

Ejercicios Tema 5: Eventos

Ejercicios Resueltos



Introducción

En el anterior tema solamente se hicieron los ejercicios atendiendo al diseño. Ahora que ya se han visto los Eventos vamos a darle funcionalidad a las aplicaciones. En **rojo** se identifica lo nuevo que hay que hacer con respecto al mismo ejercicio del tema anterior.

Ejercicio 1: Realizar un programa para convertir de grados Celsius a grados Fahrenheit. Necesitaremos un par de Etiquetas, un par de Cuadros de Texto y un par de Botones. **El resultado se mostrará en el TextField correspondiente.**



```
package es.studium.termometro;

import java.awt.Button;
import java.awt.Frame;
import java.awt.GridLayout;
import java.awt.Label;
import java.awt.TextField;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.WindowEvent;
import java.awt.event.WindowListener;

public class termometro extends Frame implements WindowListener,
ActionListener
{
    private static final long serialVersionUID = 1L;
    Label etiqueta = new Label("Celsius");
    TextField celsius = new TextField(5);
    Label etiqueta2 = new Label("Fahrenheit");
    TextField fahrenheit = new TextField(5);
    Button btnCelsius = new Button(" Celsius a Fahrenheit ");
    Button btnFahrenheit = new Button(" Fahrenheit a Celsius ");

    public termometro()
    {
        setLayout(new GridLayout(3,2));
        setTitle("Conversión de temperaturas");
        add(etiqueta);
        add(celsius);
        add(etiqueta2);
        add(fahrenheit);
        add(btnCelsius);
        add(btnFahrenheit);
        // Añadir Listener a los botones
        btnCelsius.addActionListener(this);
        btnFahrenheit.addActionListener(this);
    }
}
```

```

        //Añadir el Listener a la ventana
        addWindowListener(this);
        setSize(450,150);
        setVisible(true);
    }
    public static void main(String[] args)
    {
        new termometro();
    }
    public void windowActivated(WindowEvent we) {}
    public void windowClosed(WindowEvent we) {}
    public void windowClosing(WindowEvent we)
    {
        //Llamamos al método exit de la clase System,
        //devolviendo como código de salida un 0
        System.exit(0);
    }
    public void windowDeactivated(WindowEvent we) {}
    public void windowDeiconified(WindowEvent we) {}
    public void windowIconified(WindowEvent we) {}
    public void windowOpened(WindowEvent we) {}
    public void actionPerformed(ActionEvent we)
    {
        //Primero averiguar qué botón generó este evento
        Object boton = we.getSource();
        float resultado;
        if(boton==btnCelsius)
        {
            resultado = (float)
            ((Float.parseFloat(Celsius.getText())*9.0/5.0) + 32.0);
            Fahrenheit.setText(Float.toString(resultado));
        }
        else
        {
            resultado =
            (float) (Float.parseFloat(Fahrenheit.getText())-32.0)*(float) (5.0/9.0);
            Celsius.setText(Float.toString(resultado));
        }
    }
}

```

Ejercicio 2: Realizar una aplicación que muestre por pantalla una serie de 10 aficiones con campos de selección múltiple. **Al pulsar en un botón, se mostrará por la consola las opciones elegidas.**



```

package es.studium.aficiones;

import java.awt.Button;
import java.awt.Checkbox;
import java.awt.FlowLayout;
import java.awt.Frame;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.WindowEvent;

```

```
import java.awt.event.WindowListener;

public class aficiones extends Frame implements WindowListener,
ActionListener
{
    private static final long serialVersionUID = 1L;
    Checkbox uno = new Checkbox("Correr");
    Checkbox dos = new Checkbox("Nadar");
    Checkbox tres = new Checkbox("Andar");
    Checkbox cuatro = new Checkbox("Leer");
    Checkbox cinco = new Checkbox("Ir al Cine");
    Checkbox seis = new Checkbox("Bailar");
    Checkbox siete = new Checkbox("Fútbol");
    Checkbox ocho = new Checkbox("Tenis");
    Checkbox nueve = new Checkbox("Baloncesto");
    Checkbox diez = new Checkbox("Deportes de Vela");
    Button b = new Button("Comprobar");
    String resultado = new String("");

    public aficiones()
    {
        setLayout(new FlowLayout());
        setTitle("Aficiones");
        add(uno);
        add(dos);
        add(tres);
        add(cuatro);
        add(cinco);
        add(seis);
        add(siete);
        add(ocho);
        add(nueve);
        add(diez);
        add(b);
        // Añadir el Listener al botón
        b.addActionListener(this);
        //Añadir el Listener a la ventana
        addWindowListener(this);
        setSize(350,150);
        setVisible(true);
    }
    public static void main(String[] args)
    {
        new aficiones();
    }
    public void windowActivated(WindowEvent we) {}
    public void windowClosed(WindowEvent we) {}
    public void windowClosing(WindowEvent we)
    {
        System.exit(0);
    }
    public void windowDeactivated(WindowEvent we) {}
    public void windowDeiconified(WindowEvent we) {}
    public void windowIconified(WindowEvent we) {}
    public void windowOpened(WindowEvent we) {}
}
```

```

public void actionPerformed(ActionEvent we)
{
    resultado = "";
    if(unos.getState()==true)
    {
        resultado = resultado + "Correr\n";
    }
    if(dos.getState()==true)
    {
        resultado = resultado + "Nadar\n";
    }
    if(tres.getState()==true)
    {
        resultado = resultado + "Andar\n";
    }
    if(cuatro.getState()==true)
    {
        resultado = resultado + "Leer\n";
    }
    if(cinco.getState()==true)
    {
        resultado = resultado + "Ir al Cine\n";
    }
    if(seis.getState()==true)
    {
        resultado = resultado + "Bailar\n";
    }
    if(siete.getState()==true)
    {
        resultado = resultado + "Fútbol\n";
    }
    if(ocho.getState()==true)
    {
        resultado = resultado + "Tenis\n";
    }
    if(nueve.getState()==true)
    {
        resultado = resultado + "Baloncesto\n";
    }
    if(diez.getState()==true)
    {
        resultado = resultado + "Deportes de vela\n";
    }
    System.out.println(resultado);
}

```

Ejercicio 3: Realizar un programa que muestre por pantalla una serie de preguntas sobre vehículos. Serán campos de selección única. En primer lugar se preguntará el tipo de motorización del vehículo entre Diésel, Gasolina, Híbrido o Eléctrico. A continuación preguntará por el número de puertas entre 3, 4 ó 5. Por último preguntará si la pintura es metalizada o no. **Una vez seleccionados todos los elementos, se pulsará un botón y saldrá un presupuesto atendiendo a los siguientes elementos:**

- Diésel: 8.000 €
- Gasolina: 7.000 €
- Híbrido: 9.000 €
- Eléctrico: 8.500 €
- 3 puertas: 2.000 €
- 4 puertas: 3.000 €
- 5 puertas: 2.500 €
- Pintura normal: (Incluido en los precios anteriores)
- Pintura metalizada: 1.500 €

```
package es.studium.vehiculos;

import java.awt.Button;
import java.awt.Checkbox;
import java.awt.CheckboxGroup;
import java.awt.FlowLayout;
import java.awt.Frame;
import java.awt.Label;
import java.awt.Panel;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.WindowEvent;
import java.awt.event.WindowListener;

public class vehiculos extends Frame implements WindowListener,
ActionListener
{
    private static final long serialVersionUID = 1L;
    Panel panel1 = new Panel();
    Panel panel2 = new Panel();
    Panel panel3 = new Panel();

    Label etiqueta = new Label ("¿Tipo de motorización?");
    CheckboxGroup grupoMotor = new CheckboxGroup();
    Checkbox uno = new Checkbox("Gasolina", false, grupoMotor);
    Checkbox dos = new Checkbox("Diésel", false, grupoMotor);
    Checkbox tres = new Checkbox("Híbrido", false, grupoMotor);
    Checkbox cuatro = new Checkbox("Eléctrico", false, grupoMotor);
    Label etiqueta2 = new Label("¿Número de puertas?");
    CheckboxGroup grupoPuertas = new CheckboxGroup();
    Checkbox cinco = new Checkbox("3 puertas", false, grupoPuertas);
    Checkbox seis = new Checkbox("4 puertas", false, grupoPuertas);
    Checkbox siete = new Checkbox("5 puertas", false, grupoPuertas);
    Label etiqueta3 = new Label("¿Pintura Metalizada?");
    CheckboxGroup grupoMetal = new CheckboxGroup();
    Checkbox ocho = new Checkbox("Si", false, grupoMetal);
    Checkbox nueve = new Checkbox("No", false, grupoMetal);
    Button calcular = new Button("Calcular presupuesto");
```



```

public vehiculos()
{
    setLayout(new FlowLayout());
    setTitle("Vehículos");

    panel1.add(etiqueta);
    panel1.add(unos);
    panel1.add(dos);
    panel1.add(tres);
    panel1.add(cuatro);
    add(panel1);
    panel2.add(etiqueta2);
    panel2.add(cinco);
    panel2.add(seis);
    panel2.add(siete);
    add(panel2);
    panel3.add(etiqueta3);
    panel3.add(ocho);
    panel3.add(nueve);
    add(panel3);
    add(calcular);
    //Añadir el Listener a la ventana
    addWindowListener(this);
    //Añadir Listener al botón
    calcular.addActionListener(this);
    setSize(450,250);
    setVisible(true);
}

public static void main(String[] args)
{
    new vehiculos();
}

public void windowActivated(WindowEvent we) {}
public void windowClosed(WindowEvent we) {}
public void windowClosing(WindowEvent we)
{
    System.exit(0);
}

public void windowDeactivated(WindowEvent we) {}
public void windowDeiconified(WindowEvent we) {}
public void windowIconified(WindowEvent we) {}
public void windowOpened(WindowEvent we) {}
public void actionPerformed(ActionEvent ae)
{
    int presupuesto=0;
    if (unos.getState()==true)
    {
        presupuesto = presupuesto + 7000;
    }
    if (dos.getState()==true)
    {
        presupuesto = presupuesto + 8000;
    }
    if (tres.getState()==true)
    {
        presupuesto = presupuesto + 9000;
    }
}

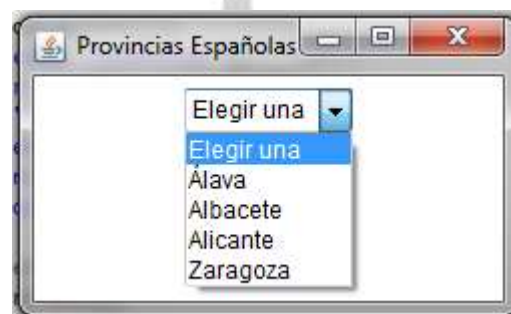
```

```

    if(cuatro.getState()==true)
    {
        presupuesto = presupuesto + 8500;
    }
    if(cinco.getState()==true)
    {
        presupuesto = presupuesto + 2000;
    }
    if(seis.getState()==true)
    {
        presupuesto = presupuesto + 3000;
    }
    if(siete.getState()==true)
    {
        presupuesto = presupuesto + 2500;
    }
    if(ocho.getState()==true)
    {
        presupuesto = presupuesto + 1500;
    }
    System.out.println(presupuesto+" €");
}

```

Ejercicio 4: Se pide diseñar un interfaz gráfico de usuario para poder elegir de una lista desplegable, las provincias españolas. **Una vez elegida, se nos mostrará el gentilicio de dicha comunidad, en masculino y femenino.**



```

package es.studium.provincias;

import java.awt.*;
import java.awt.event.ItemEvent;
import java.awt.event.ItemListener;
import java.awt.event.WindowEvent;
import java.awt.event.WindowListener;

public class provincias extends Frame implements WindowListener,
ItemListener
{
    private static final long serialVersionUID = 1L;
    Choice lista = new Choice();
    public provincias()
    {
        setLayout(new FlowLayout());
        setTitle("Provincias Españolas");
        lista.add("Elegir una");
        lista.add("Álava");
        lista.add("Albacete");
        lista.add("Alicante");
        ...
        lista.add("Zaragoza");
        add(lista);
        //Añadir el Listener a la ventana
        addWindowListener(this);
        lista.addItemListener(this);
        setSize(250,150);
        setVisible(true);
    }
}

```



```

    }
    public static void main(String[] args)
    {
        new provincias();
    }
    public void windowActivated(WindowEvent we) {}
    public void windowClosed(WindowEvent we) {}
    public void windowClosing(WindowEvent we)
    {
        System.exit(0);
    }
    public void windowDeactivated(WindowEvent we) {}
    public void windowDeiconified(WindowEvent we) {}
    public void windowIconified(WindowEvent we) {}
    public void windowOpened(WindowEvent we) {}
    public void itemStateChanged(ItemEvent ie)
    {
        String gentilicio = new String();
        if("Álava".equals(ie.getItem()))
        {
            gentilicio = "Alavés y Alavesa";
        }
        if("Albacete".equals(ie.getItem()))
        {
            gentilicio = "Albaceteño y Albaceteña";
        }
        if("Alicante".equals(ie.getItem()))
        {
            gentilicio = "Alicantino y Alicantina";
        }
        ...
        if("Zaragoza".equals(ie.getItem()))
        {
            gentilicio = "Zaragozano y Zaragozana";
        }
        System.out.println(gentilicio);
    }
}

```

Ejercicio 5: Realizar una aplicación que muestre por pantalla una lista con los equipos de la liga española de baloncesto. **Una vez elegido, indicar la ciudad a la que pertenece dicho equipo.**

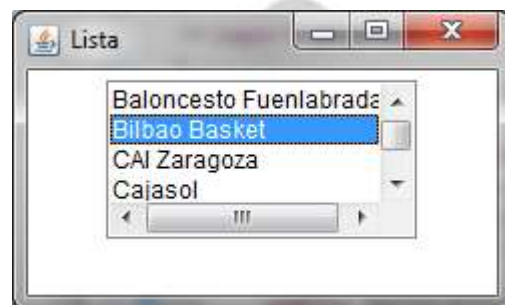
```

package es.studium.baloncesto;

import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.WindowEvent;
import java.awt.event.WindowListener;

public class baloncesto extends Frame implements WindowListener,
ActionListener
{
    private static final long serialVersionUID = 1L;
    //Creación de una lista con 5 elementos visibles y que no
    //permite multiselección
    List lista = new List(5,false);
}

```



```

public baloncesto()
{
    setLayout(new FlowLayout());
    setTitle("Lista");
    lista.add("Baloncesto Fuenlabrada");
    lista.add("Bilbao Basket");
    lista.add("CAI Zaragoza");
    lista.add("Cajasol");
    lista.add("Valencia Basket Club");
    add(lista);

    addWindowListener(this);
    lista.addActionListener(this);
    setSize(250,250);
    setVisible(true);
}

public static void main(String[] args)
{
    new baloncesto();
}

public void windowActivated(WindowEvent we) {}
public void windowClosed(WindowEvent we) {}
public void windowClosing(WindowEvent we)
{
    System.exit(0);
}

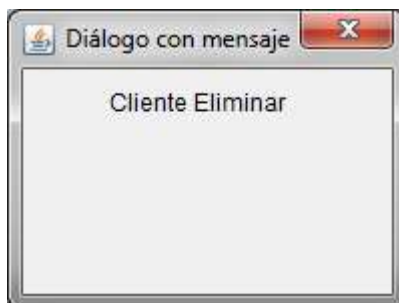
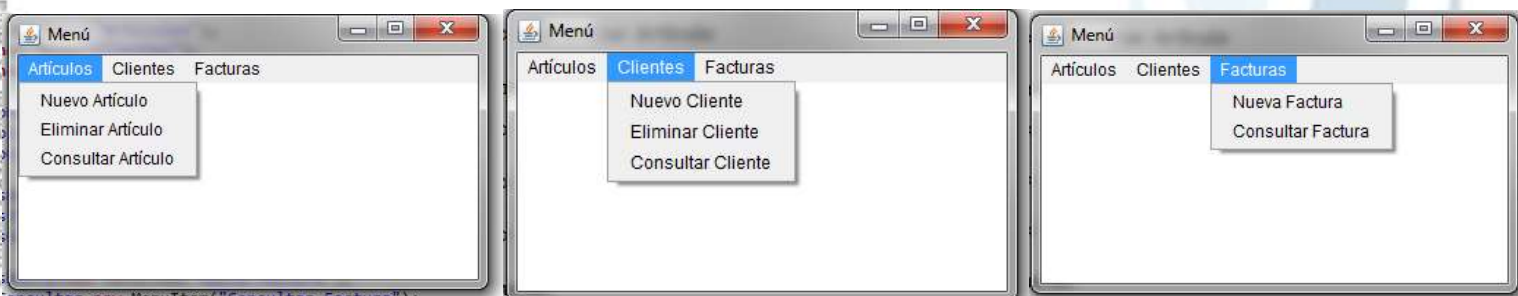
public void windowDeactivated(WindowEvent we) {}
public void windowDeiconified(WindowEvent we) {}
public void windowIconified(WindowEvent we) {}
public void windowOpened(WindowEvent we) {}
public void actionPerformed(ActionEvent ae)
{
    String ciudad = new String();
    if("Baloncesto
Fuenlabrada".equals(lista.getSelectedItem()))
    {
        ciudad = "Madrid";
    }
    if("Bilbao Basket".equals(lista.getSelectedItem()))
    {
        ciudad = "Bilbao";
    }
    if("CAI Zaragoza".equals(lista.getSelectedItem()))
    {
        ciudad = "Zaragoza";
    }
    if("Cajasol".equals(lista.getSelectedItem()))
    {
        ciudad = "Sevilla";
    }
    if("Valencia Basket Club".equals(lista.getSelectedItem()))
    {
        ciudad = "Valencia";
    }
    System.out.println(ciudad);
}
}

```

Ejercicio 6: Diseñar un programa con un menú. El menú tendrá las siguientes opciones y subopciones:

- Artículos:
 - Nuevo Artículo
 - Eliminar Artículo
 - Consultar Artículo
- Clientes:
 - Nuevo Cliente
 - Eliminar Cliente
 - Consultar Cliente
- Facturas:
 - Nueva Factura
 - Consultar Factura

Mostrar en un cuadro de diálogo, la opción elegida.



```
package es.studium.EjercicioMenu;

import java.awt.Dialog;
import java.awt.FlowLayout;
import java.awt.Frame;
import java.awt.Label;
import java.awt.Menu;
```

```
import java.awt.MenuBar;
import java.awt.MenuItem;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.WindowEvent;
import java.awt.event.WindowListener;

public class EjercicioMenu extends Frame implements WindowListener,
ActionListener
{
    private static final long serialVersionUID = 1L;
    MenuBar barraMenu = new MenuBar();

    Menu articulos = new Menu("Artículos");
    Menu clientes = new Menu("Clientes");
    Menu facturas = new Menu("Facturas");

    MenuItem articuloNuevo=new MenuItem("Nuevo Artículo");
    MenuItem articuloEliminar=new MenuItem("Eliminar Artículo");
    MenuItem articuloConsultar=new MenuItem("Consultar Artículo");

    MenuItem clientesNuevo=new MenuItem("Nuevo Cliente");
    MenuItem clientesEliminar=new MenuItem("Eliminar Cliente");
    MenuItem clientesConsultar=new MenuItem("Consultar Cliente");

    MenuItem facturasNueva=new MenuItem("Nueva Factura");
    MenuItem facturaConsultar=new MenuItem("Consultar Factura");

    Dialog d = new Dialog(this, "Diálogo con mensaje", true);
    Label e = new Label("");

    public EjercicioMenu()
    {
        setLayout(new FlowLayout());
        setTitle("Menú");

        setMenuBar(barraMenu);

        articulos.add(articuloNuevo);
        articulos.add(articuloEliminar);
        articulos.add(articuloConsultar);
        articuloNuevo.addActionListener(this);
        articuloEliminar.addActionListener(this);
        articuloConsultar.addActionListener(this);

        clientes.add(clientesNuevo);
        clientes.add(clientesEliminar);
        clientes.add(clientesConsultar);
        clientesNuevo.addActionListener(this);
        clientesEliminar.addActionListener(this);
        clientesConsultar.addActionListener(this);

        facturas.add(facturasNueva);
        facturas.add(facturaConsultar);
        facturasNueva.addActionListener(this);
        facturaConsultar.addActionListener(this);

        barraMenu.add(articulos);
        barraMenu.add(clientes);
        barraMenu.add(facturas);
    }
}
```

```

        addWindowListener(this);
        setSize(350,200);
        setVisible(true);

        d.setLayout(new FlowLayout());
        d.setSize(200,150);
        //Para poder cerrar el Diálogo
        d.addWindowListener(this);
    }
    public static void main(String[] args)
    {
        new EjercicioMenu();
    }
    public void windowActivated(WindowEvent we) {}
    public void windowClosed(WindowEvent we) {}
    public void windowClosing(WindowEvent we)
    {
        if(d.hasFocus())
        {
            d.setVisible(false);
        }
        else
        {
            System.exit(0);
        }
    }
    public void windowDeactivated(WindowEvent we) {}
    public void windowDeiconified(WindowEvent we) {}
    public void windowIconified(WindowEvent we) {}
    public void windowOpened(WindowEvent we) {}
    public void actionPerformed(ActionEvent ae)
    {
        Object a;
        a = ae.getSource();
        if(a.equals(articuloNuevo))
        {
            e.setText("Artículo Nuevo");
        }
        else if(a.equals(articuloEliminar))
        {
            e.setText("Artículo Eliminar");
        }
        else if(a.equals(articuloConsultar))
        {
            e.setText("Artículo Consultar");
        }
        else if(a.equals(clientesNuevo))
        {
            e.setText("Cliente Nuevo");
        }
        else if(a.equals(clientesEliminar))
        {
            e.setText("Cliente Eliminar");
        }
        else if(a.equals(clientesConsultar))
        {
            e.setText("Cliente Consultar");
        }
    }

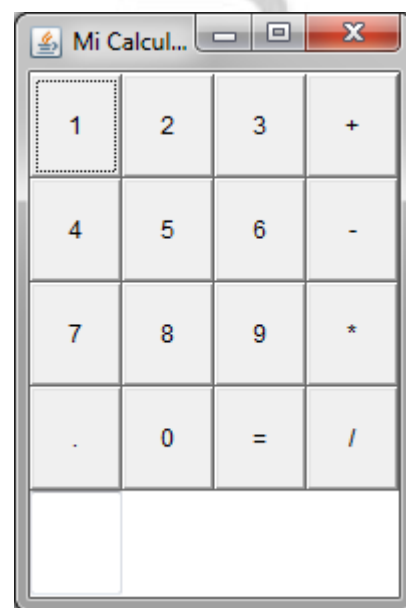
```

```

    else if (a.equals (facturasNueva))
    {
        e.setText ("Factura Nueva");
    }
    else
    {
        e.setText ("Factura Consultar");
    }
    d.add(e);
    d.setVisible(true);
}
}

```

Ejercicio 7: Realizar el diseño de una pequeña calculadora de diseño similar a la figura. Todo serán botones además de un cuadro de texto para el resultado. Tendremos los botones para los números del 0 al 9. También tendremos los botones para las operaciones de suma (+), resta (-), multiplicación (*) y división (/). Por último tendremos un botón para el igual (=) que indicará que haga la operación indicada. **Al pulsar el igual, el resultado de la operación saldrá en un cuadro de texto.**



```

package es.studium.calculadora;

import java.awt.Button;
import java.awt.Frame;
import java.awt.GridLayout;
import java.awt.TextField;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.WindowEvent;
import java.awt.event.WindowListener;

public class calculadora extends Frame implements WindowListener,
ActionListener
{
    private static final long serialVersionUID = 1L;
    Float operador1, operador2, resultado;
    int operacion;
    Boolean segundo, fin;
    Button boton1 = new Button("1");
    Button boton2 = new Button("2");
    Button boton3 = new Button("3");
    Button boton4 = new Button("+");
    Button boton5 = new Button("4");
    Button boton6 = new Button("5");
    Button boton7 = new Button("6");
    Button boton8 = new Button("-");
    Button boton9 = new Button("7");
    Button boton10 = new Button("8");
    Button boton11 = new Button("9");
    Button boton12 = new Button ("*");
    Button boton13 = new Button (".");
    Button boton14 = new Button ("0");

```



```

Button boton15 = new Button ("=");
Button boton16 = new Button ("/");
TextField t = new TextField();

public calculadora()
{
    setTitle("Mi Calculadora");
    setLayout(new GridLayout(5,3));
    segundo = false;
    fin = false;
    operacion = 0;
    add(boton1);
    boton1.addActionListener(this);
    add(boton2);
    boton2.addActionListener(this);
    add(boton3);
    boton3.addActionListener(this);
    add(boton4);
    boton4.addActionListener(this);
    add(boton5);
    boton5.addActionListener(this);
    add(boton6);
    boton6.addActionListener(this);
    add(boton7);
    boton7.addActionListener(this);
    add(boton8);
    boton8.addActionListener(this);
    add(boton9);
    boton9.addActionListener(this);
    add(boton10);
    boton10.addActionListener(this);
    add(boton11);
    boton11.addActionListener(this);
    add(boton12);
    boton12.addActionListener(this);
    add(boton13);
    boton13.addActionListener(this);
    add(boton14);
    boton14.addActionListener(this);
    add(boton15);
    boton15.addActionListener(this);
    add(boton16);
    boton16.addActionListener(this);
    add(t);

    addWindowListener(this);
    setSize(200,300);
    setVisible(true);
}

public static void main(String[] args)
{
    new calculadora();
}

public void windowActivated(WindowEvent we) {}
public void windowClosed(WindowEvent we) {}
public void windowClosing(WindowEvent we)
{
    System.exit(0);
}

```

```

public void windowDeactivated(WindowEvent we) {}
public void windowDeiconified(WindowEvent we) {}
public void windowIconified(WindowEvent we) {}
public void windowOpened(WindowEvent we) {}
public void actionPerformed(ActionEvent ae)
{
    Object a;
    a = ae.getSource();
    if(a.equals(boton1))
    {
        if(segundo)
        {
            operador2 = (float) 1;
        }
        else
        {
            operador1 = (float) 1;
        }
    }
    if(a.equals(boton2))
    {
        if(segundo)
        {
            operador2 = (float) 2;
        }
        else
        {
            operador1 = (float) 2;
        }
    }
    if(a.equals(boton3))
    {
        if(segundo)
        {
            operador2 = (float) 3;
        }
        else
        {
            operador1 = (float) 3;
        }
    }
    if(a.equals(boton4))
    {
        //Suma
        segundo = true;
        operacion = 1;
    }
    if(a.equals(boton5))
    {
        if(segundo)
        {
            operador2 = (float) 4;
        }
        else
        {
            operador1 = (float) 4;
        }
    }
    if(a.equals(boton6))
    {

```

```

        if(segundo)
        {
            operador2 = (float) 5;
        }
        else
        {
            operador1 = (float) 5;
        }
    }
    if(a.equals(boton7))
    {
        if(segundo)
        {
            operador2 = (float) 6;
        }
        else
        {
            operador1 = (float) 6;
        }
    }
    if(a.equals(boton8))
    {
        segundo = true;
        operacion = 2;
    }
    if(a.equals(boton9))
    {
        if(segundo)
        {
            operador2 = (float) 7;
        }
        else
        {
            operador1 = (float) 7;
        }
    }
    if(a.equals(boton10))
    {
        if(segundo)
        {
            operador2 = (float) 8;
        }
        else
        {
            operador1 = (float) 8;
        }
    }
    if(a.equals(boton11))
    {
        if(segundo)
        {
            operador2 = (float) 9;
        }
        else
        {
            operador1 = (float) 9;
        }
    }
}

```

```

if(a.equals(boton12))
{
    segundo = true;
    operacion = 3;
}
if(a.equals(boton13))
{
    // El decimal
}
if(a.equals(boton14))
{
    if(segundo)
    {
        operador2 = (float) 0;
    }
    else
    {
        operador1 = (float) 0;
    }
}
if(a.equals(boton15))
{
    switch(operacion)
    {
        case 1:
            resultado = operador1 + operador2;
            break;
        case 2:
            resultado = operador1 - operador2;
            break;
        case 3:
            resultado = operador1 * operador2;
            break;
        case 4:
            resultado = operador1 / operador2;
            break;
    }
    segundo = false;
    operacion = 0;
    fin = true;
}
if(a.equals(boton16))
{
    segundo = true;
    operacion = 4;
}
if(fin)
{
    t.setText(resultado.toString());
    fin = false;
    operador2 = null;
}
else
{
    if((segundo) && (operador2!=null))
    {
        t.setText(operador2.toString());
    }
}

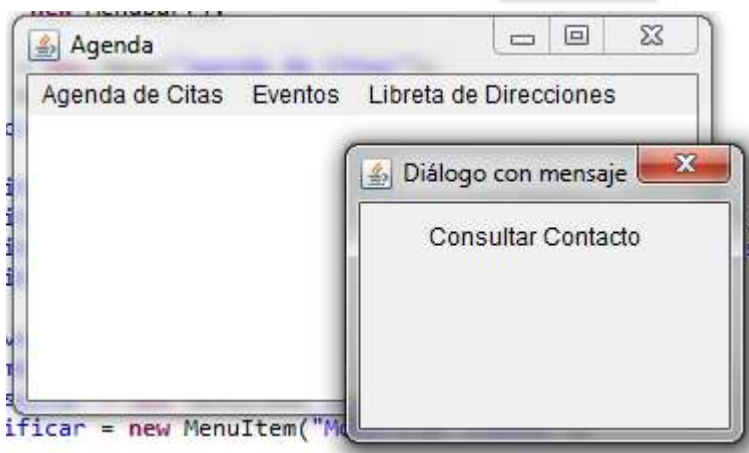
```

```

        else
        {
            t.setText(operador1.toString());
        }
    }
}

```

Ejercicio 8: Realizar el diseño de una aplicación que nos sirva de Agenda de Citas y Eventos y de Libreta de Direcciones. Tanto en una opción como en otra podremos añadir un elemento nuevo, eliminar uno existente, consultar los existentes y modificar uno de ellos. El diseño se deja a libre elección. **Mostrar en un cuadro de diálogo, la opción elegida.**



```

package es.studium.agenda;

import java.awt.Dialog;
import java.awt.FlowLayout;
import java.awt.Frame;
import java.awt.Label;
import java.awt.Menu;
import java.awt.MenuBar;
import java.awt.MenuItem;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.WindowEvent;
import java.awt.event.WindowListener;

public class agenda extends Frame implements WindowListener,
ActionListener
{
    private static final long serialVersionUID = 1L;
    MenuBar barraMenu = new MenuBar();

    Menu agendadecitas = new Menu("Agenda de Citas");
    Menu eventos = new Menu("Eventos");
    Menu libretadedirecciones = new Menu("Libreta de Direcciones");

    MenuItem agendadecitasNuevo = new MenuItem("Elemento Nuevo");
    MenuItem agendadecitasEliminar = new MenuItem("Eliminar una
Existente");
    MenuItem agendadecitasConsultar = new MenuItem("Consultar las
Existentes");
    MenuItem agendadecitasModificar = new MenuItem("Modificar
Cita");

    MenuItem eventoNuevo = new MenuItem("Evento Nuevo");
    MenuItem eventoEliminar = new MenuItem("Eliminar Evento");
    MenuItem eventoConsultar = new MenuItem("Consultar Evento");

```

```

MenuItem eventoModificar = new MenuItem("Modificar Evento");

MenuItem libretadedireccionesNueva = new MenuItem("Nueva
Dirección");
MenuItem libretadedireccionesEliminar = new MenuItem("Eliminar
Dirección");
MenuItem libretadedireccionesConsultar = new MenuItem("Consultar
Dirección");
MenuItem libretadedireccionesModificar = new MenuItem("Modificar
Dirección");

Dialog d = new Dialog(this, "Diálogo con mensaje", true);
Label e = new Label("");

public agenda()
{
    setLayout(new FlowLayout());
    setTitle("Agenda");
    setMenuBar(barraMenu);

    agendadecitas.add(agendadecitasNuevo);
    agendadecitas.add(agendadecitasEliminar);
    agendadecitas.add(agendadecitasConsultar);
    agendadecitas.add(agendadecitasModificar);
    agendadecitasNuevo.addActionListener(this);
    agendadecitasEliminar.addActionListener(this);
    agendadecitasConsultar.addActionListener(this);
    agendadecitasModificar.addActionListener(this);

    eventos.add(eventoNuevo);
    eventos.add(eventoEliminar);
    eventos.add(eventoConsultar);
    eventos.add(eventoModificar);
    eventoNuevo.addActionListener(this);
    eventoEliminar.addActionListener(this);
    eventoConsultar.addActionListener(this);
    eventoModificar.addActionListener(this);

    libretadedirecciones.add(libretadedireccionesNueva);
    libretadedirecciones.add(libretadedireccionesEliminar);
    libretadedirecciones.add(libretadedireccionesConsultar);
    libretadedirecciones.add(libretadedireccionesModificar);
    libretadedireccionesNueva.addActionListener(this);
    libretadedireccionesEliminar.addActionListener(this);
    libretadedireccionesConsultar.addActionListener(this);
    libretadedireccionesModificar.addActionListener(this);

    barraMenu.add(agendadecitas);
    barraMenu.add(eventos);
    barraMenu.add(libretadedirecciones);
    addWindowListener(this);

    d.setLayout(new FlowLayout());
    d.setSize(200,150);
    //Para poder cerrar el Diálogo
    d.addWindowListener(this);

    setSize(350,200);
    setVisible(true);
}

```



```

public static void main(String[] args)
{
    new agenda();
}
public void windowActivated(WindowEvent we) {}
public void windowClosed(WindowEvent we) {}
public void windowClosing(WindowEvent we)
{
    if(d.hasFocus())
    {
        d.setVisible(false);
    }
    else
    {
        System.exit(0);
    }
}
public void windowDeactivated(WindowEvent we) {}
public void windowDeiconified(WindowEvent we) {}
public void windowIconified(WindowEvent we) {}
public void windowOpened(WindowEvent we) {}
public void actionPerformed(ActionEvent ae)
{
    Object a;
    a = ae.getSource();
    if(a.equals(agendaDecitasNuevo))
    {
        e.setText("Nueva Cita");
    }
    else if(a.equals(agendaDecitasEliminar))
    {
        e.setText("Eliminar Cita");
    }
    else if(a.equals(agendaDecitasConsultar))
    {
        e.setText("Consultar Cita");
    }
    else if(a.equals(agendaDecitasModificar))
    {
        e.setText("Modificar Cita");
    }
    else if(a.equals(eventoNuevo))
    {
        e.setText("Nuevo Evento");
    }
    else if(a.equals(eventoEliminar))
    {
        e.setText("Eliminar Evento");
    }
    else if(a.equals(eventoConsultar))
    {
        e.setText("Consultar Evento");
    }
    else if(a.equals(eventoModificar))
    {
        e.setText("Modificar Evento");
    }
}

```

```

        else if(a.equals(libretadedireccionesNueva))
        {
            e.setText("Nuevo Contacto");
        }
        else if(a.equals(libretadedireccionesEliminar))
        {
            e.setText("Eliminar Contacto");
        }
        else if(a.equals(libretadedireccionesConsultar))
        {
            e.setText("Consultar Contacto");
        }
        else
        {
            e.setText("Modificar Contacto");
        }
        d.add(e);
        d.setVisible(true);
    }
}

```

Otra versión: Se recorren todos los menús (getMenuCount(), getMenu() y getItemCount()) hasta que se da con el que ha llamado al ActionEvent y se le saca su información (getItem() y getLabel()).

```

package es.studium.OtroMenu;

import java.awt.Dialog;
import java.awt.FlowLayout;
import java.awt.Frame;
import java.awt.Label;
import java.awt.Menu;
import java.awt.MenuBar;
import java.awt.MenuItem;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.WindowEvent;
import java.awt.event.WindowListener;

public class OtroMenu extends Frame implements WindowListener, ActionListener
{
    // En primer lugar creamos la barra de menú
    MenuBar barraMenu = new MenuBar();

    // Ahora creamos los elementos principales del menú
    Menu articulos = new Menu(" Artículos ");
    Menu clientes = new Menu(" Clientes ");
    Menu facturas = new Menu(" Facturas ");

    // Cada opción del menú
    MenuItem nuevoart = new MenuItem("Nuevo");
    MenuItem eliminarart = new MenuItem("Eliminar");
    MenuItem consultarart= new MenuItem("Consultar");

    MenuItem nuevocli = new MenuItem("Nuevo");
    MenuItem eliminarccli = new MenuItem("Eliminar");
    MenuItem consultarccli= new MenuItem("Consultar");

```

```

MenuItem nuevafac = new MenuItem("Nuevo");
MenuItem consultarfac= new MenuItem("Consultar");

Dialog mensaje=new Dialog(this, "Opción seleccionada", true);

Label opcion = new Label("");

public OtroMenu()
{
    setLayout (new FlowLayout());
    setTitle ("Menú" );

    // el menú de artículos
    articulos.add(nuevoart);
    articulos.add(eliminarart);
    articulos.add(consultarart);

    //Se añaden los Listeners a las opciones de los submenús
    //no a las opciones del menú
    nuevoart.addActionListener(this);
    eliminarart.addActionListener(this);
    consultarart.addActionListener(this);

    // el menú de clientes
    clientes.add(nuevocli);
    clientes.add(eliminarcli);
    clientes.add(consultarcli);

    //Se añaden los Listeners a las opciones de los submenús
    //no a las opciones del menú
    nuevocli.addActionListener(this);
    eliminarcli.addActionListener(this);
    consultarcli.addActionListener(this);

    // el menú de facturas
    facturas.add(nuevafac);
    facturas.add(consultarfac);

    //Se añaden los Listeners a las opciones de los submenús
    //no a las opciones del menú
    nuevafac.addActionListener(this);
    consultarfac.addActionListener(this);

    // Opciones a barramenu
    barraMenu.add(articulos);
    barraMenu.add(clientes);
    barraMenu.add(facturas);

    // Establecemos la barra de menú
    setMenuBar (barraMenu);

    // Ventana de diálogo
    mensaje.setLayout(new FlowLayout());
    mensaje.addWindowListener(this);
    mensaje.setSize(250, 100);
    mensaje.setResizable(false);
    mensaje.add(opcion);

    // Listener de ventana
    addWindowListener(this);

```

```

        setSize(400,300);
        setVisible(true);
    }
    public static void main(String[] args)
    {
        new OtroMenu();
    }
    public void actionPerformed(ActionEvent arg0)
    {
        int nmenu, nitemmenu;
        String cmenu="", citeemmenu="";
        for( nmenu=0; nmenu<barraMenu.getMenuCount();nmenu++)
        {
            for(nitemmenu=0;nitemmenu<barraMenu.getMenu(nmenu).getItemCount();nitemmenu++)
            {
                if(barraMenu.getMenu(nmenu).getItem(nitemmenu).equals(arg0.getSource()))
                {
                    cmenu=barraMenu.getMenu(nmenu).getLabel();
                    citeemmenu=barraMenu.getMenu(nmenu).getItem(nitemmenu).getLabel();
                }
            }
            opcion.setText("Opcion " + citeemmenu+ " de Menú " + cmenu);
            mensaje.setVisible(true);
        }
        public void windowActivated(WindowEvent e) {}
        public void windowClosed(WindowEvent e) {}
        public void windowClosing(WindowEvent e)
        {
            if(mensaje.hasFocus())
            {
                mensaje.setVisible(false);
            }
            else
            {
                System.exit(0);
            }
        }
        public void windowDeactivated(WindowEvent e) {}
        public void windowDeiconified(WindowEvent e) {}
        public void windowIconified(WindowEvent e) {}
        public void windowOpened(WindowEvent e) {}
    }
}

```

Una última versión: Modificando solamente el actionPerformed:

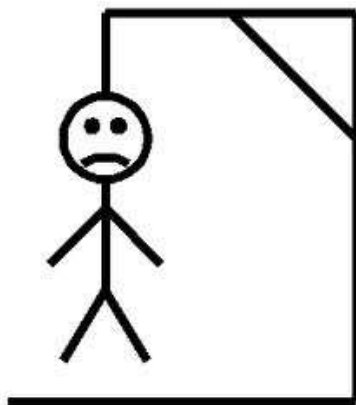
```

public void actionPerformed(ActionEvent ae)
{
    MenuItem submenu = (MenuItem)ae.getSource();
    Menu padre = (Menu)submenu.getParent();
    e.setText ("Menu: " + padre.getLabel() + "/" Opción: " +ae.getActionCommand());
    d.setVisible(true);
}

```

Ejercicio 9: Diseñar un interfaz sencillo para jugar al tres en raya. **Al pulsar una casilla se pondrá con una X, la siguiente pulsación pondrá un O, y así sucesivamente. Controlar quién gana la partida.**

Ejercicio 10: Se pide diseñar el interfaz gráfico de usuario del juego del ahorcado. En primer lugar se pedirá una palabra para adivinar. Luego se mostrarán tantas rayitas como letras tenga la palabra. A continuación se irán pidiendo letras de una en una. El juego va sustituyendo las rayitas por letras acertadas o en otro lugar las letras no acertadas. El juego acaba cuando el jugador acierta la palabra o bien cuando el muñeco se completa.



19/02/2016