Eventos

Un evento es un objeto que representa un cambio en un componente

Los eventos son generados por los usuarios u otras aplicaciones

- Para la captura de eventos es necesario relacionarlo a un escuchador (listener) con un componente que notifique al listener cuando ocurra un evento específico

Interfaces
ActionListener
AdjustmentListener
AWTEventListener
ComponentListener
ContainerListener
FocusListener
InputMethodListener
ItemListener
KeyListener
MouseListener
MouseMotionListener
TextListener
WindowListener

Clases Abstractas
ComponentAdapter
ContainerAdapter
ContainerEvent
FocusAdapter
KeyAdapter
MouseMotionAdapter
WindowAdapter

Manejador de Eventos

Las clases e interfaces para los eventos en Java se encuentran en el paquete java.awt.event

El modelo de eventos define mecanismos estándar y consistentes para generar y procesar eventos

El concepto utilizado propone: una *fuente* genera un evento y lo envía a uno o más *oyentes*

Una vez recibido, el oyente procesa el evento y luego lo retorna

La ventaja de este diseño es que la lógica que procesa los eventos está claramente separada de la lógica de la interfaz de usuario que genera esos eventos

Un elemento de la interfaz de usuario puede delegar el procesamiento de un evento a un segmento separado de código

Para el modelo de eventos, es necesario registrar una fuente (origen) con el fin de recibir una notificación del evento

Eventos

En el modelo de delegación, un evento es un objeto que describe un cambio de estado en un fuente

Puede generarse como consecuencia de la interacción de una persona con los elementos en la interfaz gráfica de usuario como presionar un botón, ingresar un carácter vía teclado, seleccionar un elemento en una lista y hace clic en el mouse

Fuente de Eventos

Una fuente de eventos es un objeto que genera un evento

Una fuente debe registrar oyentes con el fin de que reciban notificaciones acerca de un tipo específico de evento

Cada tipo de evento tiene su propio método de registro, tal como:

- public void add*Tipo*Listener(*Tipo*Listener escuchaevento)

Los oyentes(escuchadores) de los eventos

Un oyente es un objeto que notifica cuando un evento ocurre

Tiene dos requerimientos principales:

- Debe haber sido registrado por una o más fuentes para recibir notificaciones acerca de tipos específicos de eventos
- Debe implementar métodos para recibir y procesar esas notificaciones

Los métodos que reciben y procesan eventos están definidos en un conjunto de interfaces encontradas en **java.awt.event**

Clases de Eventos

Son parte fundamental del manejo de eventos en Java

En la raíz de la jerarquía de la clase evento de Java está **EventObject**, la cual está en java.util. Esta es la superclase de todos los eventos

La clase AWTEvent, definida dentro del paquete de java.awt, es una subclase de EventObject, y es la superclase (directo o indirecta) de todos los eventos basados en AWT utilizados por el modelo de delegación de eventos

Principales clases de eventos:

ActionEvent.- Se genera cuando se presiona un botón, se hace doble clic sobre una lista de elementos o se selecciona un elemento del menú

AdjustmentEvent.- Se genera cuando se maneja una barra de (scroll) desplazamiento

ComponentEvent.- Se genera cuando un componente se oculta, mueve, cambia de tamaño o se hace visible

TextEvent.- Se genera cuando el valor de un área de texto o campo de texto se cambia

1) Ejemplo con un Botón

```
……
public class VentanaForma1 extends JFrame{
        private JButton b1, b2, b3, b4, b5, b6, b7, b8, b9;
        public VentanaForma1(){
                super("Ventana de Ejemplo");
                setLayout(new GridLayout(3,3));
                b1 = new JButton("Botón 1");
                add(b1);
                b1.addActionListener(new ActionListener(){
                        public void actionPerformed(ActionEvent e){
                                System.out.println("Click Botón 1 - Ventana");
                        }
                });
        }
        ….
}
```

2) Ejemplo con un Botón

```
....
public class VentanaForma1 extends JFrame implements ActionListener{
        private JButton b1, b2, b3, b4, b5, b6, b7, b8, b9;
        public VentanaForma1(){
                super("Ventana de Ejemplo");
                setLayout(new GridLayout(3,3));
                b1 = new JButton("Botón 1");
                add(b1);
                b1.addActionListener(this);
        }
        public void actionPerformed(ActionEvent e){
                if (e.getSource() == b1){
                        System.out.println("Click Botón 1 - Ventana");
                }
        }
}
```