

本文链接: <http://blog.csdn.net/xietansheng/article/details/72814531>

Java Swing GUI 图形界面窗口开发基础教程, 本教程将系统性地详细介绍 Java Swing 开发中常用的一些组件、布局管理器等相关知识技术, 并且每章节都将通过代码实例展示实际应用。

教程总目录: [Java Swing 图形界面开发 \(目录\)](#)

## 1. Swing简介

Swing 是 Java 为图形界面应用开发提供的一组工具包, 是 Java 基础类的一部分。

Swing 包含了构建图形界面 (GUI) 的各种组件, 如: 窗口、标签、按钮、文本框等。

Swing 提供了许多比 AWT 更好的屏幕显示元素, 使用纯 Java 实现, 能够更好的兼容跨平台运行。

为了和 AWT 组件区分, Swing 组件在 `javax.swing.*` 包下, 类名均以 J 开头, 例如: JFrame、JLabel、JButton 等。

## 2. Swing组件

一个 Java 的图形界面, 由各种不同类型的“元素”组成, 例如: 窗口、菜单栏、对话框、标签、按钮、文本框等等, 这些“元素”统一被称为 **组件** (Component)。

组件按照不同的功能, 可分为 **顶层容器**、**中间容器**、**基本组件**。一个简单窗口的组成, 如下层级结构所示:

- 顶层容器
  - 菜单栏
  - 中间容器
    - 基本组件
    - 基本组件

组件类型的继承关系:

- 顶层容器 属于窗口类组件, 继承自 `java.awt.Window`;
- 中间容器 和 基本组件 继承自 `javax.swing.JComponent`。

### 2.1 顶层容器

顶层容器属于窗口类组件，可以独立显示，一个图形界面至少需要一个窗口，例如：

#	组件	描述
1	<a href="#">JFrame</a>	一个普通的窗口 (绝大多数 Swing 图形界面程序使用 JFrame 作为顶层容器)
2	<a href="#">JDialog</a>	对话框

## 2.2 中间容器

中间容器充当基本组件的载体，不可独立显示。中间容器可以添加若干基本组件（也可以嵌套添加中间容器），对容器内的组件进行管理，类似于给各种复杂的组件进行分组管理。最顶层的一个中间容器必须依托在顶层容器（窗口）内。

常用的中间容器（面板）：

#	组件	描述
1	<a href="#">JPanel</a>	一般轻量级面板容器组件
2	<a href="#">JScrollPane</a>	带滚动条的，可以水平和垂直滚动的面板组件
3	<a href="#">JSplitPane</a>	分隔面板
4	<a href="#">JTabbedPane</a>	选项卡面板
5	<a href="#">JLayeredPane</a>	层级面板

特殊的中间容器：

#	组件	描述
1	<a href="#">JMenuBar</a>	菜单栏
2	<a href="#">JToolBar</a>	工具栏
3	<a href="#">JPopupMenu</a>	弹出菜单

3	<a href="#">JMenu</a>	菜单
4	<a href="#">JInternalFrame</a>	内部窗口

## 2.3 基本组件

基本组件是直接实现人机交互的组件。

常用的简单的基本组件:

#	组件	描述
1	<a href="#">JLabel</a>	标签
2	<a href="#">JButton</a>	按钮
3	<a href="#">JRadioButton</a>	单选按钮
4	<a href="#">JCheckBox</a>	复选框
5	<a href="#">JToggleButton</a>	开关按钮
6	<a href="#">JTextField</a>	文本框
7	<a href="#">JPasswordField</a>	密码框
8	<a href="#">JTextArea</a>	文本区域
9	<a href="#">JComboBox</a>	下拉列表框
10	<a href="#">JList</a>	列表
11	<a href="#">JProgressBar</a>	进度条
12	<a href="#">JSlider</a>	滑块

选取器组件:

#	组件	描述
1	<a href="#">JFileChooser</a>	文件选取器
2	<a href="#">JColorChooser</a>	颜色选取器

其他较为复杂的基本组件:

#	组件	描述
1	<a href="#">JTable</a>	表格
2	<a href="#">JTree</a>	树

--	--	--

### 3. 布局管理器

把 **Swing** 的各种组件(**JComponent**)添加到面板容器中(**JPanel**), 需要给面板容器指定布局管理器(**LayoutManager**), 明确容器(**Container**)内的各个组件之间的排列布局方式。

常用的布局管理器:

#	布局管理器	描述
1	<a href="#">FlowLayout</a>	流式布局, 按组件加入的顺序, 按水平方向排列, 排满一行换下一行继续排列。
2	<a href="#">GridLayout</a>	网格布局, 把 Container 按指定行列数分隔出若干网格, 每一个网格按顺序放置一个控件。
3	<a href="#">GridBagLayout</a>	网格袋布局, 按网格划分 Container, 每个组件可占用一个或多个网格, 可将组件垂直、水平或沿它们的基线对齐。
4	<a href="#">BoxLayout</a>	箱式布局, 将 Container 中的多个组件按 水平或垂直

		子级组件的方式排列。
5	<a href="#">GroupLayout</a>	分组布局，将组件按层次分组（串行或并行），分别确定组件组在水平和垂直方向上的位置。
6	<a href="#">CardLayout</a>	卡片布局，将 Container 中的每个组件看作一张卡片，一次只能显示一张卡片，默认显示第一张卡片。
7	<a href="#">BorderLayout</a>	边界布局，把 Container 按方位分为 5 个区域（东、西、南、北、中），每个区域放置一个组件。
8	<a href="#">SpringLayout</a>	弹性布局，通过定义组件四条边的坐标位置来实现布局。
9	<a href="#">null</a>	绝对布局，通过设置组件在 Container 中的坐标位置来放

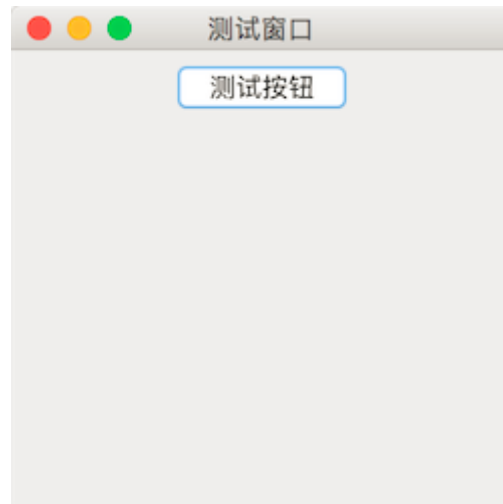
## 4. 代码实例: 一个简单的窗口程序

```
package com.xiets.swing;import
javax.swing.*;publicclassMain{publicstaticvoidmain(String[] args){// 1. 创建一个顶层
容器 (窗口)      JFrame jf =newJFrame("测试窗口");// 创建窗口
jf.setSize(250,250);// 设置窗口大小      jf.setLocationRelativeTo(null);// 把窗口位置设
置到屏幕中心      jf.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);//
当点击窗口的关闭按钮时退出程序 (没有这一句, 程序不会退出) // 2. 创建中间容器 (面
板容器)      JPanel panel =newJPanel();// 创建面板容器, 使用默认的布局管理器// 3.
创建一个基本组件 (按钮), 并添加到 面板容器 中      JButton btn =newJButton("测试
按钮");      panel.add(btn);// 4. 把 面板容器 作为窗口的内容面板 设置到 窗口
jf.setContentPane(panel);// 5. 显示窗口, 前面创建的信息都在内存中, 通过
jf.setVisible(true) 把内存中的窗口显示在屏幕上。      jf.setVisible(true);}}
```

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 10
- 11
- 12
- 13
- 14
- 15
- 16
- 17
- 18
- 19
- 20
- 21

- 22
- 23
- 24
- 25
- 26
- 27
- 28

结果展示:



result.png