EJERCICIO GUIADO. JAVA: EVENTOS DESDE CÓDIGO

Eventos desde código

La ventana de diseño de NetBeans nos permite crear cada componente, colocarlo en la ventana y acceder a los eventos que necesitemos fácilmente para programar en ellos las acciones que se tengan que realizar.

Sin embargo, si el diseño de la ventana se ha realizado directamente desde código, será necesario crear también desde código los distintos eventos a usar, y asociarlos al componente correspondiente.

Para programar eventos hay que tener en cuenta lo siguiente:

- Un evento pertenece a un objeto "Oyente" (*Listener*) o a un objeto "Adaptador" (*Adapter*)

- El objeto oyente/adaptador hay que asociarlo al elemento sobre el que sucede el evento.

Por ejemplo: Programar la pulsación de un botón *btnSumar*.

La pulsación de un botón es un evento *actionPerformed* como ya se sabe.
El evento *actionPerformed* pertenece a un objeto llamado *ActionListener.* El objeto *ActionListener* es lo que se denomina un oyente.
El objeto *ActionListener* se asociará al botón btnSumar

Programación de eventos

La programación de eventos es compleja, ya que cada evento pertenece a un oyente/adaptador, y a su vez, hay que asociar cada oyente/adaptador al componente que responde al evento.

Por otro lado, la sintaxis de programación para los eventos es distinta a lo que se ha visto hasta ahora, por lo que puede resultar bastante oscura, aunque por otro lado, siempre sigue el mismo patrón.

Para simplificar el estudio de la programación de eventos, nos limitaremos a los eventos más usados, los cuales pueden clasificarse en los siguientes grupos:

- Eventos de Acción:
    - actionPerformed
        - Activar un componente (pulsación de botón, enter en un cuadro de texto)

- Eventos de Teclado:
    - keyPressed
        - Se pulsó una tecla, pero no se soltó.

    - keyReleased
        - Se soltó una tecla.
    - keyTyped
        - Se pulsó y soltó una tecla.

- Eventos de Ratón:
    - mousePressed
        - Se pulsó un botón del ratón.
    - mouseReleased
        - Se soltó un botón del ratón.
    - mousePressed
        - Se pulsó y soltó un botón del ratón.
    - mouseEntered
        - El ratón entró en la superficie del control.
    - mouseExited
        - El ratón salió de la superficie del control.

- Eventos de Ventana:

    - windowOpened
        - Se abrió la ventana
    - windowClosing
        - Se cerró la ventana
    - windowActivated
        - Se activó la ventana
    - windowDeactivated
        - Se desactivó la ventana

En esta explicación guiada nos limitaremos a estudiar los eventos de acción.

PROGRAMACIÓN DE EVENTOS DE ACCIÓN

Un evento de acción hace referencia a la activación de un objeto (un botón, un cuadro de texto, un combo, etc...)

Solo existe un tipo de evento de acción, llamado *actionPerformed*.

El evento *actionPerformed* pertenece a un objeto oyente llamado *ActionListener*.

Eventos de Acción

Eventos		***Oyente/Adaptador***
actionPerformed	Se programan dentro de...	*ActionListener*

La forma de programar el evento *actionPerformed* de un componente xxx es la siguiente:

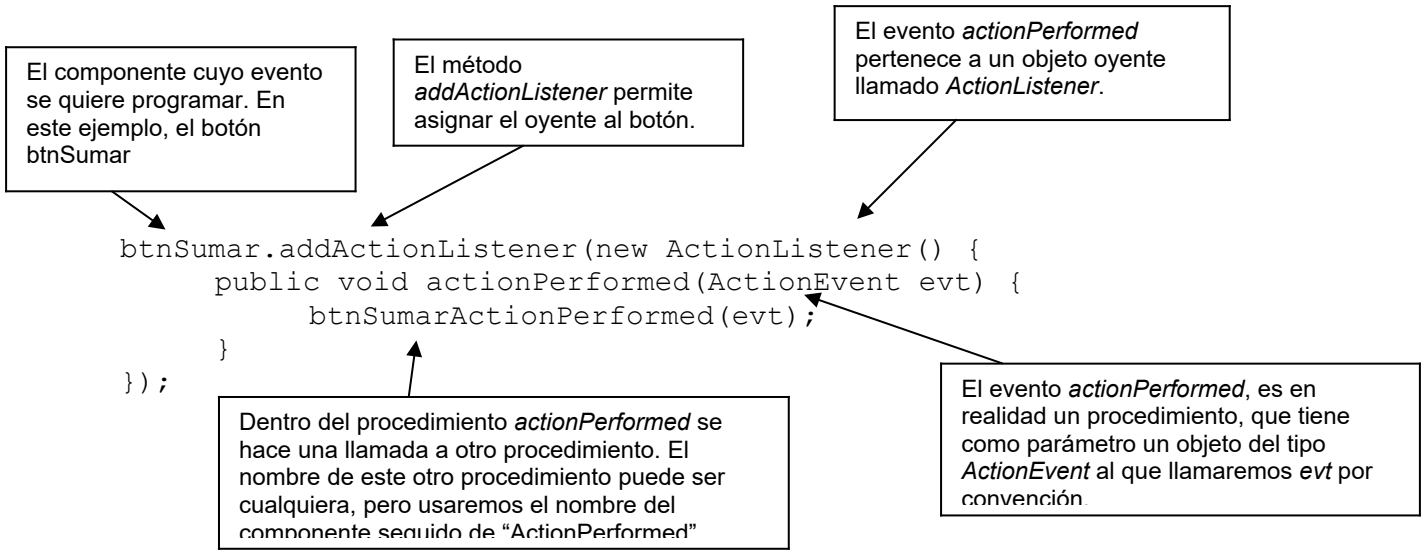
```
xxx.addActionListener(new ActionListener() {
      public void actionPerformed(ActionEvent evt) {
            xxxActionPerformed(evt);
      }
});
```

Para entender la sintaxis de la programación de un evento de acción, supongamos el siguiente ejemplo:

Se quiere programar el evento de un botón llamado btnSumar desde código. He aquí el código para crear el evento:

```
btnSumar.addActionListener(new ActionListener() {
      public void actionPerformed(ActionEvent evt) {
            btnSumarActionPerformed(evt);
      }
});
```

Una explicación del código:



El código anterior permite crear y asignar el evento *actionPerformed* al botón *btnSumar*, pero no programa el evento. Para programar el evento es necesario crear el procedimiento cuya llamada se incluye dentro del evento *actionPerformed*:

```
public void btnSumarActionPerformed(ActionEvent evt) {
       … aquí se programa el evento actionPerformed del botón btnSumar …
}
```

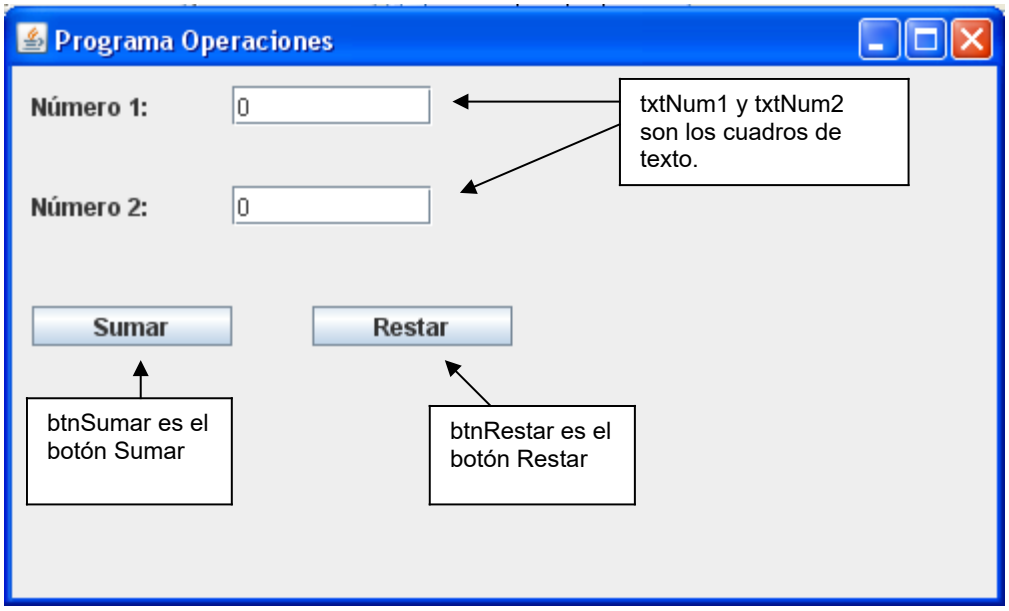
A pesar de lo complicado que resulta, hay que tener en cuenta que siempre se programa de la misma forma. Solo hay que cambiar el componente que se quiere programar y asignar un nombre a la función donde se programará el evento.

Hay que tener en cuenta que en el código expuesto antes participan nuevas clases como son ActionEvent y ActionListener, y se tendrán que agregar los import correspondientes.

EJERCICIO GUIADO

1. Abra el proyecto Operaciones que se hizo en el ejercicio guiado de la hoja anterior.

2. En el ejercicio guiado anterior se diseñó desde código la ventana de dicho proyecto. Esta ventana tiene el siguiente aspecto (Se indica también el nombre de los distintos componentes):



3. El objetivo del ejercicio es programar la pulsación del botón btnSumar para que aparezca un JOptionPane con la suma calculada. Luego haremos lo mismo con el botón btnRestar. Todo esto se hará desde código.

4. Para recordar, he aquí el código programado hasta ahora. Tenemos una llamada desde el constructor a un método *CreacionVentana* donde diseñamos cada uno de los elementos de la ventana:

```
/** Creates new form ventanaprincipal */
    public ventanaprincipal() {
        initComponents();
        CreacionVentana();
    }

    public void CreacionVentana() {

        this.setTitle("Programa Operaciones");
        this.setSize(500,300);
        this.setLocation(100,100);

        etiNum1 = new JLabel();
        etiNum1.setText("Número 1:");
        etiNum1.setBounds(10,10,100,20);
        this.getContentPane().add(etiNum1);

        etiNum2 = new JLabel();
        etiNum2.setText("Número 2:");
        etiNum2.setBounds(10,60,100,20);
        this.getContentPane().add(etiNum2);

        txtNum1 = new JTextField();
        txtNum1.setText("0");
        txtNum1.setBounds(110,10,100,20);
        this.getContentPane().add(txtNum1);

        txtNum2 = new JTextField();
        txtNum2.setText("0");
        txtNum2.setBounds(110,60,100,20);
        this.getContentPane().add(txtNum2);

        btnSumar = new JButton();
        btnSumar.setText("Sumar");
        btnSumar.setBounds(10,120,100,20);
        this.getContentPane().add(btnSumar);

        btnRestar = new JButton();
        btnRestar.setText("Restar");
        btnRestar.setBounds(150,120,100,20);
        this.getContentPane().add(btnRestar);

    }
```

En el constructor hacemos una llamada a nuestro método *CreacionVentana*

Programamos algunos detalles de la ventana principal…

Creamos la etiqueta “Número 1”…

Creamos la etiqueta “Número 2”…

Creamos el primer cuadro de texto…

Creamos el segundo…

Creamos el botón Sumar…

Creamos el botón Restar…

5. Ahora asignaremos un evento *actionPerformed* al botón *btnSumar*. Esto lo haremos al final del método *CreacionVentana*, El código que debe añadir para el evento *actionPerformed* del botón *btnSumar* es el siguiente:

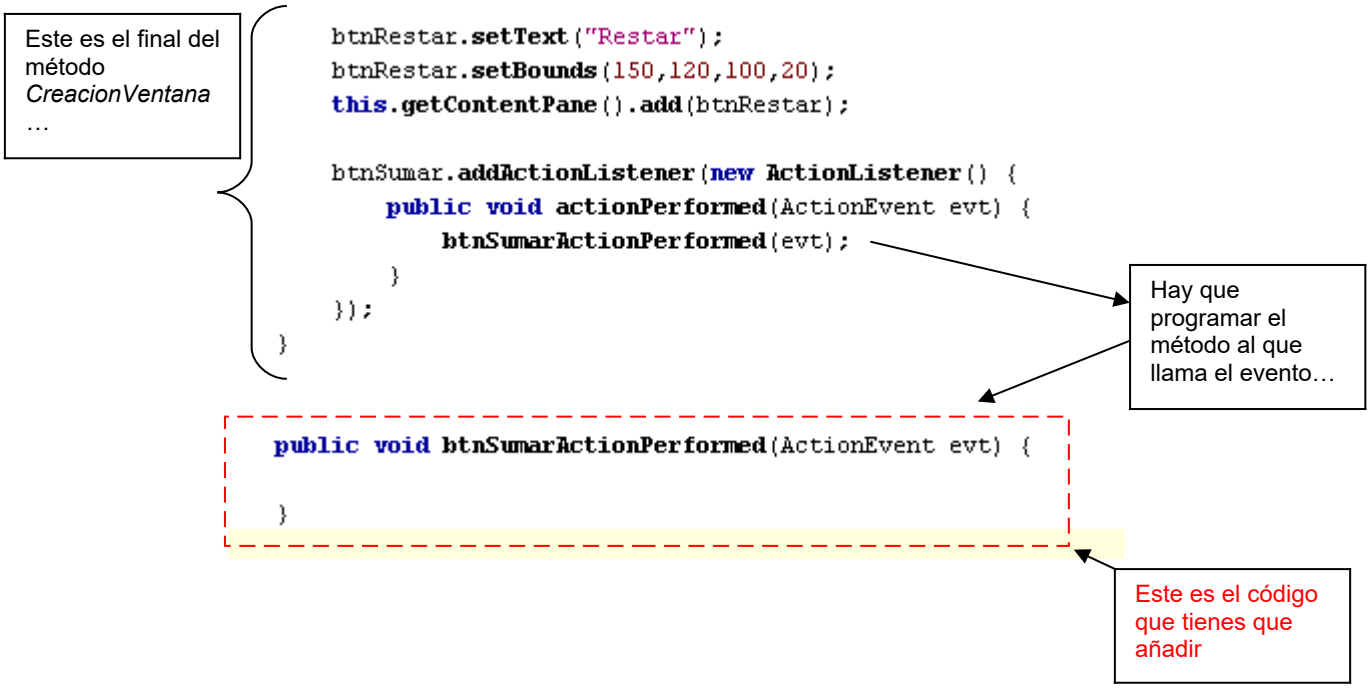
```
btnSumar.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent evt) {
        btnSumarActionPerformed(evt);
    }
});
```

(Recuerda que el código es siempre igual, solo hay que indicar el nombre del botón y el nombre del procedimiento al que se llama)

6. Será necesario añadir un import para el objeto oyente ActionListener y para la clase ActionEvent.

7. Ahora que se ha definido un evento *actionPerformed* para el botón btnSumar, será necesario programarlo. Esto se hace creando el procedimiento al que llama el código del actionPerformed. A este procedimiento le hemos dado de nombre *btnSumarActionPerformed*.

   Así pues, añade el siguiente procedimiento a la clase. (No te vayas a equivocar, el siguiente código está fuera del método C*reacionVentana*)



8. Es precisamente en este nuevo procedimiento que hemos creado donde se programa el evento *actionPerformed* del botón *btnSumar*. Lo que se pretende que haga el programa es que aparezca un JOptionPane con la suma de los números introducidos. Para ello, añade el siguiente código:

```
        this.getContentPane().add(btnRestar);

        btnSumar.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent evt) {
                btnSumarActionPerformed(evt);
            }
        });
    }

    public void btnSumarActionPerformed(ActionEvent evt) {
        double a,b,s;

        a=Double.parseDouble(txtNum1.getText());
        b=Double.parseDouble(txtNum2.getText());
        s=a+b;
        JOptionPane.showMessageDialog(null,"La suma es "+s);
    }
```

Programa el código de lo que tiene que hacer el evento.

9. Bien, hemos programado la pulsación del botón Sumar. Hemos seguido dos pasos:

    a. Primero hemos creado el evento *actionPerformed* y se lo hemos asignado al botón. Esto lo hemos hecho dentro del método *CreacionVentana*. Este evento hace una llamada al procedimiento donde se programará la respuesta al evento.

    b. Luego hemos creado ese procedimiento y hemos programado en él el código correspondiente a la pulsación del botón.

    Ejecuta el programa y comprueba el funcionamiento del botón Sumar.

10. Ahora programaremos el botón btnRestar. Primero se tendrá que definir el evento *actionPerformed* y asignárselo al botón. Esto se hace añadiendo el código de creación del evento al final del método *CreacionVentana*.

```
btnRestar.addActionListener(new ActionListener() {
      public void actionPerformed(ActionEvent evt) {
          btnRestarActionPerformed(evt);
      }
  });
```

    (Observa que es idéntico al código de asignación del evento del botón btnSumar, solo cambia el nombre del botón y el nombre del procedimiento al que se llama)

11. Y ahora, hay que programar el procedimiento al que llama el evento *actionPerformed*. Es aquí donde se programa la respuesta al evento. En nuestro ejemplo, queremos que al pulsar el botón Restar se resten los números introducidos.

```
    public void btnRestarActionPerformed(ActionEvent evt) {
            double a,b,r;

            a=Double.parseDouble(txtNum1.getText());
            b=Double.parseDouble(txtNum2.getText());
            r=a-b;
            JOptionPane.showMessageDialog(null,"La resta es "+r);
    }
```

12. Ejecuta el programa y comprueba el funcionamiento del botón Restar.

13. Resumiendo una vez más. Para programar un evento sobre un componente, primero hay que enlazar el oyente del evento con el componente, y luego programar el método al que llama el evento. Observa el código que hemos programado:

```
    btnSumar.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent evt) {
            btnSumarActionPerformed(evt);
        }
    });
```

Asignación de un *actionPerformed* al botón btnSumar.

(Se hace una llamada al método *btnSumarActionPerformed*)

```
    btnRestar.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent evt) {
            btnRestarActionPerformed(evt);
        }
    });

}
public void btnSumarActionPerformed(ActionEvent evt) {
    double a,b,s;

    a=Double.parseDouble(txtNum1.getText());
    b=Double.parseDouble(txtNum2.getText());
    s=a+b;
    JOptionPane.showMessageDialog(null,"La suma es "+s);
}
```

Asignación de un *actionPerformed* al botón btnRestar.

(Se hace una llamada al método *btnRestarActionPerformed*)

Programación del método *btnSumarActionPerformed*

(respuesta a la pulsación del botón btnSumar)

```
public void btnRestarActionPerformed(ActionEvent evt) {
    double a,b,r;

    a=Double.parseDouble(txtNum1.getText());
    b=Double.parseDouble(txtNum2.getText());
    r=a-b;
    JOptionPane.showMessageDialog(null,"La resta es "+r);
}
```

Programación del método *btnRestarActionPerformed*

(respuesta a la pulsación del botón btnRestar)

CONCLUSIÓN

Cada evento pertenece a un objeto oyente, y es el oyente el que se asigna al componente de la ventana que se quiere programar.

El evento *actionPerformed* pertenece al oyente *ActionListener*.

Para enlazar el oyente *ActionListener* a un componente XXX hay que usar el siguiente código:

```
XXX.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent evt) {
                XXXActionPerformed(evt);
        }
});
```

El enlace de un evento a un componente se hace en el constructor. Esta asignación hace una llamada a un procedimiento XXXActionPerformed, (donde XXX representa el nombre del componente que se programa) Es en este procedimiento donde realmente se programa el evento.