

A hand in a suit jacket points towards the word "Java" written in a large, bold, dark blue font. The background is a light blue gradient with a grid of hexagonal icons containing various symbols like a location pin, envelope, globe, alarm clock, magnifying glass, gear, house, phone, mobile phone, refresh, and location pin. The word "Java" is the central focus, with a hand pointing at it from the left.

Java

JAVA程序设计



中国科技出版传媒股份有限公司
China Science Publishing & Media Ltd. (CSPM)
科学出版社

目录 CONTENTS

1 学习指南

2 难点重点

3 知识内容

4 本章小结

```
h3 {font-size: 20px !important;}
h4 {font-size: 16px; text-align: left;}
hr {margin: 3px !important; padding: 0px !important; padding-top: 5px !important; border-top: 1px solid #ccc !important;}

#container {margin: auto; width: 850px; padding-top: 90px;}

#info_bar_line1 {font-weight: bold; font-size: 20px; margin: 0; padding: 0; text-align: left;}
#info_bar_line2 {font-size: 14px; margin: 0; text-align: left;}
.info_bar {width: 100%; background-color: #4288c4; position: fixed; padding: 10px 20px; z-index: 10;}
.info_bar p {color: #ffffff !important;}

.hide {display: none;}

.field_information {cursor: pointer; float: left; margin: 1px 0 0 5px;}
.field_information_container {float: left;}
.label {font-size: 12x !important;}
.btn_copy_text {width: 110px;}
.btn_get_first {width: 110px;}

.title {width: 701px !important;}
.description {width: 701px !important; height: 73px !important;}

.tag_editor {line-height: 25px !important; height: 225px; padding: 3px 0px !important; border: 1px solid #ccc !important; border-radius: 4px;}
.tag_editor_delete {height: 25px !important;}
.tag_editor_delete i {line-height: 25px !important;}
.tag_editor_spacer {width: 10px !important;}

@btn_settings { webkit-user-select: none; -ms-user-select: none; user-select: none; -ms-user-select: none; user-select: none; transition: all 0.5s ease-out 0s;}
@btn_settings:shower {cursor: pointer; transform: rotate(180deg); transition: all 0.5s ease-out 0s;}

$select_themes_container {width: 280px;}
$google_api_key {width: 400px;}
$get_first_n_value {width: 50px;}
.simple_text {text-decoration: none !important;}
.panel_settings {padding: 10px !important;}
.panel_settings_container {margin-bottom: 5px !important;}

$google_translate_api_info {font-size: 10px; margin-left: 35px;}
.checkbox_comment {font-size: 10px;}
.btn_default .badge {margin-left: 3px; border-radius: 5px !important;}
na*k {padding: 0 !important;}

$add_and_translate {font-size: 10px;}

.tooltipster-box {background: #fff !important;}
.tooltipster-arrow-background {border-top-color: #fff !important;}
.tooltipster-box {-webkit-box-shadow: 0 1px 4px rgba(0,0,0,.2); box-shadow: 0 1px 4px rgba(0,0,0,.2)}
```



1

学习指南

```
port class MainActivity;
@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
}
public void onClick(View view) {
    Intent i = new Intent("net.l
```

本章主要通过详尽的实例，配以合理的练习，详细介绍Java中常用的AWT和Swing组件，GUI的相关概念和使用方法，另外介绍了三种布局管理器，介绍了GUI中常用的类库以及常用类。通过本章的学习，读者需要掌握使用合理的布局管理我们的AWT和Swing中常用组件以及掌握图形用户界面GUI程序的编写。

```
h3 {font-size: 20px !important;}
h4 {font-size: 16px; text-align: left;}
hr {margin: 3px !important; padding: 0px !important; padding-top: 5px !important; border-top: 1px solid #ccc !important;}
```

```
#container {margin: auto; width: 850px; padding-top: 90px;}
#info_bar_line1 {font-weight: bold; font-size: 20px; margin: 0; padding: 0; text-align: left;}
#info_bar_line2 {font-size: 14px; margin: 0; text-align: left;}
.info_bar {width: 100%; background-color: #4288c4; position: fixed; padding: 10px 20px; z-index: 10;}
.info_bar p {color: #ffffff !important;}
```

```
.hide {display: none;}
.field_information {cursor: pointer; float: left; margin: 1px 0 0 5px;}
.field_information_container {float: left;}
.label {font-size: 12px !important;}
.btn_copy_text {width: 110px;}
.btn_get_first {width: 110px;}
```

```
.title {width: 70px !important;}
.description {width: 70px !important; height: 73px !important;}
```

```
.tag_editor {line-height: 25px !important; height: 225px; padding: 5px 0px !important; border: 1px solid #ccc !important; border-radius: 4px;}
.tag_editor_delete {height: 25px !important;}
.tag_editor_delete i {line-height: 25px !important;}
.tag_editor_spacer {width: 10px !important;}
```

```
@btn_settings { webkit-user-select: none; -ms-user-select: none; user-select: none; -ms-user-select: none; user-select: none; transition: all 0.5s ease-out 0s;}
```

```
#select_theme_container {width: 280px;}
#google_api_key {width: 400px;}
#get_first_value {width: 50px;}
.simple_text {text-decoration: none !important;}
.pamel_settings {padding: 10px !important;}
.pamel_settings_container {margin-bottom: 5px !important;}
```

```
#google_translate_api_info {font-size: 10px; margin-left: 35px;}
.checkbox_comment {font-size: 10px;}
.btn_default .badge {margin-left: 3px; border-radius: 5px !important;}
na*k {padding: 0 !important;}
```

```
#add_and_translate {font-size: 10px;}
```

```
.tooltipster-box {background: #fff !important;}
.tooltipster-arrow-background {border-top-color: #fff !important;}
.tooltipster-box {-webkit-box-shadow: 0 1px 4px rgba(0,0,0,.2); box-shadow: 0 1px 4px rgba(0,0,0,.2)}
```



2

难点重点

难点重点



常用组件：按钮、菜单、标签、几种选择组件



颜色和字体的设置



图形界面窗口的设计



几种容器的使用和区别



三种布局管理器



事件模型及事件处理



```
h3 {font-size: 20px !important;}
h4 {font-size: 16px; text-align: left;}
hr {margin: 3px !important; padding: 0px !important; padding-top: 5px !important; border-top: 1px solid #ccc !important;}

#container {margin: auto; width: 850px; padding-top: 90px;}

#info_bar_line1 {font-weight: bold; font-size: 20px; margin: 0; padding: 0; text-align: left;}
#info_bar_line2 {font-size: 14px; margin: 0; text-align: left;}
.info_bar {width: 100%; background-color: #4288c4; position: fixed; padding: 10px 20px 0 10px;}
.info_bar p {color: #ffffff !important;}

.hide {display: none;}

.field_information {cursor: pointer; float: left; margin: 1px 0 0 5px;}
.field_information_container {float: left;}
.label {font-size: 12px !important;}
.btn_copy_text {width: 110px;}
.btn_get_first {width: 110px;}

.title {width: 70px !important;}
.description {width: 70px !important; height: 75px !important;}

.tag_editor {line-height: 25px !important; height: 225px; padding: 5px 0px !important; border: 1px solid #ccc !important; border-radius: 4px;}
.tag_editor_delete {height: 25px !important;}
.tag_editor_delete i {line-height: 25px !important;}
.tag_editor_spacer {width: 10px !important;}

#btn_settings {width: user-select: none; height: user-select: none; width: user-select: none; height: user-select: none; user-select: none; transition: all 0.5s ease-out 0s;}
#btn_settings:hover {cursor: pointer; transform: rotate(180deg); transition: all 0.5s ease-out 0s;}

#select_theme_container {width: 280px;}
#google_api_key {width: 400px;}
#get_first_value {width: 50px;}
.simple_text {text-decoration: none !important;}
.panel_settings {padding: 10px !important;}
.panel_settings_container {margin-bottom: 5px !important;}

#google_translate_api_info {font-size: 10px; margin-left: 35px;}
.checkbox_comment {font-size: 10px;}
.btn_default .badge {margin-left: 5px; border-radius: 5px !important;}
na*k {padding: 0 !important;}

#add_and_translate {font-size: 10px;}

.tooltipster-box {background: #fff !important;}
.tooltipster-arrow-background {border-top-color: #fff !important;}
.tooltipster-box {width: box-shadow: 0 1px 4px rgba(0,0,0,.2); box-shadow: 0 1px 4px rgba(0,0,0,.2)}}
```



3

知识内容

1. 图形用户界面

1.1 Java的图形用户界面软件包

Java Foundation Classes的五个部分：

- (1) AWT组件：旧有的窗口组件包。
- (2) Swing组件：新的窗口组件包。
- (3) Accessibility API：提供一种更先进的沟通界面。例如语音输入或触摸屏。
- (4) Java 2D API：提供强大的图形处理函数。
- (5) 支持Drag and Drop：类似于Windows中开两个窗口做文件的Drag and Drop (Copy and Paste) 功能，即在两个相同的java界面，甚至是Java与其他应用程序界面作数据交换的操作。

1. 图形用户界面

AWT和Swing的总结

AWT 是基于本地方法的C/C++程序，其运行速度比较快；Swing是基于AWT 的Java程序，其运行速度比较慢。对于一个嵌入式应用来说，目标平台的硬件资源往往非常有限，而应用程序的运行速度又是项目中至关重要的因素。在这种矛盾的情况下，简单而高效的AWT 当然成了嵌入式Java的第一选择。而在普通的基于PC或者是工作站的标准Java应用中，硬件资源对应用程序所造成的限制往往不是项目中的关键因素，所以在标准版的Java中则提倡使用Swing，也就是通过牺牲速度来实现应用程序的功能。

1. 图形用户界面

1.2 窗口

`Frame()`构造一个最初不可见的 `Frame` 新实例 ()。

`Frame(String title)`构造一个新的、最初不可见的、具有指定标题的 `Frame` 对象。

`setBounds(int x,int y,int width,int height)`从 `Windows`继承而来，移动窗口并调整其大小。由 `x` 和 `y` 指定左上角的新位置，由 `width` 和 `height` 指定新的大小。

`setSize(int width,int height)`从 `Windows`继承而来，调整窗口的大小，使其宽度为 `width`，高度为 `height`。

1. 图形用户界面

1.2 窗口

setBackground(Color c) 从类 `java.awt.Component` 继承的方法，设置组件的背景色。

setVisible(boolean b) 从类 `java.awt.Window` 继承的方法，根据参数 `b` 的值显示或隐藏此 `Window`。

pack() 从类 `java.awt.Window` 继承的方法，调整此窗口的大小，以适合其子组件的首选大小和布局。如果该窗口和/或其所有者还不可显示，则在计算首选大小之前都将变得可显示。在计算首选大小之后，将会验证该窗口。

1. 图形用户界面

1.3 标签与按钮

同时，Label类提供3种方法来改变对象的属性：

(1) 返回标签对象的文本值(public String getText());

(2) 重置标签对象的对齐方式(public void setAlignment(int alignment);

(3) 重置标签对象的文本值(public void setText(String text)。

1. 图形用户界面

1.3 标签与按钮

几种常用的方法：

addNotify()用来创建一个按钮的实体。

getLabel()返回按钮的标签。

getActionCommand()返回由按钮发出的action事件的命令名称。

addActionListener(ActionListener)用来添加指定的事件接收器，并同时接收按钮发出的action动作。

1. 图形用户界面

1.4 文本框和文本区

文本框的建立：

TextField tf1, tf2, tf3, tf4;

文本框所在位置java.awt.TextField ;

空文本 tf1 = new TextField();

允许最多输入20个字符的空文本框 tf2 = new TextField("", 20);

默认显示“Hello”的文本框 tf3 = new TextField("Hello!");

**允许最多输入30个字符且默认显示“Hello”的文本框 tf4 = new
TextField("Hello", 30);**

1. 图形用户界面

1.4 文本框和文本区

常用方法清单：

- ① `TextField()` 构造新文本字段。
- ② `TextField(int columns)` 构造具有指定列数的新空文本字段。
- ③ `TextField(String text)` 构造使用指定文本初始化的新文本字段。
- ④ `TextField(String text, int columns)` 构造使用要显示的指定文本初始化的新文本字段，宽度足够容纳指定列数
- ⑤ `setText(String t)` 将此文本组件显示的文本设置为指定文本。

1. 图形用户界面

1.5 复选框、单选按钮、列表框和下拉列表

Checkbox类用于创建复选框按钮和单选按钮，它们是具有开关或真假值状态的按钮。

创建复选按钮框的构造函数是：`public Checkbox (Strng s);`
创建的复选框具有一个标签s，Checkbox对象初始化时是未选中的。

1. 图形用户界面

1.5 复选框、单选按钮、列表框和下拉列表

为了响应及处理Checkbox事件，applet程序要实现接口ItemListener;并为Checkbox 对象接上监听器AddItemListener (this);

事件处理的专门方法为itemStateChanged (ItemEvent e)。

当某个复选框被选中后，方法getState()返回true，再次点击被选中的复选框，则选择被取消，getState()返回false

1. 图形用户界面

1.5 复选框、单选按钮、列表框和下拉列表

单选按钮是在一组按钮中，选择其中一个，创建一组单选按钮的构造器是：

```
public Checkbox (String s,CheckboxGroup c,boolean state)
```

参数s是单选按钮的标签名，该按钮属于按钮组c。在创建单选按钮之前，先按钮组CheckboxGroup的类对象C。state是初始时单选按钮的状态，其值为true或false 在一组单选按钮中仅允许有一个的初态为true，其余均为false。

1. 图形用户界面

1.5 复选框、单选按钮、列表框和下拉列表

单选按钮的事件响应及事件处理方式与复选按钮框相同。在事件处理— `itemStateChanged()` 中，利用 `Itemevent` 的方法 `getSource()`，返回所选择的单选按钮对象名，把选择结果保存在一个全局性的类成员变量中。经过执行 `repaint()` 之后，`paint()` 方法根据这个全局变量的值，执行所选择单选按钮的要求功能。

1. 图形用户界面

1.5 复选框、单选按钮、列表框和下拉列表

列表 (List)显示一系列的选项，用户可以从中选择一项或多项。

List类继承于Component类，用于创建List的类对象。

List的构造函数是：

```
public List (int items , boolean ms)
```

其中，items表示一次显示几个选项，如果选项多于items的值，显示窗口会出现垂直的滚动条，允许翻页寻找。逻辑值ms为false时，表示这个List为单项选择，为true时，允许多项选择。

1. 图形用户界面

1.5 复选框、单选按钮、列表框和下拉列表

下拉列表框可以为用户提供多个选项，单击下拉列表框右侧的按钮可以打开或关闭选择列表。

下拉列表框的建立。java.awt包中的Choice类是用来建立下拉列表框的，即Choice类创建的一个对象就是一个下拉列表框。

1. 图形用户界面

1.5 复选框、单选按钮、列表框和下拉列表

Choice类的常用方法：

- ①Choice()构造方法。构造下拉列表框。
- ②add(String s)。向下拉列表中增加一个选项。
- ③getSelectedIndex()。返回当前选项的索引。
- ④getSelectedItem()。返回当前选项的字符串。
- ⑤insert(String s , int n)。将字符串插入下拉列表中的指定位置。
- ⑥remove(int n)。从下拉列表中删除指定的选项。
- ⑦removeAll()。删除全部选项。
- ⑧addItemListener(ItemListener l)。向下拉列表框增加监视器。

1. 图形用户界面

1.6 菜单

菜单的创建需要菜单条、菜单和菜单项。菜单项添加到菜单里，菜单添加到菜单条里，菜单条通过Frame的setMenuBar(MenuBar mb)方法添加到窗口里。

菜单项：java.awt.MenuItem负责创建菜单项，菜单项放在菜单里。其常用方法...

菜单：java.awt.Menu创建菜单对象，其主要方法...

菜单条：java.awt.MenuBar负责创建菜单条对象。

1. 图形用户界面

1.7 对话框

在java.awt.Window中窗口一共分两种：

框架（Frame）和对话框（Dialog），两者都直接继承Window类，对话框是一种简单窗口，任务栏中只有一个关闭按钮，常见于对窗口错误操作时跳出来的错误提示，就是用对话框来实现的。

Dialog 的默认布局为 BorderLayout。

1. 图形用户界面

1.8 容器

在Swing中，我们有四种可以使用的顶层容器：

JFrame:用来设计类似于Windows系统中的窗口形式的应用程序。

JDialog:和JFrame类似，只不过JDialog是用来设计对话框。

JApplet:用来设计可以在嵌入在网页中的Java小程序。

JWindow可以显示在用户桌面上的任何位置。它没有标题栏、窗口管理按钮或者其他与JFrame关联的修饰，但它仍然是用户桌面的“一类居民”，可以存在于桌面上的任何位置。

1. 图形用户界面

1.9 布局设计

布局管理是决定容器中组件的大小和位置的过程。布局管理器 (Layout Manager) 负责管理容器中组件的布局。它指明了容器中构件的位置和尺寸大小。当创建一个容器时, Java自动地为它创建并分配一个默认的布局管理器, 它确定容器中控件的布置。可以根据需要在应用中为不同的容器创建不同的布局管理器, 从而达到客户所需要的页面效果。

1. 图形用户界面

1.9 布局设计

在java.awt包中定义了五种布局类: FlowLayout、BorderLayout、CardLayout、GridLayout、GridBagLayout。

在Sun公司新版的SL-275教材上,删除了有关CardLayout和GridBagLayout的内容。

1. 图形用户界面

1.10 流式布局

FlowLayout类是流布局（顺序布局）管理器类，流布局管理器可以自动依据窗口的大小，将组件按照由左到右、由上到下的顺序排列

构造方法 :FlowLayout () :

构造一个新的FlowLayout，居中对齐，默认的水平垂直间隙是5个单位。

1. 图形用户界面

1.10 流式布局

FlowLayout(int align) : 构造一个新的 FlowLayout , 对齐方式是指定的 , 默认的水平和垂直间隙是 5 个单位。其中align参数可以为 FlowLayout.LEFT (按左对齐)、FlowLayout.RIGHT (按右对齐)、FlowLayout.CENTER (中心对齐) 。

FlowLayout(int align, int hgap, int vgap) : 创建一个新的流布局管理器 , 具有指定的对齐方式以及指定的水平和垂直间隙。与上面第二个构造不同的是多了两个参数 , 其中hgap、vgap参数分别以像素为单位设置组件之间水平和垂直间隔。

1. 图形用户界面

1.11 边框布局

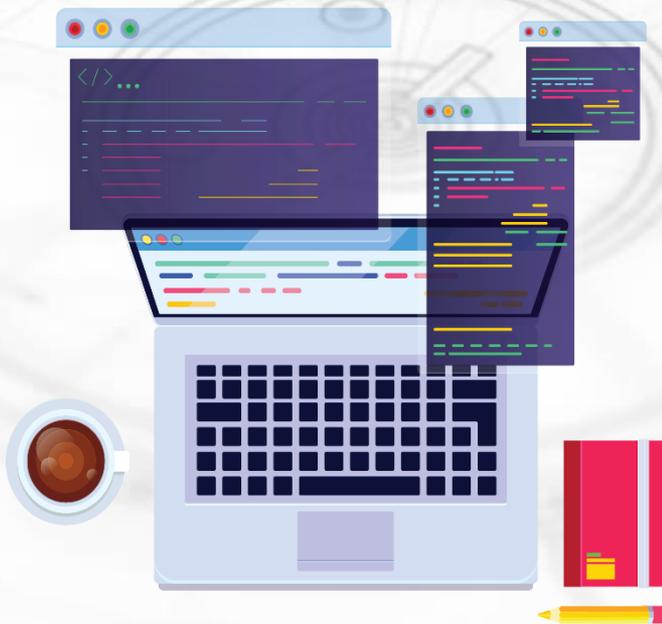
BorderLayout (边界布局管理器) , 把容器内的空间简单地划分为东、南、西、北、中 5 个区域 , 然后按各区域放置组件。

当加入组件时 , 使用 **add()** 方法。形式如下 : **add (Component comp , Object regin)** 。

1. 图形用户界面

1.12 网络布局

网格布局管理器用于将容器区域划分为一个矩形网格（区域），其组件按行和列排列，每个组件占一格。



1. 图形用户界面

1.12 网络布局

GridLayout的构造方法包括：

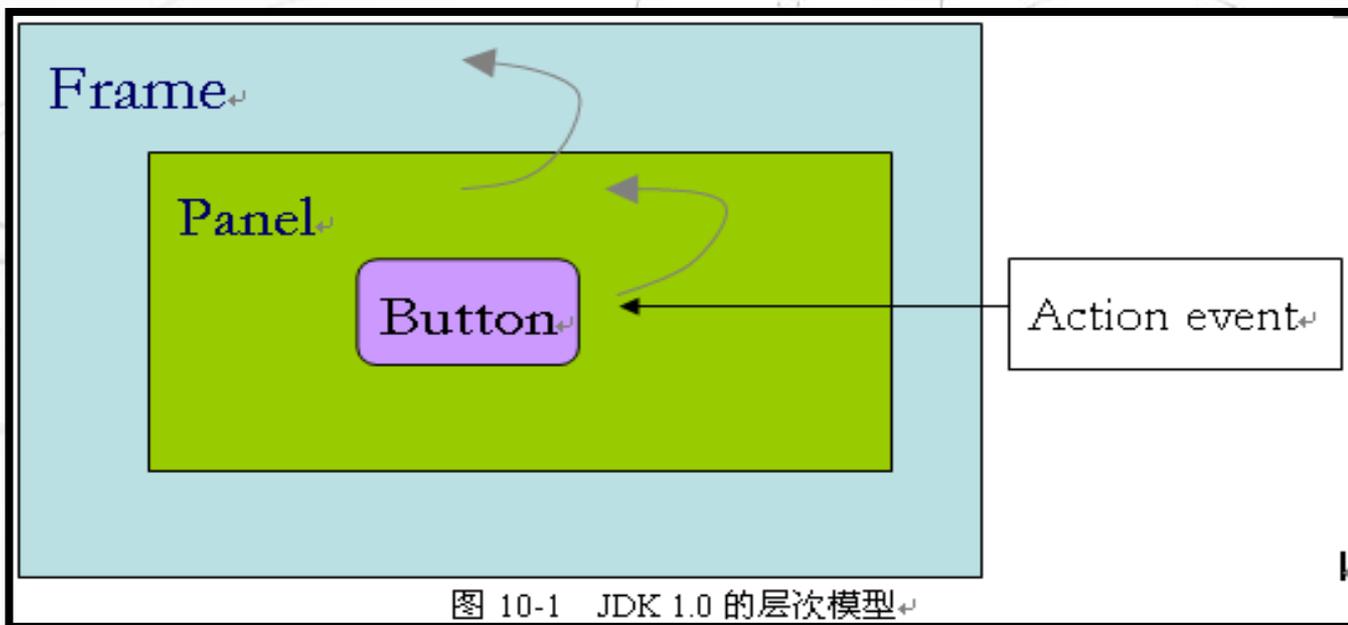
GridLayout() 创建具有默认值的网格布局，即每个组件占据一行一列。

GridLayout(int rows, int cols) 创建具有指定行数和列数的网格布局。

GridLayout(int rows, int cols, int hgap, int vgap) 创建具有指定行数和列数的网格布局。分别设定组件之间的水平间距和垂直间距。创建网格布局管理器时，可以在程序中给出网格大小，即行数和列数。

1. 图形用户界面

1.13 事件处理

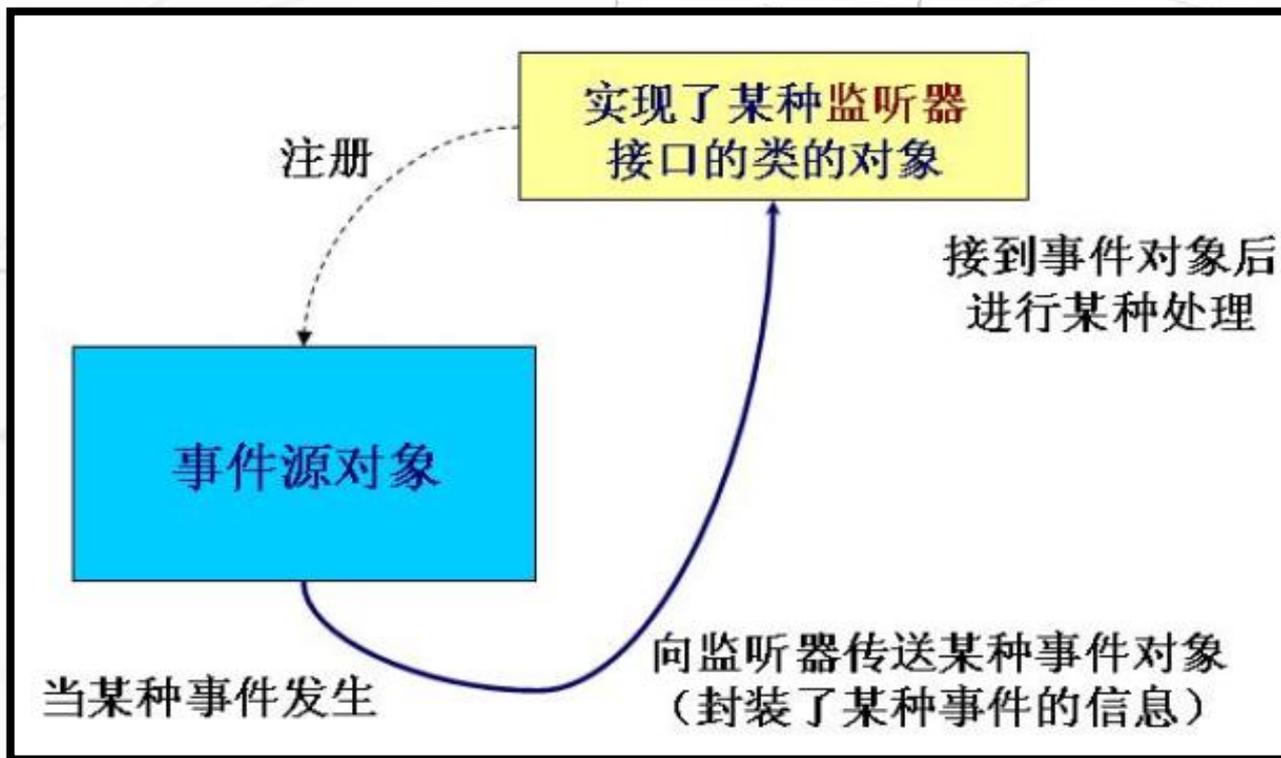


层次型模型的优点：

简单，对中小规模的应用比较合适。

1. 图形用户界面

事件处理的委托模型



1. 图形用户界面

事件处理的委托模型

在这种模式中，事件的产生者和事件的处理者分离开来了，它们可以是不同的对象。事件的处理者，即那些listeners，是一些实施了Listener接口的类。当事件传到登记的listener时，该listener中必须有相应的方法来接受这种类型的事件并对它进行处理。一个组件如没有登记的listeners，则它产生的事件就不会被传递。

1. 图形用户界面

1.14 事件与监听者

打个比方：

一个小偷在郑州火车站偷人钱包被安装在广场某处的监控器拍摄到了，结果被公安民警及时发现，抓捕小偷追回赃款。在这个例子里有我们要讲的事件源：郑州火车站广场某角落；监听器：监控摄像头；事件处理对象：民警抓小偷。

我们把监听器安装在了郑州火车站广场某角落，在这个角落里发生了盗窃事件，由监听器把该事件的详细情况传给了事件处理对象——公安民警，民警同志做的处理就是逮捕小偷归案。

1. 图形用户界面

1.15 常用事件处理

事件的分类：

语义 (Semantic) 事件

低级 (Low-Level) 事件

监听器接口概况：

`java.awt.event.ActionListener`用于接收操作事件的侦听器接口。

`java.awt.event.AdjustmentListener`用于接收调整事件的侦听器接口。

1. 图形用户界面

监听器接口：

java.awt.event. ActionListener该侦听器接口用于接收指派给对象的事件的通知。

java.awt.event. ComponentListener用于接收组件事件的侦听器接口。

java.awt.event. ContainerListener用于接收容器事件的侦听器接口。

1. 图形用户界面

监听器接口：

java.awt.event. FocusListener用于接收组件上的键盘焦点事件的侦听器接口。

java.awt.event. HierarchyBoundsListener用于接收祖先的移动和大小调整事件的侦听器接口。

java.awt.event. MouseMotionListener用于接收组件上的鼠标移动事件的侦听器接口。

1. 图形用户界面

监听器接口：

java.awt.event. MouseWheelListener用于接收组件上的鼠标滚轮事件的侦听器接口。

java.awt.event. TextListener用于接收文本事件的侦听器接口。

java.awt.event. WindowFocusListener用于接收 WindowEvent (包括 WINDOW_GAINED_FOCUS 和 WINDOW_LOST_FOCUS 事件) 的侦听器接口。

java.awt.event. WindowListener用于接收窗口事件的侦听器接口。

java.awt.event. WindowStateListener用于接收窗口状态事件的侦听器接口。

1. 图形用户界面

1.16 综合实例

我们运用我们学习过的GUI知识做一个简易的计算器，重点查考一下，组件、事件等相关知识。



1. 图形用户界面

1.17 颜色与字体设置

颜色常量	颜色	RGB 值
Color.black	黑色	0.0.0
Color.blue	蓝色	0.0.255
Color.green	绿色	0.255.0
Color.cyan	青色	0.255.255
Color.darkGray	深灰色	64.64.64
Color.gray	灰色	128.128.128
Color.lightGray	浅灰色	192.192.192
Color.red	红色	255.0.0
Color.magenta	深红色	255.0.255
Color.pink	粉红色	255.75.175
Color.orange	橘黄色	255.00.0
Color.yellow	黄色	255.255.0
Color.white	白色	255.255.255

1. 图形用户界面

1.18 颜色与字体设置

使用Color类的构造函数可以创建自己喜欢的颜色对象。

```
public Color ( int r , int g , int b );
```

```
public Color (float r1 , float g1 , float b1 );
```

整数r、g、b分别表示红、绿、蓝的含量，取值为0~255。浮点数

r1、g1、b1的取值范围是0.0~1.0。

以下的三个方法返回当前颜色的三个分量值:

```
public int getRed();
```

```
public int getGreen();
```

```
public int getBlue();
```

1. 图形用户界面

字体类：

Java提供的Font类，可以决定所要显示文字的字体、大小和位置，使输出的言窒于变化，更具特色，也更实用。

言先创建Font类的一个对象，构造函数是：

`Font (String name , int style , int size)`

字型名称name有Courier、Helvetica、TimesRoman或是宋、楷体等。

1. 图形用户界面

字体类：

字体风格style指字的外观，有三个字体风格的静态常量：

Font.PLAIN (正常字体)、 Font.BOLD (黑体)、 Font.ITALIC (斜体)。

字体风格可以组合使用，如：

Font.BOLD + Font.ITALIC

字体大小size以点来衡量，一个点(point)是1/2英寸。

然后使用Graphics类的方法void setFont (Font font) ，设置字体。

1. 图形用户界面

字体类：

经常用到Font类的以下几个方法

`int getStyle()`返回当前字体风格的整数值。

`int getSize()`返回当前字体大小的整数值。

`String getName()`返回当前字体名称的字符串。

`String getFamily()`返回当前字体家族名称的字符串。

`boolean isPlain()`测试当前字体是否是正常字体。

`boolean isBold()`测试当前字体是否是黑体。

`boolean isItalic()`测试当前字体是否是斜体。

1. 图形用户界面

字体类：

经常用到Font类的以下几个方法

`int getStyle()`返回当前字体风格的整数值。

`int getSize()`返回当前字体大小的整数值。

`String getName()`返回当前字体名称的字符串。

`String getFamily()`返回当前字体家族名称的字符串。

`boolean isPlain()`测试当前字体是否是正常字体。

`boolean isBold()`测试当前字体是否是黑体。

`boolean isItalic()`测试当前字体是否是斜体。

```
h3 {font-size: 20px !important;}
h4 {font-size: 16px; text-align: left;}
hr {margin: 3px !important; padding: 0px !important; padding-top: 5px !important; border-top: 1px solid #ccc !important;}
```

```
#container {margin: auto; width: 850px; padding-top: 90px;}
#info_bar_line1 {font-weight: bold; font-size: 20px; margin: 0; padding: 0; text-align: left;}
#info_bar_line2 {font-size: 14px; margin: 0; text-align: left;}
.info_bar {width: 100%; background-color: #4288c4; position: fixed; padding: 10px 20px; z-index: 10;}
.info_bar p {color: #ffffff !important;}
```

```
.hide {display: none;}
.field_information {cursor: pointer; float: left; margin: 1px 0 0 5px;}
.field_information_container {float: left;}
.label {font-size: 12px !important;}
.btn_copy_text {width: 110px;}
.btn_get_first {width: 110px;}
```

```
.title {width: 70px !important;}
.description {width: 70px !important; height: 75px !important;}
```

```
.tag_editor {line-height: 25px !important; height: 225px; padding: 5px 0px !important; border: 1px solid #ccc !important; border-radius: 4px;}
.tag_editor_delete {height: 25px !important;}
.tag_editor_delete i {line-height: 25px !important;}
.tag_editor_spacer {width: 10px !important;}
```

```
@btn_settings { webkit-user-select: none; -ms-user-select: none; user-select: none; -ms-user-select: none; user-select: none; transition: all 0.5s ease-out 0s;}
```

```
#select_theme_container {width: 280px;}
#google_api_key {width: 400px;}
#get_first_value {width: 50px;}
.simple_text {text-decoration: none !important;}
.pamel_settings {padding: 10px !important;}
.pamel_settings_container {margin-bottom: 5px !important;}
```

```
#google_translate_api_info {font-size: 10px; margin-left: 35px;}
.checkbox_comment {font-size: 10px;}
.btn_default .badge {margin-left: 3px; border-radius: 5px !important;}
na*k {padding: 0 !important;}
```

```
#add_and_translate {font-size: 10px;}
```

```
.tooltipster-box {background: #fff !important;}
.tooltipster-arrow-background {border-top-color: #fff !important;}
.tooltipster-box {-webkit-box-shadow: 0 1px 4px rgba(0,0,0,.2); box-shadow: 0 1px 4px rgba(0,0,0,.2)}
```



4

本章小结

本章小结

本章详细介绍Java中常用的AWT和Swing组件，GUI的相关概念和使用方法，另外介绍了三种布局管理器，介绍了GUI中常用的类库以及常用类。



THANK YOU



中国科技出版传媒股份有限公司
China Science Publishing & Media Ltd. (CSPM)
科学出版社