

二级Java考试辅导教程：5.2AWT事件处理模型 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__E4_BA_8C_E7_BA_A7Java_c97_138770.htm

5.2 AWT事件处理模型 上一节中的主要内容是如何放置各种组件，使图形界面更加丰富多彩，但是还不能响应用户的任何操作，要能够让图形界面接收用户的操作，就必须给各个组件加上事件处理机制。在事件处理的过程中，主要涉及三类对象：Event-事件，用户对界面操作在java语言上的描述，以类的形式出现，例如键盘操作对应的事件类是KeyEvent。Event Source-事件源，事件发生的场所，通常就是各个组件，例如按钮Button。Event handler-事件处理者，接收事件对象并对其进行处理的对象 例如，如果用户用鼠标单击了按钮对象button，则该按钮button就是事件源，而java运行时系统会生成ActionEvent类的对象actionE，该对象中描述了该单击事件发生时的一些信息，然后，事件处理者对象将接收由java运行时系统传递过来的事件对象actionE并进行相应的处理。由于同一个事件源上可能发生多种事件，因此java采取了授权处理机制(Delegation Model)，事件源可以把在其自身所有可能发生的事件分别授权给不同的事件处理者来处理。比如在Canvas对象上既可能发生鼠标事件，也可能发生键盘事件，该Canvas对象就可以授权给事件处理者一来处理鼠标事件，同时授权给事件处理者二来处理键盘事件。有时也将事件处理者称为监听器，主要原因也在于监听器时刻监听着事件源上所有发生的事件类型，一旦该事件类型与自己所负责处理的事件类型一致，就马上进行处理。授权模型把事件的处理委托给外部的处理实

体进行处理，实现了将事件源和监听器分开的机制。事件处理者（监听器）通常是一个类，该类如果要能够处理某种类型的事件，就必须实现与该事件类型相对的接口。例如例5.9中类ButtonHandler之所以能够处理ActionEvent事件，原因在于它实现了与ActionEvent事件对应的接口ActionListener。每个事件类都有一个与之相对应的接口。来源

：www.examda.com 将事件源对象和事件处理器（事件监听器）分开。如图5.2所示 打个不太恰当的比喻，比如说有一位李先生，李先生可能会发生很多法律纠纷，可能是民事法律纠纷，也可能是刑事法律纠纷，那么李先生可以请律师，他可以授权王律师负责帮他打民事法律的官司，同时也可以授权张律师帮他打刑事法律的官司。这个请律师的过程从李先生的角度来看，就是授权的过程，而从王律师和张律师的角度来看，一旦被授权，他们就得时刻对李先生负责，“监听”着李先生，一旦发生民事纠纷了，王律师就要马上去处理，而一旦发生刑事纠纷了，张律师就要马上进行处理。此时此刻，李先生就是事件源，王律师是一个事件处理者，张律师是另外一个事件处理者，民事纠纷和刑事纠纷就是不同类型的事件。使用授权处理模型进行事件处理的一般方法归纳如下：
1.对于某种类型的事件XXXEvent, 要想接收并处理这类事件，必须定义相应的事件监听器类，该类需要实现与该事件相对应的接口XXXListener；
2.事件源实例化以后，必须进行授权，注册该类事件的监听器，使

用addXXXListener(XXXListener)方法来注册监听器。5.2.1 事件类 与AWT有关的所有事件类都由java.awt.AWTEvent类派生，它也是EventObject类的子类。AWT事件共有10类，可以归为

两大类：低级事件和高级事件。 `java.util.EventObject`类是所有事件对象的基础父类，所有事件都是由它派生出来的。AWT的相关事件继承于`java.awt.AWTEvent`类，这些AWT事件分为两大类：低级事件和高级事件，低级事件是指基于组件和容器的事件，当一个组件上发生事件，如：鼠标的进入，点击，拖放等，或组件的窗口开关等，触发了组件事件。高级事件是基于语义的事件，它可以不和特定的动作相关联，而依赖于触发此事件的类，如在`TextField`中按`Enter`键会触发`ActionEvent`事件,滑动滚动条会触发`AdjustmentEvent`事件，或是选中项目列表的某一条就会触发`ItemEvent`事件。低级事件

- `ComponentEvent`（组件事件：组件尺寸的变化，移动）
- `ContainerEvent`（容器事件：组件增加，移动）
- `WindowEvent`（窗口事件：关闭窗口，窗口闭合，图标化）
- `FocusEvent`（焦点事件：焦点的获得和丢失）
- `KeyEvent`（键盘事件：键按下、释放）
- `MouseEvent`（鼠标事件：鼠标单击，移动）

高级事件（语义事件）

- `ActionEvent`（动作事件：按钮按下，`TextField`中按`Enter`键）
- `AdjustmentEvent`（调节事件：在滚动条上移动滑块以调节数值）
- `ItemEvent`（项目事件：选择项目，不选择"项目改变"）
- `TextEvent`（文本事件，文本对象改变）

5.2.2 事件监听器

每类事件都有对应的事件监听器，监听器是接口，根据动作来定义方法。例如，与键盘事件`KeyEvent`相对应的接口是：

```
public interface KeyListener
extends EventListener {
    public void keyPressed(KeyEvent ev);
    public void keyReleased(KeyEvent ev);
    public void keyTyped(KeyEvent ev);
}
```

注意到在本接口中有三个方法，那么`java`运行时系统何时调用哪个方法？其实根据这三个方法的方法名就能够知道

应该是什么时候调用哪个方法执行了。当键盘刚按下去时，将调用keyPressed()方法执行，当键盘抬起来时，将调用keyReleased()方法执行，当键盘敲击一次时，将调用keyTyped()方法执行。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