

JAVA基础：浅谈Java与C#的事件处理机制 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/145/2021_2022_JAVA_E5_9F_BA_E7_A1_80_c104_145434.htm 以上两个方法的前面均冠

以synchronized，是因为运行在多线程环境时，可能同时有几个对象同时要进行注册和注销操作，使用synchronized来确保它们之间的同步。开发工具或程序员使用这两个方法建立源与监听者之间的事件流。

```
protected void notifyModelChanged()  
{/*事件源使用本方法通知监听者发生了modelChanged事件*/  
Vector l. EventObject e = new EventObject(this). 首先要把监听者拷贝到l数组中，冻结EventListeners的状态以传递事件。这样来确保在事件传递到所有监听者之前，已接收了事件的目标监听者的对应方法暂不生效。  
synchronized(this) { l = (Vector)listeners.clone(). }  
for (int i = 0. i /* 依次通知注册在监听者队列中的每个监听者发生了modelChanged事件，并把事件状态对象e作为参数传递给监听者队列中的每个监听者*/  
((ModelChangeListener)l.elementAt(i)).modelChanged(e). }  
}在程序中可见事件源Model类显式地调用了接口中的modelChanged方法，实际是把事件状态对象e作为参数，传递给了监听者类中的modelChanged方法。
```

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