Práctica 2: Interfaz Grafica - Eventos

Objetivo:

- Conocer y utilizar los eventos en componentes de la interfaz grafica.

Tiempo destinado: 2hrs.

Fundamentos:

En Java, cada evento está representado por un objeto que es una subclase de la clase **EventObject** en el paquete **java.util**. Cada subclase de **EventObject** representa un tipo de evento particular. Por ejemplo:

- **MouseEvent**, para representar acciones del ratón: mover, arrastrar, hacer clic en un botón del ratón, etc.
- **KeyEvent**, para representar acciones del teclado, esto es, pulsar teclas.
- **ActionEvent**, para representar un acción del usuario en la interfaz, por ejemplo, pulsar un botón en la pantalla.

No obstante, en el modelo de eventos de Java, los **EventObject** no realizan acciones por si mismos, sino que esos eventos son enviados a otro objeto encargado de responder a un tipo de evento en particular. Estos objetos son los que conocemos como **listeners** o "escuchadores", y existen diferentes **listeners** que "escuchan" a los diferentes **eventos**, tales como los **mouse listeners**, **key listeners** o **action listeners**.

Los **listener** no se implementan como clases en java, sino como **interfaces**. Un interface sería entonces una colección de métodos que definen un comportamiento en particular. De este modo, cualquier clase que suministre información para dichos métodos puede declarar que implementa a dicho interface.

La gran ventaja de implementar interfaces frente a heredar de otras clases es que una misma clase puede implementar varios interfaces simultáneamente pero solo puede heredar de una.

Los listeners que usaremos estarán en el paquete java.awt.event (que debemos importar antes de implementarlos) y nos podemos encontrar los siguientes:

Nombre: ActionListener
Descripción: Se produce al hacer click en un componente, también si se pulsa Enter teniendo el foco en el componente.
Método:
public void actionPerformed(ActionEvent e)
Eventos:
JButton: click o pulsar Enter con el foco activado en él.
JList: doble click en un elemento de la lista.
JMenuItem: selecciona una opción del menú.
JTextField: al pulsar Enter con el foco activado.

Nombre: KeyListener
Descripción: Se produce al pulsar una tecla. según el método cambiara la forma de pulsar la tecla.
Métodos:
public void keyTyped(KeyEvent e)
public void keyPressed(KeyEvent e)
public void keyReleased(KeyEvent e)
Eventos: Cuando pulsamos una tecla, según el listener:
keyTyped: al pulsar y soltar la tecla.
keyPressed : al pulsar la tecla.
keyReleased : al soltar la tecla.

Nombre: FocusListener
Descripción: Se produce cuando un componente gana o pierde el foco, es decir, que esta seleccionado.
Métodos: Recibir o perder el foco.

Nombre: MouseListener
Descripción: Se produce cuando realizamos una acción con el ratón.
Métodos:
public void mouseClicked(MouseEvent e)
public void mouseEntered(MouseEvent e)
public void mouseExited(MouseEvent e)
public void mousePressed(MouseEvent e)
public void mouseReleased(MouseEvent e)
Eventos: Según el listener:
mouseClicked: pinchar y soltar.
mouseEntered: entrar en un componente con el puntero.
mouseExited: salir de un componente con el puntero
mousePressed: presionar el botón.
mouseReleased: soltar el botón.

Nombre: MouseMotionListener
Descripción: Se produce con el movimiento del mouse.
Métodos:
public void mouseDragged(MouseEvent e)
public void mouseMoved(MouseEvent e)
Eventos: Según el listener:
mouseDragged: click y arrastrar un componente.
mouseMoved: al mover el puntero sobre un elemento.

De esta forma, los listener se pueden invocar de esta manera:

```
import java.awt.event.*;

boton1.addActionListener(new ActionListener(){
  public void actionPerformed (){
  //Acciones
  }
});
```

o de esta otra:

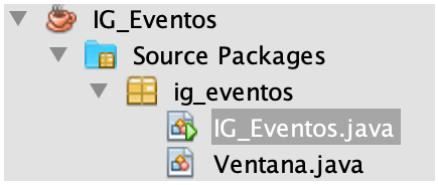
```
import java.awt.event.*;

ActionListener al=new ActionListener(){
  public void actionPerformed (){
  //Acciones
  }
};
boton1.addActionListener(al);
```

Instrucciones:

1. Realice correctamente cada paso en la sección de desarrollo, documente la práctica con capturas de pantalla, mostrando cada paso realizado y los resultados obtenidos.
2. Suba su reporte al espacio asignado en el aula virtual.

Desarrollo:

1. Genere un nuevo proyecto de Java en NetBeans, llamado IG_Eventos con la siguiente estructura de clases:



   a. Dentro de la clase Ventana capture el siguiente código:

```
package ig_eventos;

import java.awt.event.*;
import javax.swing.*;
public class Ventana extends JFrame implements ActionListener {

    JButton boton1;

    public Ventana() {
        super("Ventana usando Eventos");
        //Layout absoluto
        setLayout(null);
        setSize(300, 300);

        //No redimensionable
        setResizable(false);

        //Cerrar proceso al cerrar la ventana
        setDefaultCloseOperation(EXIT_ON_CLOSE);
```

```
        //Botón
        boton1 = new JButton("Finalizar");
        boton1.setBounds(100, 100, 100, 30);
        add(boton1);
        boton1.addActionListener(this);

        //Muestro JFrame (lo último para que lo pinte todo correctamanete)
        setVisible(true);

    }

    public void actionPerformed(ActionEvent e) {
        if (e.getSource() == boton1) {
            System.exit(0);
        }

    }

}
```

b. En la función main de la clase IG_Eventos, creamos un objeto de la clase Ventana para visualizar el resultado:

```
package ig_eventos;

/**
 *
 * @author jjpg
 */
public class IG_Eventos {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // TODO code application logic here
        new Ventana();
    }

}
```

2. Ejecute su aplicación, deberá de observar una ventana que contiene un boton en el JFrame (JFrame pertenece a swing y es una mejora de Frame). Presione el botón y pruebe el cerrar la ventana con el botón del JFrame (busque los botones de minimizar, maximizar y cerrar)

3. Ahora dentro del mismo proyecto genere otra clase llamada Ventana2 con el siguiente código:

```
package ig_eventos;


import java.awt.event.*;
import javax.swing.*;
```

```
public class Ventana2 extends JFrame implements ActionListener{

    private JButton boton1, boton2, boton3;

    public Ventana2() {

        //Layout absoluto
        setLayout(null);

        //Tamaño de la ventana
        setBounds(0, 0, 350, 200);

        //Título
        setTitle("Ejemplo 2: Botones");

        //No redimensionable
        setResizable(false);

        //Cerrar proceso al cerrar la ventana
        setDefaultCloseOperation(EXIT_ON_CLOSE);
        //Botones
        boton1 = new JButton("1");
        boton1.setBounds(10, 100, 90, 30);
        add(boton1);
        boton1.addActionListener(this);
        boton2 = new JButton("2");
        boton2.setBounds(110, 100, 90, 30);
        add(boton2);
        boton2.addActionListener(this);
        boton3 = new JButton("3");
        boton3.setBounds(210, 100, 90, 30);
        add(boton3);
        boton3.addActionListener(this);

        //Muestro JFrame (lo último para que lo pinte todo correctamanete)
        setVisible(true);
    }

    public void actionPerformed(ActionEvent e) {
        if (e.getSource() == boton1) {
            setTitle("boton 1");
        }
        if (e.getSource() == boton2) {
            setTitle("boton 2");
        }
        if (e.getSource() == boton3) {
            setTitle("boton 3");
        }
    }
}
```

Modifique la funcion main, para crear un objeto de ventana2:

```
public static void main(String[] args) {
    // TODO code application logic here
    new Ventana2();
}
```

4. Pruebe el funcionamiento.

5. Analice cada código y documente sus observaciones y conclusiones.