

介绍Swing线程相关处理Java认证考试 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/644/2021_2022__E4_BB_8B_E7_BB_8DSwin_c104_644449.htm 我们知道这将在非Swing线程中调用，因为该事件是直接在LookupManager中触发的，这将在Swing线程中执行。因为所有的代码功能上都是异步的（我们不必等待监听器方法允许结束后才调用其它代码），我们可以通过SwingUtilities.invokeLater（）将这些代码改道到Swing线程。下面是新的方法，传入一个匿名Runnable到SwingUtilities.invokeLater（）：

```
public void  
lookupCompleted(final LookupEvent e) { //notice the threading  
SwingUtilities.invokeLater( new Runnable() { public void run() {  
outputTA.setText(""). String[] results = e.getResults(). for (int i = 0. i  
&lt;. results.length. i ) { String result = results[i].  
outputTA.setText(outputTA.getText() "" result). } } } ). } 如果任  
何LookupListener不是在Swing线程中执行，我们可以在调用线  
程中执行监听器代码。作为一个原则，我们希望所有的监听  
器都迅速地接到通知。所以，如果你有一个监听器需要很多  
时间来处理自己的功能，你应该创建一个新的线程或者把耗  
时代码放入ThreadPool中等待执行。最后的步骤是  
让LookupManager在非Swing线程中执行lookup.当前  
，LookupManager是在JButton的 ActionListener的Swing线程中  
被调用的。现在是我们做出决定的时候，或者我们在JButton  
的ActionListener中引入一个新的线程，或者我们可以保  
证lookup自己在非Swing线程中执行，自己开始一个新的线程  
。我选择尽可能和Swing类贴近地管理Swing线程。这有助于
```

把所有Swing逻辑封装在一起。如果我们把Swing线程逻辑添加到LookupManager，我们将引入了一层不必要的依赖。并且，对于LookupManager在非Swing线程环境中孵化自己的线程是完全没有必要的，比如一个非绘图的用户界面，在我们的例子中，就是Logger.产生不必要的新线程将损害到你应用的性能，而不是提高性能。LookupManager执行的很好，不管Swing线程与否所以，我喜欢把代码集中在那儿。现在我们需要将JButton的ActionListener执行lookup的代码放在一个非Swing线程中。我们创建一个匿名的Thread，使用一个匿名的Runnable执行这个lookup.

```
private void  
searchButton_actionPerformed() { new Thread(){ public void run()  
{ lookupManager.lookup(searchTF.getText()). }.start(). } 这就完  
成了我们的Swing线程。简单地在actionPerformed () 方法中  
添加线程，确保监听器在新的线程中执行照顾到了整个线程  
问题。注意，我们不用处理像第一个例子那样的任何问题。  
通过把时间花费在定义一个事件驱动的体系，我们在和Swing  
线程相关处理上节约了更多的时间。 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
```