

EJERCICIO GUIADO. JAVA: EVENTOS DESDE CÓDIGO. EVENTOS DE RATÓN

Eventos desde código

Se estudió en la hoja de teoría anterior la forma de programar eventos desde código. Concretamente se estudió como programar eventos de acción. En esta hoja se estudiará como programar eventos de ratón.

Los eventos de ratón son los siguientes:

- mouseEntered
 - o Indica que el ratón entró en la superficie del elemento (etiqueta, botón, etc)
- mouseExited
 - o Indica que el ratón salió de la superficie del elemento (etiqueta, botón, etc)
- mousePressed
 - o Indica que un botón del ratón ha sido presionado sobre el elemento.
- mouseReleased
 - o Indica que un botón del ratón ha sido soltado sobre el elemento.
- mouseClicked
 - o Indica que se ha pulsado/soltado un botón del ratón.

Como se dijo en la hoja anterior, todos los eventos pertenecen a un objeto “oyente” o “adaptador”. En este caso concreto, todos los eventos del ratón pertenecen a un objeto “adaptador” llamado MouseAdapter.

Cada uno de los eventos se programará dentro del adaptador MouseAdapter. Cada evento es en realidad un procedimiento que recibe un parámetro con información sobre el evento. Este parámetro, en el caso de los eventos del ratón, es del tipo MouseEvent.

Cada evento a su vez hace una llamada a un procedimiento, y es en este procedimiento donde realmente se programa la respuesta al evento.

Todo esto resulta complicado al principio, pero verás que en realidad siempre es igual. Solo hay que conocer el nombre de cada evento, el adaptador u oyente al que pertenece y el tipo de parámetro que recibe cada evento.

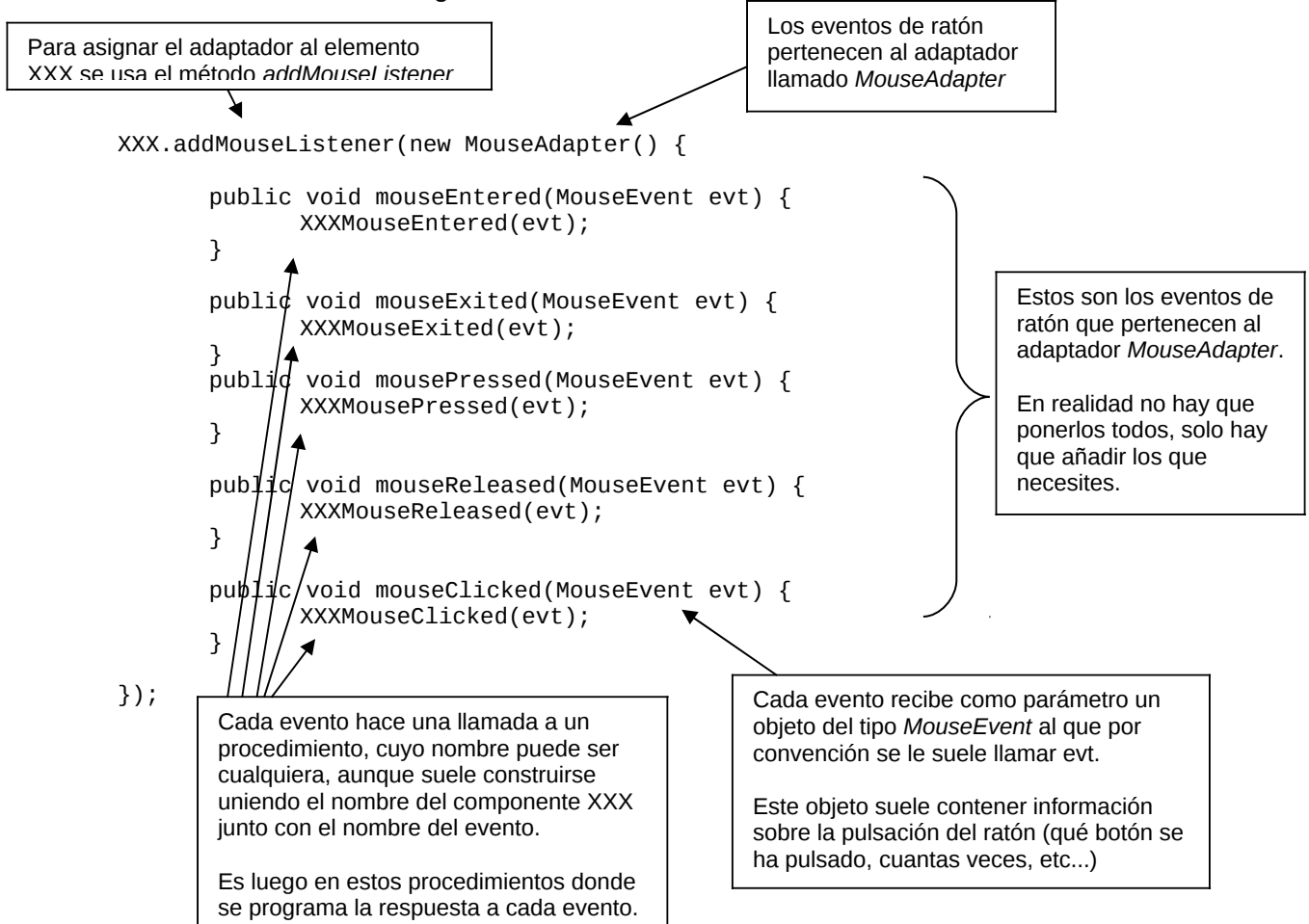
Como enlazar eventos de ratón con un componente

Supongamos que tenemos un componente (botón, etiqueta, etc...) llamado XXX.

Si queremos asociarle eventos de ratón, tendremos que añadir el siguiente código dentro del constructor:

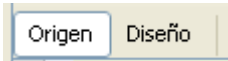
```
XXX.addMouseListener(new MouseAdapter() {  
  
    public void mouseEntered(MouseEvent evt) {  
        XXXMouseEntered(evt);  
    }  
  
    public void mouseExited(MouseEvent evt) {  
        XXXMouseExited(evt);  
    }  
  
    public void mousePressed(MouseEvent evt) {  
        XXXMousePressed(evt);  
    }  
  
    public void mouseReleased(MouseEvent evt) {  
        XXXMouseReleased(evt);  
    }  
  
    public void mouseClicked(MouseEvent evt) {  
        XXXMouseClicked(evt);  
    }  
  
});
```

Veamos este mismo código comentado:



EJERCICIO GUIADO

1. Crea un nuevo proyecto.
2. Accede al código del proyecto.



3. Inserta una llamada dentro del constructor a un procedimiento *CreacionVentana* de la siguiente forma:

```
public ventanaprincipal() {  
    initComponents();  
    CreacionVentana();  
}  
  
public void CreacionVentana() {  
  
}
```

La llamada...

Y el procedimiento.

4. La ventana va a contener dos etiquetas, que tendremos que declarar en la zona de variables globales:

```
public class ventanaprincipal extends javax.swing.JFrame {  
  
    JLabel etiTexto;  
    JLabel etiZona;
```

5. Ahora vamos a diseñar la ventana. Introduce el siguiente código dentro del procedimiento *CreacionVentana*:

```
public void CreacionVentana() {  
  
    this.setTitle("Programacion de eventos de ratón");  
    this.setSize(400,400);  
    this.setLocation(100,100);  
  
    etiTexto = new JLabel();  
    etiTexto.setBounds(50,10,300,20);  
    etiTexto.setText("---");  
    this.getContentPane().add(etiTexto);  
  
    etiZona = new JLabel();  
    etiZona.setBounds(50,50,300,250);  
    etiZona.setText("");  
    etiZona.setOpaque(true);  
    etiZona.setBackground(Color.RED);  
    this.getContentPane().add(etiZona);  
  
}
```

Detalles de la ventana

Se diseña la primera etiqueta

Se diseña la segunda etiqueta

6. Ejecuta ya el programa para observar el resultado de nuestro diseño. El aspecto debe ser el siguiente:



7. El objetivo del programa es simple. Queremos que al entrar el ratón dentro de la superficie de la etiqueta roja, aparezca un mensaje en la otra etiqueta indicándolo. También queremos que aparezca un mensaje al salir el ratón de la etiqueta. Y también al pulsar un botón del ratón sobre la etiqueta.
8. Para hacer esto, necesitamos crear un *MouseAdapter* que contenga los siguientes eventos: *mouseEntered*, *mouseExited* y *mousePressed*. Observa que no es necesario que contenga el *mouseReleased* ni el *mouseClicked*.

Luego, el *MouseAdapter* se asignará a la etiqueta roja *etiZona*.

9. Primero programaremos el enlace del adaptador *MouseAdapter* con la etiqueta *etiZona*. Programa al final del método *CreacionVentana* lo siguiente:

```
etiZona.addMouseListener(new MouseAdapter() {
    public void mouseEntered(MouseEvent evt) {
        etiZonaMouseEntered(evt);
    }

    public void mouseExited(MouseEvent evt) {
        etiZonaMouseExited(evt);
    }

    public void mousePressed(MouseEvent evt) {
        etiZonaMousePressed(evt);
    }
});
```

10. Observa el código:

- Se añade un `MouseAdapter` al objeto `etiZona`.
- El `MouseAdapter` que se añade solo contiene los tres eventos que necesitamos: `mouseEntered`, `mouseExited` y `mousePressed`.
- Será necesario añadir *imports* para las clases `MouseAdapter` y `MouseEvent`.
- Es normal que las llamadas a los procedimientos de cada evento den error ya que todavía no han sido programadas.

11. Ahora se programará cada procedimiento que es llamado desde cada evento.

Recuerda que esto debe hacerse fuera del método *CreaciónVentana*. Aquí tienes lo que debes programar:

```
} ← (Llave del método CreacionVentana)

public void etiZonaMouseEntered(MouseEvent evt) {
    etiTexto.setText("El ratón acaba de entrar en la etiqueta");
}

public void etiZonaMouseExited(MouseEvent evt) {
    etiTexto.setText("El ratón acaba de salir de la etiqueta");
}

public void etiZonaMousePressed(MouseEvent evt) {
    etiTexto.setText("Se acaba de pulsar un botón del ratón");
}
```

Debes programar este código

12. Ahora ya puedes ejecutar el programa y comprobar lo que sucede cuando interactúas con el ratón sobre la etiqueta de color rojo.

13. Vamos a añadir una pequeña mejora que nos servirá para estudiar ese objeto llamado `evt` del tipo `MouseEvent` que llevan todos los eventos de ratón como parámetro.

Se dijo antes que este parámetro contiene información sobre la acción del ratón. Por ejemplo, nos puede servir para saber cuantas veces se pulsó el ratón, con qué botón, etc. Aprovecharemos este parámetro para indicar qué botón del ratón es el que se ha pulsado sobre la etiqueta.

Cambia el procedimiento `etiZonaMousePressed` y déjalo de la siguiente forma:

```
public void etiZonaMousePressed(MouseEvent evt) {  
  
    if (evt.getButton()==MouseEvent.BUTTON1) {  
        etiTexto.setText("Se acaba de pulsar el botón izquierdo");  
    } else if (evt.getButton()==MouseEvent.BUTTON2) {  
        etiTexto.setText("Se acaba de pulsar el botón central");  
    } else if (evt.getButton()==MouseEvent.BUTTON3) {  
        etiTexto.setText("Se acaba de pulsar el botón derecho");  
    }  
}
```

14. En el código anterior aprovechamos los métodos que tiene el objeto `MouseEvent` para averiguar que botón se ha pulsado. Usamos el método `getButton` que nos dice el botón pulsado. Aprovechamos las constantes de la clase `MouseEvent` para saber si se pulsó el botón izquierdo, central o derecho.
15. Ejecuta el programa y comprueba el funcionamiento.

CONCLUSIÓN

Hay grupos de eventos que pertenecen a un mismo oyente o adaptador. Por ejemplo, los eventos de ratón pertenecen al adaptador llamado `MouseListener`.

Si se quiere asignar eventos de ratón a un componente XXX, hay que usar el siguiente código:

```
XXX.addMouseListener(new MouseAdapter() {  
  
    public void mouseEntered(MouseEvent evt) {  
        XXXMouseEvent(evt);  
    }  
  
    ... otros eventos de ratón...  
});
```

Ten en cuenta que solo añades al adaptador aquellos eventos que necesites.

Cada evento de ratón lleva como parámetro un objeto `evt` del tipo `MouseEvent` que contiene información sobre el evento: qué botón del ratón se pulsó, cuantas veces, etc.