AWT BUTTON CLASS

Introduction

Button is a control component that has a label and generates an event when pressed. When a button is pressed and released, AWT sends an instance of ActionEvent to the button, by calling processEvent on the button. The button's processEvent method receives all events for the button; it passes an action event along by calling its own processActionEvent method. The latter method passes the action event on to any action listeners that have registered an interest in action events generated by this button.

If an application wants to perform some action based on a button being pressed and released, it should implement ActionListener and register the new listener to receive events from this button, by calling the button's addActionListener method. The application can make use of the button's action command as a messaging protocol.

Class declaration

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

```
public class Button
   extends Component
      implements Accessible
```

Class constructors

S.N.	Constructor & Description
1	**Button** Constructs a button with an empty string for its label.
2	**Button*Stringtext*** Constructs a new button with specified label.

Class methods

S.N.	Method & Description
1	**void addActionListener*ActionListenerl*** Adds the specified action listener to receive action events from this button.
2	**void addNotify** Creates the peer of the button.
3	**AccessibleContext getAccessibleContext** Gets the AccessibleContext associated with this Button.

4

String getActionCommand

Returns the command name of the action event fired by this button.

5

ActionListener[] getActionListeners

Returns an array of all the action listeners registered on this button.

6

String getLabel

Gets the label of this button.

7

<T extends EventListener> T[] getListeners*Class < T > listenerType*

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

8

protected String paramString

Returns a string representing the state of this Button.

9

protected void processActionEvent*ActionEvente*

Processes action events occurring on this button by dispatching them to any registered ActionListener objects.

10

protected void processEvent*AWTEvente*

Processes events on this button.

11

void removeActionListener*ActionListenerl*

Removes the specified action listener so that it no longer receives action events from this button.

12

void setActionCommand*Stringcommand*

Sets the command name for the action event fired by this button.

13

void setLabel*Stringlabel*

Sets the button's label to be the specified string.

Methods inherited

This class inherits methods from the following classes:

- java.awt.Component

- java.lang.Object

Button 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.showButtonDemo();
   }

   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 showButtonDemo(){
      headerLabel.setText("Control in action: Button");

      Button okButton = new Button("OK");
      Button submitButton = new Button("Submit");
      Button cancelButton = new Button("Cancel");

      okButton.addActionListener(new ActionListener() {
         public void actionPerformed(ActionEvent e) {
            statusLabel.setText("Ok Button clicked.");
         }
      });

      submitButton.addActionListener(new ActionListener() {
         public void actionPerformed(ActionEvent e) {
```

```
            statusLabel.setText("Submit Button clicked.");
         }
      });

      cancelButton.addActionListener(new ActionListener() {
         public void actionPerformed(ActionEvent e) {
            statusLabel.setText("Cancel Button clicked.");
         }
      });

      controlPanel.add(okButton);
      controlPanel.add(submitButton);
      controlPanel.add(cancelButton);

      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