# 单元9 源代码

## 任务9.1 GUI基本概念和组件

1. 创建一个新窗口。

package chapter09;

import javax.swing.JFrame;

public class Example9\_1

{

public static void main(String[] args)

{

JFrame newFrame=new JFrame("第一个窗口");//创建窗口实例

newFrame.setSize(300,400);//设置窗口大小

newFrame.setLocation(200,300);//设置窗口显示左上角的位置

newFrame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);//设置窗口关闭操作

newFrame.setVisible(true);//设置显示属性

}

}

1. dimension对象实例对例9-1中的窗口设置大小。

import javax.swing.JFrame;

import java.awt.Dimension;

public class Example9\_2

{

public static void main(String[] args)

{

JFrame newFrame=new JFrame("第二个窗口");//创建窗口实例

Dimension d=new Dimension();

d.setSize(500,700);

newFrame.setSize(d);//设置窗口大小

newFrame.setLocation(200,300);//设置窗口显示左上角的位置

newFrame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);//设置窗口关闭操作

newFrame.setVisible(true);//设置显示属性

}

}

1. 在例9-2基础上增加一个中间容器JPanel，并且设置其背景为黄色，注意下面黑体代码。

import java.awt.Color;

import javax.swing.JFrame;

import java.awt.Dimension;

import javax.swing.JPanel;

public class Example9\_3

{

public static void main(String[] args)

{

JFrame newFrame=new JFrame("第三个窗口");//创建窗口实例

Dimension d=new Dimension();

d.setSize(500,700);

newFrame.setSize(d);//设置窗口大小

newFrame.setLocation(200,300);//设置窗口显示左上角的位置

newFrame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);//设置窗口关闭操作

newFrame.setVisible(true);//设置显示属性

JPanel p=new JPanel();//创建JPanel对象

p.setBackground(Color.yellow);//设置Jpanel背景颜色

newFrame.add(p);

}

}

1. 在窗口中添加一个中间容器JPanel，在JPanel中添加一个标签对象。

import java.awt.Color;

import javax.swing.JPanel;

import javax.swing.JFrame;

import javax.swing.JLabel;

import java.awt.Font;

public class Example9\_4

{

public static void main(String[] args)

{ //1.创建顶级容器对象

JFrame newWin=new JFrame("窗口测试标签");

newWin.setSize(400,500);

newWin.setLocation(0, 0);//设置窗口左上角的坐标

newWin.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

newWin.setVisible(true);//显示窗口

//2.创建JPanel中间容器

JPanel panelObj=new JPanel();

panelObj.setBackground(Color.gray);

//3.创建标签对象

JLabel labelObj=new JLabel("我是一个标签！！");

labelObj.setFont(new Font("隶书",Font.BOLD,40));//设置字体

labelObj.setForeground(Color.BLUE);//设置字体颜色

//4.根据容器包含与被包含的顺序，将标签对象添加到中间容器中，将中间容器添加到顶级容器中

panelObj.add(labelObj);

newWin.add(panelObj);

}

}

【例9-5】在JPanel中再添加一个JTextField对象

import java.awt.Color;

import javax.swing.JPanel;

import javax.swing.JFrame;

import javax.swing.JLabel;

import javax.swing.JTextField;

import java.awt.Font;

public class Example9\_5

{

public static void main(String[] args)

{ //1.创建顶级容器对象

JFrame newWin=new JFrame("窗口测试标签");

newWin.setSize(400,500);

newWin.setLocation(0, 0);//设置窗口左上角的坐标

newWin.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

//2.创建JPanel中间容器

JPanel panelObj=new JPanel();

panelObj.setBackground(Color.gray);

//3.创建标签对象

JLabel labelObj=new JLabel("请输入姓名：");

labelObj.setFont(new Font("隶书",Font.BOLD,20));//设置字体

labelObj.setForeground(Color.BLUE);//设置字体颜色

//4.创建JTextField对象

JTextField textFiled=new JTextField(20);

textFiled.setHorizontalAlignment(JTextField.RIGHT);// 设置字体的文本格式

//5.根据容器包含与被包含的顺序，将标签对象添加到中间容器中，将中间容器添加到顶级容器中

panelObj.add(labelObj);

panelObj.add(textFiled);

newWin.add(panelObj);

newWin. setVisible(true);//显示窗口

}

}

1. 在例9-5基础上增加一个JButton组件。

import java.awt.Color;

import javax.swing.JPanel;

import javax.swing.JFrame;

import javax.swing.JLabel;

import javax.swing.JTextField;

import javax.swing.JButton;

import java.awt.Font;

public class Example9\_6

{

public static void main(String[] args)

{

//1.创建顶级容器对象

JFrame newWin=new JFrame("窗口测试标签");

newWin.setSize(400,500);

newWin.setLocation(0, 0);//设置窗口左上角的坐标

newWin.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

//2.创建JPanel中间容器

JPanel panelObj=new JPanel();

panelObj.setBackground(Color.gray);

//3.创建标签对象

JLabel labelObj=new JLabel("请输入姓名：");

labelObj.setFont(new Font("隶书",Font.BOLD,20));//设置字体

labelObj.setForeground(Color.BLUE);//设置字体颜色

//4.创建JTextField对象

JTextField textFiled=new JTextField(20);

textFiled.setHorizontalAlignment(JTextField.RIGHT);// 设置字体的文本格式

//5.创建JButton对象

JButton button=new JButton("提交");

//6.根据容器包含与被包含的顺序，将标签对象添加到中间容器中，将中间容器添加到顶级容器中

panelObj.add(labelObj);

panelObj.add(textFiled);

panelObj.add(button);

newWin.add(panelObj);

newWin.setVisible(true);//显示窗口

}

}

1. 创建一个使用JCheckBox。

import java.awt.Color;

import javax.swing.JPanel;

import javax.swing.JFrame;

import javax.swing.JLabel;

import javax.swing.JTextField;

import javax.swing.JButton;

import javax.swing.JCheckBox;

import java.awt.Font;

public class Example9\_7

{

public static void main(String[] args)

{

//1.创建顶级容器对象

JFrame newWin=new JFrame("窗口测试标签");

newWin.setSize(400,500);

newWin.setLocation(0, 0);//设置窗口左上角的坐标

newWin.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

//2.创建JPanel中间容器

JPanel panelObj=new JPanel();

panelObj.setBackground(Color.gray);

//3.创建标签对象

JLabel labelObj=new JLabel("爱好：");

labelObj.setFont(new Font("隶书",Font.BOLD,20));//设置字体

labelObj.setForeground(Color.BLUE);//设置字体颜色

//4.创建CheckBox对象

JCheckBox cb1=new JCheckBox("足球",true) ;

JCheckBox cb2=new JCheckBox("篮球") ;

JCheckBox cb3=new JCheckBox("田径") ;

//5.根据容器包含与被包含的顺序，将标签对象添加到中间容器中，将中间容器添加到顶级容器中

panelObj.add(labelObj);

panelObj.add(cb1);

panelObj.add(cb2);

panelObj.add(cb3);

newWin.add(panelObj);

newWin.setVisible(true);//显示窗口

}

}

1. 在例9-7基础上增加一个选择性别的单选操作功能。

import java.awt.Color;

import javax.swing.JPanel;

import javax.swing.JFrame;

import javax.swing.JLabel;

import javax.swing.JTextField;

import javax.swing.JButton;

import javax.swing.JCheckBox;

import javax.swing.JRadioButton;

import javax.swing.ButtonGroup;

import java.awt.Font;

public class Example9\_8

{

public static void main(String[] args)

{

//1.创建顶级容器对象

JFrame newWin=new JFrame("窗口测试标签");

newWin.setSize(400,500);

newWin.setLocation(0, 0);//设置窗口左上角的坐标

newWin.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

//2.创建JPanel中间容器

JPanel panelObj=new JPanel();

panelObj.setBackground(Color.gray);

//3.创建标签对象

JLabel labelObj=new JLabel("爱好：");

labelObj.setFont(new Font("隶书",Font.BOLD,20));//设置字体

labelObj.setForeground(Color.BLUE);//设置字体颜色

//4.创建CheckBox对象

JCheckBox cb1=new JCheckBox("足球") ;

JCheckBox cb2=new JCheckBox("篮球") ;

JCheckBox cb3=new JCheckBox("田径") ;

//5.添加一个单选按钮对象

JLabel labelObj2=new JLabel("性别：");//创建提示标签

labelObj2.setFont(new Font("隶书",Font.BOLD,20));

JRadioButton jrb1=new JRadioButton("男",true);//创建两个单选按钮

JRadioButton jrb2=new JRadioButton("女");

ButtonGroup sexSelect=new ButtonGroup();//创建按钮组对象且将单选按钮添加到按钮组

sexSelect.add(jrb1);

sexSelect.add(jrb2);

//6.根据容器包含与被包含的顺序，将标签对象添加到中间容器中，将中间容器添加到顶级容器中

panelObj.add(labelObj);

panelObj.add(cb1);

panelObj.add(cb2);

panelObj.add(cb3);

panelObj.add(labelObj2);

panelObj.add(jrb1);

panelObj.add(jrb2);

newWin.add(panelObj);

newWin.setVisible(true);//显示窗口

}

}

1. 创建一个列表。

import javax.swing.JPanel;

import javax.swing.JFrame;

import javax.swing.JLabel;

import javax.swing.JList;

public class Example9\_9

{

public static void main(String args[])

{ //创建窗口框架并设置窗口的大小与显示属性

JFrame f = new JFrame("测试列表");

f.setSize(300,500);

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

//创建面板对象，并将面板加入到窗口框架中

JPanel p=new JPanel();

f.add(p);

//创建一个标签提示与一个列表对象

Object[] dataList={"博士","硕士","大学","大专","中专","高中","初中","小学"};

JLabel l=new JLabel("请选择");

JList list=new JList(dataList);

//将上面创建的二个对象按显示的先后次序加入到面板对象中

p.add(l);

p.add(list);

f.setVisible(true);

}

}

1. JToolBar组件应用案例。

import javax.swing.\*;

//工具栏的使用案例

public class Example9\_10 extends javax.swing.JFrame {

private JToolBar myJToolBar;

private JButton jB\_file;

private JButton jB\_edit;

private JButton jB\_tools;

private JButton jB\_help;

public static void main(String[] args) {

SwingUtilities.invokeLater(new Runnable() {

public void run() {

Example9\_10 inst = new Example9\_10();

inst.setLocationRelativeTo(null);

inst.setVisible(true);

}

});//黑体部分是定义了一个线程匿名实例对象

}

public Example9\_10()

{

super();

initGUI();

}

private void initGUI() {

try {

setDefaultCloseOperation(WindowConstants.DISPOSE\_ON\_CLOSE);

getContentPane().setLayout(null);

getContentPane().setBackground(new java.awt.Color(255, 128, 255));

{

myJToolBar = new JToolBar();

getContentPane().add(myJToolBar);

myJToolBar.setBounds(29, 12, 320, 38);

myJToolBar.setBackground(new java.awt.Color(255, 255, 255));

{

jB\_file = new JButton();//创建新的按钮实例对象

myJToolBar.add(jB\_file);//将按钮实例对象添加到JToolBar对象中

jB\_file.setText("文件操作");//设置按钮对象显示的文字信息

jB\_file.setPreferredSize(new java.awt.Dimension(80, 34)); //设置按钮对象外观尺寸

jB\_file.setFont(new java.awt.Font("楷体", 0, 14));//设置按钮显示字符

jB\_file.setToolTipText("这是文件操作按钮");//设置提示信息，即鼠标移动到当前按钮时，显示"这是文件操作按钮"这行信息

jB\_file.setBackground(new java.awt.Color(255, 255, 128));//设置按钮背景颜色

}

{

jB\_edit = new JButton();

myJToolBar.add(jB\_edit);

jB\_edit.setText("编辑操作");

//jB\_edit.setToolTipText("点此每次新增一个结点");

jB\_edit.setFont(new java.awt.Font("楷体", 0, 14));

jB\_edit.setBackground(new java.awt.Color(255, 255, 128));

jB\_edit.setToolTipText("这是编程操作按钮");

jB\_edit.setPreferredSize(new java.awt.Dimension(78, 34));

}

{

jB\_tools = new JButton();

myJToolBar.add(jB\_tools);

jB\_tools.setText("工具按钮");

jB\_tools.setFont(new java.awt.Font("楷体", 0, 14));

jB\_tools.setBackground(new java.awt.Color(255, 255, 128));

jB\_tools.setToolTipText("这是工具按钮");

jB\_tools.setPreferredSize(new java.awt.Dimension(94, 34));

}

{

jB\_help = new JButton();

myJToolBar.add(jB\_help);

jB\_help.setText("帮助操作");

jB\_help.setFont(new java.awt.Font("楷体", 0, 14));

jB\_help.setBackground(new java.awt.Color(255, 255, 128));

jB\_help.setToolTipText("这是帮助操作按钮");

jB\_help.setPreferredSize(new java.awt.Dimension(95, 34));

}

}

pack();

setSize(400, 300);

}

catch (Exception e) {

e.printStackTrace();

}

}

}

1. 弹出一个针对用户主目录的文件选择器，只显示.jpg和.gif图像。

import java.io.\*;

import java.awt.\*;

import java.awt.event.\*;

import javax.swing.\*;

import javax.swing.SwingUtilities;

import javax.swing.filechooser.\*;

public class Example9\_11 extends JPanel implements ActionListener {

static private final String newline = "\n";

JButton openButton, saveButton;

JTextArea log;

JFileChooser fc;

public Example9\_11() {

super(new BorderLayout());

log = new JTextArea(5,20);

log.setMargin(new Insets(5,5,5,5));

log.setEditable(false);

JScrollPane logScrollPane = new JScrollPane(log);

//创建文件过滤器实例对象

fc = new JFileChooser();

//创建一个打开文件的按钮

openButton = new JButton("Open a File...");

openButton.addActionListener(this);

//创建一个保存文件的按钮

saveButton = new JButton("Save a File...");

saveButton.addActionListener(this);

//创建中间容器JPanel，并将两个按钮添加到JPanel中

JPanel buttonPanel = new JPanel(); //use FlowLayout

buttonPanel.add(openButton);

buttonPanel.add(saveButton);

//将JPanel添加到窗口容器中

add(buttonPanel, BorderLayout.PAGE\_START);

add(logScrollPane, BorderLayout.CENTER);

}

//声明事件监听处理代码

public void actionPerformed(ActionEvent e) {

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

int returnVal = fc.showOpenDialog(Example9\_11.this);

if (returnVal == JFileChooser.APPROVE\_OPTION) {

File file = fc.getSelectedFile();

log.append("Opening: " + file.getName() + "." + newline);

}

else {

log.append("Open command cancelled by user." + newline);

}

log.setCaretPosition(log.getDocument().getLength());

}

else if (e.getSource() == saveButton) {

int returnVal = fc.showSaveDialog(Example9\_11.this);

if (returnVal == JFileChooser.APPROVE\_OPTION) {

File file = fc.getSelectedFile();

log.append("Saving: " + file.getName() + "." + newline);

}

else {

log.append("Save command cancelled by user." + newline);

}

log.setCaretPosition(log.getDocument().getLength());

}

}

/\*\* 如果图片文件不存在或无效，进行必要的信息提示\*/

protected static ImageIcon createImageIcon(String path) {

java.net.URL imgURL = Example9\_11.class.getResource(path);

if (imgURL != null) {

return new ImageIcon(imgURL);

} else {

System.err.println("Couldn't find file: " + path);

return null;

}

}

/\*\*

\* 创建图形界面

\*/

private static void createAndShowGUI() {

//创建窗口

JFrame frame = new JFrame("FileChooserDemo");

frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

//将Example9\_11中的panel容器添加到窗口中

frame.add(new Example9\_11());

//显示窗口

frame.pack();

frame.setVisible(true);

}

public static void main(String[] args) {

SwingUtilities.invokeLater(new Runnable() {

public void run() {

UIManager.put("swing.boldMetal", Boolean.FALSE);

createAndShowGUI();

}

});//黑体部分是线程的匿名实例对象

}

}

**任务实施**

import java.awt.Color;

import javax.swing.\*;

import java.awt.Font;

public class Task901 {

public static void main(String[] args) {

//创建顶级容器----窗口，设置窗口的显示大小、默认关闭操作、显示属性

JFrame myWin=new JFrame("注册窗口");

myWin.setSize(900,400);

myWin.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);//设置窗口默认关闭操作

myWin.setVisible(true);

//创建三个中级容器，将这两个中级容器添加到顶级容器中

JPanel desk=new JPanel();

JPanel header=new JPanel();

JPanel body=new JPanel();

header.setBackground(Color.yellow);

body.setBackground(Color.PINK);

myWin.add(desk);

desk.add(header);

desk.add(body);

//向Header容器添加标签实例

JLabel jlHeader=new JLabel("注册",JLabel.CENTER);

Font font=new Font("楷体",Font.BOLD+Font.PLAIN,35);//设置字体格式对象

jlHeader.setFont(font);

header.add(jlHeader);

//向body容器中添加相应组件

body.add(new JLabel("姓名："));

body.add(new JTextField(20));

body.add(new JLabel("性别："));

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

JRadioButton sexWoman=new JRadioButton("女");

ButtonGroup sexButtonGroup=new ButtonGroup();

sexButtonGroup.add(sexMan);

sexButtonGroup.add(sexWoman);

body.add(sexMan);

body.add(sexWoman);

body.add(new JLabel("密码："));

body.add(new JPasswordField(20));

body.add(new JButton("提交"));

body.add(new JButton("取消"));

}

}

任务演练

package project901;

import javax.swing.\*;

import java.awt.\*;

public class Project901 {

public static void main(String[] args) {

//创建顶级容器----窗口，设置窗体的显示大小、默认关闭操作、显示属性

JFrame myWin=new JFrame("计算器窗口");

myWin.setSize(900,400);

myWin.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);//设置窗体默认关闭操作

myWin.setVisible(true);

//创建三个中级容器，将这两个中级容器添加到顶级容器中

JPanel deskPanel=new JPanel();

JPanel headPanel=new JPanel();

JPanel bodyPanel=new JPanel();

headPanel.setBackground(Color.yellow);

bodyPanel.setBackground(Color.PINK);

myWin.add(deskPanel);

deskPanel.add(headPanel);

deskPanel.add(bodyPanel);

//向Header容器添加计算显示的输入文本框

JTextField txtCalculate=new JTextField(50);

headPanel.add(txtCalculate);

//向body容器中添加相应组件

String txtSymbol="0123456789+-\*/" ;

for(int index=0;index<14;index++)

{

bodyPanel.add(new JButton(String.valueOf(txtSymbol.charAt(index))));

}

}

}

## 任务9.2 布局管理器

1. 用FlowLayout作布局管理器。

import java.awt.\*;

import javax.swing.\*;

import java.awt.event.\*;

class Example9\_12

{

public static void main(String args[])

{ JFrame window=new JFrame("我的窗口");

window.setSize(200,200);

window.setVisible(true);

JPanel desk=new JPanel();

window.add(desk);

//创建一些组件

JButton b1=new JButton("打开");

JButton b2=new JButton("关闭");

JButton b3=new JButton("退出");

JButton b4=new JButton("修改");

//创建流式管理器

FlowLayout mylayout=new FlowLayout(FlowLayout.LEFT,10,20);

//设置面板的布局管理

desk.setLayout(mylayout);

//向面板加入组件

desk.add(b1);

desk.add(b2);

desk.add(b3);

desk.add(b4);

//为窗口设置监听器

window.addWindowListener(new WindowAdapter()

{ public void windowClosing(WindowEvent e)

{ System.exit(1); }

});

}

}

1. 用BorderLayout作布局管理器

import java.awt.\*;

import javax.swing.\*;

import java.awt.event.\*;

public class Example9\_13

{

public static void main(String args[])

{ JFrame window=new JFrame("我的窗口");

window.setSize(200,200);

window.setVisible(true);

Panel desk=new Panel();

window.add(desk);

//创建一些组件

JButton b1=new JButton("东");

JButton b2=new JButton("南");

JButton b3=new JButton("西");

JButton b4=new JButton("北");

JButton b5=new JButton("中");

BorderLayout mylayout=new BorderLayout(10,10); //创建边界管理器

desk.setLayout(mylayout); //设置面板的布局管理

//向面板加入组件

desk.add("East",b1);

desk.add("South",b2);

desk.add("West",b3);

desk.add("North",b4);

desk.add("Center",b5);

window.addWindowListener(new WindowAdapter()//为窗口设置监听器

{ public void windowClosing(WindowEvent e)

{ System.exit(1); }

});

}

}

1. 网格布袋布局管理器应用简例.

import java.awt.\*;

import javax.swing.\*;

import java.awt.event.\*;

class Example9\_14

{ //定义对象变量

JFrame window;

JPanel panelobject;

GridBagLayout layoutObject;

GridBagConstraints con;

JButton b1;

JButton b2;

JButton b3;

JTextField tf;

JTextArea ta;

JLabel l;

public Example9\_14 ()

{ //创建窗口对象实例并设置大小、可见性与监听

window=new JFrame("测试网格布袋布局");

window.setSize(300,300);

window.setVisible(true);

window.addWindowListener(new WindowAdapter()

{ public void windowClosing(WindowEvent e)

{ System.exit(0);}

});

//创建其它对象

panelobject=new JPanel();

b1=new JButton("按钮1");

b2=new JButton("按钮2");

b3=new JButton("按钮3");

tf=new JTextField("这是一个测试布袋网格布局");

ta=new JTextArea("这是一个测试布袋网格布局",10,5);

l=new JLabel("我是标签");

//为突出显示，设置标签的前后背景色

l.setBackground(Color.BLUE);

l.setForeground(Color.white);

//创建布局

layoutObject=new GridBagLayout();

//创建GirdBagConstraints对象

con=new GridBagConstraints();

//将面板加入到窗口中，并设置面板的布局

window.add(panelobject);

panelobject.setLayout(layoutObject);

//定义属性并将组件加入到相应的空间

//加入第一个组件

con.gridx=0;

con.gridy=0;

con.anchor=GridBagConstraints.EAST;

con.fill=GridBagConstraints.BOTH;

con.gridwidth=20;

con.gridheight=1;//定义属性值

layoutObject.setConstraints(b1,con);//设置布局属性

panelobject.add(b1);//将组件加入到容器

//后面均按上面的模块进行设置、添加操作

con.gridx=0;

con.gridy=1;

con.anchor=GridBagConstraints.SOUTHEAST;

con.fill=GridBagConstraints.BOTH;

con.gridwidth=10;

con.gridheight=1;

layoutObject.setConstraints(b2,con);

panelobject.add(b2);

con.gridx=10;

con.gridy=1;

con.anchor=GridBagConstraints.SOUTH;

con.fill=GridBagConstraints.HORIZONTAL;

con.gridwidth=10;

con.gridheight=1;

layoutObject.setConstraints(b3,con);

panelobject.add(b3);

con.gridx=0;

con.gridy=2;

con.anchor=GridBagConstraints.NORTHEAST;

con.fill=GridBagConstraints.BOTH;

con.gridwidth=15;

con.gridheight=10;

layoutObject.setConstraints(ta,con);

panelobject.add(ta);

con.gridx=15;

con.gridy=2;

con.anchor=GridBagConstraints.NORTHEAST;

con.fill=GridBagConstraints.BOTH;

con.gridwidth=5;

con.gridheight=5;

layoutObject.setConstraints(l,con);

panelobject.add(l);

con.gridx=15;

con.gridy=7;

con.anchor=GridBagConstraints.NORTHEAST;

con.fill=GridBagConstraints.BOTH;

con.gridwidth=5;

con.gridheight=5;

layoutObject.setConstraints(tf,con);

panelobject.add(tf);

}

public static void main(String args[])

{ new Gridbaglayouttest(); } //创建一个类的匿名对象

}

**任务实施**

import java.awt.Color;

import javax.swing.\*;

import java.awt.\*;

public class Task902 {

public static void main(String[] args) {

//创建顶级容器----窗口，设置窗口的显示大小、默认关闭操作、显示属性

JFrame myWin=new JFrame("注册窗口");

myWin.setSize(300,500);

myWin.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);//设置窗口默认关闭操作

myWin.setVisible(true);

//创建三个中级容器，将这两个中级容器添加到顶级容器中

JPanel desk=new JPanel();

JPanel header=new JPanel();

JPanel body=new JPanel();

header.setBackground(Color.yellow);

myWin.add(desk);

//设置desk边界布局管理器

desk.setLayout(new BorderLayout());

desk.add(header,BorderLayout.NORTH);

desk.add(body,BorderLayout.CENTER);

//向Header容器添加标签实例

JLabel jlHeader=new JLabel("注册",JLabel.CENTER);

Font font=new Font("楷体",Font.BOLD+Font.PLAIN,35);//设置字体格式对象

jlHeader.setFont(font);

header.add(jlHeader);

//设置Body布局为网格布局管理

body.setLayout(new GridLayout(4,2));

//向body容器中添加相应组件

body.add(new JLabel("姓名："));

body.add(new JTextField(20));

body.add(new JLabel("性别："));

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

JRadioButton sexWoman=new JRadioButton("女");

ButtonGroup sexButtonGroup=new ButtonGroup();

sexButtonGroup.add(sexMan);

sexButtonGroup.add(sexWoman);

//创建一个保存性别单选按钮的容器

JPanel sexPanel=new JPanel();

sexPanel.add(sexMan);

sexPanel.add(sexWoman);

body.add(sexPanel);

body.add(new JLabel("密码："));

body.add(new JPasswordField(20));

body.add(new JButton("提交"));

body.add(new JButton("取消"));

}

}

任务演练

package project902;

import javax.swing.\*;

import java.awt.\*;

public class Project902 {

public static void main(String[] args) {

//创建顶级容器----窗口，设置窗体的显示大小、默认关闭操作、显示属性

JFrame myWin=new JFrame("计算器窗口");

myWin.setSize(400,500);

myWin.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);//设置窗体默认关闭操作

myWin.setVisible(true);

Font font=new Font("楷体",Font.BOLD,20);

//创建三个中级容器，将这两个中级容器添加到顶级容器中

JPanel deskPanel=new JPanel();

JPanel headPanel=new JPanel();

JPanel bodyPanel=new JPanel();

headPanel.setBackground(Color.yellow);

bodyPanel.setBackground(Color.PINK);

myWin.add(deskPanel);

//设置deskPanel布局管理器为边界布局管理器

deskPanel.setLayout(new BorderLayout());

deskPanel.add(headPanel,BorderLayout.NORTH);

deskPanel.add(bodyPanel,BorderLayout.CENTER);

//向Header容器添加计算显示的输入文本框

JTextField txtCalculate=new JTextField(30);

txtCalculate.setFont(font);

headPanel.add(txtCalculate);

//设置bodyPanel布局管理器为网格布局管理器

bodyPanel.setLayout(new GridLayout(4,4));

//向body容器中添加相应组件

String txtSymbol="0123456789.+-\*/=" ;

for(int index=0;index<16;index++)

{

JButton b1=new JButton(String.valueOf(txtSymbol.charAt(index)));

b1.setFont(font);

bodyPanel.add(b1);

}

}

}

## 任务9.3 事件处理机制

1. 为窗口添加windowListener。

import java.awt.\*;//包含AWT组件类

import javax.swing.\*;

import java.awt.event.\*;//包含事件监听器类

//定义继承WindowListener的类TestEvent

class Example9\_15 implements WindowListener

{

JFrame myFrame;//定义窗口框架

public Example9\_15 ()//构造方法方法,设置窗口的大小与可见性

{

myFrame=new JFrame("测试窗口监听器");

myFrame.setSize(200,200);

myFrame.setVisible(true);

//为窗口注册WindowListener监听器

myFrame.addWindowListener(this);

}

public static void main(String args[])

{

Example9\_15 myWindow=new Example9\_15 ();}//创建窗口

//实现WindowListener接口中的方法

public void windowActivated(WindowEvent e) { }

public void windowClosed(WindowEvent e) { }

public void windowClosing(WindowEvent e)

{ System.exit(1); }//为相应方法编写功能代码

public void windowDeactivated(WindowEvent e) { }

public void windowDeiconified(WindowEvent e) { }

public void windowIconified(WindowEvent e) { }

public void windowOpened(WindowEvent e) { }

}

1. 用内部类实现事件处理。在上例的基础上，定义一个按钮监听内部类，实现对按钮动作的监听：即当单击相应按钮时，在标签中显示单击的按钮名称信息。

import java.awt.\*;//包含AWT组件类

import javax.swing.\*;

import java.awt.event.\*;//包含事件监听器类

//定义继承WindowListener的类TestEvent

class Example9\_16 implements WindowListener

{ JFrame myFrame;//定义窗口框架

JButton b1;//定义三个按钮

JButton b2;

JButton b3;

JPanel p;//定义面板

JLabel L;//定义标签

public Example9\_16 ()

{ //创建窗口、并设置窗口大小、可见性

myFrame=new JFrame("测试窗口监听器");

myFrame.setSize(200,200);

myFrame.setVisible(true);

p=new JPanel();//创建面板对象

//创建三个按钮对象

b1=new JButton("按钮1");

b2=new JButton("按钮2");

b3=new JButton("按钮3");

L=new JLabel("未按下任何按钮"); //创建标签对象

myFrame.add(p); //将面板加入到窗口中

//将其它组件加入到面板中

p.add(b1);

p.add(b2);

p.add(b3);

p.add(L);

myFrame.addWindowListener(this); //为窗口注册WindowListener监听器

//为三个按钮注册监听器

b1.addActionListener(new buttonListener());

b2.addActionListener(new buttonListener());

b3.addActionListener(new buttonListener());

}

public static void main(String args[])//创建窗口

{ Example9\_16 myWindow=new Example9\_16 (); }

//实现WindowListener接口中的方法

public void windowActivated(WindowEvent e){}

public void windowClosed(WindowEvent e) {}

public void windowClosing(WindowEvent e)

{ System.exit(1); }

public void windowDeactivated(WindowEvent e) {}

public void windowDeiconified(WindowEvent e) {}

public void windowIconified(WindowEvent e) {}

public void windowOpened(WindowEvent e) {}

//定义一个buttonListener内部类，继承ActionListener接口

class buttonListener implements ActionListener

{

public void actionPerformed(ActionEvent e)

{ //获取发生事件的事件源

Object source=e.getSource();

if(source==b1)

L.setText("按下的是按钮1");

else if(source==b2)

L.setText("按下的是按钮2");

else if(source==b3)

L.setText("按下的是按钮3");

}

}

}

1. 用适配器方式实现例9-15。

import java.awt.\*;//包含AWT组件类

import javax.swing.\*;

import java.awt.event.\*;//包含事件监听器类

//定义类Example9\_17继承窗口的适配器

class Example9\_17 extends WindowAdapter

{ JFrame myFrame;//定义窗口框架

public Example9\_17 ()//构造方法方法,设置窗口的大小与可见性

{ myFrame=new JFrame("测试窗口监听器");

myFrame.setSize(200,200);

myFrame.setVisible(true);

//为窗口注册WindowListener监听器

myFrame.addWindowListener(this);

}

//此处只重写了方法windowClosing

public void windowClosing(WindowEvent e)

{ System.exit(1); }

public static void main(String args[])//创建窗口

{ Example9\_17 myWindow=new Example9\_17 (); }

}

1. 使用匿名类对例9-17窗口监听代码改写。

import java.awt.\*;//包含AWT组件类

import javax.swing.\*;

import java.awt.event.\*;//包含事件监听器类

class Example9\_18

{ JFrame myFrame;//定义窗口框架

public Example9\_18 ()//构造方法方法,设置窗口的大小与可见性

{ myFrame=new JFrame("测试窗口监听器");

myFrame.setSize(200,200);

myFrame.setVisible(true);

//为窗口注册WindowListener监听器

myFrame.addWindowListener(

new WindowAdapter()

{ public void windowClosing(WindowEvent e)

{ System.exit(1); }

} );

}

public static void main(String args[])//创建窗口

{ Example9\_18 myWindow=new Example9\_18 (); }

}

**任务实施**

import java.awt.Color;

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.\*;

public class Task903 {

//将主方法中的组件声明改成类的成员变量

JFrame myWin;

JPanel desk,header,body,footer;

JTextField txtName;

JLabel showInfo;

JPasswordField passwrd;

JButton btnCancel,btnConfirm;

JRadioButton sexMan,sexWoman;

Task903()//定义构造方法器

{ //创建顶级容器----窗口，设置窗口的显示大小、默认关闭操作、显示属性

myWin=new JFrame("注册窗口");

myWin.setSize(300,500);

myWin.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);//设置窗口默认关闭操作

myWin.setVisible(true);

//创建三个中级容器，将这两个中级容器添加到顶级容器中

desk=new JPanel();

header=new JPanel();

body=new JPanel();

footer=new JPanel();

header.setBackground(Color.yellow);

myWin.add(desk);

//设置desk边界布局管理器

desk.setLayout(new BorderLayout());

desk.add(header,BorderLayout.NORTH);

desk.add(body,BorderLayout.CENTER);

desk.add(footer,BorderLayout.SOUTH);

//向Header容器添加标签实例

JLabel jlHeader=new JLabel("注册",JLabel.CENTER);

Font font=new Font("楷体",Font.BOLD+Font.PLAIN,35);//设置字体格式对象

jlHeader.setFont(font);

header.add(jlHeader);

//设置Body布局为网格布局管理

body.setLayout(new GridLayout(4,2));

//向body容器中添加相应组件

body.add(new JLabel("姓名："));

txtName=new JTextField(20);

body.add(txtName);

body.add(new JLabel("性别："));

sexMan=new JRadioButton("男",true);

sexWoman=new JRadioButton("女");

ButtonGroup sexButtonGroup=new ButtonGroup();

sexButtonGroup.add(sexMan);

sexButtonGroup.add(sexWoman);

//创建一个保存性别单选按钮的容器

JPanel sexPanel=new JPanel();

sexPanel.add(sexMan);

sexPanel.add(sexWoman);

body.add(sexPanel);

body.add(new JLabel("密码："));

passwrd=new JPasswordField(20);

body.add(passwrd);

btnCancel=new JButton("取消");

btnConfirm=new JButton("提交");

body.add(btnConfirm);

body.add(btnCancel);

//对两个按钮注册事件监听

btnCancel.addActionListener(new MyBtnEvent());

btnConfirm.addActionListener(new MyBtnEvent());

//将显示标签添加到footer里

showInfo=new JLabel();

showInfo.setFont(font);

footer.add(showInfo);

}

public static void main(String[] args) {

new Task903();

}

//声明行为事件监听内部类

class MyBtnEvent implements ActionListener

{

public void actionPerformed(ActionEvent e)//重写继承的方法

{

Object source=e.getSource();//获取事件源对象

if(source==btnConfirm)//如果事件源是确定按钮

{

String info="姓名："+txtName.getText()+" "+"性别："+(sexMan.isSelected()?"男":"女")+" "+"密码："+passwrd.getText();

showInfo.setText(info);

//调用清除方法

clearFunction();

}

else if(source==btnCancel)

{

clearFunction();

showInfo.setVisible(false);

}

}

//声明一个能够清除图形界面中输入内容的组件。

void clearFunction()

{

sexMan.setSelected(true);

txtName.setText("");

passwrd.setText("");

}

}

}

任务演练

package project903;

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.\*;

public class Project903 {

JFrame myWin;

JPanel deskPanel,headPanel,bodyPanel;

JTextField txtCalculate;

boolean flag=false;

Project903()

{

//创建顶级容器----窗口，设置窗体的显示大小、默认关闭操作、显示属性

myWin=new JFrame("计算器窗口");

myWin.setSize(400,500);

myWin.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);//设置窗体默认关闭操作

myWin.setVisible(true);

Font font=new Font("楷体",Font.BOLD,20);

//创建三个中级容器，将这两个中级容器添加到顶级容器中

deskPanel=new JPanel();

headPanel=new JPanel();

bodyPanel=new JPanel();

headPanel.setBackground(Color.yellow);

bodyPanel.setBackground(Color.PINK);

myWin.add(deskPanel);

//设置deskPanel布局管理器为边界布局管理器

deskPanel.setLayout(new BorderLayout());

deskPanel.add(headPanel,BorderLayout.NORTH);

deskPanel.add(bodyPanel,BorderLayout.CENTER);

//向Header容器添加计算显示的输入文本框

txtCalculate=new JTextField(30);

txtCalculate.setFont(font);

headPanel.add(txtCalculate);

//设置bodyPanel布局管理器为网格布局管理器

bodyPanel.setLayout(new GridLayout(4,4));

//向body容器中添加相应组件

String txtSymbol="0123456789.+-\*/=" ;

for(int index=0;index<16;index++)

{

JButton b1=new JButton(String.valueOf(txtSymbol.charAt(index)));

b1.setFont(font);

bodyPanel.add(b1);

b1.addActionListener(new MyActionListener());

}

}

public static void main(String[] args) {

new Project903();

}

//定义事件内部类

class MyActionListener implements ActionListener

{

public void actionPerformed(ActionEvent e)

{

if (!flag)

txtCalculate.setText("");

JButton btn=(JButton)e.getSource();

char btnName=btn.getText().charAt(0);

switch(btnName)

{

case '0':

case '1':

case '2':

case '3':

case '4':

case '5':

case '6':

case '7':

case '8':

case '9':

case '.':

case '+':

case '-':

case '\*':

case '/':

if(!flag)

flag=true;

txtCalculate.setText(txtCalculate.getText()+String.valueOf(btnName));

break;

case '=':

txtCalculate.setText(calculate());

flag=false;

}

}

//此处是调用一个专门用于计算算术表达式的方法，其中引用的Calculator类是从网上直接复制下来的，如果有想自己实现此算法的，也可以尝试编写相应的代码

public String calculate()

{

String result="";

result=String.valueOf(Calculator.conversion(txtCalculate.getText()));

return result;

}

}

}