

第九章 GUI编程

本章内容

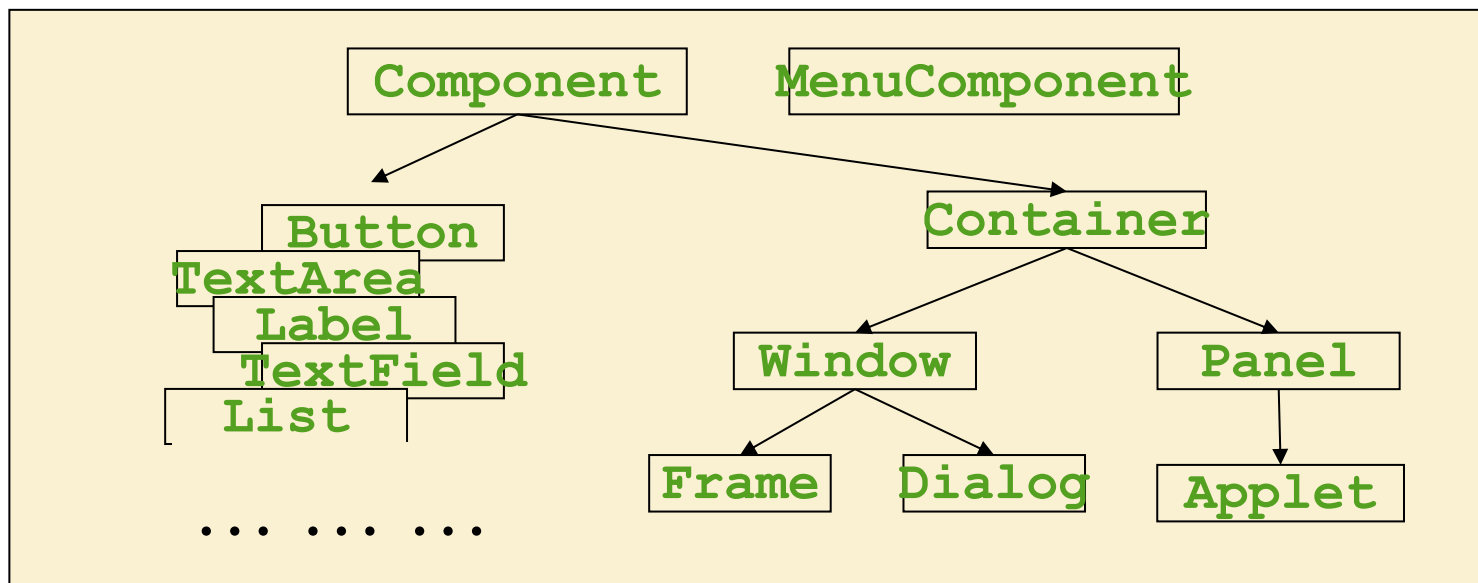
- ▶ AWT (SWING、SWT)
- ▶ 组件和容器
- ▶ 布局管理器
- ▶ 事件处理

GUI -- Graphical User Interface

- ▶ **AWT** (抽象窗口工具集) AbstractWindow Toolkit
 - 为简单的applet程序设计
 - AWT没有提供足够的桌面端的程序所需的特性
 - 它不适宜用来构建丰富的桌面图形界面
- ▶ **Swing**
 - Swing是AWT的扩展和功能加强
 - 是构建java图形界面标准的API
- ▶ **SWT** (Standard Widget Toolkit)
 - 是一个开源的GUI编程框架
 - SWT是IBM为它的Eclipse集成开发环境而开发的图形用户界面工具

AWT

- ▶ AWT(Abstract Window Toolkit)包括了很多类和接口，用于Java Application 的GUI（Graphics User Interface 图形用户界面）编程。
- ▶ GUI的各种元素（如：窗口，按钮，文本框等）由Java 类来实现。
- ▶ 使用AWT所涉及的类一般在java.awt包及其子包中。
- ▶ Container和Component是AWT中的两个核心类。Modal Modeless



Component & Container

- ▶ Java的图形用户界面的最基本组成部分是Component，Component类及其子类的对象用来描述以图形化的方式显示在屏幕上并能与用户进行交互的GUI元素，例如，一个按钮，一个标签等。
 - ▶ 一般的Component对象不能独立地显示出来，必须将“放在”某一的Container对象中才可以显示出来。
- Container是Component子类，Container子类对象可以“容纳”别的Component对象。
 - Container对象可使用方法add(..)向其中添加其他Component对象。
 - Container是Component的子类，因此Container对象也可以被当作Component对象添加到其他Container对象中。
 - 有两种常用的Container：
 - Window: 其对象表示自由停泊的顶级窗口
 - Panel: 其对象可作为容纳其它Component对象，但不能独立存在，必须被添加到其它Container中(如Window或Applet)

Frame

- ▶ Frame是Window的子类，由Frame或其子类创建的对象为一个窗体。
- ▶ Frame的常用构造方法：
 - Frame()
 - Frame(String s) 创建标题栏为字符串s的窗口。

setBounds(int x,int y,int width,int height)

设置窗体位置和大小，x，y是左上角坐标，
width和height是宽度和高度

setSize(int width,int height)

设置窗体的位置，x，y是左上角坐标

setLocation(int x,int y)

设置窗体的大小，width和height分别是宽度和高度。

setBackground(Color c)

设置背景颜色，参数为Color对象。

setVisible(boolean b)设置是否可见。

setTitle(String name) String getTitle()

setResizable(boolean b)设置是否可以调整大小。

TestFrame.java

基本窗口显示

TestMultiFrame.java

显示多个不同背景色的
窗口

Panel

- ▶ Panel对象可以看成可以容纳Component的空间
- ▶ Panel对象可以拥有自己的布局管理器
- ▶ Panel类拥有从其父类继承来的
 - `setBounds(int x,int y,int width,int height)`
 - `setSize(int width,int height)`
 - `setLocation(int x,int y)`
 - `setBackground(Color c)`
 - `setLayout(LayoutManager mgr)` 等方法。
- ▶ Panel的构造方法为：
 - `Panel()` 使用默认的 `FlowLayout`类布局管理器初始化。
 - `Panel(LayoutManager layout)`使用指定的布局管理器初始化。

TestPanel.java
Panel基础用法
TestMultiPanel.java
多个Panel

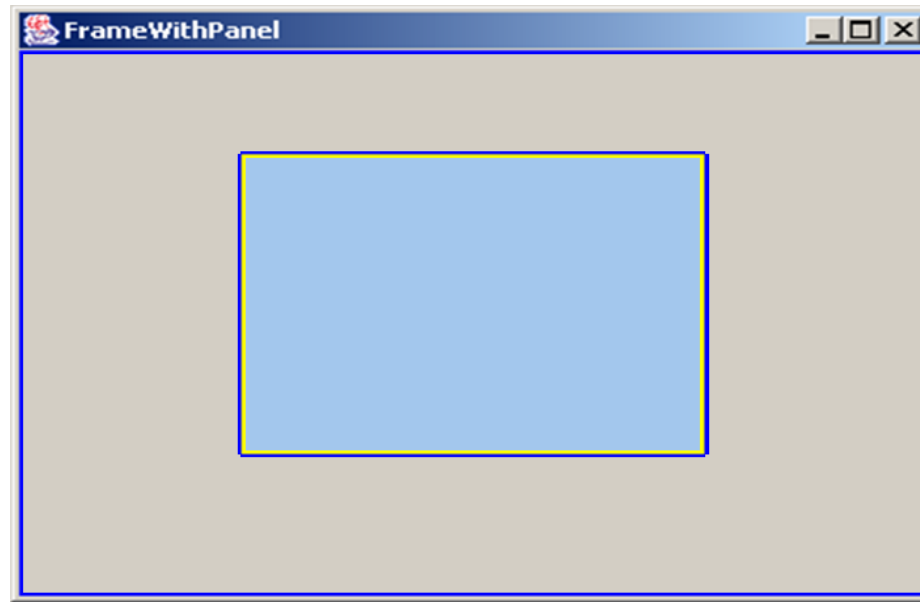
课堂练习

课堂练习



CenterPanel.java

设计一个含有Panel的自定义的Frame类含有，形式如下图：

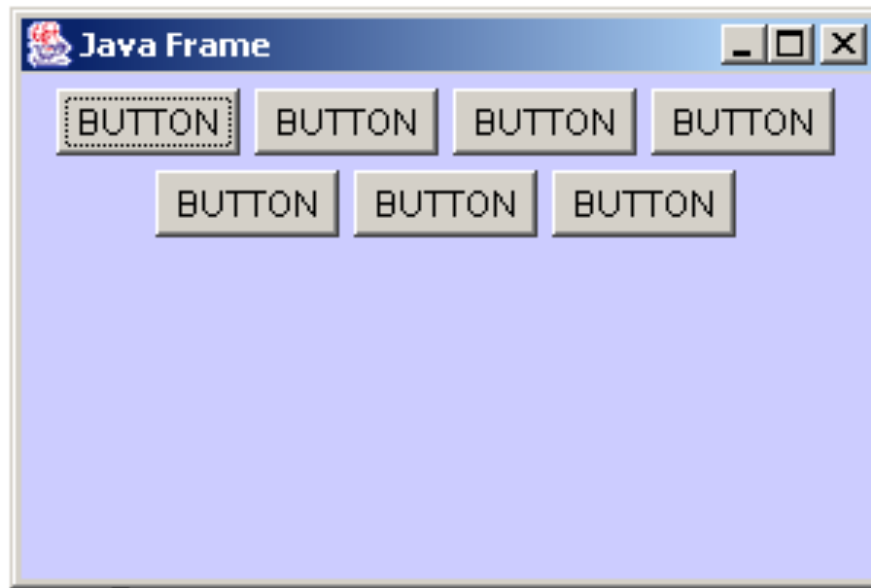


布局管理器

- ▶ Java语言中，提供了布局管理器类的对象可以管理
 - 管理Component在Container中的布局，不必直接设置Component位置和大小。
 - 每个Container都有一个布局管理器对象，当容器需要对某个组件进行定位或判断其大小尺寸时，就会调用其对应的布局管理器,调用Container的setLayout方法改变其布局管理器对象。
- ▶ Awt提供了5种布局管理器类：
 - FlowLayout
 - BorderLayout
 - GridLayout
 - CardLayout
 - GridBagLayout

FlowLayout布局管理器

- ▶ FlowLayout是Panel类的默认布局管理器。
 - FlowLayout布局管理器对组件逐行定位，行内从左到右，一行排满后换行。
 - 不改变组件的大小，按组件原有尺寸显示组件，可设置不同的组件间距,行距以及对齐方式。
- ▶ FlowLayout布局管理器默认的对齐方式是居中。



FlowLayout 的构造方法

- ▶ `new FlowLayout(FlowLayout.RIGHT,20,40);`
 - 右对齐，组件之间水平间距20个像素，垂直间距40个像素。
- ▶ `new FlowLayout(FlowLayout.LEFT);`
 - 左对齐，水平和垂直间距为缺省值（5）。
- ▶ `new FlowLayout();`
 - 使用缺省的居中对齐方式，水平和垂直间距为缺省值（5）。

TestFlowLayout.java
FlowLayout初步用法
TestFlowLayout2.java
FlowLayout多个Button

BorderLayout 布局管理器

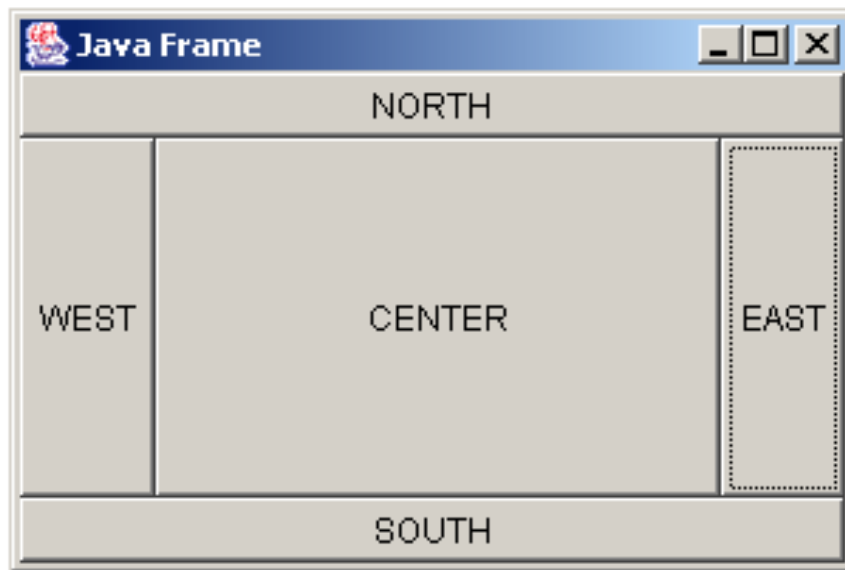
- ▶ BorderLayout是Frame类的默认布局管理器。
- ▶ BorderLayout将整个容器的布局划分成
 - 东 (EAST)
 - 西 (WEST)
 - 南 (SOUTH)
 - 北 (NORTH)
 - 中 (CENTER) 五个区域，组件只能被添加到指定的区域。
- ▶ 如不指定组件的加入部位，则默认加入到CENTER区。
- ▶ 每个区域只能加入一个组件，如加入多个，则先前加入的会被覆盖。

BorderLayout 布局管理器

► BorderLayout型布局容器尺寸缩放原则：

- 北、南两个区域在水平方向缩放。
- 东、西两个区域在垂直方向缩放。
- 中部可在两个方向上缩放。

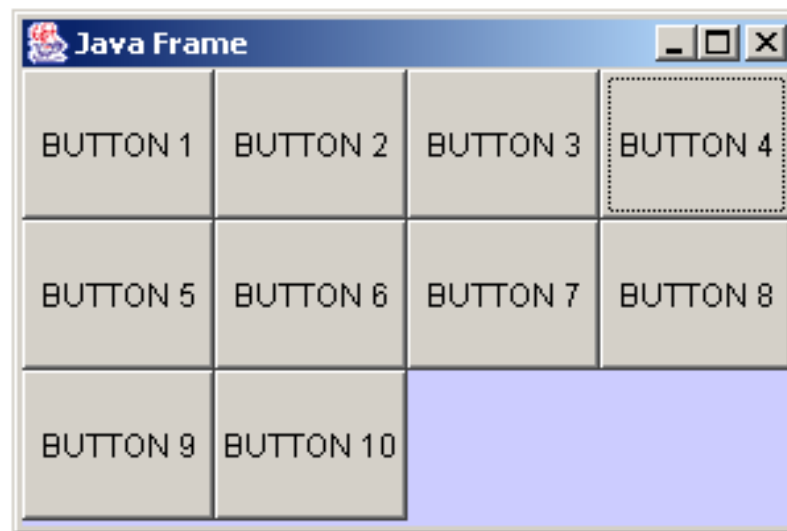
TestBorderLayout.java



GridLayout 布局管理器

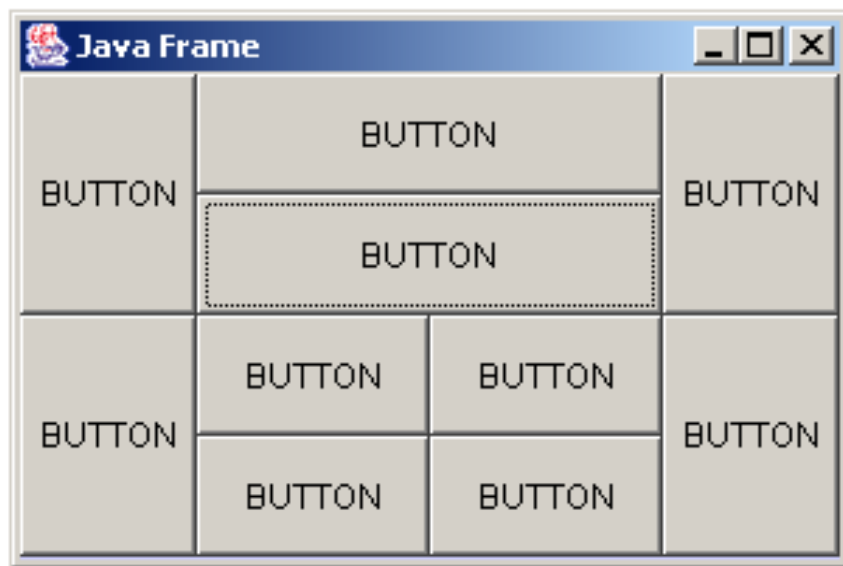
- ▶ GridLayout型布局管理器将空间划分成规则的矩形网格，每个单元格区域大小相等。组件被添加到每个单元格中，先从左到右添满一行后换行，再从上到下。
- ▶ 在 GridLayout 构造方法中指定分割的行数和列数：
 - 如：GridLayout(3,4)

TestGridLayout.java



课堂练习

- 使用Container的嵌套实现下面布局。

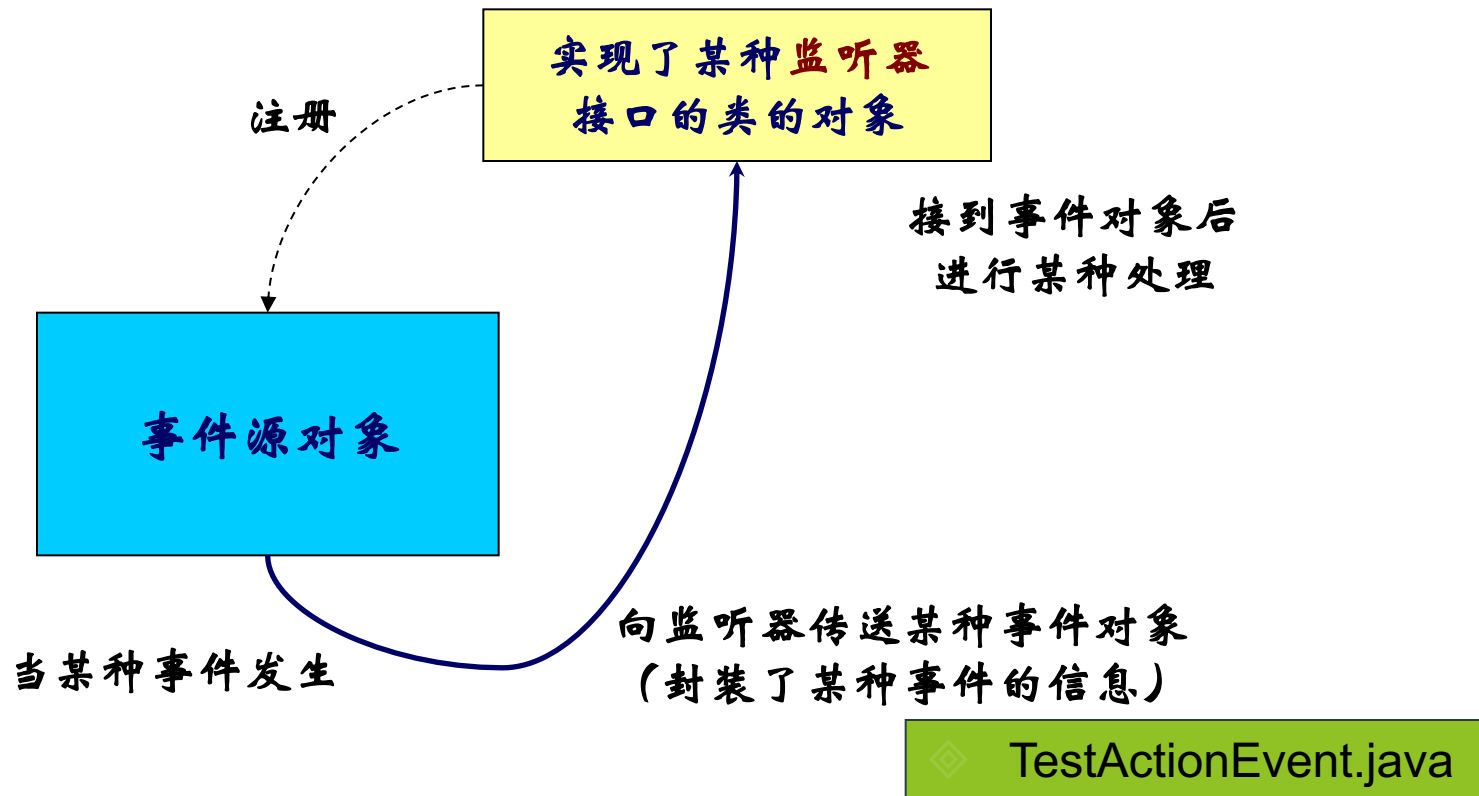


TenButtons.java

布局管理器总结

- ▶ Frame是一个顶级窗口，Frame的缺省布局管理器为BorderLayout
- ▶ Panel无法单独显示，必须添加到某个容器中。
 - ▶ Panel的缺省布局管理器为FlowLayout。
- ▶ 当把Panel作为一个组件添加到某个容器中后，该Panel仍然可以有自己的布局管理器。
- ▶ 使用布局管理器时，布局管理器负责各个组件的大小和位置，因此用户无法在这种情况下设置组件大小和位置属性，如果试图使用Java语言提供的setLocation()，setSize()，setBounds()等方法，则都会被布局管理器覆盖。
- ▶ 如果用户确实需要亲自设置组件大小或位置，则应取消该容器的布局管理器，方法为：
 - ▶ `setLayout(null)`

事件监听



TextField 类

- ▶ java.awt.TextField类用来创建文本框对象。
- ▶ TextField 有如下常用方法：

```
TextField()  
TextField(int columns)  
TextField(String text)  
TextField(String text, int columns)
```

```
public void setText(String t)  
public String getText()  
public void setEditable(boolean b)  
public boolean isEditable()  
public void setBackground(Color c)  
public void select(int selectionStart,int  
selectionEnd)  
public void selectAll()  
public void addActionListener(ActionListener l)  
    添加动作监听器。
```

TextField事件监听

- ▶ TextField 对象可能发生Action（光标在文本框内敲回车）事件。与该事件对应的事件类是java.awt.event.ActionEvent。
- ▶ 用来处理 ActionEvent事件是实现了java.awt.event. ActionListener 接口的类的对象。ActionListener接口定义有方法：
 - public void actionPerformed(ActionEvent e)
- ▶ 实现该接口的类要在该方法中添加处理该事件（Action）的语句。
- ▶ 使用 addActionListener(ActionListener l)方法为TextField 对象注册一个ActionListener对象，当TextField对象发生Action事件时，会生成一个ActionEvent对象，该对象作为参数传递给ActionListener对象的actionPerformed方法在方法中可以获取该对象的信息，并做相应的处理。

```
TFActionEvent.java //TextField事件监听初步  
TFMath / 2 / 3 / 4.java //数学运算及内部类初步
```

内部类

- ▶ 在Java中，允许一个类的定义位于另一个类的内部，前者称为内部类
- ▶ 内部类和外层封装它的类之间存在逻辑上的所属关系
- ▶ Inner class的名字不能与包含它的类名相同；
- ▶ Inner class可以访问包含它的类的所有成员；

内部类总结

- ▶ 好处:
 - 可以方便的访问包装类成员
 - 可以更清楚的组织逻辑，防止不应该被其它类访问的类进行访问
- ▶ 何时使用
 - 该类不允许或者不需要其它类进行访问时候