

# JCheckBox

JCheckBox es una parte del paquete Java Swing. JCheckBox se puede seleccionar o deselectar. Muestra su estado al usuario. JCheckBox es una implementación de checkbox. JCheckBox hereda la clase JToggleButton.

## Los constructores de la clase son:

1. **JCheckBox()** : crea una nueva casilla de verificación sin texto ni icono
2. **JCheckBox(Icon i)** : crea una nueva casilla de verificación con el icono especificado
3. **JCheckBox(Icono de icono, s booleana)**: crea una nueva casilla de verificación con el icono especificado y el valor booleano especifica si está seleccionado o no.
4. **JCheckBox(String t)**: crea una nueva casilla de verificación con la cadena especificada
5. **JCheckBox(String text, boolean selected)** :crea una nueva casilla de verificación con la cadena especificada y el valor booleano especifica si está seleccionada o no.
6. **JCheckBox(Texto de cadena, icono de icono)**: crea una nueva casilla de verificación con la cadena y el icono especificados.
7. **JCheckBox(String text, Icon icon, boolean selected)**: crea una nueva casilla de verificación con la cadena y el icono especificados y el valor booleano especifica si está seleccionado o no.

## Métodos para agregar el agente de escucha de elementos a la casilla de verificación.

1. **addActionListener(ItemListener l)**: agrega el agente de escucha de elementos al componente
2. **itemStateChanged(ItemEvent e)**: función abstracta invocada cuando cambia el estado del elemento al que se aplica el agente de escucha
3. **getItem()** : devuelve el objeto específico del componente asociado al elemento cuyo estado ha cambiado.
4. **getStateChange()** : devuelve el nuevo estado del elemento. La clase ItemEvent define dos estados: SELECTED y DESELECTED.
5. **getSource()**: devuelve el componente que desencadenó el evento item.

## Métodos comúnmente utilizados:

1. **setIcon(Icon i)** : establece el icono de la casilla de verificación en el icono dado
2. **setText(String s)**: establece el texto de la casilla de verificación en el texto dado
3. **setSelected(boolean b)**: establece la casilla de verificación en selected si el valor booleano pasado es true o viceversa
4. **getIcon()** : devuelve la imagen de la casilla de verificación
5. **getText()** : devuelve el texto de la casilla de verificación
6. **updateUI()**: restablece la propiedad UI con un valor de la apariencia actual.
7. **getUI()**: devuelve el objeto de aspecto que representa este componente.
8.  **paramString()** : devuelve una representación de cadena de este JCheckBox.
9. **getUIClassID()**: devuelve el nombre de la clase Look and feel que representa este componente.
10. **getAccessibleContext()** : obtiene el AccessibleContext asociado a este JCheckBox.
11. **isBorderPaintedFlat()** : obtiene el valor de la propiedad **borderPaintedFlat**.
12. **setBorderPaintedFlat(boolean b)** : establece la propiedad **borderPaintedFlat**,

## Los siguientes programas ilustrarán el uso de JCheckBox

### 1. Programa para crear una casilla de verificación simple usando JCheckBox

```
// java Program to create a simple checkbox using JCheckBox
import java.awt.event.*;
import java.awt.*;
import javax.swing.*;
class solve extends JFrame {
```

```
// frame
static JFrame f;

// main class
public static void main(String[] args)
{
    // create a new frame
    f = new JFrame("frame");

    // set layout of frame
    f.setLayout(new FlowLayout());

    // create checkbox
    JCheckBox c1 = new JCheckBox("checkbox 1");
    JCheckBox c2 = new JCheckBox("checkbox 2");

    // create a new panel
    JPanel p = new JPanel();

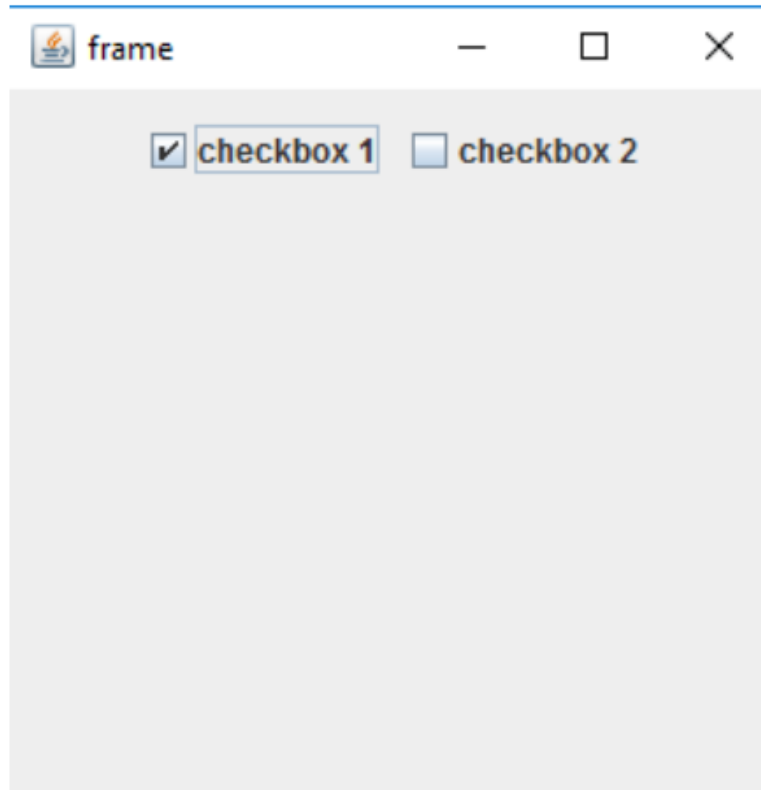
    // add checkbox to panel
    p.add(c1);
    p.add(c2);

    // add panel to frame
    f.add(p);

    // set the size of frame
    f.setSize(300, 300);

    f.show();
}
}
```

**Salida:**



## 2. Programa para crear una casilla de verificación con un icono .

```
// java Program to create a checkbox with a icon .
import java.awt.event.*;
import java.awt.*;
import javax.swing.*;
class solve extends JFrame {

    // frame
    static JFrame f;

    // main class
    public static void main(String[] args)
    {
        // create a new frame
        f = new JFrame("frame");

        // set layout of frame
        f.setLayout(new FlowLayout());

        // create checkbox
        JCheckBox c1 = new JCheckBox("checkbox with image", new ImageIcon("f:/gfg.jpg"), true);
        JCheckBox c2 = new JCheckBox("checkbox 2");

        // create a new panel
        JPanel p = new JPanel();

        // add checkbox to panel
        p.add(c1);
```

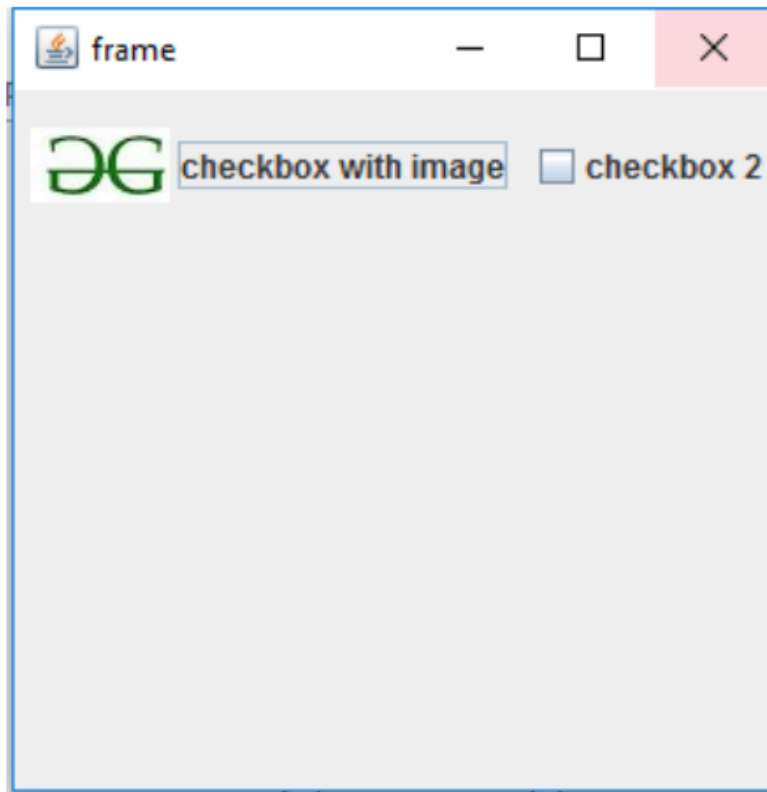
```
p.add(c2);

// add panel to frame
f.add(p);

// set the size of frame
f.setSize(300, 300);

f.show();
}
```

**Salida:**



## Bibliografía:

- Extraído de <<Java Swing - JCheckBox con ejemplos>>. Consultado el 28 de noviembre del 2022