

37 - Swing - JMenuBar, JMenu, JMenuItem [Listado completo de tutoriales](#)

Ahora veremos como crear un menú de opciones y la captura de eventos de los mismos.

Cuando necesitamos implementar un menú horizontal en la parte superior de un JFrame requerimos de un objeto de la clase JMenuBar, uno o más objetos de la clase JMenu y por último objetos de la clase JMenuItem.

Par la captura de eventos debemos implementar la interface ActionListener y asociarlo a los controles de tipo JMenuItem, el mismo se dispara al presionar con el mouse el JMenuItem.

Problema 1:

Confeccionaremos un menú de opciones que contenga tres opciones que permita cambiar el color de fondo del JFrame a los colores: rojo, verde y azul.



Programa:

[Ver video](#)

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
public class Formulario extends JFrame implements Act
    private JMenuBar mb;
    private JMenu menu1;
    private JMenuItem mi1,mi2,mi3;
    public Formulario() {
        setLayout(null);
        mb=new JMenuBar();
        setJMenuBar(mb);
```

```

        menu1=new JMenu("Opciones");
        mb.add(menu1);
        mi1=new JMenuItem("Rojo");
        mi1.addActionListener(this);
        menu1.add(mi1);
        mi2=new JMenuItem("Verde");
        mi2.addActionListener(this);
        menu1.add(mi2);
        mi3=new JMenuItem("Azul");
        mi3.addActionListener(this);
        menu1.add(mi3);
    }

    public void actionPerformed(ActionEvent e) {
        Container f=this.getContentPane();
        if (e.getSource()==mi1) {
            f.setBackground(new Color(255,0,0));
        }
        if (e.getSource()==mi2) {
            f.setBackground(new Color(0,255,0));
        }
        if (e.getSource()==mi3) {
            f.setBackground(new Color(0,0,255));
        }
    }

    public static void main(String[] ar) {
        Formulario formulario1=new Formulario();
        formulario1.setBounds(10,20,300,200);
        formulario1.setVisible(true);
        formulario1.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }
}

```

Importamos el paquete javax.swing ya que en el mismo se encuentran las tres clases JMenuBar, JMenu y JMenuItem:

```
import javax.swing.*;
```

Importamos java.awt donde se encuentra la clase Color:

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

Para la captura de eventos mediante la interface ActionListener debemos importar el paquete java.awt.event:

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

Declaramos la clase Formulario, heredamos de la clase JFrame e indicamos que implementaremos la interface ActionListener:

```
public class Formulario extends JFrame implements ActionListener{
```

Definimos un objeto de la clase JMenuBar (no importa que tan grande sea un menú de opciones solo se necesitará un solo objeto de esta clase):

```
    private JMenuBar mb;
```

Definimos un objeto de la clase JMenu (esta clase tiene por objeto desplegar un conjunto de objetos de tipo JMenuItem u otros objetos de tipo JMenu:

```
    private JMenu menu1;
```

Definimos tres objetos de la clase JMenuItem (estos son los que disparan eventos cuando el operador los selecciona:

```
    private JMenuItem mi1,mi2,mi3;
```

En el constructor creamos primero el objeto de la clase JMenuBar y lo asociamos al JFrame llamando al método setJMenuBar:

```
        mb=new JMenuBar();  
        setJMenuBar(mb);
```

Seguidamente creamos un objeto de la clase JMenu, en el constructor pasamos el String que debe mostrar y asociamos dicho JMenu con el JMenuBar llamando al método add de objeto de tipo JMenuBar (Es decir el objeto de la clase JMenu colabora con la clase JMenuBar):

```
        menu1=new JMenu("Opciones");  
        mb.add(menu1);
```

Ahora comenzamos a crear los objetos de la clase JMenuItem y los añadimos al objeto de la clase JMenu (también mediante la llamada al método addActionListener indicamos al JMenuItem que objeto procesará el clic):

```
        mi1=new JMenuItem("Rojo");  
        mi1.addActionListener(this);
```

```
menu1.add(mi1);
```

Lo mismo hacemos para los otros dos JMenuItem:

```
mi2=new JMenuItem("Verde");  
mi2.addActionListener(this);  
menu1.add(mi2);  
mi3=new JMenuItem("Azul");  
mi3.addActionListener(this);  
menu1.add(mi3);
```

En el método actionPerformed primero obtenemos la referencia al panel asociado con el JFrame:

```
public void actionPerformed(ActionEvent e) {  
    Container f=this.getContentPane();
```

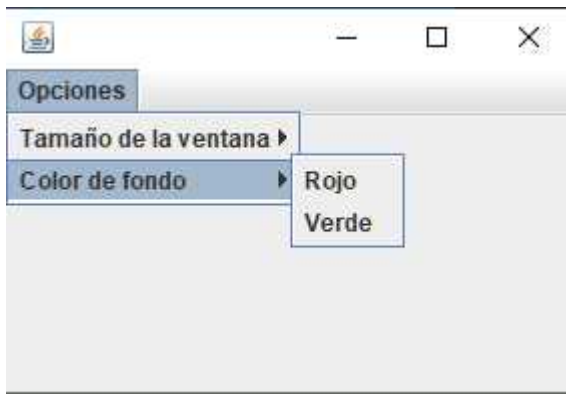
Luego mediante if verificamos cual de los tres JMenuItem fue seleccionado y a partir de esto llamamos al método setBackground del objeto de la clase Container):

```
    if (e.getSource()==mi1) {  
        f.setBackground(new Color(255,0,0));  
    }  
    if (e.getSource()==mi2) {  
        f.setBackground(new Color(0,255,0));  
    }  
    if (e.getSource()==mi3) {  
        f.setBackground(new Color(0,0,255));  
    }  
}
```

Problema 2:

Confeccionaremos un menú de opciones que contenga además del JMenu de la barra otros dos objetos de la clase JMenu que dependan del primero.

Uno debe mostrar dos JMenuItem que permitan modificar el tamaño del JFrame y el segundo también debe mostrar dos JMenuItem que permitan cambiar el color de fondo.



Programa:

[Ver video](#)

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
public class Formulario extends JFrame implements ActionListener {
    private JMenuBar mb;
    private JMenu menu1, menu2, menu3;
    private JMenuItem mi1, mi2, mi3, mi4;
    public Formulario() {
        setLayout(null);
        mb = new JMenuBar();
        setJMenuBar(mb);
        menu1 = new JMenu("Opciones");
        mb.add(menu1);
        menu2 = new JMenu("Tamaño de la ventana");
        menu1.add(menu2);
        menu3 = new JMenu("Color de fondo");
        menu1.add(menu3);
        mi1 = new JMenuItem("640*480");
        menu2.add(mi1);
        mi1.addActionListener(this);
        mi2 = new JMenuItem("1024*768");
        menu2.add(mi2);
        mi2.addActionListener(this);
        mi3 = new JMenuItem("Rojo");
        menu3.add(mi3);
        mi3.addActionListener(this);
        mi4 = new JMenuItem("Verde");
        menu3.add(mi4);
        mi4.addActionListener(this);
    }

    public void actionPerformed(ActionEvent e) {
```

```

        if (e.getSource()==mi1) {
            setSize(640,480);
        }
        if (e.getSource()==mi2) {
            setSize(1024,768);
        }
        if (e.getSource()==mi3) {
            getContentPane().setBackground(new Color(
        }
        if (e.getSource()==mi4) {
            getContentPane().setBackground(new Color(
        }
    }

    public static void main(String[] ar) {
        Formulario formulario1=new Formulario();
        formulario1.setBounds(0,0,300,200);
        formulario1.setVisible(true);
        formulario1.setDefaultCloseOperation(JFrame.E
    }

```

Definimos un objeto de la clase JMenuBar, 3 objetos de la clase JMenu y finalmente 4 objetos de la clase JMenuItem:

```

private JMenuBar mb;
private JMenu menu1,menu2,menu3;
private JMenuItem mi1,mi2,mi3,mi4;

```

Es importante notar el orden de creación de los objetos y como los relacionamos unos con otros.

Primero creamos el JMenuBar y lo asociamos con el JFrame:

```

mb=new JMenuBar();
setJMenuBar(mb);

```

Creamos el primer JMenu y lo pasamos como parámetro al JMenuBar mediante el método add:

```

menu1=new JMenu("Opciones");
mb.add(menu1);

```

Ahora creamos el segundo objeto de la clase JMenu y lo asociamos con el primer JMenu creado:

```
menu2=new JMenu("Tamaño de la ventana");  
menu1.add(menu2);
```

En forma similar creamos el tercer objeto de la clase JMenu y lo asociamos con el primer JMenu creado:

```
menu3=new JMenu("Color de fondo");  
menu1.add(menu3);
```

Finalmente comenzamos a crear los objetos de la clase JMenuItem y los dos primeros los asociamos con el segundo JMenu:

```
mi1=new JMenuItem("640*480");  
menu2.add(mi1);  
mi1.addActionListener(this);  
mi2=new JMenuItem("1024*768");  
menu2.add(mi2);  
mi2.addActionListener(this);
```

También hacemos lo mismo con los otros dos objetos de tipo JMenuItem pero ahora los asociamos con el tercer JMenu:

```
mi3=new JMenuItem("Rojo");  
menu3.add(mi3);  
mi3.addActionListener(this);  
mi4=new JMenuItem("Verde");  
menu3.add(mi4);  
mi4.addActionListener(this);
```

En el método actionPerformed si se presiona el mi1 procedemos a redimensionar el JFrame llamando al método setSize y le pasamos dos parámetros que representan el nuevo ancho y alto de la ventana:

```
if (e.getSource()==mi1) {  
    setSize(640,480);  
}
```

De forma similar si se presiona el segundo JMenuItem cambiamos el tamaño de la ventana a 1024 píxeles por 768:

```
if (e.getSource()==mi2) {  
    setSize(1024,768);  
}
```

Para cambiar de color de forma similar al problema anterior mediante el método getContentPane obtenemos la referencia al objeto de la clase Container y llamamos al método setBackground para fijar un nuevo color de fondo:

```
if (e.getSource()==mi3) {  
    getContentPane().setBackground(new Color(255,0,0));  
}  
if (e.getSource()==mi4) {  
    getContentPane().setBackground(new Color(0,255,0));  
}
```

Problema propuesto

1. Mediante dos controles de tipo JTextField permitir el ingreso de dos números. Crear un menú que contenga una opción que redimensione el JFrame con los valores ingresados por teclado. Finalmente disponer otra opción que finalice el programa (Finalizamos un programa java llamando al método exit de la clase System: System.exit(0))

[Ver video](#)

[Solución](#)

Retornar