

## 38 - Swing - JCheckBox

[Listado completo de tutoriales](#)

El control JCheckBox permite implementar un cuadro de selección (básicamente un botón de dos estados)

### Problema 1:

Confeccionar un programa que muestre 3 objetos de la clase JCheckBox con etiquetas de tres idiomas. Cuando se lo selecciona mostrar en el título del JFrame todos los JCheckBox seleccionados hasta el momento.



### Programa:

[Ver video](#)

```
import javax.swing.*;
import javax.swing.event.*;
public class Formulario extends JFrame implements Cha
    private JCheckBox check1,check2,check3;
    public Formulario() {
        setLayout(null);
        check1=new JCheckBox("Inglés");
        check1.setBounds(10,10,150,30);
        check1.addChangeListener(this);
        add(check1);
        check2=new JCheckBox("Francés");
        check2.setBounds(10,50,150,30);
        check2.addChangeListener(this);
        add(check2);
        check3=new JCheckBox("Alemán");
        check3.setBounds(10,90,150,30);
        check3.addChangeListener(this);
        add(check3);
```

```

    }

    public void stateChanged(ChangeEvent e) {
        String cad="";
        if (check1.isSelected()==true) {
            cad=cad+"Inglés-";
        }
        if (check2.isSelected()==true) {
            cad=cad+"Francés-";
        }
        if (check3.isSelected()==true) {
            cad=cad+"Alemán-";
        }
        setTitle(cad);
    }

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

```

Lo primero y más importante que tenemos que notar que para capturar el cambio de estado del JCheckBox hay que implementar la interface ChangeListener que se encuentra en el paquete:

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

y no en el paquete:

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

Cuando declaramos la clase JFrame indicamos que implementaremos la interface ChangeListener:

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

Definimos tres objetos de la clase JCheckBox:

```
private JCheckBox check1,check2,check3;
```

En el constructor creamos cada uno de los objetos de la clase JCheckBox y llamamos al método addChangeListener indicando quien procesará el evento de

cambio de estado:

```
check1=new JCheckBox("Inglés");  
check1.setBounds(10,10,150,30);  
check1.addChangeListener(this);  
add(check1);
```

El método que debemos implementar de la interface ChangeListener es:

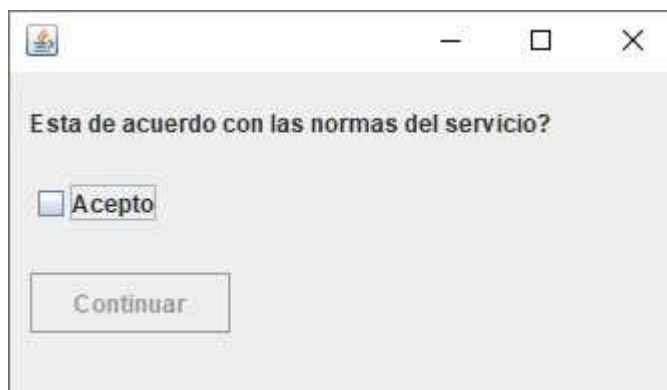
```
public void stateChanged(ChangeEvent e){
```

En este mediante tres if verificamos el estado de cada JCheckBox y concatenamos los String con los idiomas seleccionados:

```
String cad="";  
if (check1.isSelected()==true) {  
    cad=cad+"Inglés-";  
}  
if (check2.isSelected()==true) {  
    cad=cad+"Francés-";  
}  
if (check3.isSelected()==true) {  
    cad=cad+"Alemán-";  
}  
setTitle(cad);
```

## Problema 2:

Disponer un control JLabel que muestre el siguiente mensaje: "Esta de acuerdo con las normas del servicio?", luego un JCheckBox y finalmente un objeto de tipo JButton desactivo. Cuando se tilde el JCheckBox debemos activar el botón.



## Programa:

[Ver video](#)

```
import javax.swing.*;
import javax.swing.event.*;
import java.awt.event.*;
public class Formulario extends JFrame implements ActionListener {
    private JLabel label1;
    private JCheckBox check1;
    private JButton boton1;
    public Formulario() {
        setLayout(null);
        label1=new JLabel("Esta de acuerdo con las no");
        label1.setBounds(10,10,400,30);
        add(label1);
        check1=new JCheckBox("Acepto");
        check1.setBounds(10,50,100,30);
        check1.addChangeListener(this);
        add(check1);
        boton1=new JButton("Continuar");
        boton1.setBounds(10,100,100,30);
        add(boton1);
        boton1.addActionListener(this);
        boton1.setEnabled(false);
    }

    public void stateChanged(ChangeEvent e) {
        if (check1.isSelected()==true) {
            boton1.setEnabled(true);
        } else {
            boton1.setEnabled(false);
        }
    }

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

    public static void main(String[] ar) {
        Formulario formulario1=new Formulario();
        formulario1.setBounds(0,0,350,200);
        formulario1.setVisible(true);
        formulario1.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }
}
```

Importamos los paquetes donde se encuentran las interfaces para captura de eventos de objetos de tipo JButton y JCheckBox:

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

También importamos el paquete donde están definidas las clase JFrame, JButton y JCheckBox:

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

Como debemos implementar dos interfaces las debemos enumerar después de la palabra implements separadas por coma:

```
public class Formulario extends JFrame implements ActionListener, ChangeListener{  
    Definimos los tres objetos:
```

```
        private JLabel label1;  
        private JCheckBox check1;  
        private JButton boton1;
```

En el constructor creamos el objeto de tipo JLabel:

```
    public Formulario() {  
        setLayout(null);  
        label1=new JLabel("Esta de acuerdo con las normas del servicio  
        label1.setBounds(10,10,400,30);  
        add(label1);
```

El objeto de tipo JCheckBox:

```
        check1=new JCheckBox("Acepto");  
        check1.setBounds(10,50,100,30);  
        check1.addChangeListener(this);  
        add(check1);
```

y también creamos el objeto de tipo JButton y llamando al método setEnabled con un valor false luego el botón aparece desactivo:

```
        boton1=new JButton("Continuar");  
        boton1.setBounds(10,100,100,30);  
        add(boton1);  
        boton1.addActionListener(this);  
        boton1.setEnabled(false);
```

Cuando se cambia el estado del control JCheckBox se ejecuta el método `stateChanged` donde verificamos si está seleccionado procediendo a activar el botón en caso negativo lo desactivamos:

```
public void stateChanged(ChangeEvent e) {  
    if (check1.isSelected()==true) {  
        boton1.setEnabled(true);  
    } else {  
        boton1.setEnabled(false);  
    }  
}
```

El método `actionPerformed` se ejecuta cuando se presiona el objeto de tipo `JButton` (debe estar activo para poder presionarlo):

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

## Problema propuesto

1. Disponer tres objetos de la clase `JCheckBox` con nombres de navegadores web. Cuando se presione un botón mostrar en el título del `JFrame` los programas seleccionados.

[Ver video](#)

[Solución](#)

**Retornar**