

SWING - JCHECKBOX CLASS

http://www.tutorialspoint.com/swing/swing_jcheckbox.htm

Copyright © tutorialspoint.com

Introduction

The class **JCheckBox** is an implementation of a check box - an item that can be selected or deselected, and which displays its state to the user.

Class declaration

Following is the declaration for **javax.swing.JCheckBox** class –

```
public class JCheckBox
    extends JToggleButton
    implements Accessible
```

Field

Following are the fields for **javax.swing.JCheckBox** class –

- **static String BORDER_PAINTED_FLAT_CHANGED_PROPERTY** – Identifies a change to the flat property.

Class constructors

S.N.	Constructor & Description
1	JCheckBox Creates an initially unselected check box button with no text, no icon.
2	JCheckBoxActiona Creates a check box where properties are taken from the Action supplied.
3	JCheckBoxIconicon Creates an initially unselected check box with an icon.
4	JCheckBoxIconicon, booleanselected Creates a check box with an icon and specifies whether or not it is initially selected.
5	JCheckBoxStringtext Creates an initially unselected check box with text.
6	JCheckBoxStringtext, booleanselected Creates a check box with text and specifies whether or not it is initially selected.
7	JCheckBoxStringtext, Iconicon Creates an initially unselected check box with the specified text and icon.
8	JCheckBoxStringtext, Iconicon, booleanselected

Creates a check box with text and icon, and specifies whether or not it is initially selected.

Class methods

S.N.	Method & Description
1	AccessibleContext getAccessibleContext Gets the AccessibleContext associated with this JCheckBox.
2	String getUIClassID Returns a string that specifies the name of the L&F class that renders this component.
3	boolean isBorderPaintedFlat Gets the value of the borderPaintedFlat property.
4	protected String paramString Returns a string representation of this JCheckBox.
5	void setBorderPaintedFlatbooleanb Sets the borderPaintedFlat property, which gives a hint to the look and feel as to the appearance of the check box border.
6	void updateUI Resets the UI property to a value from the current look and feel.

Methods inherited

This class inherits methods from the following classes:

- javax.swing.AbstractButton
- javax.swing.JToggleButton
- javax.swing.JComponent
- java.awt.Container
- java.awt.Component
- java.lang.Object

JCheckBox Example

Create the following java program using any editor of your choice in say **D:/ > SWING > com > tutorialspoint > gui >**

SwingControlDemo.java

```
package com.tutorialspoint.gui;  
  
import java.awt.*;  
import java.awt.event.*;
```

```

import javax.swing.*;

public class SwingControlDemo {

    private JFrame mainFrame;
    private JLabel headerLabel;
    private JLabel statusLabel;
    private JPanel controlPanel;

    public SwingControlDemo(){
        prepareGUI();
    }

    public static void main(String[] args){
        SwingControlDemo swingControlDemo = new SwingControlDemo();
        swingControlDemo.showCheckBoxDemo();
    }

    private void prepareGUI(){
        mainFrame = new JFrame("Java Swing Examples");
        mainFrame.setSize(400,400);
        mainFrame.setLayout(new GridLayout(3, 1));
        mainFrame.addWindowListener(new WindowAdapter() {
            public void windowClosing(WindowEvent windowEvent){
                System.exit(0);
            }
        });
        headerLabel = new JLabel("", JLabel.CENTER);
        statusLabel = new JLabel("", JLabel.CENTER);

        statusLabel.setSize(350,100);

        controlPanel = new JPanel();
        controlPanel.setLayout(new FlowLayout());

        mainFrame.add(headerLabel);
        mainFrame.add(controlPanel);
        mainFrame.add(statusLabel);
        mainFrame.setVisible(true);
    }

    private void showCheckBoxDemo(){

        headerLabel.setText("Control in action: CheckBox");

        final JCheckBox chkApple = new JCheckBox("Apple");
        final JCheckBox chkMango = new JCheckBox("Mango");
        final JCheckBox chkPeer = new JCheckBox("Peer");

        chkApple.setMnemonic(KeyEvent.VK_C);
        chkMango.setMnemonic(KeyEvent.VK_M);
        chkPeer.setMnemonic(KeyEvent.VK_P);

        chkApple.addItemListener(new ItemListener() {
            public void itemStateChanged(ItemEvent e) {
                statusLabel.setText("Apple Checkbox: "
                    + (e.getStateChange()==1?"checked":"unchecked"));
            }
        });

        chkMango.addItemListener(new ItemListener() {
            public void itemStateChanged(ItemEvent e) {
                statusLabel.setText("Mango Checkbox: "
                    + (e.getStateChange()==1?"checked":"unchecked"));
            }
        });

        chkPeer.addItemListener(new ItemListener() {
            public void itemStateChanged(ItemEvent e) {

```

```

        statusLabel.setText("Peer Checkbox: "
+ (e.getStateChange()==1?"checked":"unchecked"));
    }
});

controlPanel.add(chkApple);
controlPanel.add(chkMango);
controlPanel.add(chkPeer);

mainFrame.setVisible(true);
}
}

```

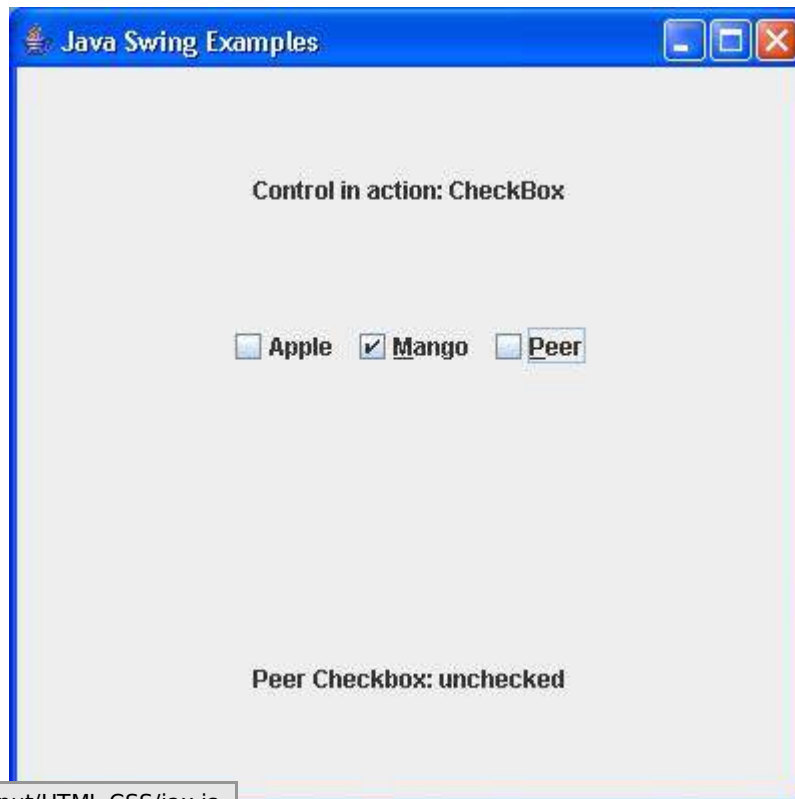
Compile the program using command prompt. Go to **D:/ > SWING** and type the following command.

```
D:\SWING>javac com\tutorialspoint\gui\SwingControlDemo.java
```

If no error comes that means compilation is successful. Run the program using following command.

```
D:\SWING>java com.tutorialspoint.gui.SwingControlDemo
```

Verify the following output



Loading [Mathjax]/jax/output/HTML-CSS/jax.js