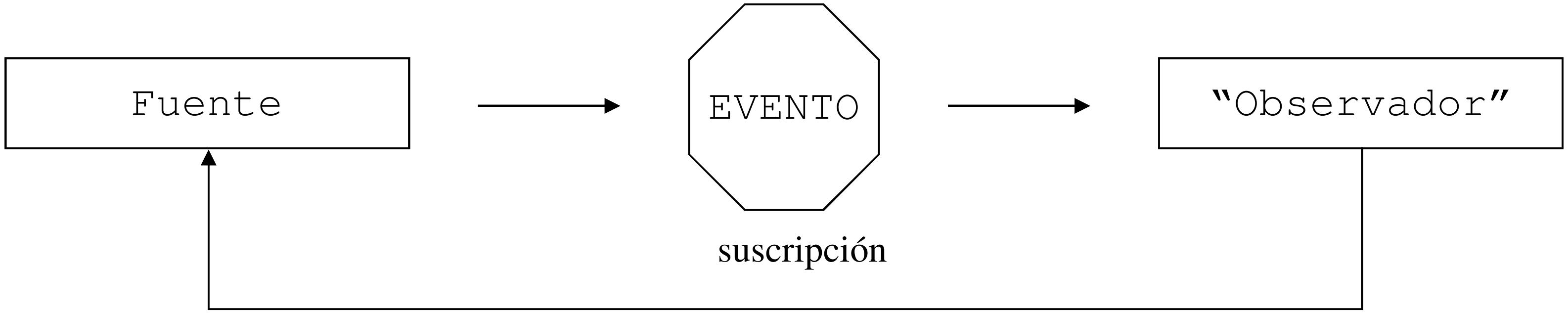
Interacción con GUIs

Un uso concreto del patrón “Delegation Event Model”

Más conocido como “observer” o “publish/subscribe”

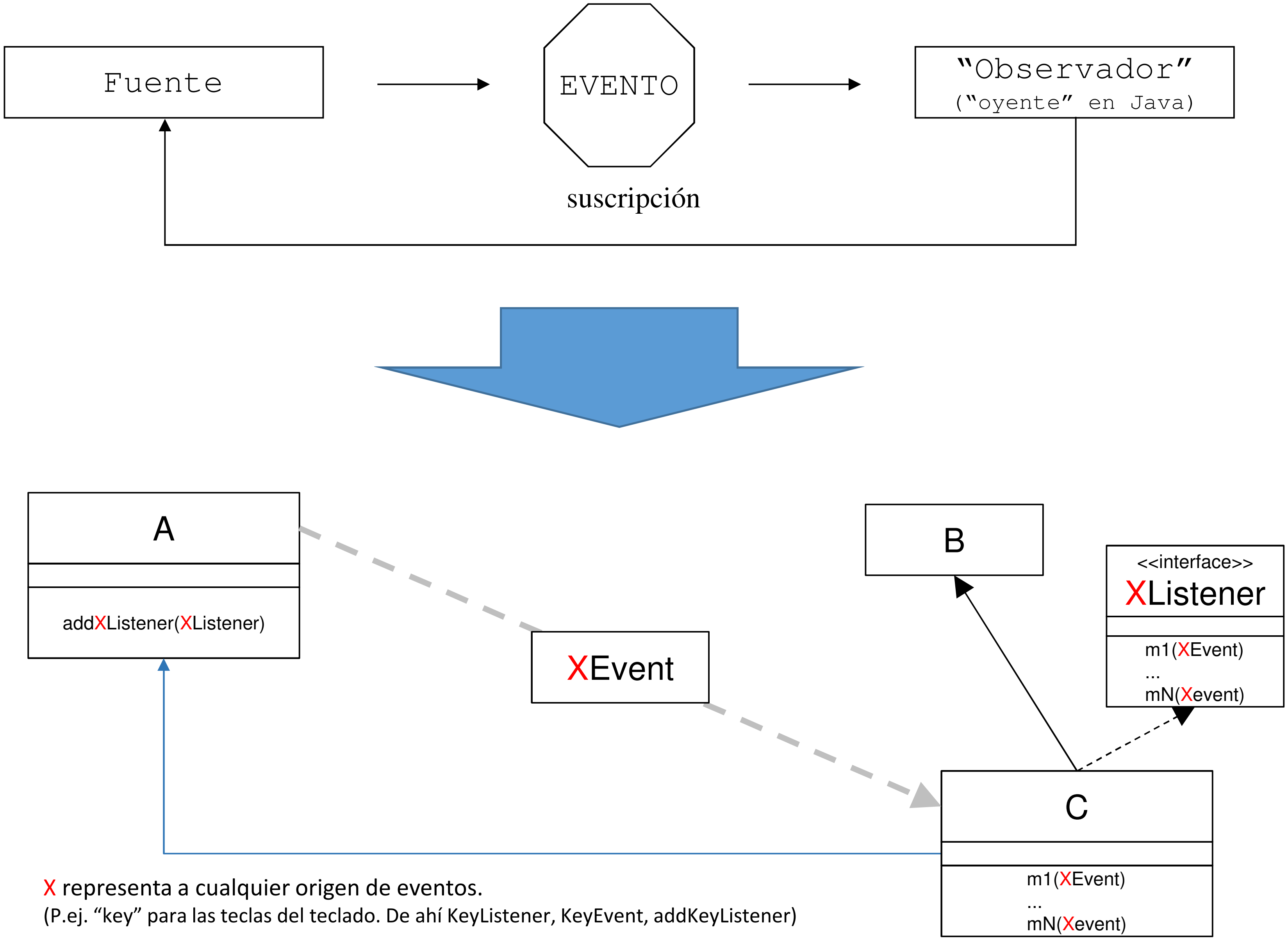


Fuente

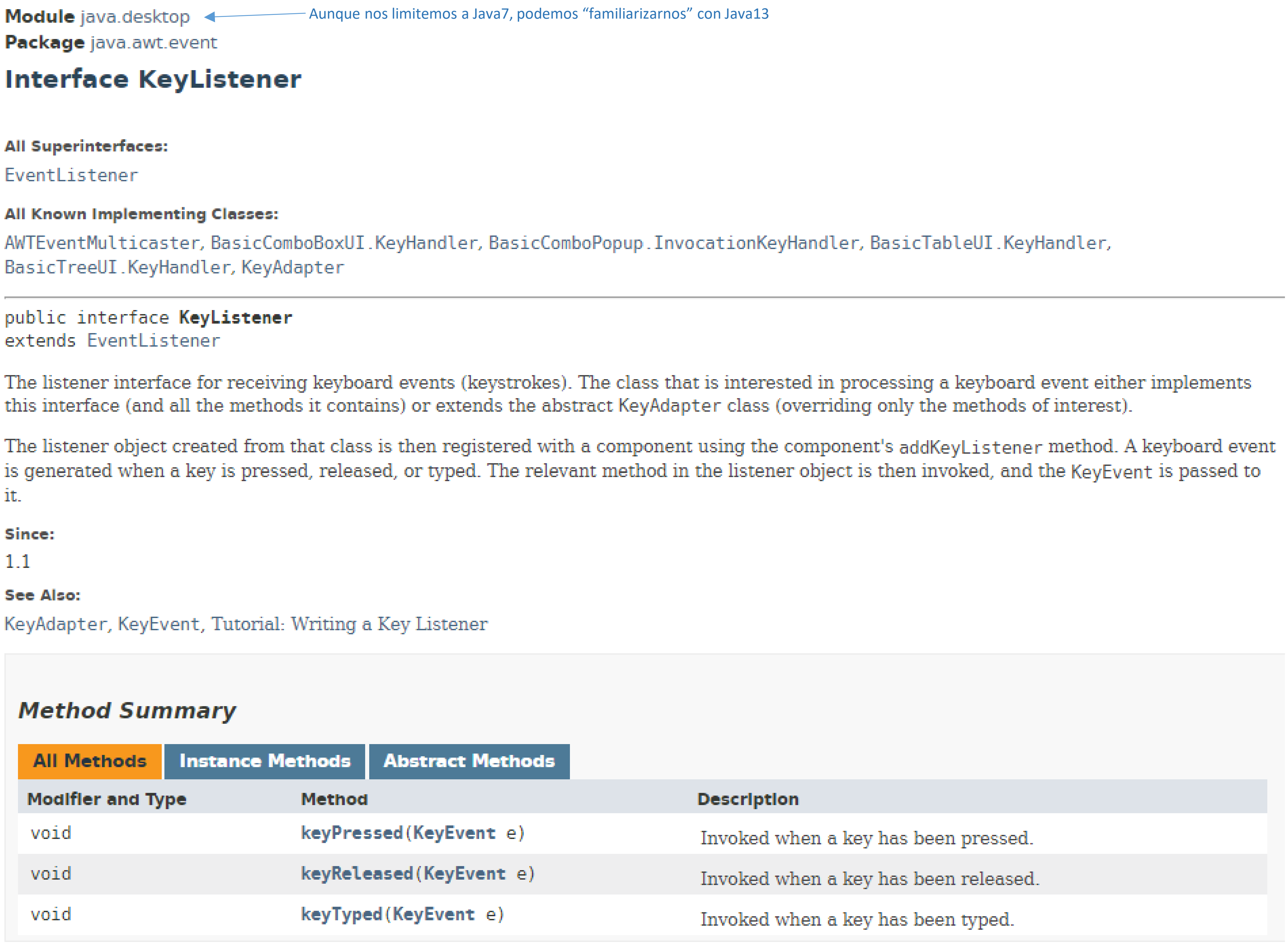
EVENTO

“Observador”

suscripción



X representa a cualquier origen de eventos.
(P.ej. "key" para las teclas del teclado. De ahí KeyListener, KeyEvent, addKeyListener)



Module java.desktop ← Aunque nos limitemos a Java7, podemos “familiarizarnos” con Java13

Package java.awt.event

Interface KeyListener

All Superinterfaces:

`EventListener`

All Known Implementing Classes:

`AWTEventMulticaster`, `BasicComboBoxUI.KeyHandler`, `BasicComboPopup.InvocationKeyHandler`, `BasicTableUI.KeyHandler`, `BasicTreeUI.KeyHandler`, `KeyAdapter`

```
public interface KeyListener
extends EventListener
```

The listener interface for receiving keyboard events (keystrokes). The class that is interested in processing a keyboard event either implements this interface (and all the methods it contains) or extends the abstract `KeyAdapter` class (overriding only the methods of interest).

The listener object created from that class is then registered with a component using the component's `addKeyListener` method. A keyboard event is generated when a key is pressed, released, or typed. The relevant method in the listener object is then invoked, and the `KeyEvent` is passed to it.

Since:

1.1

See Also:

`KeyAdapter`, `KeyEvent`, Tutorial: Writing a Key Listener

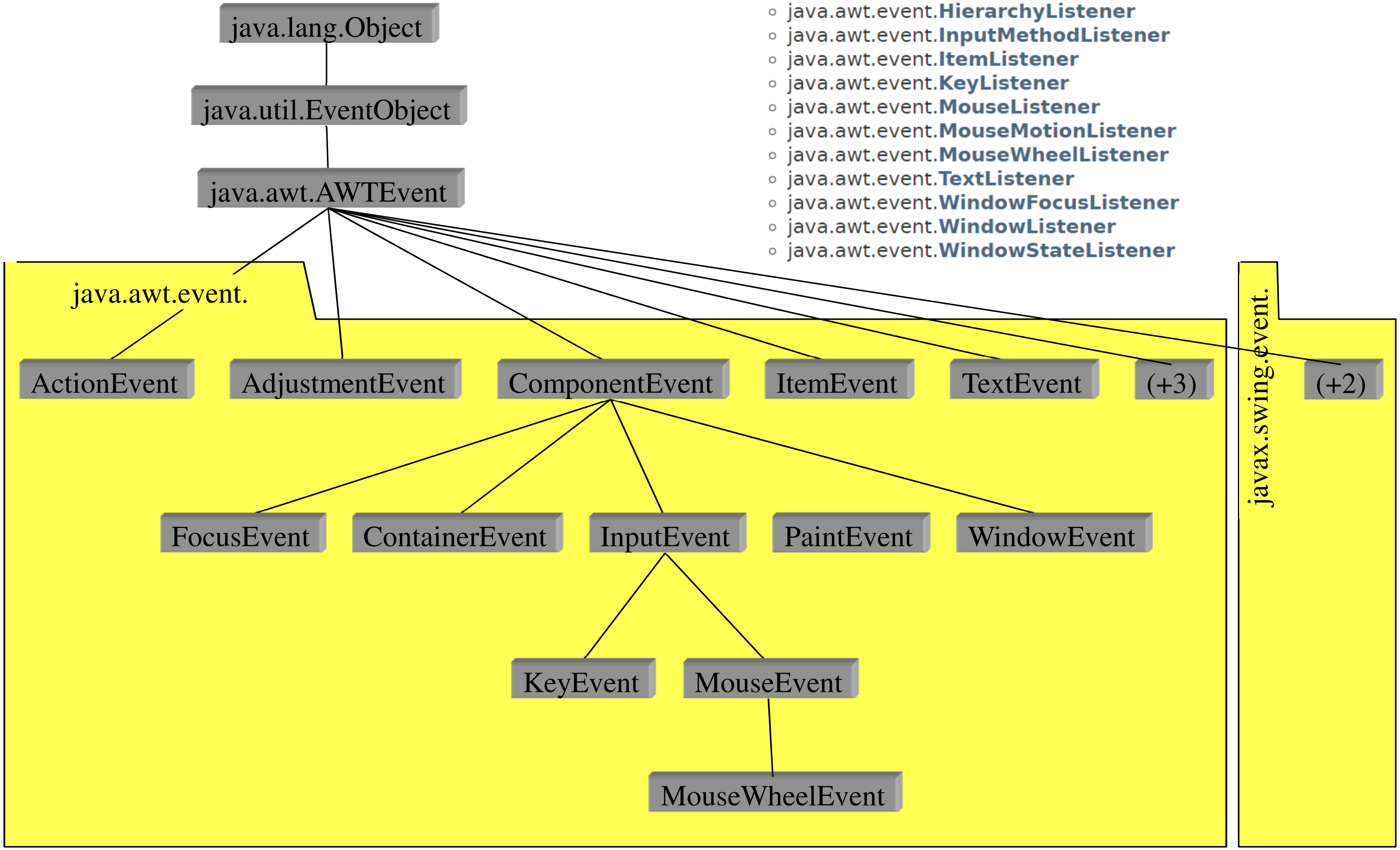
Method Summary

All Methods | **Instance Methods** | **Abstract Methods**

Modifier and Type	Method	Description
`void`	**`keyPressed`**`(KeyEvent e)`	Invoked when a key has been pressed.
`void`	**`keyReleased`**`(KeyEvent e)`	Invoked when a key has been released.
`void`	**`keyTyped`**`(KeyEvent e)`	Invoked when a key has been typed.

“Listeners” y eventos disponibles en un GUI con awt/swing

- java.util.**EventListener**
  - java.awt.event.**ActionListener**
  - java.awt.event.**AdjustmentListener**
  - java.awt.event.**AWTEventListener**
  - java.awt.event.**ComponentListener**
  - java.awt.event.**ContainerListener**
  - java.awt.event.**FocusListener**
  - java.awt.event.**HierarchyBoundsListener**
  - java.awt.event.**HierarchyListener**
  - java.awt.event.**InputMethodListener**
  - java.awt.event.**ItemListener**
  - java.awt.event.**KeyListener**
  - java.awt.event.**MouseListener**
  - java.awt.event.**MouseMotionListener**
  - java.awt.event.**MouseWheelListener**
  - java.awt.event.**TextListener**
  - java.awt.event.**WindowFocusListener**
  - java.awt.event.**WindowListener**
  - java.awt.event.**WindowStateListener**



java.lang.Object

java.util.EventObject

java.awt.AWTEvent

java.awt.event.

ActionEvent

AdjustmentEvent

ComponentEvent

ItemEvent

TextEvent

(+3)

FocusEvent

ContainerEvent

InputEvent

PaintEvent

WindowEvent

KeyEvent

MouseEvent

MouseWheelEvent

javax.swing.event.

(+2)

Métodos de los principales Listeners

ActionListener	actionPerformed(ActionEvent)
ItemListener	itemStateChanged(ItemEvent)

MouseListener	mousePressed(MouseEvent)
	mouseReleased(MouseEvent)
	mouseEntered(MouseEvent)
	mouseExited(MouseEvent)
	mouseClicked(MouseEvent)

MouseMotionListener	mouseDragged(MouseEvent)
	mouseMoved(MouseEvent)

KeyListener	keyPressed(KeyEvent)
	keyReleased(KeyEvent)
	keyTyped(KeyEvent)

FocusListener	focusGained(FocusEvent)
	focusLost(FocusEvent)

AdjustmentListener	adjustmentValueChanged(AdjustmentEvent)

ComponentListener	componentMoved(ComponentEvent)
	componentHidden(ComponentEvent)
	componentResized(ComponentEvent)
	componentShown(ComponentEvent)

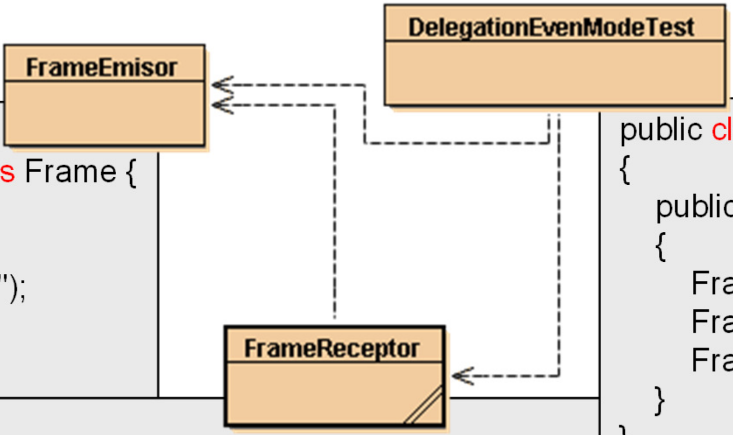
WindowListener	windowClosing(WindowEvent)
	windowOpened(WindowEvent)
	windowIconified(WindowEvent)
	windowDeiconified(WindowEvent)
	windowClosed(WindowEvent)
	windowActivated(WindowEvent)
	windowDeactivated(WindowEvent)

ContainerListener	componentAdded(ContainerEvent)
	componentRemoved(ContainerEvent)

TextListener	textValueChanged(TextEvent)

Un ejemplo hecho “a mano”

```
import java.awt.*;
public class FrameEmisor extends Frame {
    Button boton;
    public FrameEmisor() {
    boton = new Button ("Púlsame");
      add (boton);
      setVisible(true);
      pack();
    }
}
```

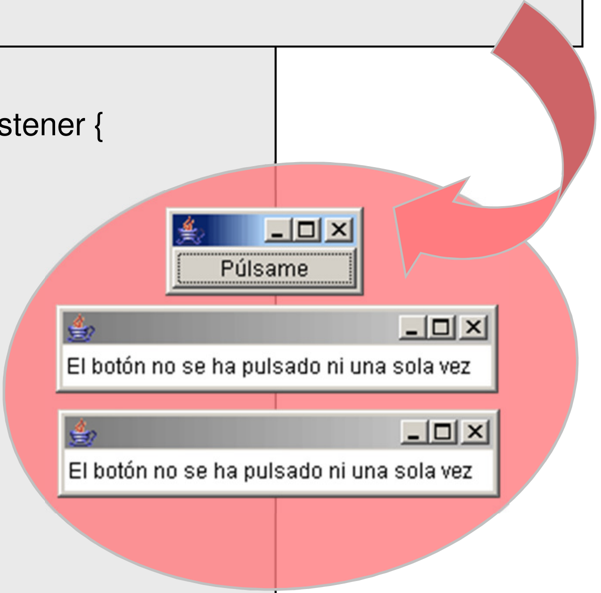


```
public class DelegationEvenModeTest
{
    public static void main(String[ ] args)
    {
        FrameEmisor fe=new FrameEmisor();
        FrameReceptor fr=new FrameReceptor(fe);
        FrameReceptor fr2=new FrameReceptor(fe);
    }
}
```

```
import java.awt.*;
import java.awt.event.*;

public class FrameReceptor extends java.awt.Frame implements MouseListener {
      private int toques=0;
      Label etiqueta
      public FrameReceptor(FrameEmisor f) {
        etiqueta = new Label ("El botón no se ha pulsado ni una sola vez");
        add (etiqueta);
        setVisible(true);
        pack();

        f.boton.addMouseListener(this);
      }

      public void mouseClicked (java.awt.event.MouseEvent evt) {
        etiqueta.setText ("El botón se ha pulsado "+(++toques)+((toques==1)?" vez":" veces"));
      }

      public void mousePressed (MouseEvent evt) { }
      public void mouseReleased (MouseEvent evt) { }
      public void mouseEntered (MouseEvent evt) { }
      public void mouseExited (MouseEvent evt) { }
}
```

Suscripción

MouseListener

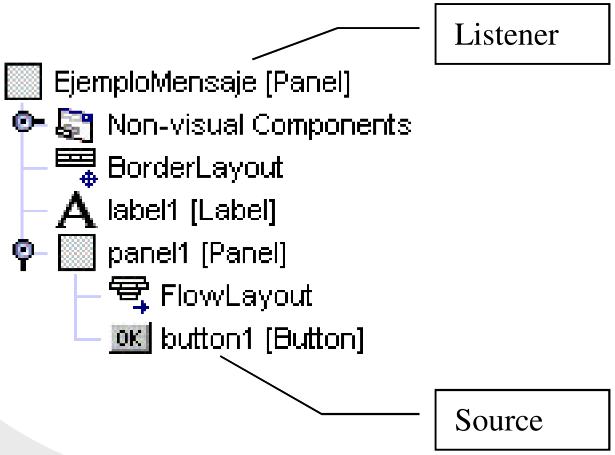


El botón no se ha pulsado ni una sola vez

El botón no se ha pulsado ni una sola vez

Un ejemplo (a estudiar) hecho con ayuda de Netbeans

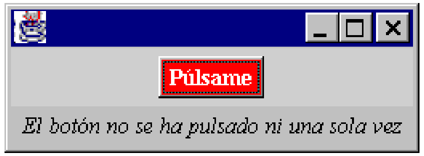
```
import java.awt.*;
public class EjemploMensaje extends java.awt.Panel {

    public EjemploMensaje() {
        initComponents ();
    }

  private void initComponents () {
    label1 = new java.awt.Label ();
    panel1 = new java.awt.Panel ();
    button1 = new java.awt.Button ();
    setLayout (new java.awt.BorderLayout ());

    label1.setFont (new java.awt.Font ("Book Antiqua", 2, 14));
    label1.setName ("label6");
    label1.setBackground (new java.awt.Color (204, 204, 204));
    label1.setForeground (java.awt.Color.black);
    label1.setText ("El bot\u00f3n no se ha pulsado ni una sola vez");
    label1.setAlignment (java.awt.Label.CENTER);

    add (label1, java.awt.BorderLayout.CENTER);

    panel1.setFont (new java.awt.Font ("Dialog", 0, 11));
    panel1.setName ("panel8");
    panel1.setBackground (new java.awt.Color (204, 204, 204));
    panel1.setForeground (java.awt.Color.black);

    button1.setFont (new java.awt.Font ("Arial", 1, 14));
    button1.setLabel ("P\u00falsame");
    button1.setName ("button6");
    button1.setBackground (new java.awt.Color (255, 0, 51));
    button1.setForeground (java.awt.Color.white);

(continúa ...)
```



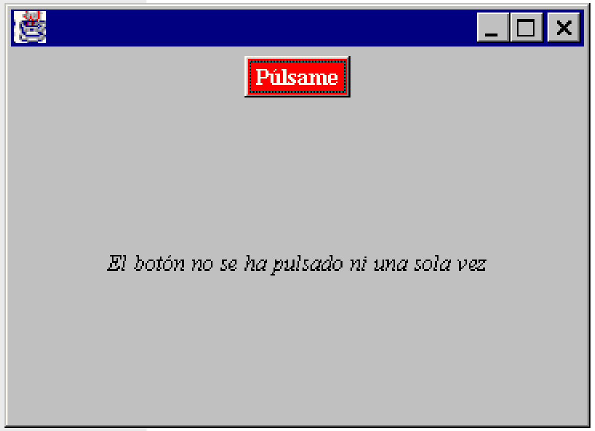
Listener

```
EjemploMensaje [Panel]
  Non-visual Components
  BorderLayout
  label1 [Label]
  panel1 [Panel]
    FlowLayout
    button1 [Button]
```

Source



El botón no se ha pulsado ni una sola vez



El botón no se ha pulsado ni una sola vez button1.addMouseListener (nuevo objeto MouseAdapter definido aquí mismo);

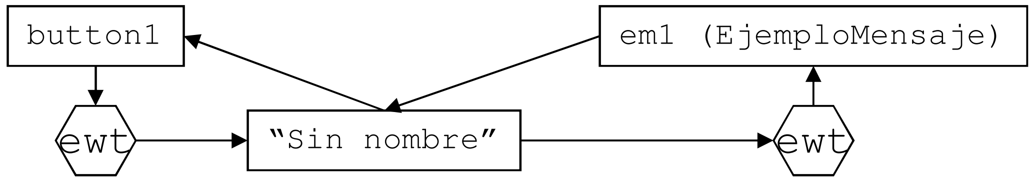
Source

Adapter "sin nombre"

```
    button1.addMouseListener (new java.awt.event.MouseAdapter () {
      public void mouseClicked (java.awt.event.MouseEvent evt) {
        button1MouseClicked (evt);
        }
      }
      );
    panel1.add (button1);

    add (panel1, java.awt.BorderLayout.NORTH);

    }

  private void button1MouseClicked (java.awt.event.MouseEvent evt) {
    label1.setText ("El botón se ha pulsado "+(++toques)+((toques==1)?" vez":" veces"));
  }


private java.awt.Label label1;
private java.awt.Panel panel1;
private java.awt.Button button1;

private int toques=0;
}
```

El "oyente" del adapter es el propio Panel

Rutina "oyente" del adaptador



```
public class FrameTEst extends java.awt.Frame {

    public FrameTEst() {
        initComponents ();
        ejm=new EjemploMensaje();
        add(ejm);
        pack ();
    }

    private void initComponents () {
        addWindowListener (new java.awt.event.WindowAdapter () {
            public void windowClosing (java.awt.event.WindowEvent evt) {
                exitForm (evt);
            }
        }
        );
    }

    private void exitForm(java.awt.event.WindowEvent evt) {
        System.exit (0);
    }

    public static void main (String args[]) {
        new FrameTEst ().show ();
    }

    EjemploMensaje ejm;
}
```