

什么是事件？

一个对象的状态变化被称为事件，即事件描述源状态的变化。事件产生的结果与用户交互的图形用户界面组件。例如，点击一个按钮，移动鼠标，通过键盘输入一个字符，从列表中选择个项目，滚动页面的活动，使一个事件的发生。

事件的类型

事件可以被大致分为两类：

- **前台事件 - 这些事件需要用户直接互动。在图形用户界面中的图形组件交互的人产生的后果。例如，点击一个按钮，移动鼠标，通过键盘输入一个字符，从列表中选择个项目，滚动页面等**
- **后台事件 - 这些事件，需要最终用户的交互是已知的作为背景的事件。操作系统的中断，硬件或软件故障，定时器到期时，操作完成的背景事件的例子。**

事件处理是什么？

事件处理机制，控制的事件，并决定如果一个事件发生时，会发生什么。这种机制被称为事件处理程序，在事件发生时执行的代码。[Java](#)使用代理事件模型来处理事件。该模型定义了标准的机制来生成和处理事件。让我们简要介绍这种模式。

代理事件模型具有以下的主要参与者，即：

- **源 - 源是一个对象，在该对象上的事件发生。它的处理器提供发生事件的信息来源是可靠的。JAVA提供与源对象的类。**
- **监听器 - 它也被称为作为事件handler.Listener是负责产生响应一个事件。从Java实现的角度来看，监听器也是一个对象。等待直到它接收到一个事件监听器。一旦收到事件，监听器进程的事件就返回。**

这种方法的好处是，用户界面逻辑完全分开，生成该事件的逻辑。用户界面元素是能够委派的事件处理单独的一段代码。在这个模型中，需要与源对象注册监听使侦听器能够接收事件通知。这是一个有效的方式处理事件，因为这些事件通知只发送给那些监听器要接收他们。

参与事件处理的步骤

- 用户单击该按钮时产生该事件。
- 现在有关事件类的对象是自动创建的信息源和事件在同一对象得到填充。
- 事件对象被转发注册监听器类的方法。
- 该方法现在得到执行并返回。

要记住的要点有关监听器

- 为了设计一个监听类，我们必须开发一些监听器接口。这些监听器接口预测一些公共的抽象监听器类必须实现的回调方法。
- 如果不执行任何预定义的接口，类不能作为源对象的监听器类。

回调方法

这些方法所提供的API提供程序，被定义为应用程序员和应用程序开发者调用。这里的回调方法代表一个事件的方法。在响应一个事件的Java JRE将触发回调方法。所有这些回调方法的监听器接口。

如果一个组件需要一些监听器会听的事件源必须注册自己监听。

一、事件监听器

事件监听器代表负责处理事件的接口。Java 提供了各种事件监听器类，但我们将讨论那些被更频繁地使用。每一个事件监听器方法具有方法的EventObject类的子类的对象，这是作为一个单独的参数。例如，鼠标事件侦听器方法将接受MouseEvent的实例，其中派生的事件的EventObject。

EventListener 接口

它是一个标记接口，每一个监听器接口扩展。这个类定义在java.util包。

类声明

以下是声明java.util.EventListener接口：

```
public interface EventListener
```

Swing事件监听器接口：

以下是常用的的事件监听器列表。

Sr. No.	Control & Description
1	ActionListener 该接口用于接收动作事件。
2	ComponentListener 该接口用于接收组件事件。
3	ItemListener 该接口用于接收项目事件。
4	KeyListener 该接口用于接收按键事件。
5	MouseListener 该接口用于接收鼠标事件。
6	WindowListener 该接口用于接收窗口事件。
7	AdjustmentListener 该接口用于接收调整事件。
8	ContainerListener 该接口用于接收容器事件。
9	MouseMotionListener 此接口用于接收鼠标移动事件。
10	FocusListener 该接口用于接收焦点事件。

二、事件适配器

适配器是抽象类，用于接收各种事件。这些类中的方法是空的。这些类存在的目的是方便创建监听器对象。

SWING适配器：

以下是常用的适配器列表，监听SWING GUI事件。

Sr. No.	适配器&说明
1	FocusAdapter 接收焦点事件的抽象适配器类。
2	KeyAdapter 接收按键事件的抽象适配器类。
3	MouseAdapter 接收鼠标事件的抽象适配器类。

4	MouseMotionAdapter 接收鼠标移动事件的抽象适配器类。
5	WindowAdapter 接收窗口事件的抽象适配器类。