

## EJERCICIO GUIADO. JAVA: EVENTOS DESDE CÓDIGO. IDEAS GENERALES

### Eventos desde código. Consideraciones Generales

Se finalizará el estudio de los eventos desde código, planteando de forma general el código necesario para programarlos.

A la hora de programar eventos sobre un componente concreto, será necesario tener en cuenta lo siguiente:

- Primero hay que enlazar el componente con el oyente o adaptador correspondiente al evento.
- El oyente o adaptador contendrá aquellos eventos que interesen programar.
- Cada evento tiene un parámetro asociado con información sobre el evento.
- Cada evento realiza una llamada a un procedimiento y es en este procedimiento donde se programa la respuesta al evento.

La sintaxis general de programación de eventos es la siguiente:

```
Componente.MetodoParaAñadirOyente(new TipoOyente() {  
  
    public void evento1(TipoEvento evt) {  
        LlamadaAProcedimiento1(evt);  
    }  
  
    public void evento2(TipoEvento evt) {  
        LlamadaAProcedimiento2(evt);  
    }  
  
    ...otros eventos...  
});
```

Para poder programar un evento es necesario conocer lo siguiente:

- El componente sobre el que se programará.
- El tipo oyente o adaptador al que pertenece el evento.
- El método usado para asignar el oyente al componente.
- El nombre del evento.
- El tipo de evento para el parámetro.

## **Ejemplo 1. Eventos del Ratón**

Los eventos de ratón son: *mouseEntered*, *mouseExited*, *mousePressed*, *mouseReleased*, *mouseClicked*.

Pertenecen al adaptador *MouseAdapter*.

El adaptador se asigna al componente con el método *addMouseListener*.

Los métodos llevan como parámetro un objeto del tipo *MouseEvent*.

Conociendo esto, si queremos programar los eventos del ratón de una etiqueta llamada *etiTexto*, solo tenemos que hacer lo siguiente:

```
etiTexto.addMouseListener(new MouseAdapter() {  
  
    public void mouseEntered(MouseEvent evt) {  
        etiTextoMouseEntered(evt);  
    }  
  
    public void mouseExited(MouseEvent evt) {  
        etiTextoMouseExited(evt);  
    }  
  
    public void mousePressed(MouseEvent evt) {  
        etiTextoMousePressed(evt);  
    }  
  
    public void mouseReleased(MouseEvent evt) {  
        etiTextoMouseReleased(evt);  
    }  
  
    public void mouseClicked(MouseEvent evt) {  
        etiTextoMouseClicked(evt);  
    }  
  
});
```

## **Ejemplo 2. Eventos del teclado**

Los eventos de teclado controlan las pulsaciones de tecla. Entre ellos, podemos mencionar a los siguientes:

- *keyPressed*, que define la pulsación de una tecla.
- *keyReleased*, que define el momento en que se suelta una tecla.
- *keyTyped*, que define el pulsar-soltar una tecla.

Los eventos de teclado pertenecen al adaptador *KeyAdapter*.

El adaptador anterior se asigna a un componente a través del método *addKeyListener*.

Los eventos de teclado llevan como parámetro un objeto del tipo *KeyEvent*.

Sabiendo lo anterior, si se quisieran programar los eventos de teclado de un cuadro de texto llamado txtDato el código necesario sería el siguiente:

```
txtDato.addKeyListener(new KeyAdapter() {  
  
    public void keyPressed(KeyEvent evt) {  
        txtDatoKeyPressed(evt);  
    }  
  
    public void keyReleased(KeyEvent evt) {  
        txtDatoKeyReleased(evt);  
    }  
  
    public void keyTyped(KeyEvent evt) {  
        txtDatoKeyTyped(evt);  
    }  
  
});
```

Como puedes observar, la programación de eventos siempre se hace de la misma forma, solo tienes que conocer el adaptador/oyente, el método para asignar dicho adaptador/oyente, el nombre de los eventos, y el tipo de objeto que llevan como parámetro.

Sabiendo estos datos, la programación siempre se hace igual.

El problema que se plantea en cambio es el siguiente:

Si surge un nuevo evento X el cual quiero usar desde código, ¿cómo puedo saber cual es su adaptador/oyente, el método para asignar dicho adaptador/oyente al componente, y el tipo de datos del parámetro?

Para este caso, se aconseja crear un proyecto de prueba desde la ventana de diseño y asignar el evento X que se quiere estudiar a un componente. Luego simplemente hay que analizar el código generado por el NetBeans.

--- NO HAY EJERCICIO GUIADO EN ESTA HOJA ---

## CONCLUSIÓN

Para programar desde código un evento es necesario conocer lo siguiente:

- \* El nombre del evento / eventos del mismo grupo.
- \* El oyente/adaptador al que pertenece el evento / eventos.
- \* El método para asignar el oyente/adaptador al componente sobre el que actúa el evento.
- \* El tipo de datos de objeto recibido como parámetro por el evento.

Conociendo estos datos, la programación siempre se hace igual:

```
Componente.MetodoParaAñadirOyente(new NombreOyenteAdaptador() {  
    public void nombreEvento1(TipoDatoParametro evt) {  
        LlamadaAProcedimientoEvento1(evt);  
    }  
  
    public void nombreEvento2(TipoDatoParametro evt) {  
        LlamadaAProcedimientoEvento2(evt);  
    }  
});
```

Se puede aprender mucho sobre los eventos asociándolos desde diseño y luego mirando en el código generado.