AWT MENUITEM CLASS

Introduction

The MenuBar class represents the actual item in a menu. All items in a menu should derive from class MenuItem, or one of its subclasses. By default, it embodies a simple labeled menu item.

Class declaration

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

```
public class MenuItem
   extends MenuComponent
      implements Accessible
```

Class constructors

S.N.	Constructor & Description
1	**MenuItem** Constructs a new MenuItem with an empty label and no keyboard shortcut.
2	**MenuItem***Stringlabel* Constructs a new MenuItem with the specified label and no keyboard shortcut.
3	**MenuItem***Stringlabel, MenuShortcuts* Create a menu item with an associated keyboard shortcut.

Class methods

S.N.	Method & Description
1	**void addActionListener***ActionListenerl* Adds the specified action listener to receive action events from this menu item.
2	**void addNotify** Creates the menu item's peer.
3	**void deleteShortcut** Delete any MenuShortcut object associated with this menu item.
4	

void disable

Deprecated. As of JDK version 1.1, replaced by setEnabled*boolean*.

5

protected void disableEvents*longeventsToDisable*

Disables event delivery to this menu item for events defined by the specified event mask parameter.

6

void enable

Deprecated. As of JDK version 1.1, replaced by setEnabled*boolean*.

7

void enable*booleanb*

Deprecated. As of JDK version 1.1, replaced by setEnabled*boolean*.

8

protected void enableEvents*longeventsToEnable*

Enables event delivery to this menu item for events to be defined by the specified event mask parameter.

9

AccessibleContext getAccessibleContext

Gets the AccessibleContext associated with this MenuItem.

10

String getActionCommand

Gets the command name of the action event that is fired by this menu item.

11

ActionListener[] getActionListeners

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

12

String getLabel

Gets the label for this menu item.

13

EventListener[] getListeners*ClasslistenerType*

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

14

MenuShortcut getShortcut

Get the MenuShortcut object associated with this menu item.

15

boolean isEnabled

Checks whether this menu item is enabled.

16

String paramString

Returns a string representing the state of this MenuItem.

17

protected void processActionEvent*ActionEvente*

Processes action events occurring on this menu item, by dispatching them to any registered ActionListener objects.

18

protected void processEvent*AWTEvente*

Processes events on this menu item.

19

void removeActionListener*ActionListenerl*

Removes the specified action listener so it no longer receives action events from this menu item.

20

void setActionCommand*Stringcommand*

Sets the command name of the action event that is fired by this menu item.

21

void setEnabled*booleanb*

Sets whether or not this menu item can be chosen.

22

void setLabel*Stringlabel*

Sets the label for this menu item to the specified label.

23

void setShortcut*MenuShortcuts*

Set the MenuShortcut object associated with this menu item.

Methods inherited

This class inherits methods from the following classes:

- java.awt.MenuComponent
- java.lang.Object

MenuItem Example

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

AWTMenuDemo.java

```
package com.tutorialspoint.gui;
```

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

public class AWTMenuDemo {
   private Frame mainFrame;
   private Label headerLabel;
   private Label statusLabel;
   private Panel controlPanel;

   public AWTMenuDemo(){
      prepareGUI();
   }

   public static void main(String[] args){
      AWTMenuDemo  awtMenuDemo = new AWTMenuDemo();
      awtMenuDemo.showMenuDemo();
   }

   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 showMenuDemo(){
      //create a menu bar
      final MenuBar menuBar = new MenuBar();

      //create menus
      Menu fileMenu = new Menu("File");
      Menu editMenu = new Menu("Edit");
      final Menu aboutMenu = new Menu("About");

      //create menu items
      MenuItem newMenuItem =
         new MenuItem("New",new MenuShortcut(KeyEvent.VK_N));
      newMenuItem.setActionCommand("New");

      MenuItem openMenuItem = new MenuItem("Open");
      openMenuItem.setActionCommand("Open");

      MenuItem saveMenuItem = new MenuItem("Save");
      saveMenuItem.setActionCommand("Save");

      MenuItem exitMenuItem = new MenuItem("Exit");
      exitMenuItem.setActionCommand("Exit");

      MenuItem cutMenuItem = new MenuItem("Cut");
      cutMenuItem.setActionCommand("Cut");
```

```
      MenuItem copyMenuItem = new MenuItem("Copy");
      copyMenuItem.setActionCommand("Copy");

      MenuItem pasteMenuItem = new MenuItem("Paste");
      pasteMenuItem.setActionCommand("Paste");

      MenuItemListener menuItemListener = new MenuItemListener();

      newMenuItem.addActionListener(menuItemListener);
      openMenuItem.addActionListener(menuItemListener);
      saveMenuItem.addActionListener(menuItemListener);
      exitMenuItem.addActionListener(menuItemListener);
      cutMenuItem.addActionListener(menuItemListener);
      copyMenuItem.addActionListener(menuItemListener);
      pasteMenuItem.addActionListener(menuItemListener);

      final CheckboxMenuItem showWindowMenu =
         new CheckboxMenuItem("Show About", true);
      showWindowMenu.addItemListener(new ItemListener() {
         public void itemStateChanged(ItemEvent e) {
            if(showWindowMenu.getState()){
               menuBar.add(aboutMenu);
            }else{
               menuBar.remove(aboutMenu);
            }
         }
      });

      //add menu items to menus
      fileMenu.add(newMenuItem);
      fileMenu.add(openMenuItem);
      fileMenu.add(saveMenuItem);
      fileMenu.addSeparator();
      fileMenu.add(showWindowMenu);
      fileMenu.addSeparator();
      fileMenu.add(exitMenuItem);

      editMenu.add(cutMenuItem);
      editMenu.add(copyMenuItem);
      editMenu.add(pasteMenuItem);

      //add menu to menubar
      menuBar.add(fileMenu);
      menuBar.add(editMenu);
      menuBar.add(aboutMenu);

      //add menubar to the frame
      mainFrame.setMenuBar(menuBar);
      mainFrame.setVisible(true);
}

   class MenuItemListener implements ActionListener {
      public void actionPerformed(ActionEvent e) {
         statusLabel.setText(e.getActionCommand()
            + " MenuItem clicked.");
      }
   }
}
```

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

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

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

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

Verify the following output. *ClickonFileMenu*. *Selectanymenuitem*.