ACTION LISTENER (JAVA)

Generalmente, cuando se produce una acción sobre uno de los JButton o JCheckBox asociados a un ActionListener, suele ejecutarse el método public void actionPerformed (ActionEvent e){...}.

Este método estará dentro de un objeto, el cual se indicará expresamente que será el encargado de capturar el evento y para ello deberá implementar ActionListener.

La clase encargada de capturar el evento:

```
public class BtnListener implements ActionListener {
    ...
    @Override
    public void actionPerformed (ActionEvent e){
        //acciones a ejecutar cuando se pulsa el botón
        ...
    }
    ...
}
```

La clase encargada de generar el evento: (no tiene por qué implementar ni heredar de nada)

```
public class GeneradoraEventos {
    JButton j;
    BtnListener l;
    ...
    // Generalmente en el contructor de la clase.
    //Se crea el elemento que generará el evento.
    j= new JButton (" ");
    //Se crea el objeto de la clase que capturará el evento.
    l= new BtnListener ();
    //Se relaciona el evento del botón con la clase encargada
    // de gestionar el evento.
    j.addActionListener (l);

    ...
}
```

A veces, la propia clase que contiene el botón implementa la interfaz ActionListener y por lo tanto contiene el método actionPerformed. En este caso la clase encargada de generar y gestionar el evento quedaría como:

```
public class GeneradoraEventos implements ActionListener{
    JButton j;
    //Generalmente en el constructor.
    j = new JButton(" ");
    j.addActionListener(this);
                    ↳ se pasa el propio objeto, ya que es
                      él el que posee el método a ejecutar
                      cuando el evento es generado.
    ...
    @Override
    public void actionPerformed(ActionEvent e){
        //acciones a ejecutar cuando este elemento
        //produce un evento
    ...
}
```

ACTIONLISTENER-2 (JAVA)

En el momento en que se ejecuta el metodo actionBrformed es porque se ha producido un evento.

Cuando la clase que gestiona el evento, lo hace sobre más de un componente, es interesante saber quién ha generado el evento.

```
JButton jbtn;
JCheckBox jchk;
...
public void actionBrformed(ActionEvent e){
if (e.getSource()==jbtn){
   //Se está detectando si el generador del evento es el JButton.}
   ...
}
```

e.getSource: devuelve quien ha generado el evento.

e.paramString(): devuelve una serie de parámetros:

ACTION PERFORMED, cmd = Ctrl-chck, when = . . .

↑ nombre del JCheckBox cuando hacemos new JCheckBox ("Ctrl-chck");

De esta manera, si la clase que gestiona el evento no es la propia que lo genera, para evitar tener un array de componentes y buscar qué componente ha generado el evento se puede hacer lo siguiente.

```
if(e.paramString().regionMatches(21,"Añadir fila",int,int))
```

↑ cogemos a partir de la posición 21 que es donde empieza el nombre del componente, y en función de su nombre así haremos.