx, ipady);
gridbag.setConstraints(panel1, c);
contentPane.add(panel1);
gridx = 0;
gridy = 1;
c = new GridBagConstraints(gridx, gridy, gridwidth, gridheight,
    weightx, weighty, anchor, fill, inset, ipadx, ipady);
gridbag.setConstraints(panel2, c);
contentPane.add(panel2);

gridx = 2;
gridy = 0;
gridwidth = 1; // 占一单位宽度
gridheight = 2; // 占两单位高度
c = new GridBagConstraints(gridx, gridy, gridwidth, gridheight,
    weightx, weighty, anchor, fill, inset, ipadx, ipady);
gridbag.setConstraints(panel3, c);
contentPane.add(panel3);

f.pack();
f.setVisible(true);

f.addWindowListener(new WindowAdapter() {
    public void windowClosing(WindowEvent e) {
        System.exit(0);
    }
});
});
}

public static void main(String[] args) {
    new Demo();
}
```

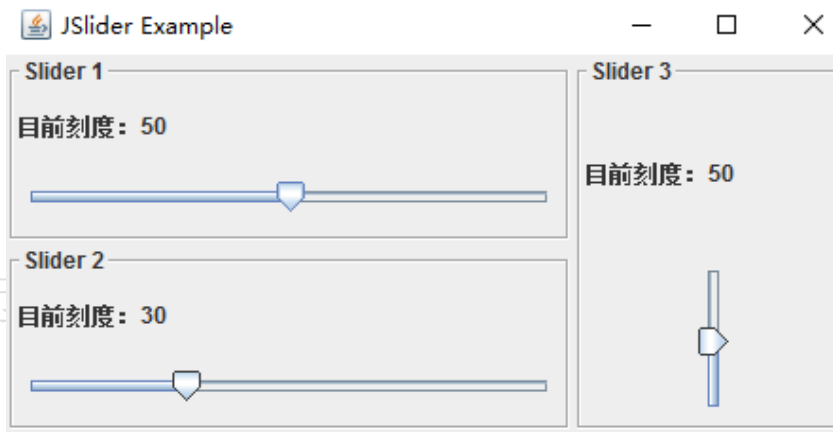
```
}

// 处理 ChangeEvent 事件,当用户移动滑动杆时,label 上的值会随着用户的移动而改变.
public void stateChanged(ChangeEvent e) {
    if ((JSlider) e.getSource() == slider1)
        label1.setText("目前刻度: " + slider1.getValue());
    if ((JSlider) e.getSource() == slider2)
        label2.setText("目前刻度: " + slider2.getValue());
    if ((JSlider) e.getSource() == slider3)
        label3.setText("目前刻度: " + slider3.getValue());
}
}
```

要求: 理解掌握 **JSlider** 及其应用

【实验结果与分析】

实验结果:



实验分析:

JSlider 不仅可以置于面版的任何地方,也可以在 JSlider 上标上刻度与数字

首先看下 JSlider 的构造函数:

JSlider():建立一个水平的 JSlider 对象,刻度从 0-100,初始刻度为 50.

JSlider(BoundedRangeModel brm):使用默认模式建立一个水平的 JSlider 对象.

JSlider(int orientation):建立一个自定义方向的 JSlider 对象,刻度从 0-100,初始刻度为 50.

JSlider(int min,int max):建立一个水平的 JSlider 对象,自定义刻度,从 min-max,初始刻度为 50.

JSlider(int min, int max,int value):建立一个水平的 JSlider 的对象,自定义刻度与初始值.

JSlider(int orientation,int min,int max,int value):建立一个自定义方向,刻度与刻度初始值的 JSlider 对象.

要使用 JSlider 组件就不得不提到 JSlider 事件的处理.当用户在 JSlider 上滑动杆时,就会产生 ChangeEvent 事件,若我们要处理 ChangeEvent 事件就必须实作 ChangeListener 界面,此界面定义了一个方法,那就是 stateChanged().通常我们在这个方法上会取得或置滑动杆的相关信息,例如滑动杆的延伸区(extent),最大最小值或滑动杆目前所在刻度等等。

本程序定义了两个水平放置的 JSlider 和一个垂直放置的 JSlider, 并且设置了不同的刻度范围。

4、编译运行下面程序, 理解掌握在颜色选择器 JColorChooser

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

```
public class Demo extends JFrame
{
    Container c;
    JButton btn=new JButton("选背景色");
    Color color=new Color(200,200,200);
    public Demo()
    {
        c=getContentPane();
        c.setLayout(new FlowLayout());
        c.add(btn);
        btn.addActionListener(new ActionListener()
        {
            public void actionPerformed(ActionEvent e)
            {
                color=JColorChooser.showDialog(null,"请选择你喜欢的颜色",color);
                if (color==null )
                    color=Color.lightGray;
                c.setBackground(color);
                c.repaint();
            }
        });
        setDefaultCloseOperation(WindowConstants.DISPOSE_ON_CLOSE);
        setSize(new Dimension(400,300));
        this.setVisible(true);
    }
}
```

要求：编辑上面的程序，给出运行结果。

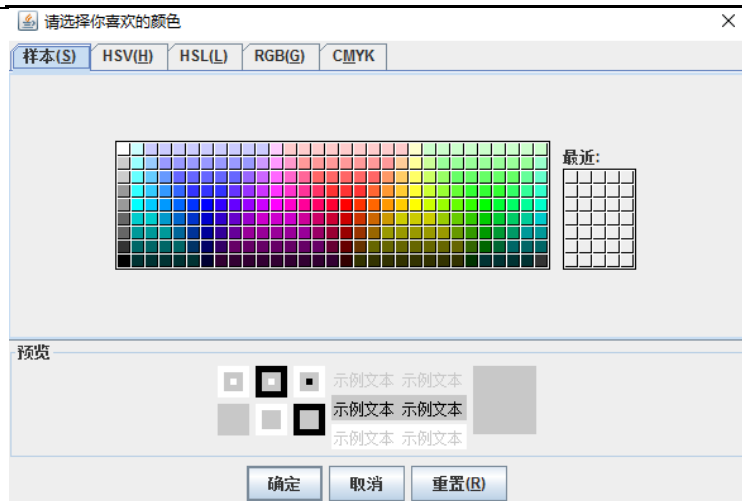
【实验结果与分析】

实验结果：

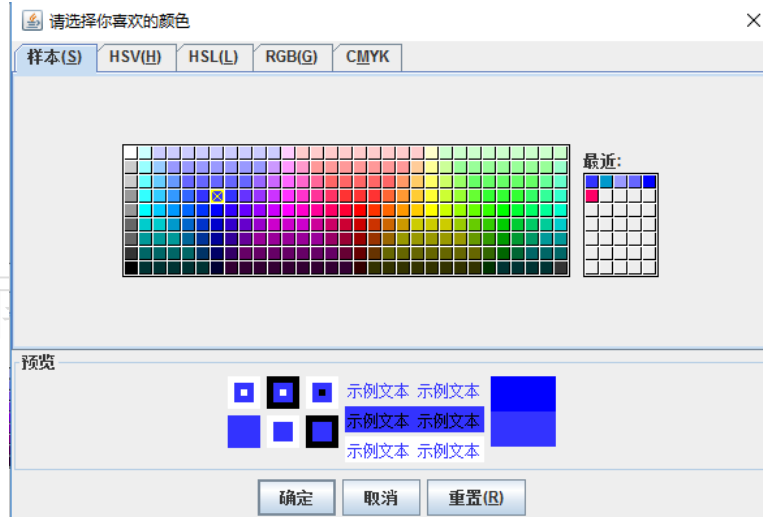
初始时无色：



按下选背景色选取颜色：



选几个颜色之后:



最终选取蓝色，按下确定：



实验分析：

JFrame 有一个 Content Pane，窗口能显示的所有组件都是添加在这个 Content Pane 中。JFrame 提供了两个方法：getContentPane 和 setContentPane 就是用于获取和设置其 Content Pane 的。所以本程序先定义 JFrame 然后调用 getContentPane 获取 Content Pane，接着鼠标点击自己喜欢的颜色就能设置背景颜色了。

5、编译运行下面程序，理解掌握在 JLabel 中显示图像

```
import javax.swing.*;
import java.awt.*;
public class Demo {

    public static void main(String[] args) {
        JFrame frame=new JFrame("Imagelcon Test");

        ImageIcon imagelcon = new ImageIcon("src/images/panda.jpg");
        JLabel l=new JLabel("test");

        l.setIcon(imagelcon);

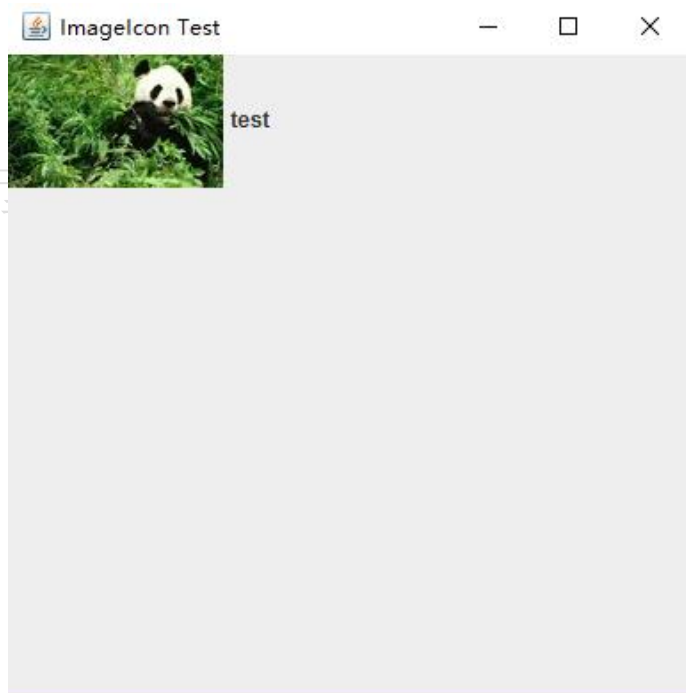
        frame.getContentPane().add(l,BorderLayout.NORTH);

        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setSize(400,400);
        frame.setVisible(true);}
}
```

提醒： 图片可放在当前目录的子目录 images 中，且要指明路径 “src/images/panda.jpg”
因为系统会从当前项目所在的路径进行寻址。

【实验结果与分析】

实验结果：



实验分析：

JLabel 对象可以显示文本、图像或同时显示二者。可以通过设置垂直和水平对齐方式，指定标签显示区中标签内容在何处对齐。默认情况下，标签在其显示区内垂直居中对齐。默认情况下，只显示文本的标签是开始边对齐；而只显示图像的标签则水平居中对齐。还可以指定文本相对于图像的位置。默认情况下，文本位于图像的结尾边上，文本和图像都垂直对齐。

构造方法介绍：

JLabel() 创建无图像并且其标题为空字符串的 JLabel。

JLabel(Icon image) 创建具有指定图像的 JLabel 实例。

JLabel(Icon image, int horizontalAlignment) 创建具有指定图像和水平对齐方式的 JLabel 实例。

`JLabel(String text)` 创建具有指定文本的 `JLabel` 实例。

`JLabel(String text, Icon icon, int horizontalAlignment)` 创建具有指定文本、图像和水平对齐方式的 `JLabel` 实例。

`JLabel(String text, int horizontalAlignment)` 创建具有指定文本和水平对齐方式的 `JLabel` 实例。

常用方法：

`getHorizontalAlignment()` 返回标签内容沿 X 轴的对齐方式。

`getHorizontalTextPosition()` 返回标签的文本相对其图像的水平位置。

`getIcon()` 返回该标签显示的图形图像（字形、图标）。 `getText()` 返回该标签所显示的文本字符串。

`setHorizontalAlignment(int alignment)` 设置标签内容沿 X 轴的对齐方式。

`setHorizontalTextPosition(int textPosition)` 设置标签的文本相对其图像的水平位置。

`setIcon(Icon icon)` 定义此组件将要显示的图标。

`setText(String text)` 定义此组件将要显示的单行文本。 `setUI(LabelUI ui)` 设置呈现此组件的 L&F 对象。

`setVerticalAlignment(int alignment)` 设置标签内容沿 Y 轴的对齐方式。

`setVerticalTextPosition(int textPosition)` 设置标签的文本相对其图像的垂直位置。

6、编译运行下面程序，理解掌握在 `JTabbedPane`

```
import javax.swing.Icon;
import javax.swing.ImageIcon;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JTabbedPane;

public class Demo {
    private static void createAndShowGUI() {
        // Create and set up the window.
        final JFrame frame = new JFrame("Split Pane Example");

        // Display the window.
        frame.setSize(500, 300);
        frame.setVisible(true);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        // set grid layout for the frame
        frame.getContentPane().setLayout(new GridLayout(1, 1));

        JTabbedPane tabbedPane = new JTabbedPane(JTabbedPane.TOP);

        // add tab with title
        tabbedPane.addTab("Tab 1", new JLabel("This is tab 1"));

        // add tab with title and icon
        Icon icon = new ImageIcon("icon.gif");
        tabbedPane.addTab("Tab 2", icon, new JLabel("This is tab 2"));

        // add tab with title, icon and tooltip
        String tooltip = "This is a tab";
        tabbedPane.addTab("Tab 3", icon, new JLabel("This is tab 3"), tooltip);

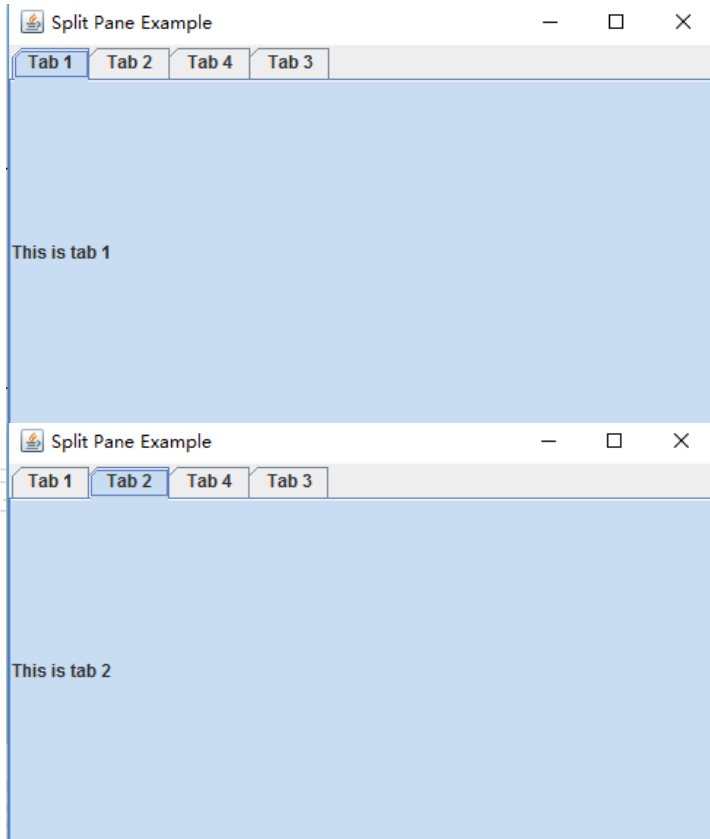
        // insert tab after second tab
        int index = 2;
        tabbedPane.insertTab("Tab 4", icon, new JLabel("This is tab 4"), tooltip, index);

        frame.getContentPane().add(tabbedPane);
    }
}
```

```
public static void main(String[] args) {  
    //Schedule a job for the event-dispatching thread:  
    //creating and showing this application's GUI.  
    javax.swing.SwingUtilities.invokeLater(new Runnable() {  
        public void run() {  
            createAndShowGUI();  
        }  
    });  
}
```

【实验结果与分析】

实验结果:



实验分析:

JTabbedPane 在 Swing 中实现选项窗格，它可以把多个组件放在多个选项卡中，从而使页面不致拥挤，其选项卡的形式也能为程序增色不少。选项窗格和分隔窗格类似，创建出来需要添加到设置好布局的面板中，通常我们可以设置布局为 1*1 的网格布局或是边界布局。选项窗格内部可以放置多个选项页，每个选项页都可以容纳一个 JPanel 作为子组件，我们只要设计好需要添加到选项页的面板即可。

本程序共创建了 4 个选项窗格，每个窗格里面都写有文本。

7、编译运行下面程序，理解掌握在 GridBagLayout 布局管理器

GridBagLayout 是最复杂的，功能也是最强大的，它所提供了很多的可配置选项，几乎可以完全地控制容器的布局方式

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

```
import javax.swing.JFrame;

public class Demo {
    final static boolean shouldFill = true;
    final static boolean shouldWeightX = true;
    final static boolean RIGHT_TO_LEFT = false;

    public static void addComponentsToPane(Container pane) {
        if (RIGHT_TO_LEFT) {
            pane.setComponentOrientation(ComponentOrientation.RIGHT_TO_LEFT);
        }

        JButton button;
        pane.setLayout(new GridBagLayout());
        GridBagConstraints c = new GridBagConstraints();
        if (shouldFill) {
            //natural height, maximum width
            c.fill = GridBagConstraints.HORIZONTAL;
        }

        button = new JButton("Button 1");
        if (shouldWeightX) {
            c.weightx = 0.5;
        }
        c.fill = GridBagConstraints.HORIZONTAL;
        c.gridx = 0;
        c.gridy = 0;
        pane.add(button, c);

        button = new JButton("Button 2");
        c.fill = GridBagConstraints.HORIZONTAL;
        c.weightx = 0.5;
        c.gridx = 1;
        c.gridy = 0;
        pane.add(button, c);

        button = new JButton("Button 3");
        c.fill = GridBagConstraints.HORIZONTAL;
        c.weightx = 0.5;
        c.gridx = 2;
        c.gridy = 0;
        pane.add(button, c);

        button = new JButton("Long-Named Button 4");
        c.fill = GridBagConstraints.HORIZONTAL;
        c.ipady = 40; //make this component tall
        c.weightx = 0.0;
        c.gridwidth = 3;
        c.gridx = 0;
        c.gridy = 1;
        pane.add(button, c);

        button = new JButton("5");
        c.fill = GridBagConstraints.HORIZONTAL;
        c.ipady = 0; //reset to default
        c.weighty = 1.0; //request any extra vertical space
        c.anchor = GridBagConstraints.PAGE_END; //bottom of space
        c.insets = new Insets(10,0,0,0); //top padding
        c.gridx = 1; //aligned with button 2
        c.gridwidth = 2; //2 columns wide
        c.gridy = 2; //third row
        pane.add(button, c);
    }

    /**
```

```
* Create the GUI and show it. For thread safety,
* this method should be invoked from the
* event-dispatching thread.
*/
private static void createAndShowGUI() {
    //Create and set up the window.
    JFrame frame = new JFrame("GridBagLayoutDemo");
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

    //Set up the content pane.
    addComponentsToPane(frame.getContentPane());

    //Display the window.
    frame.setSize(350,280);
    frame.pack();
    frame.setLocation(300,300);

    frame.setVisible(true);
}

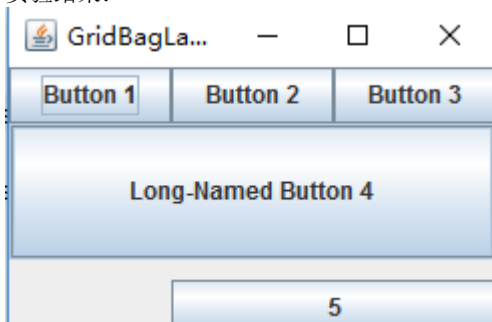
public static void main(String[] args) {
    //Schedule a job for the event-dispatching thread:
    //creating and showing this application's GUI.
    javax.swing.SwingUtilities.invokeLater(new Runnable() {
        public void run() {
            createAndShowGUI();
        }
    });
}
```

说明：更详细的用法参考：深入学习 GridBagLayout

http://www.360doc.com/content/15/0411/16/22423467_462424998.shtml

【实验结果与分析】

实验结果：



实验分析：

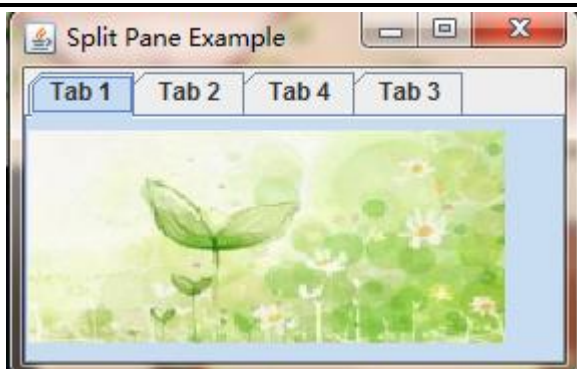
GridBagLayout 是没有任何约束或限制的网格布局(GridLayout),一个组件可以占据若干行和列，而且大小设定也是自由的。

一般通过以下步骤为界面进行 GridBagLayout 布局

- 1).设置主界面的布局管理器为 GridBagLayout(不用指定行和列)
- 2).为界面中的每一个组件(这里是 JPanel 对象)指定一个 GridBagConstraints 对象,通过设置该对象的属性值指出组件在管理器中的布局方案
- 3).通过下面的调用添加组件及其约束条件(GridBagConstraints 对象)
add(Component,constraints);

7、编写程序，设计一个简单的图片浏览器，使用 JTabbedPane，利用 JLabel 显示图像，单击不同的选项卡显示不同的图片。

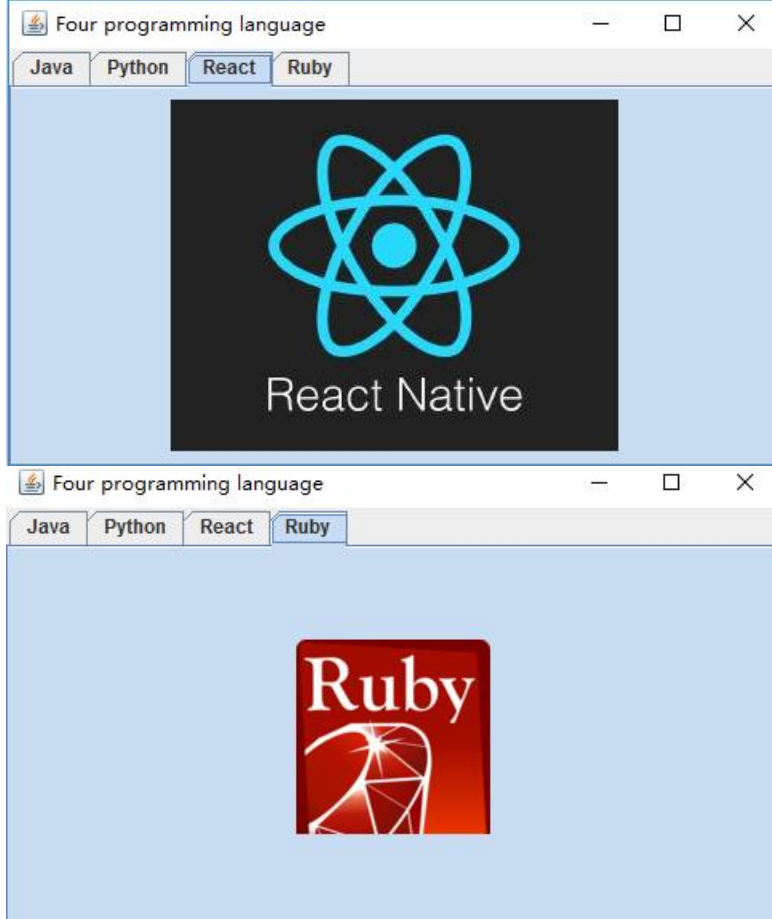
要求：编写程序，实现如下程序效果



【实验结果与分析】

A、程序运行结果





B、程序源代码

```
import java.awt.*;
import javax.swing.Icon;
import javax.swing.ImageIcon;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JTabbedPane;

public class DemoJT {

    private static void createAndShowGUI() {

        // Create and set up the window.
        final JFrame frame = new JFrame("Four programming language");

        // Display the window.
        frame.setSize(500, 300);
        frame.setVisible(true);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        // set grid layout for the frame
        frame.getContentPane().setLayout(new GridLayout(1, 1));
```

```
JTabbedPane tabbedPane = new JTabbedPane(JTabbedPane.TOP);

Icon image = new ImageIcon("image/java.gif");
tabbedPane.addTab("Java", new JLabel(image, JLabel.CENTER));

image = new ImageIcon("image/python.gif");
tabbedPane.addTab("Python", new JLabel(image, JLabel.CENTER));

image = new ImageIcon("image/react.gif");
tabbedPane.addTab("React", new JLabel(image, JLabel.CENTER));

image = new ImageIcon("image/ruby.gif");
tabbedPane.addTab("Ruby", new JLabel(image, JLabel.CENTER));

frame.getContentPane().add(tabbedPane);

}

public static void main(String[] args) {
    //Schedule a job for the event-dispatching thread:
    //creating and showing this application's GUI.
    javax.swing.SwingUtilities.invokeLater(new Runnable() {
        public void run() {
            createAndShowGUI();
        }
    });
}
```

6、设计应用程序，程序运行与操作结果如下图所示。



图 1 登录界面



图 2 如果不输入密码登录, 弹出相应对话框



图 3 如果用户名与密码不正确, 弹出相应对话框



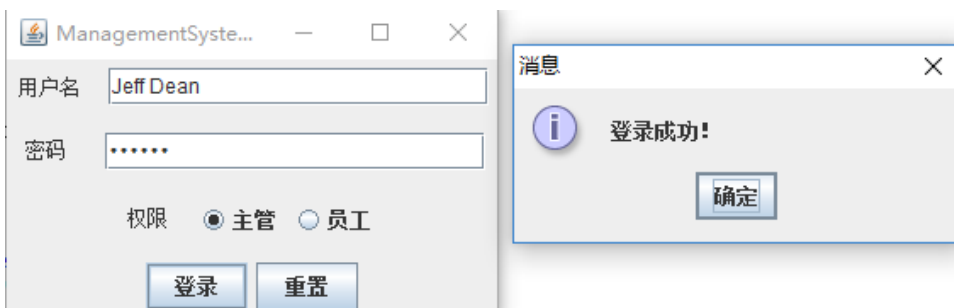
图 4 如果用户名与密码均正确, 弹出相应对话框

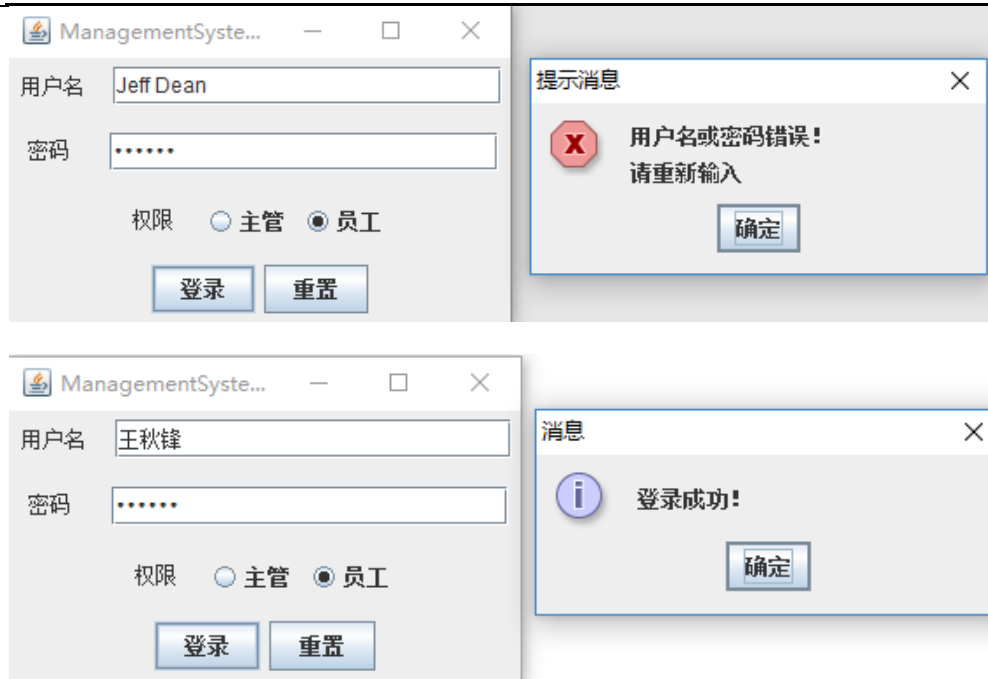
要求与提示:

1. 用户名与密码可以存储在数据库中;
2. 如果不使用数据库, 可将用户名与密码直接以字符串常量形式存储, 如果存储多个用户, 可以考虑使用字符串数组;

【实验结果与分析】

A、程序运行结果





B、程序源代码

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.JPanel;
import javax.swing.JFrame;
import javax.swing.JButton;
import javax.swing.JTextField;
import javax.swing.JOptionPane;
import javax.swing.ButtonGroup;
import javax.swing.JRadioButton;
import javax.swing.JPasswordField;

public class ManagementSystem implements ActionListener {
    public void actionPerformed(ActionEvent e) {
        String password = new String(JP.getPassword());
        if (e.getSource() == b1) {
            if (password.equals("")) {
                JOptionPane.showMessageDialog(fr, "请输入密码!", "提示消息",
                JOptionPane.WARNING_MESSAGE);
                return ;
            }
            if (jr1.isSelected()) {
                String[] director = {"Jeff Dean", "tensorflow", "caffe", "theano"};
                String[] director_password = {"888888", "888888", "654321",
                "654321"};
                boolean fg = false;
                for(int i=0; i<4; i++){
                    if ( JT.getText().equals(director[i]) &&
                    password.equals(director_password[i]) ) {
```

```
JOptionPane.showMessageDialog(fr, "登录成功!");
fg = true;
    }
}

if(!fg)JOptionPane.showMessageDialog(fr, "用户名或密码错误! \n请重新输入", "提示消息", JOptionPane.ERROR_MESSAGE);

}
else {
    String[] staff = {"Alice", "Bob", "王秋锋", "Cyberdebut"};
    String[] staff_password = {"123456", "1234546", "666666",
"666666"};
    boolean fg = false;
    for(int i=0; i<4; i++){
        if ( JT.getText().equals(staff[i]) &&
password.equals(staff_password[i]) ) {
            JOptionPane.showMessageDialog(fr, "登录成功!");
            fg = true;
        }
    }

    if(!fg)JOptionPane.showMessageDialog(fr, "用户名或密码错误! \n请重新输入", "提示消息", JOptionPane.ERROR_MESSAGE);
}
}
else {
    JT.setText("");
    JP.setText("");
}
}

JFrame fr = new JFrame("ManagementSystem display");
JTextField JT;
ButtonGroup bg;
JPasswordField JP;
JButton b1, b2;
JRadioButton jr1, jr2;
JPanel p1, p2, p3, p4;

ManagementSystem() {
    fr.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    fr.setLayout(new GridLayout(4, 1));

    p1 = new JPanel();
    p1.setLayout(new FlowLayout());
```

```
p1.add(new JLabel("用户名"));
JT = new JTextField(20);
p1.add(JT);

p2 = new JPanel(); p2.setLayout(new FlowLayout());
p2.add(new JLabel(" 密码  "));
JP = new JPasswordField(20);
p2.add(JP);

p3 = new JPanel(); p3.setLayout(new FlowLayout());
p3.add(new JLabel("权限"));
jr1 = new JRadioButton("主管", true);
jr2 = new JRadioButton("员工");
bg = new ButtonGroup();
bg.add(jr1); bg.add(jr2);
p3.add(jr1); p3.add(jr2);

p4 = new JPanel(); p4.setLayout(new FlowLayout());
b1 = new JButton("登录"); b2 = new JButton("重置");
b1.addActionListener(this);
b2.addActionListener(this);
p4.add(b1); p4.add(b2);

fr.add(p1); fr.add(p2);
fr.add(p3); fr.add(p4);
fr.pack();
fr.setVisible(true);
}
public static void main(String []args) {
    new ManagementSystem();
}
}
```