事件处理模式

1.事件

　　事件是用户在界面上的一个操作（通常使用各种输入设备，如：鼠标、键盘等来完成）。

　　当一个事件发生时，该事件用一个事件对象来表示。事件对象有对应的事件类。不同的事件类描述不同类型的用户动作。事件类包含在java.awt.event和javax.swing.event包中。

2.事件源

　　产生事件的组件叫事件源。在一个按钮上单击鼠标时，该按钮就是事件源，会产生一个ActionEvent类型的事件。

3.事件处理器（事件处理方法）

　　事件处理器是一个接收事件对象并进行相应处理的方法。事件处理器包含在一个类中，这个类的对象负责检查事件是否发生，若发生就激活事件处理器进行处理。

4.事件监听器类

　　包含事件处理器，并负责检查事件是否发生，若发生就激活事件处理器进行处理的类叫做事件监听器类。其实例就是事件监听器对象。事件监听器类必须实现事件监听器接口或继承事件监听器适配器类。

　　事件监听器接口定义了处理事件必须实现的方法。事件监听器适配器类是对事件监听器接口的简单实现。目的是为了减少编程的工作量。

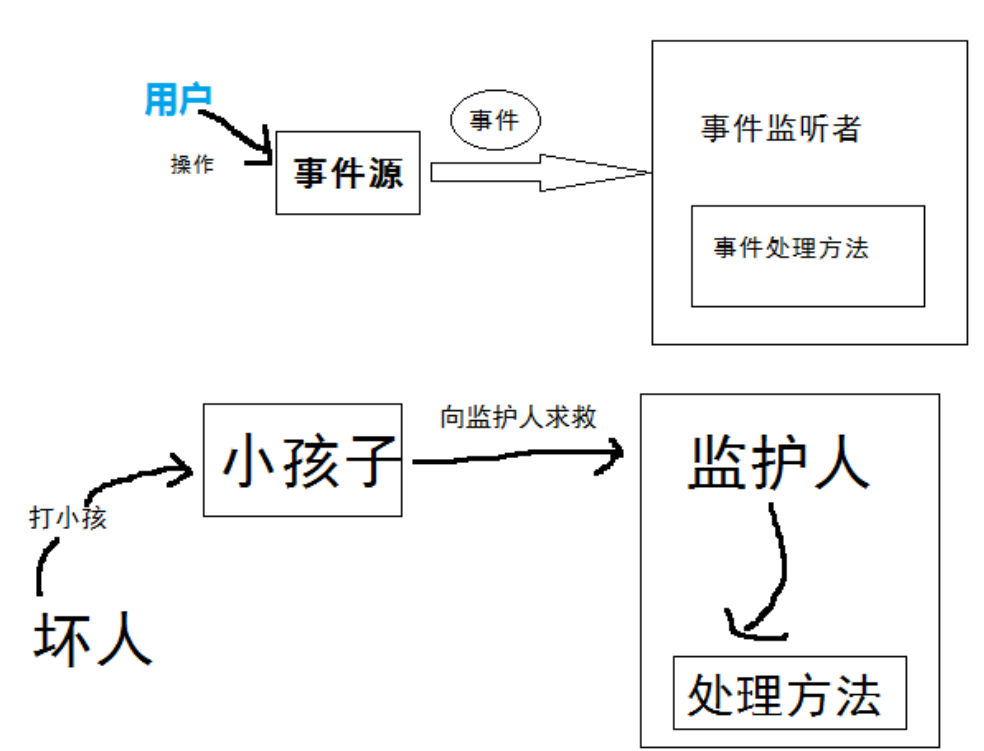
　　事件监听器接口和事件监听器适配器类也都包含在java.awt.event和javax.swing.event包中。

5.注册事件监听器

　　为了能够让事件监听器检查某个组件（事件源）是否发生了某些事件，并且在发生时激活事件处理器进行相应的处理，必须在事件源上注册事件监听器。这是通过使用事件源组件的以下方法来完成的：

　　addXxxListener（事件监听器对象）

　　Xxx对应相应的事件类。



思路：创建一个该监视器对象的类——>声明ActionListener接口——>

重写actionPerformed方法——>再创建一个[任意命名]类从而创建监视器——>\_\_\_通过addActionListener有监视器的类（监视器）服务于\_\_\_的对象

编写一个监听器类，该监听器类实现了XXXListener接口，（假定监听器类的类名为MyXXXListener)；

在MyXXXListener中的相应事件处理方法中编写事件处理代码（事件处理器）。

调用GUI组件c的addXXXListener()方法注册事件监听器对象。 c.addXXXListener(new MyXXXListener());

要处理C上的多种事件，可以编写多个相应的监听器类，进行多次注册。也可以编写一个实现了多个监听器接口的类，进行一次注册。

事件、事件监听器接口、事件监听器类:

每一类事件有一个相应的事件监听器接口，该接口定义了接收和

处理事件的抽象方法。实现该接口的类，就是监听器类。其对象可

作为监听器对象向相应的组件注册。

事件的类名通常为：XxxEvent

对应的事件监听器接口名通常为：XxxListener

一个监听器接口定义了一种以上的抽象事件处理方法（事件处理

器）。

事件监听器类实现事件监听器接口，其类名可以由我们自己取。事

件监听器类需要我们自己编写。

一个组件可以注册一个或多个监听器对象，事件对象只向已注册的

监听器对象报告事件的发生。

事件监听器类的编写要点

事件监听器类应该包括以下两部分内容：

1.在事件监听器类的声明中指定要实现的监听器接口名，如：

　　public class MyListener implements XxxListener {

　　… }

2.实现监听器接口中的事件处理方法，如：

　　public void 事件处理方法名(XxxEvent e) {

　　...//处理某个事件的代码... }

　　然后，在一个或多个组件上可以进行监听器类的实例的注册。如：组件对象.addXxxListener(MyListener对象);

java.awt.event 包中最常用的语义事件类：

•ActionEvent (对应按钮点击、 菜单选择、选择列表项或在文本框中 ENTER);

•AdjustmentEvent (用户调节滚动条)；

•ItemEvem (用户从复选框或列表框中选择一项）。

常用的 5 个底层事件类是：

•KeyEvent (一个键被按下或释放)；

•MouseEvent (鼠标键被按下、 释放、 移动或拖动)；

•MouseWheelEvent (鼠标滚轮被转动)；

•FocusEvent (某个组件获得焦点或失去焦点)；

•WindowEvent ( 窗口状态被改变）

所有的Listener接口都继承于EventListener,位于java.util包中。

AWT事件接口及处理方法（java.awt.event包）

接口 方法 参数 / 访问方法

ActionListener actionPerformed ActionEvent

• getActionCommand

• getModifiers

事件源 描述信息

AbstractButton 单击按钮、单击菜单项文本框内按回车等动作

JComboBox

JTextField

Timer

AdjustmentListener adjustmentValueChanged AdjustmentEvent

•getAdjustable

•getAdjustmentType

•getValue JScrollbar 移动了滚动条等组件

ItemListener itemStateChanged ItemEvent

•getltem

•getltemSelectable

•getStateChange

AbstractButton 选择了可选项的内容

JComboBox

FocusListener focusGained FocusEvent

focusLost •isTemporary

Component 组件得到或失去焦点

KeyListener keyPressed KeyEvent

keyReleased •getKeyChar

keyTyped •getKeyCode

•getKeyModifiersText

•getKeyText

•isActionKey

Component 键盘输入

MouseListener mousePressed MouseEvent

mouseReleased •getClickCount

mouseEntered •getX、getY

mouseExited •translatePoint

mouseClicked •getPoint

Component 鼠标点击等

MouseMotionListener mouseDragged MouseEven

mouseMoved

Component 鼠标移动

MouseWheelListener mouseWheelMoved MouseWheelEvent

•getWheelRotation

•getScrollAmount

Component 鼠标滚轮被转动

WindowListener windowClosing WindowEvent

windowOpened •getWindow

windowlconified

windowDeiconified

windowClosed

windowActivated

windowDeactivated

Window 窗口事件

WindowFocusListener windowGainedFocus WindowEvent

windowLostFocus •getOppositeWindow

Window 窗体得到或失去焦点

WindowStateListener windowStateChanged WindowEvent

•getOldState

•getNewState

Window 窗体状态

Document getDocument DocumentEvent

changedUpdata(DocumentEvent e)

RemoveUpdata(DocumentEvent e)

insertUpdata(DocumentEvent e)

文本区所维护的文档