

全面剖析Java6中新型模态对话框API PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/253/2021_2022__E5_85_A8_E9_9D_A2_E5_89_96_E6_c104_253557.htm

一、引言 对话框，是指一个最顶层的拥有标题和边框的弹出窗口，典型地应用于用户进行某种形式的输入操作。在JDK 5.0和早期版本中，构建一个对话框时，它必须拥有一个作为它的所有者窗口的框架窗口或另一个对话框，即使窗口是不可见的。当用户最小化一个可见对话框的所有者窗口时会自动隐藏该对话框；而当用户随后恢复所有者窗口时，该对话框再次出现。一个对话框可以是无模式的也可以是模式的。除了该对话框的所有者的窗口之外，模式对话框将阻止在应用程序中其它顶层窗口的输入。模式对话框捕获窗口焦点直到其被关闭为止（经常响应于一次按钮点击）。另一方面，一个无模式对话框允许用户改变它的状态，而此时其它窗口仍然可拥有焦点。后者常用于工具栏窗口中，例如你在一个图像编辑程序中所见的。在JDK 5.0和早期版本中的模态模型有一些局限。而且，这种模态模型也存在一些问题。最著名的问题涉及到JavaHelp工具窗口。JavaHelp，Java应用程序中提供帮助信息的API，使用独立的窗口来显示所有的必要信息。然而，如果应用程序显示任何模式对话框，例如一标准Save As对话框，那么该对话框将阻止用户与JavaHelp工具窗口进行交互。但是Java 6，代码名称为Mustang，已经通过一种新的抽象窗口工具箱(AWT)模态模型解决了此问题以及其它几个问题。这种新型模型允许开发者根据其选择的模态类型指定范围或限制一个对话框的模态阻断。如此模态类型也允许窗口和对

对话框成为真正的无父窗口，也即，拥有一个null父窗口，它可以帮助限制窗口的范围和对话框的模态。

二、模态类型

Java 6支持四种模态类型：

- 无模式。无模式对话框在自己为可见时并不阻断任何其它窗口。
- 文档-模式。文档-模式对话框阻断所有的来自同一文档的窗口，除了那些来自于它的子层次上的窗口外。在此意义中，一个文档是指一个窗口层次-框架窗口，对话框等等，它们共享一个文档根窗口。文档根窗口是所有没有所有者的最顶级窗口。
- 应用程序-模式。应用程序-模式对话框能够阻断同一应用程序中的所有窗口，除了那些来自于它的子层次上的窗口外。如果在浏览器中激活若干applet，那么浏览器将把它们当作独立的应用程序或者作为一个单一的应用程序。具体情况依赖实现的具体环境。
- 工具箱-模式。工具箱-模式对话框能够阻断所有的运行于同样的工具箱中的窗口，除了那些来自于它的子层次上的窗口外。如果激活多个applet，那么它们都运行于同样的工具箱上。因此，从applet中显示的工具箱-模式对话框可能影响其它的applet。正如以前的JDK一样，一个对话框在缺省情况下是无模式的。但是如果你在Mustang中构造一个模式对话框的话，现在它将缺省使用应用程序-模式类型。另外，模式和无模式对话框的行为已经在Mustang作了改变，它们可以一直出现在其父窗口的顶部。

模态优先权是由阻断强度决定的。这种模态优先权帮助处理两个对话框可见并且能够彼此阻断的情形。优先权按升序排列分别是：无模式，文档-模式，应用程序-模式和工具箱-模式。这种优先权自然地反映了一个对话框的范围阻断的嵌套情形。一个无模式对话框有一个空范围的阻断。文档-模式对话框的范围阻断是在特定的应用程序中

完成的，并且所有的应用程序是运行于一种工具箱中。图1展示了这样的例子。注意，这种新型的模态模型并没有实现一个系统模态，这将会阻断(Java或其它)所有被显示在桌面上的应用程序，而只有一个模式对话框处于活动状态。三、了解新的构造器能提供真正无父窗口而不中断向后兼容对于AWT开发小组来说是一个挑战。在JDK 5.0及以前的版本中，允许传递null作为JDialog或JWindow的父窗口。这意味着，一不可见的共享所有者的框架窗口将成为这个对话框或窗口的父窗口。这个共享所有者的框架窗口的创建是为了创建无父窗口对话框。这种情况直到Java SE 6才得以成功-Java SE 6引入了新型的文档-模式对话框能够阻止所有的窗口使用相同的文档。因而，工具箱会知道这样的对话框或窗口是没有null父窗口的。在Mustang中，仍然允许传递null作为一个父窗口到旧式的JDialog或JWindow构造器中。并且这可以完成相同的事情：其成为共享的所有者框架窗口而不是成为父窗口，这也照顾了向后兼容问题。然而，现在能把null传递到Dialog或Window构造器中，而且也能传递到新的JDialog或JWindow构造器中，这意味着这些对话框能够真正成为无父窗口。下面列举的是一些比较有用的构造器：JDialog(Dialog owner) 创建一个无标题并用一个指定的对话框作为其父窗口的无模式对话框 JDialog(Dialog owner, boolean modal) 用指定的所有者Dialog和模态创建一个具有指定的模态和所有者对话框的对话框 JDialog(Dialog owner, String title) 创建一个具有指定的标题和所有者对话框的无模式对话框 JDialog(Dialog owner, String title, boolean modal) 创建一个具有指定的标题，模态和所有者对话框的对话框 JDialog(Dialog owner, String title

, boolean modal , GraphicsConfiguration gc) 创建一个具有指定的标题 , 模态和所有者对话框和GraphicsConfiguration的对话框 JDialog(Frame owner) 创建一个无标题但用指定的框架作为其所有者的无模式对话框 JDialog(Window owner , String title , Dialog.ModalityType modalityType) 创建一个具有指定的标题 , 模态和所有者窗口的对话框 有关这里每个构造器的更多细节请参考Mustang有关文档。 四、 与所有者窗口一同工作 如前面所提及的 , 当一个窗口或一个对话框是另一个窗口的父亲时 , 就称为该父组件"拥有"它的孩子。在此 , 当在新的模态模式下与所有者窗口一同工作时 , 你应该注意几件事情 : 创建一个没有所有者的文档-模式对话框。在这种情况下 , 因为Dialog是Window的一个子类 ; 所以 , 如果一个Dialog实例没有所有者的话 , 它自动地成为该文档的根。这样 , 如果这个对话框是文档-模式的 , 那么它的阻断范围是空的 , 并且其行为就象一个无模式对话框。 创建一个有所有者的应用程序-模式或工具箱-模式对话框。 一个应用程序-模式或文档-模式对话框的阻断范围并不依靠它的所有者。在这种情况下 , 所有者所唯一影响的是Z-顺序 (顶级组件的相对顺序) 。如果你有两个窗口 , 一个遮住另一个 , 第一个窗口位于第二窗口上面 , 那么最顶端的窗口通常是一个活动的窗口。 一个相关概念是"总是位于顶层" , 这时一个窗口总是出现在系统中所有其它窗口之上。对话框总是会位于它的所有者的上部。

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