

JAVA教程第七讲Swing用户界面设计7.1 PDF转换可能丢失图片或格式，建议阅读原文

[https://www.100test.com/kao\\_ti2020/138/2021\\_2022\\_JAVA\\_E6\\_95\\_99\\_E7\\_A8\\_8B\\_c97\\_138733.htm](https://www.100test.com/kao_ti2020/138/2021_2022_JAVA_E6_95_99_E7_A8_8B_c97_138733.htm) 7.1 Swing简介7.1.1 简介 第五

讲中我们学习了AWT，AWT是Swing的基础。Swing的产生主要原因就是AWT不能满足图形化用户界面发展的需要。AWT设计的初衷是支持开发小应用程序的简单用户界面。例如AWT缺少剪贴板、打印支持、键盘导航等特性，而且原来的AWT甚至不包括弹出式菜单或滚动窗格等基本元素。此外AWT还存在着严重的缺陷，人们使AWT适应基于继承的、具有很大伸缩性的事件模型，基于同位体的体系结构也成为其致命的弱点。随着发展的需要，Swing出现了，Swing组件几乎都是轻量组件，与重量组件相比，没有本地的对等组件，不像重量组件要在它们自己的本地不透明窗体中绘制，轻量组件在它们的重量组件的窗口中绘制。这一讲我们讲一下基本的Swing组件使用方法和使用Swing组件创建用户界面的初步方法。Swing是由100%纯Java实现的，Swing组件是用Java实现的轻量级（light-weight）组件，没有本地代码，不依赖操作系统的支持，这是它与AWT组件的最大区别。由于AWT组件通过与具体平台相关的对等类（Peer）实现，因此Swing比AWT组件具有更强的实用性。Swing在不同的平台上表现一致，并且有能力提供本地窗口系统不支持的其它特性。

#### 7.1.4 MVC(Model-View-Control)体系结构

Swing胜过AWT的主要优势在于MVC体系结构的普遍使用。在一个MVC用户界面中，存三个通讯对象：模型、视图和控件。模型是指定的逻辑表示法，视图是模型的可视化表示法，而控件则指定了如何处

理用户输入。当模型发生改变时，它会通知所有依赖它的视图，视图使用控件指定其相应机制。为了简化组件的设计工作，在Swing组件中视图和控件两部分合为一体。每个组件有一个相关的分离模型和它使用的界面（包括视图和控件）。比如，按钮JButton有一个存储其状态的分离模型ButtonModel对象。组件的模型是自动设置的，例如一般都使用JButton而不是使用ButtonModel对象。另外，通过Model类的子类或通过实现适当的接口，可以为组件建立自己的模型。把数据模型与组件联系起来用setModel()方法。MVC是现有的编程语言中制作图形用户界面的一种通用的思想，其思路是把数据的内容本身和显示方式分离开，这样就使得数据的显示更加灵活多样。比如，某年级各个班级的学生人数是数据，则显示方式是多种多样的，可以采用柱状图显示，也可以采用饼图显示，也可以采用直接的数据输出。因此在设计的时候，就考虑把数据和显示方式分开，对于实现多种多样的显示是非常有帮助的。

#### 7.1.5 可存取性支持

所有Swing组件都实现了Accessible接口，提供对可存取性的支持，使得辅助功能如屏幕阅读器能够十分方便的从Swing组件中得到信息。

#### 7.1.6 支持键盘操作

在Swing组件中，使用JComponent类的registerKeyboardAction()方法，能使用户通过键盘操作来替代鼠标驱动GUI上Swing组件的相应动作。有些类还为键盘操作提供了更便利的方法。其实这就相当于热键，使得用户可以只用键盘进行操作。

#### 7.1.7 设置边框

对Swing组件可以设置一个和多个边框。Swing中提供了各式各样的边框供用户选用，也能建立组合边框或自己设计边框。一种空白边框可以增大组件，协助布局管理器对容器中的组件进行合理的布局。

100Test 下载频道开通，各类考试题目直接下载。详细请访问  
[www.100test.com](http://www.100test.com)