

AWT CHECKBOX CLASS

http://www.tutorialspoint.com/awt/awt_checkbox.htm

Copyright © tutorialspoint.com

Introduction

Checkbox control is used to turn an option on *true* or off *false*. There is label for each checkbox representing what the checkbox does. The state of a checkbox can be changed by clicking on it.

Class declaration

Following is the declaration for **java.awt.Checkbox** class:

```
public class Checkbox
    extends Component
    implements ItemSelectable, Accessible
```

Class constructors

S.N.	Constructor & Description
1	Checkbox Creates a check box with an empty string for its label.
2	CheckboxStringlabel Creates a check box with the specified label.
3	CheckboxStringlabel, booleanstate Creates a check box with the specified label and sets the specified state.
4	CheckboxStringlabel, booleanstate, CheckboxGroupgroup Constructs a Checkbox with the specified label, set to the specified state, and in the specified check box group.
5	CheckboxStringlabel, CheckboxGroupgroup, booleanstate Creates a check box with the specified label, in the specified check box group, and set to the specified state.

Class methods

S.N.	Method & Description
1	void addItemListenerItemListenerl Adds the specified item listener to receive item events from this check box.

2

void addNotify

Creates the peer of the Checkbox.

3

AccessibleContext getAccessibleContext

Gets the AccessibleContext associated with this Checkbox.

4

CheckboxGroup getCheckboxGroup

Determines this check box's group.

5

ItemListener[] getItemListeners

Returns an array of all the item listeners registered on this checkbox.

6

String getLabel

Gets the label of this check box.

7

<T extends EventListener>T[] getListenersClass < T > listenerType

Returns an array of all the objects currently registered as FooListeners upon this Checkbox.

8

Object[] getSelectedObjects

Returns an array *length1* containing the checkbox label or null if the checkbox is not selected.

9

boolean getState

Determines whether this check box is in the **on** or **off** state.

10

protected String paramString

Returns a string representing the state of this Checkbox.

11

protected void processEventAWTEvente

Processes events on this check box.

12

protected void processItemEventItemEvente

Processes item events occurring on this check box by dispatching them to any registered ItemListener objects.

13

void removeItemListener*ItemListenerl*

Removes the specified item listener so that the item listener no longer receives item events from this check box.

14

void setCheckboxGroup*CheckboxGroupg*

Sets this check box's group to the specified check box group.

15

void setLabel*Stringlabel*

Sets this check box's label to be the string argument.

16

void setState*booleanstate*

Sets the state of this check box to the specified state.

Methods inherited

This class inherits methods from the following classes:

- java.awt.Component
- java.lang.Object

CheckBox Example

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

AwtControlDemo.java

```
package com.tutorialspoint.gui;

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

public class AwtControlDemo {

    private Frame mainFrame;
    private Label headerLabel;
    private Label statusLabel;
    private Panel controlPanel;

    public AwtControlDemo(){
        prepareGUI();
    }

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

    private void prepareGUI(){
        mainFrame = new Frame("Java AWT 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 Label();
headerLabel.setAlignment(Label.CENTER);
statusLabel = new Label();
statusLabel.setAlignment(Label.CENTER);
statusLabel.setSize(350,100);

controlPanel = new Panel();
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");

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

    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:/ > AWT** and type the following command.

```
D:\AWT>javac com\tutorialspoint\gui\AwtControlDemo.java
```

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

```
D:\AWT>java com.tutorialspoint.gui.AwtControlDemo
```

Verify the following output



Control in action: CheckBox

☒ Apple ☐ Mango ☐ Peer

Apple Checkbox: checked

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