

Swing学习笔记

实际使用 Java 开发图形界面程序时，很少使用 AWT 组件，绝大部分时候都是用 Swing 组件开发的。Swing是由100%纯 Java实现的，不再依赖于本地平台的 GUI，因此可以在所有平台上都保持相同的界面外观。独立于本地平台的Swing组件被称为*轻量级组件*；而依赖于本地平台的 AWT 组件被称为*重量级组件*。

Swing的特征：

\1. Swing 组件采用 MVC(Model-View-Controller，即模型一视图一控制器)设计模式：

- 模型(Model): 用于维护组件的各种状态
- 视图(View): 是组件的可视化表现；
- 控制器(Controller):用于控制对于各种事件、组件做出响应。

当模型发生改变时，它会通知所有依赖它的视图，视图会根据模型数据来更新自己。

Swing使用UI代理来包装视图和控制器，还有一个模型对象来维护该组件的状态。例如，按钮JButton有一个维护其状态信息的模型ButtonModel对象。Swing组件的模型是自动设置的，因此一般都使用JButton，而无须关心ButtonModel对象。

\2. Swing在不同的平台上表现一致，并且有能力提供本地平台不支持的显示外观。由于Swing采用 MVC 模式来维护各组件，所以当组件的外观被改变时，对组件的状态信息(由模型维护)没有任何影响。因此，Swing可以使用插拔式外观感觉 (Pluggable Look And Feel, PLAF)来控制组件外观，使得 Swing图形界面在同一个平台上运行时能拥有不同的外观，用户可以选择自己喜欢的外观。相比之下，在 AWT 图形界面中，由于控制组件外观的对等类与具体平台相关，因此 AWT 组件总是具有与本地平台相同的外观。

