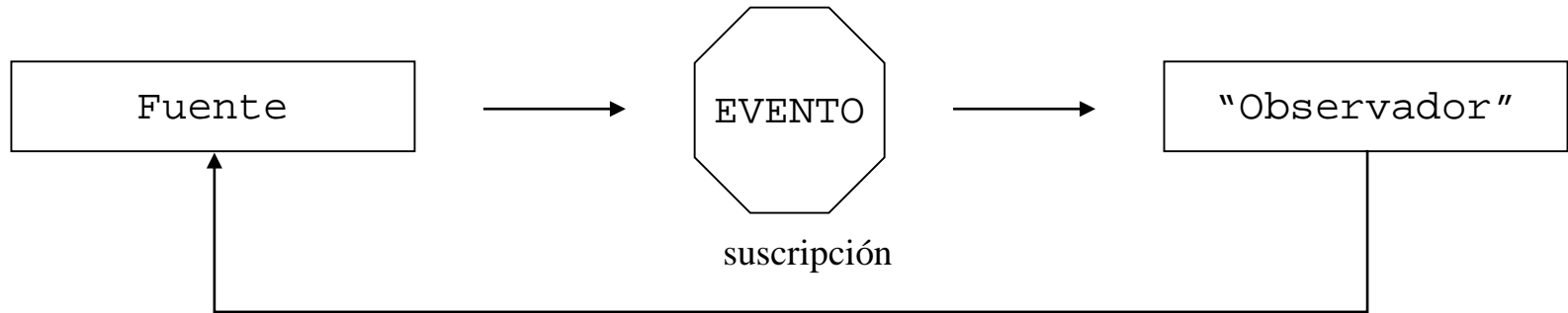
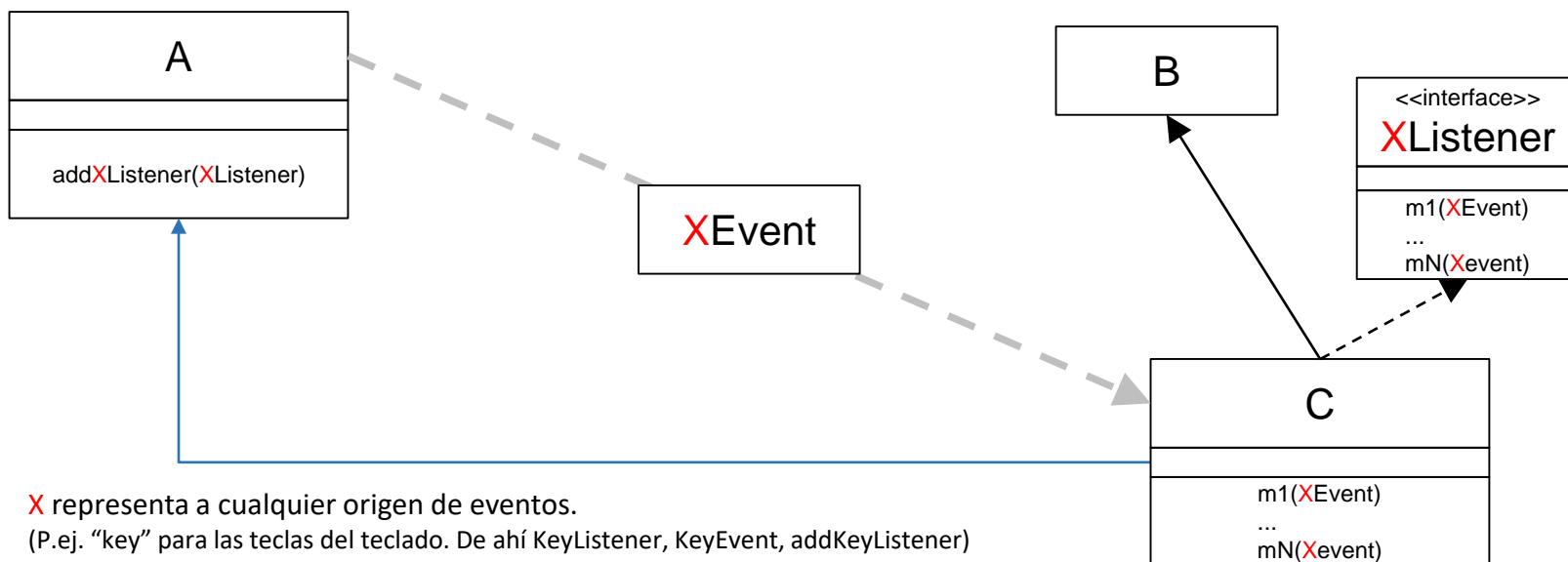
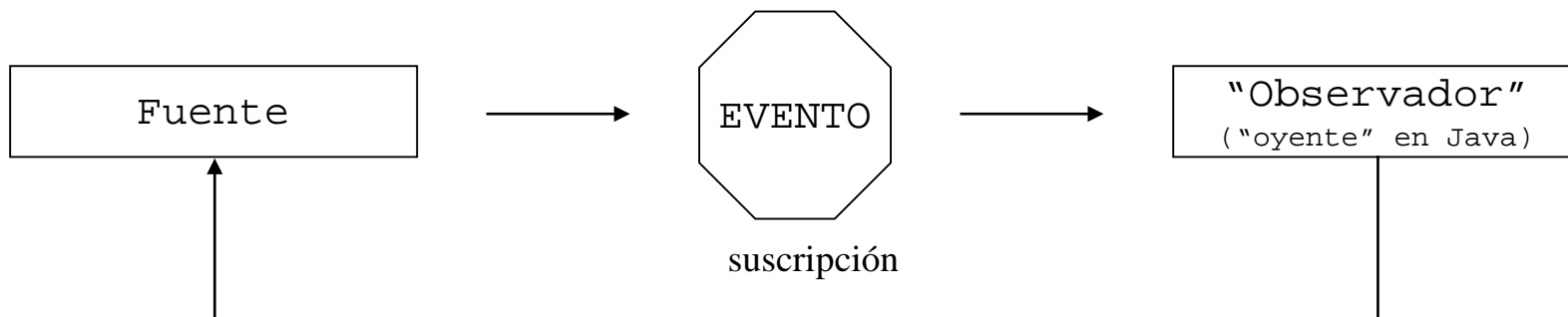


Interacción con GUIs

Un uso concreto del patrón “Delegation Event Model”

Más conocido como “[observer](#)” o “publish/subscribe”





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

Aunque nos limitemos a Java7, podemos “familiarizarnos” con Java13

Module java.desktop

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.

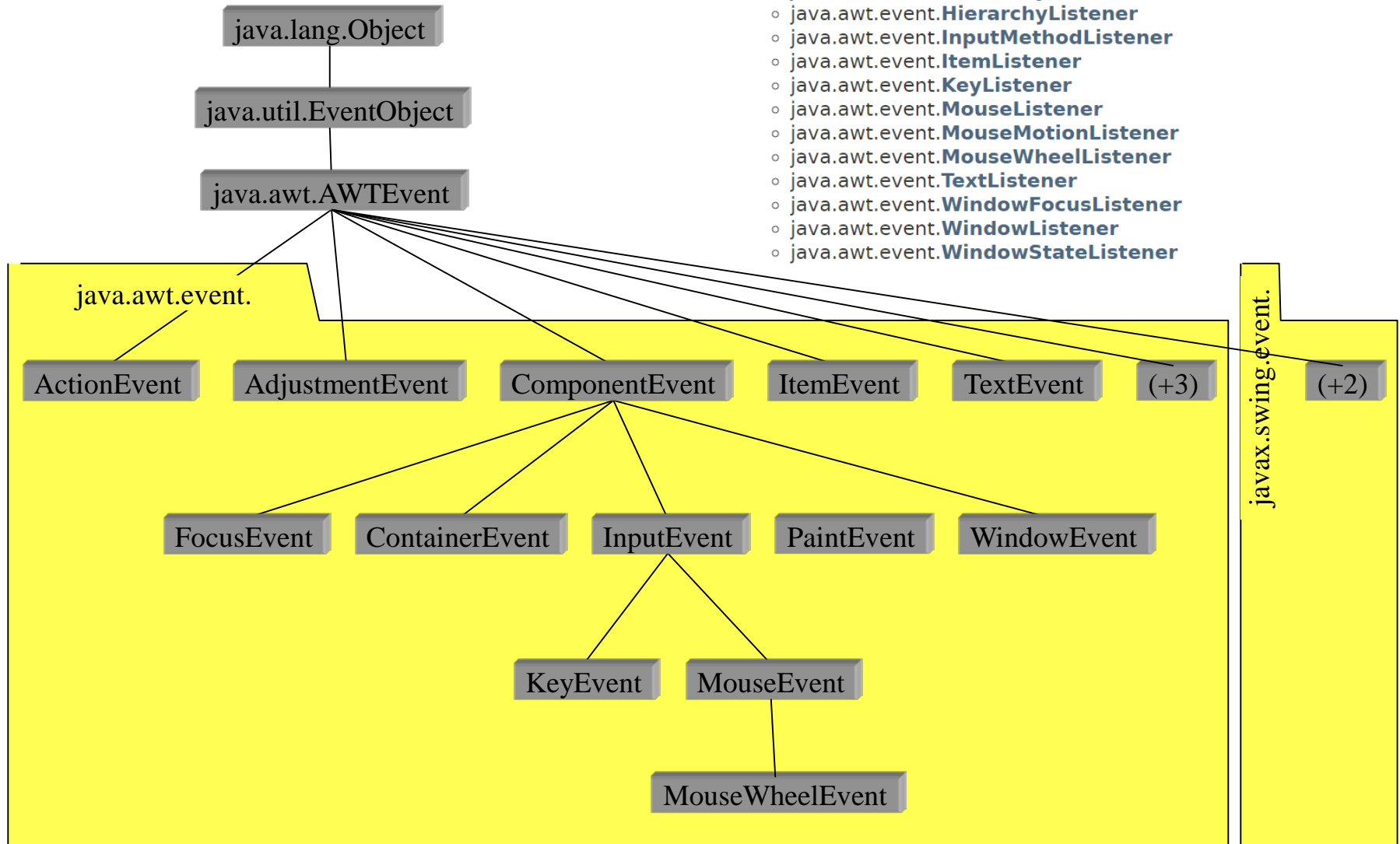
void

keyTyped(KeyEvent e)

Invoked when a key has been typed.

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

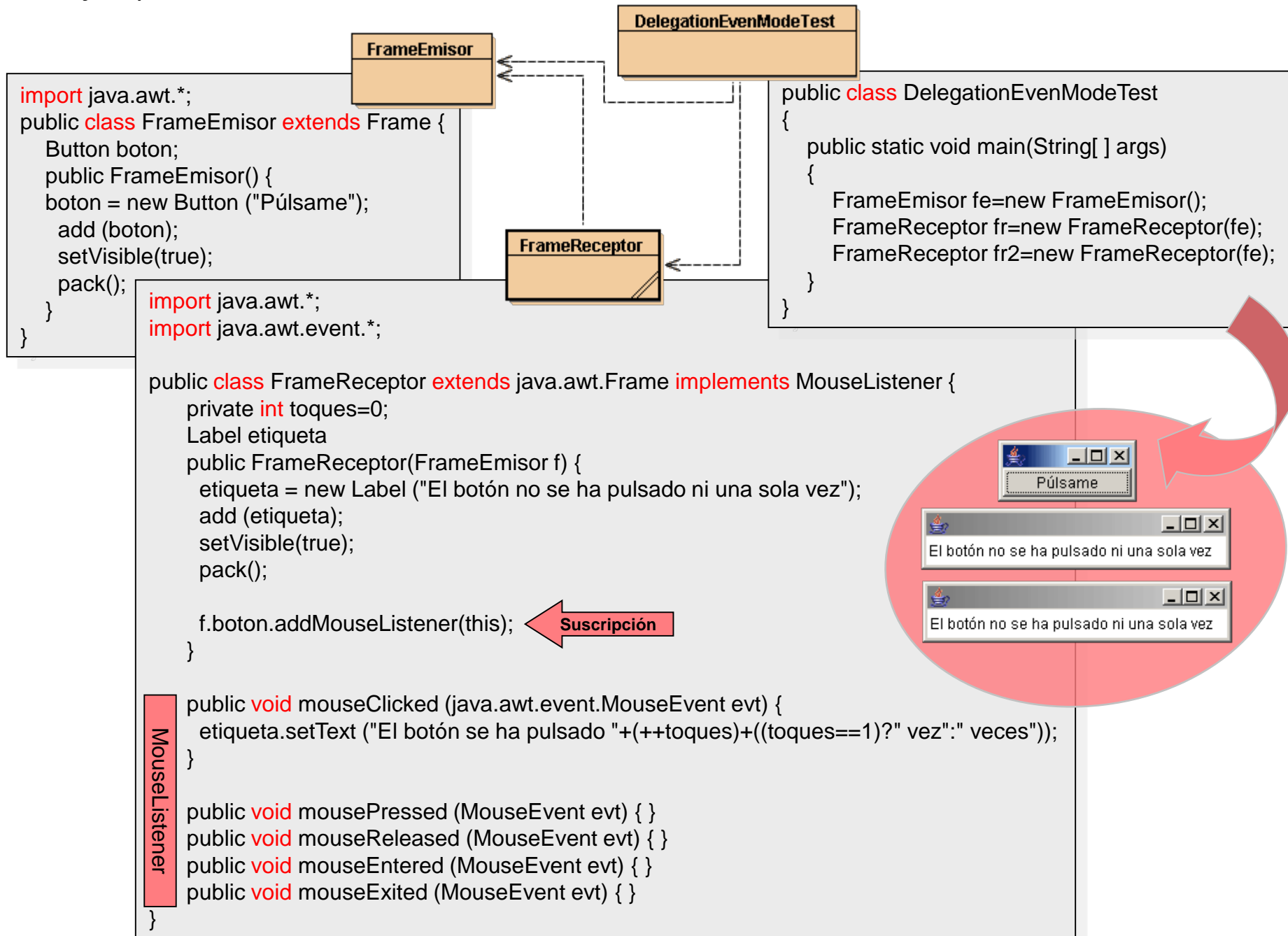
- java.util.**EventListener**
 - java.awt.event.**ActionListener**
 - java.awt.event.**AdjustmentListener**
 - java.awt.event.**AWTEventListener**
 - java.awt.event.**ComponentListener**
 - java.awt.event.**ContainerListener**
 - java.awt.event.**FocusListener**
 - java.awt.event.**HierarchyBoundsListener**
 - 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**



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"



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 () {
        labell = new java.awt.Label ();
        panell = new java.awt.Panel ();
        boton1 = new java.awt.Button ();
        setLayout (new java.awt.BorderLayout ());

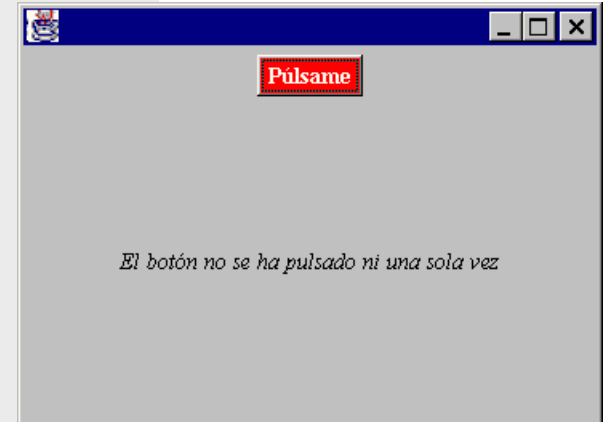
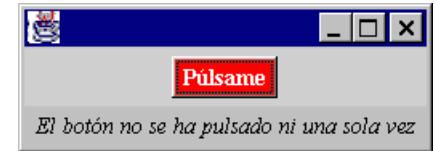
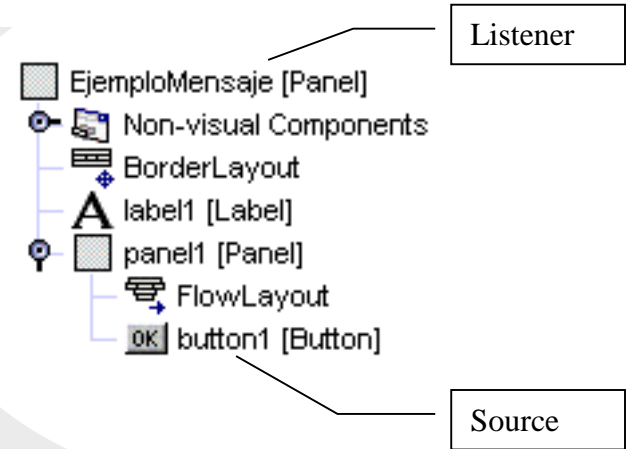
        labell.setFont (new java.awt.Font ("Book Antiqua", 2, 14));
        labell.setName ("label6");
        labell.setBackground (new java.awt.Color (204, 204, 204));
        labell.setForeground (java.awt.Color.black);
        labell.setText ("El bot\u00f3n no se ha pulsado ni una sola vez");
        labell.setAlignment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

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

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

        boton1.setFont (new java.awt.Font ("Arial", 1, 14));
        boton1.setLabel ("P\u00falsame");
        boton1.setName ("button6");
        boton1.setBackground (new java.awt.Color (255, 0, 51));
        boton1.setForeground (java.awt.Color.white);
```

(continúa ...)



```
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 );  
    }  
} );
```

El "oyente" del adapter es el propio Panel

```
panell.add ( button1 );
```

```
add ( panell, java.awt.BorderLayout.NORTH );
```

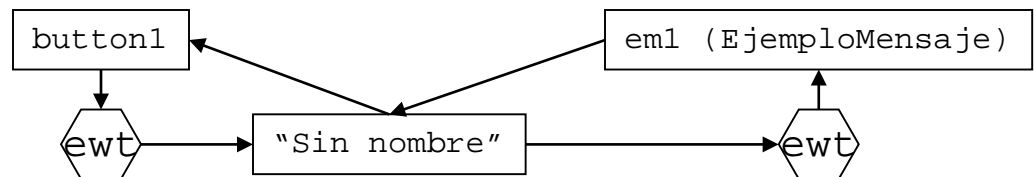
```
}
```

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

Rutina "oyente" del adaptador

```
private java.awt.Label labell;  
private java.awt.Panel panell;  
private java.awt.Button button1;
```

```
private int toques=0;  
}
```




```
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;
}
```