



Java 程序设计

第9章 Swing 用户界面组件

田英鑫 tyx@hit.edu.cn

哈尔滨工业大学软件学院



第9章 Swing 用户界面组件

2/50

& 本章导读

- n 9.1 用户界面组件简介
- n 9.2 布局管理概述
- n 9.3 组件简介
- n 9.4 简单组件
- n 9.5 文本输入
- n 9.6 选择组件
- n 9.7 菜单
- n 9.8 对话框





第9章 Swing 用户界面组件

3/50

& 本章重点

- n 9.2 布局管理概述
- n 9.4 简单组件
- n 9.5 文本输入

& 本章难点

- n 9.7 菜单
- n 9.8 对话框



9.1 用户界面组件简介

n 容器类

n JFrame

n JDialog

n JPanel

n JApplet

扩展自 java.awt.Container 类

n 组件类

n JButton、JTextField、JMenu ...

n 辅助类

n LayoutManager

n Color、Font、FontMetrics、Dimension ...



9.2 布局管理概述

n 布局管理器

- n Java的UI组件在容器中放置，每个容器中都有布局管理器来安排UI组件的布局
- n 使用setLayout方法为容器对象设置布局管理器

n 常用的布局方式

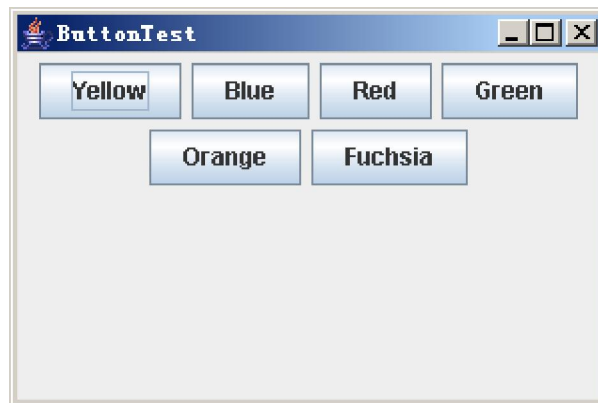
- n 流布局 (FlowLayout)
- n 边界布局 (BorderLayout)
- n 网格布局 (GridLayout)
- n 卡片布局 (CardLayout)
- n 不使用布局管理器



9.2 布局管理概述

n 流布局 (FlowLayout)

- n JPanel容器的默认布局为流布局
- n 在尽量少的行上排列组件，当一行没有足够的空间则开始新的一行
- n 组件的大小由其所显示的文本大小决定





9.2 布局管理概述

n 构造流布局管理器

n **FlowLayout()**

- n 构造一个FlowLayout对象，默认的对齐方式是居中，默认的水平间距和垂直间距都是5个像素

n **FlowLayout(int alignment)**

- n 根据指定的对齐方式构造一个FlowLayout对象，默认的水平间距和垂直间距都是5个像素

n **FlowLayout(int align,int hGap,int vGap)**

- n 根据指定的对齐方式、水平间距和垂直间距构造一个FlowLayout对象



9.2 布局管理概述

n 流布局 (FlowLayout)

```
public ButtonPanel()  
{  
    JButton yellowButton = new JButton("Yellow");  
    JButton blueButton = new JButton("Blue");  
    JButton redButton = new JButton("Red");  
    JButton greenButton = new JButton("Green");  
    JButton orangeButton = new JButton("Orange");  
    JButton fuchsiaButton = new JButton("Fuchsia");  
    add(yellowButton);  
    add(blueButton);  
    add(redButton);  
    add(greenButton);  
    add(orangeButton);  
    add(fuchsiaButton);  
}
```




9.2 布局管理概述

n 边界布局 (BorderLayout)

n 边界布局将容器划分为五个区域

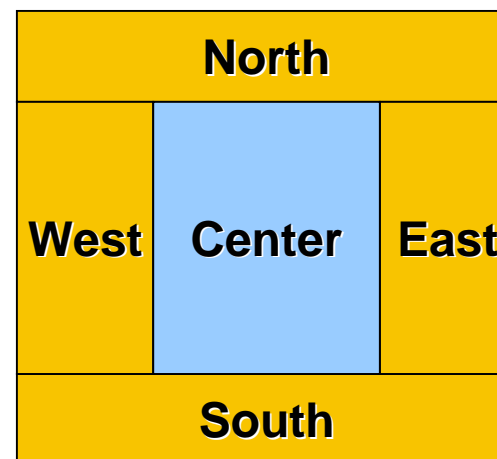
n 上(北)、下(南)、左(西)、右(东)、中央

n JFrame容器的默认布局为边界布局

n West和East区的宽度不变

n North和South区的高度不变

n Center区自动填充其他区域





9.2 布局管理概述

n 边界布局 (BorderLayout)

n 设置边界布局

- n `panel.setLayout(new BorderLayout());`

n 向使用边界布局管理器的容器中增加一个组件， 可以指定区域

- n `add(Component, constraint);`

n 组件被约束在如下区域

- n `BorderLayout.EAST`

- n `BorderLayout.SOUTH`

- n `BorderLayout.WEST`

- n `BorderLayout.NORTH`

- n `BorderLayout.CENTER` (默认值)



9.2 布局管理概述

n 边界布局 (BorderLayout)

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

public class BorderLayoutFrame extends JFrame {
    public BorderLayoutFrame() {
        Container container = getContentPane();
        container.setLayout(new BorderLayout(5,10));
        container.add(new JButton("East"),BorderLayout.EAST);
        container.add(new JButton("South"),BorderLayout.SOUTH);
        container.add(new JButton("West"), BorderLayout.WEST);
        container.add(new JButton("North"),BorderLayout.NORTH);
        container.add(new JButton("Center"),BorderLayout.CENTER);
    }
}
```



9.2 布局管理概述

n 面板 (JPanel)

n 可以使用面板组合 (嵌套) 构造灵活的用户界面

```
public ButtonFrame()  
{  
    ...  
    JPanel p = new JPanel();  
    ButtonPanel panel = new ButtonPanel();  
    p.setLayout(new BorderLayout());  
    p.add(panel, BorderLayout.SOUTH);  
    add(p);  
}
```

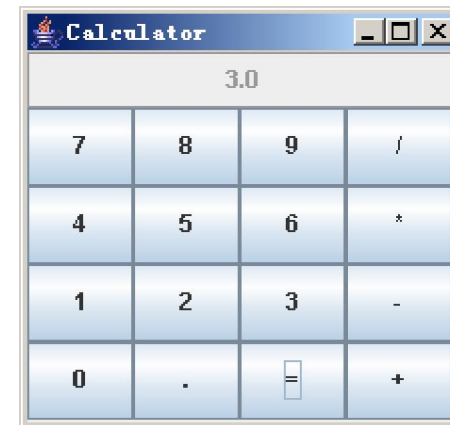




9.2 布局管理概述

n 网格布局 (GridLayout)

- n 网格布局根据指定的行数和列数将组件排列成网格形式
- n 网格内的每个空间大小相等，并充满整个单元格
- n 组件在网格中将被从左到右从第一行开始，然后是第二行 ... 依次放置





9.2 布局管理概述

n 构造网格布局管理器

n `public GridLayout(int rows,int columns)`

n 根据指定的行数和列数构造一个GirdLayout对象

n `public GridLayout(int rows, int columns, int hGap, int vGap)`

n 根据指定的行数和列数构造一个GirdLayout对象

n 组件在容器上的水平和垂直间距也按指定值设置

参见例9-1:Calculator.java



9.2 布局管理概述

n 卡片布局 (CardLayout)

n 卡片布局管理器将组件安放在一系列卡片上，一次只能看到一张卡片

n 构造方法

n `CardLayout()`

n 使用下面的方法将组件添加到容器中:

n `void add(Component com,String name)`

n 在卡片容器中显示组件

n `void first(container)`

n `void last(container)`

n `void next(container)`

n `void previous(container)`

n `void show(container, String name)`



9.2 布局管理概述

n 卡片布局 (CardLayout)

```
private CardLayout queue = new CardLayout();  
private JPanel cardPanel = new JPanel();  
for (int i=1; i<=15; i++)  
    cardPanel.add(  
        new JLabel(new ImageIcon("images/L"+i+".gif")),  
        String.valueOf(i)  
    );  
queue.first(cardPanel);  
queue.last(cardPanel);  
queue.previous(cardPanel);  
queue.next(cardPanel);  
queue.show(cardPanel, (String)e.getItem());
```

参见例:ShowCardLayout.java



9.2 布局管理概述

n 不使用布局管理器

- n 利用setLayout(null)语句说明不使用布局管理器
- n 组件必须通过它的实例方法setBounds放置到容器中的指定位置

```
private JButton jbt1 = new JButton("Cancel");  
private JButton jbt2 = new JButton("Ok");  
setLayout(null);  
add(jbt1);  
add(jbt2);  
jbt1.setBounds(100, 150, 100, 50);  
jbt2.setBounds(200, 150, 100, 50);
```

参见例:ShowNoLayout.java



9.3 组件简介

- n 组件的父类

- n JComponent

- n JComponent 类常用属性

- n toolTipTextfont

- n background

- n foreground

- n border

- n preferredSize

- n minimumSize

- n maximumSize



9.4 简单组件

n 按钮 (JButton)

n 按钮是一种点击时触发ActionEvent事件的组件

n 按钮的非默认构造方法

n JButton(String text)

n JButton(String text, Icon icon)

n JButton(Icon icon)

n 按钮属性

n text、icon、mnemonic (快捷键)

n horizontalAlignment、verticalAlignment

n horizontalTextPosition、verticalTextPosition



9.4 简单组件

n 按钮 (JButton)

n 响应按钮事件

n 事件类型 : `ActionEvent`

n 监听器类型 : `ActionListener`

```
public void actionPerformed(ActionEvent e)
{
    // 得到一个按钮标签
    String actionCommand = e.getActionCommand();
    // 确信事件源是一个按钮
    if (e.getSource() instanceof JButton)
        // 确信它是正确的按钮
        if ("My Button".equals(actionCommand))
            System.out.println ("Button pressed!");
}
```



9.4 简单组件

n 标签 (JLabel)

- n 标签是显示一小段文字、一幅图片或二者皆有的区域
- n 标签的非默认构造方法
 - n JLabel(String text)
 - n JLabel(String text, int horizontalAlignment)
 - n JLabel(Icon icon)
 - n JLabel(Icon icon, int horizontalAlignment)

参见例:LabelDemo.java



9.5 文本输入

n 文本域 (JTextField)

- n 文本域是一个用户可以输入字符的输入区
- n 文本域允许用户输入各种数据
- n 文本域的非默认的构造方法
 - n JTextField(int columns)
 - n JTextField(String text)
 - n JTextField(String text, int columns)
- n 文本域属性
 - n text、horizontalAlignment
 - n Editable、columns



9.5 文本输入

n 文本域 (JTextField)

n 文本域方法

n `getText()`

n 从文本域返回的是字符

n `setText(String text)`

n 将给出的字符放入文本域中

n `setEditable(boolean editable)`

n 使文本域可以编辑或不可以编辑，默认情况下是允许编辑的

n `setColumns(int)`

n 设置文本域的列数

参见例9-2: `TextTest.java`、`TextFieldDemo.java`



9.5 文本输入

n 密码域 (JPasswordField)

n 密码域是一种特殊的文本域，密码域将输入的字
符不直接显示出来，而用特殊的回显字符表示

n 构造密码域

n JPasswordField(String text,int columns);

n text: 将要显示的文本

n columns: 列数

n 密码域的方法

n void setEchoChar(char echo);

n char[] getPassword();



9.5 文本输入

n 文本区 (JTextArea)

n 文本区，它可以允许用户输入多行文字

n 文本区的构造方法

n `JTextArea(int rows,int columns)`

n 用指定的行数和列数创建一个文本区

n `JTextArea(String s,int rows,int columns)`

n 创建一个指定文本、行数和列数的文本区

n 文本区的属性

n `text`、`editable`、`columns`、`lineWrap`

n `wrapStyleWord`、`rows`、`lineCount`

n `tabSize`

参见例9-4: `TextAreaTest.java`、`TextAreaDemo.java`



9.5 文本输入

n 文本区 (JTextArea)

- n 文本区没有滚动条，如果需要滚动条，可以把文本区放到一个滚动窗格 (ScrollPane) 中
- n 滚动窗格是一个支持不要编码自动滚动的组件

参见例:ScrollPaneDemo.java



9.6 选择组件

n 复选框 (JCheckBox)

n 复选框是一种能够打开、关闭选项的组件

n 复选框自动带有标识标签

n 复选框构造方法

n JCheckBox(String text)

n JCheckBox(String text, boolean selected)

n JCheckBox(Icon icon)

n JCheckBox(String text, Icon icon)

n JCheckBox(String text, Icon icon, boolean selected)



9.6 选择组件

n 复选框 (JCheckBox)

n 复选框除了有在按钮中的所有属性以外，还有下列属性

n selected

n 复选框方法

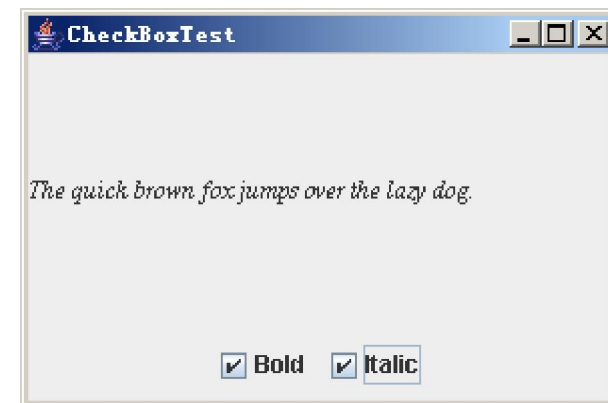
n void setSelected(boolean state);

n boolean isSelected();

n 复选框事件

n(ActionEvent)

n(ItemEvent)



参见例9-5:CheckBoxTest.java



9.6 选择组件

n 单选按钮 (JRadioButton)

n 单选按钮是一种变化的选择框，让用户从一组选项中选择唯一的一个选项

n 单选按钮的构造方法

n JRadioButton()

n JRadioButton(String text)

n JRadioButton(String text, boolean selected)

n JRadioButton(Icon icon)

n JRadioButton(String text, Icon icon)

n JRadioButton(String text, Icon icon, boolean selected)



9.6 选择组件

n 单选按钮 (JRadioButton)

n 单选按钮属性

n selected

n 单选按钮组

```
ButtonGroup btg = new ButtonGroup();  
btg.add(jrb1);  
btg.add(jrb2);
```



参见例9-6:RadioButtonTest.java、 RadioButtonDemo.java



9.6 选择组件

n 组合框 (JComboBox)

- n 组合框 (combo box) 也叫下拉列表框，它是一些项目的简单列表
- n 组合框可以限制用户的选择范围并可以避免对输入数据有效性验证
- n 构造方法
 - n JComboBox()
 - n JComboBox(Object[] stringItems)



9.6 选择组件

n 组合框 (JComboBox)

n 组合框属性

n selectedIndex

n selectedItem

n 组合框方法

n addItem(Object item)

n getItemAt(int index)

n getSelectedIndex()

n removeItem(Object anObject)

n removeAllItems()



9.6 选择组件

n 组合框 (JComboBox)

n 组合框事件

n ActionEvent事件

n ItemEvent事件，监听器需要重写itemStateChanged方法处理该事件

```
public void itemStateChanged(ItemEvent e)
{
    // Make sure the source is a combo box
    if (e.getSource() instanceof JComboBox)
        String s = (String)e.getItem();
}
```

参见例9-8: [ComboBoxTest.java](#)、[ComboBoxDemo.java](#)



9.6 选择组件

n 列表框 (JList)

n 列表框的作用与组合框基本相同，但它允许用户同时选择一项或多项

n 列表框的构造方法

n JList()

n 创建一个空的列表框

n JList(Object[] stringItems)

n 创建的一个新列表框，并使用一组字符串来初始化

n 列表框的属性

n selectedIndex、selectedIndices

n selectedValue、selectedValues

n selectionMode、visibleRowCount



9.6 选择组件

- n 列表框 (JList)

- n 列表框事件

- n ListSelectionEvent事件

- n 监听器需要重写valueChanged方法处理该事件

```
public void valueChanged(ListSelectionEvent e)
{
    .....
}
```

参见例:ListDemo.java



9.7 菜单

n 菜单概述

n JFrame或者JApplet 可以拥有一个菜单。菜单由用户可以选择（或开关）的菜单项组成。菜单栏作为支持菜单结构进行显示

n Java在框架中提供了五个实现菜单的类：

- n JMenuBar //窗口顶部的菜单栏
- n JMenu //菜单
- n JMenuItem //菜单项
- n JCheckBoxMenuItem //复选框菜单项
- n JRadioButtonMenuItem //单选按钮菜单项



9.7 菜单

n 菜单栏 (JMenuBar)

- n 菜单栏中保存菜单，菜单栏只能被增加到框架中，下面的代码是创建和增加菜单栏到框架中

```
JFrame frame = new JFrame();  
JMenuBar menuBar = new JMenuBar(); //构造菜单栏对象  
frame.setJMenuBar(menuBar);        //将菜单栏添加到窗口
```



9.7 菜单

n 菜单 (JMenu)

- n 需要为每个菜单建立一个菜单对象，然后添加到菜单栏中
- n 下面代码创建了一个Edit菜单，并将它添加到JMenuBar 实例menuBar中

```
JMenu editMenu = new JMenu("Edit");  
menuBar.add(editMenu);
```



9.7 菜单

n 菜单项 (JMenuItem)

- n 可以将菜单项增加到菜单中，下面的代码是增加菜单项并对菜单项进行分组

```
fileMenu.add(new JMenuItem("new"));  
JMenuItem pasteItem = new JMenuItem("Paste");  
editMenu.add(pasteItem);  
editMenu.addSeparator(); //增加分割线
```



9.7 菜单

n 子菜单

- n 可以添加子菜单到菜单项目中
- n 下面的代码是在菜单项目“Options”中添加子菜单“Unix”和“Win95”

```
JMenu optionMenu = new JMenu("Options");  
editMenu.add(optionMenu);  
// 可以继续给子菜单optionMenu添加下级菜单项
```




9.7 菜单

n 复选框菜单项

- n JCheckBoxMenuItem是JMenuItem的子类，它在JMenuItem上添加一个布尔状态，当状态为真时，该菜单项前显示对号
- n 下面代码在菜单中加入一个复选框菜单项

```
JCheckBoxMenuItem raadonlyItem =  
    new JCheckBoxMenuItem("Read-only");  
optionMenu.add(raadonlyItem);
```



9.7 菜单

n 单选按钮菜单项

- n 使用JRadioButtonMenuItem可以在菜单中加入单选按钮
- n 单选按钮菜单常用于菜单中一组相互排斥的选项

```
ButtonGroup group = new ButtonGroup();
JRadioButtonMenuItem insertItem =
    new JRadioButtonMenuItem("Insert");
insert.setSelected(true);
JRadioButtonMenuItem overtypeItem =
    new JRadioButtonMenuItem("Overtime");
group.add(insertItem);
group.add(overtimeItem);
optionsMenu.add(insertItem);
optionsMenu.add(overtimeItem);
```

参见例9-11:MenuTest.java



9.7 菜单

n 工具栏 (JToolBar)

n 创建工具栏

n `JToolBar bar = new JToolBar();`

n 将工具栏添加到窗口

n `frame.add(bar, BorderLayout.NORTH);`

参见例9-12: `ToolBarTest.java`、`ToolBarDemo.java`



9.8 对话框

n 选项对话框

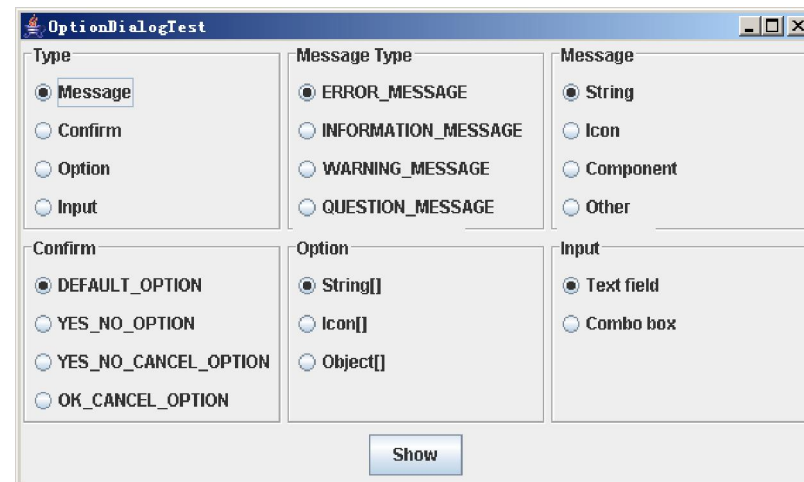
n JOptionPane类的四个用于构造简单对话框的静态方法

n showMessageDialog

n showConfirmDialog

n showOptionDialog

n showInputDialog



参见例9-18:OptionDialogTest.java

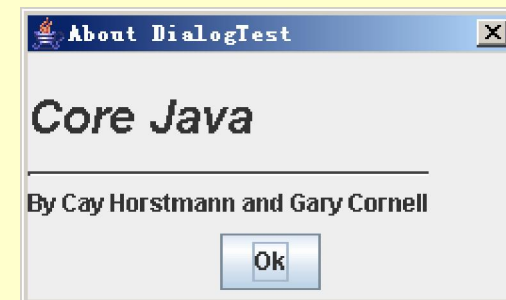


9.8 对话框

n 创建对话框

n 实现自定义对话框需要从JDialog派生一个类

```
public AboutDialog extends JDialog {  
    public AboutDialog(JFrame owner) {  
        super(owner, "About DialogTest", true);  
        add(new JLabel(...), BorderLayout.CENTER);  
        JPanel panel = new JPanel();  
        JButton ok = new JButton("Ok");  
        ok.addActionListener(  
            ...  
        );  
        panel.add(ok);  
        add(panel, BorderLayout.SOUTH);  
        setSize(250, 150);  
    }  
}
```



参见例9-19:DialogTest.java



9.8 对话框

n 数据交换

参见例9-20:DateExchangeTest.java



9.8 对话框

n 文件对话框 (JFileChooser)

n 构造文件对话框对象

n `JFileChooser chooser = new JFileChooser();`

n 设置当前目录

n `setCurrentDirectory`

n 设置默认选择的文件名

n `setSelectedFile`

n 允许选择多个文件

n `setMultiSelectionEnabled`

n 设置可以选择目录

n `setFileSelectionMode`



9.8 对话框

n 文件对话框 (JFileChooser)

n 显示对话框

- n showOpenDialog

- n showSaveDialog

n 获取返回信息

- n getSelectedFile

- n getSelectedFiles

- n `String filename = getSelectedFile().getPath();`

n 设置过滤器

- n setFileFilter

参见例9-21:FileChooserTest.java



9.8 对话框

- n 颜色选择器 (JColorChooser)

- n 显示颜色选择器模式对话框

- n Color selectedColor;

- n selectedColor = JColorChooser.showDialog(...);

参见例9-22:ColorChooserTest.java



第9章 Swing 用户界面组件

50/50



Any Question?