**BUTTON:-**

The button class is used to create a labeled button that has platform independent implementation.

Package-import java.awt.Button;

ActionEvent class and ActionListener interface is associated with button component.

|  |  |  |  |
| --- | --- | --- | --- |
| **GUI Component** | **Event class** | **Listener Interface** | **Method (abstract method)** |
| Button | ActionEvent | ActionListener | public void actionPerformed(ActionEvent e) |

**constructors**

|  |  |
| --- | --- |
| **S.N.** | **Constructor** |
| 1 | **Button()**  Constructs a button with an empty string for its label. |
| 2 | **Button(String text)**  Constructs a new button with specified label. |

**methods**

|  |  |
| --- | --- |
| **S.N.** | **Method** |
| 1 | **void addActionListener(ActionListener l)**  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(ActionEvent e)**  Processes action events occurring on this button by dispatching them to any registered ActionListener objects. |
| 10 | **protected void processEvent(AWTEvent e)**  Processes events on this button. |
| 11 | **void removeActionListener(ActionListener l)**  Removes the specified action listener so that it no longer receives action events from this button. |
| 12 | **void setActionCommand(String command)**  Sets the command name for the action event fired by this button. |
| 13 | **void setLabel(String label)**  Sets the button's label to be the specified string. |

***Example:-***

*import java.awt.\*;*

*public class MyButton {*

*public static void main(String[] args) {*

*Frame f=new Frame("Button Example-");*

*Button b=new Button("Click Here");*

*b.setBounds(50,100,80,30);*

*f.add(b);*

*f.setSize(400,400);*

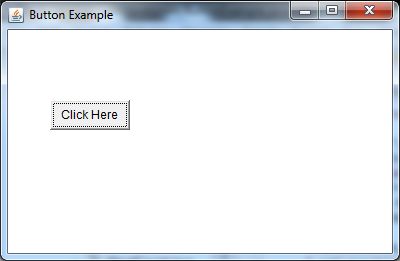
*f.setLayout(null);*

*f.setVisible(true);*

*}*

*}*

**Output:-**

****