

SWING - JMENUITEM CLASS

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

Copyright © tutorialspoint.com

Introduction

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

Class declaration

Following is the declaration for **javax.swing.JMenuItem** class:

```
public class JMenuItem
    extends AbstractButton
    implements Accessible, MenuElement
```

Class constructors

S.N.	Constructor & Description
1	JMenuItem Creates a JMenuItem with no set text or icon.
2	JMenuItemAction Creates a menu item whose properties are taken from the specified Action.
3	JMenuItemIcon Creates a JMenuItem with the specified icon.
4	JMenuItemStringtext Creates a JMenuItem with the specified text.
5	JMenuItemStringtext, Icon Creates a JMenuItem with the specified text and icon.
6	JMenuItemStringtext, int mnemonic Creates a JMenuItem with the specified text and keyboard mnemonic.

Class methods

S.N.	Method & Description
1	protected void actionPerformed(ActionEvent, String propertyName) Updates the button's state in response to property changes in the associated action.
2	void addMenuDragMouseListener(MenuDragMouseListener)

Adds a MenuDragMouseListener to the menu item.

3 **void addMenuKeyListener***MenuKeyListenerl*

Adds a MenuKeyListener to the menu item.

4 **protected void configurePropertiesFromAction***Actiona*

Sets the properties on this button to match those in the specified Action.

5 **protected void fireMenuDragMouseDragged***MenuDragMouseEventevent*

Notifies all listeners that have registered interest for notification on this event type.

6 **protected void fireMenuDragMouseEntered***MenuDragMouseEventevent*

Notifies all listeners that have registered interest for notification on this event type.

7 **protected void fireMenuDragMouseExited***MenuDragMouseEventevent*

Notifies all listeners that have registered interest for notification on this event type.

8 **protected void fireMenuDragMouseReleased***MenuDragMouseEventevent*

Notifies all listeners that have registered interest for notification on this event type.

9 **protected void fireMenuKeyPressed***MenuKeyEventevent*

Notifies all listeners that have registered interest for notification on this event type.

10 **protected void fireMenuKeyReleased***MenuKeyEventevent*

Notifies all listeners that have registered interest for notification on this event type.

11 **protected void fireMenuKeyTyped***MenuKeyEventevent*

Notifies all listeners that have registered interest for notification on this event type.

12 **KeyStroke getAccelerator**

Returns the KeyStroke which serves as an accelerator for the menu item.

13 **AccessibleContext getAccessibleContext**

Returns the AccessibleContext associated with this JMenuItem.

14 **Component getComponent**

Returns the java.awt.Component used to paint this object.

15 **MenuDragMouseListener[] getMenuDragMouseListeners**

Returns an array of all the MenuDragMouseListeners added to this JMenuItem with addMenuDragMouseListener.

16 **MenuKeyListener[] getMenuKeyListeners**

Returns an array of all the MenuKeyListeners added to this JMenuItem with addMenuKeyListener.

17 **MenuElement[] getSubElements**

This method returns an array containing the sub-menu components for this menu component.

18 **String getUIClassID**

Returns the suffix used to construct the name of the L&F class used to render this component.

19 **protected void initStringtext, Iconicon**

Initializes the menu item with the specified text and icon.

20 **boolean isArmed**

Returns whether the menu item is "armed".

21 **void menuSelectionChangedbooleanisIncluded**

Called by the MenuSelectionManager when the MenuElement is selected or unselected.

22 **protected String paramString**

Returns a string representation of this JMenuItem.

23 **void processKeyEventKeyEvente, MenuElement[]path, MenuSelectionManagermanager**

Processes a key event forwarded from the MenuSelectionManager and changes the menu selection, if necessary, by using MenuSelectionManager's API.

24 **void processMenuDragMouseEventMenuDragMouseEvente**

Handles mouse drag in a menu.

25 **void processMenuKeyEventMenuKeyEvente**

Handles a keystroke in a menu.

26 **void processMouseEventMouseEvente, MenuElement[]path, MenuSelectionManagermanager**

Processes a mouse event forwarded from the MenuSelectionManager and changes the menu selection, if necessary, by using the MenuSelectionManager's API.

27 **void removeMenuDragMouseListenerMenuDragMouseListenerl**

Removes a MenuDragMouseListener from the menu item.

28 **void removeMenuKeyListenerMenuKeyListenerl**

Removes a MenuKeyListener from the menu item.

29 **void setAcceleratorKeyStrokekeyStroke**

Sets the key combination which invokes the menu item's action listeners without navigating the menu hierarchy.

30 **void setArmed***booleanb*

Identifies the menu item as "armed".

31 **void setEnabled***booleanb*

Enables or disables the menu item.

32 **void setModel***ButtonModelnewModel*

Sets the model that this button represents.

33 **void setUI***MenuItemUIui*

Sets the look and feel object that renders this component.

34 **void updateUI**

Resets the UI property with a value from the current look and feel.

Methods inherited

This class inherits methods from the following classes:

- javax.swing.JAbstractButton
- javax.swing.JComponent
- java.awt.Container
- java.awt.Component
- java.lang.Object

JMenuItem Example

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

SwingMenuDemo.java

```
package com.tutorialspoint.gui;

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

public class SwingMenuDemo {
    private JFrame mainFrame;
    private JLabel headerLabel;
    private JLabel statusLabel;
    private JPanel controlPanel;

    public SwingMenuDemo(){
        prepareGUI();
    }

    public static void main(String[] args){
        SwingMenuDemo swingMenuDemo = new SwingMenuDemo();
        swingMenuDemo.showMenuDemo();
    }
}
```

```

}

private void prepareGUI(){
    mainFrame = new JFrame("Java SWING Examples");
    mainFrame.setSize(400,400);
    mainFrame.setLayout(new GridLayout(3, 1));

    headerLabel = new JLabel("",JLabel.CENTER );
    statusLabel = new JLabel("",JLabel.CENTER);

    statusLabel.setSize(350,100);
    mainFrame.addWindowListener(new WindowAdapter() {
        public void windowClosing(WindowEvent windowEvent){
            System.exit(0);
        }
    });
    controlPanel = new JPanel();
    controlPanel.setLayout(new FlowLayout());

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

private void showMenuDemo(){
    //create a menu bar
    final JMenuBar menuBar = new JMenuBar();

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

    //create menu items
    JMenuItem newMenuItem = new JMenuItem("New");
    newMenuItem.setMnemonic(KeyEvent.VK_N);
    newMenuItem.setActionCommand("New");

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

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

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

    JMenuItem cutMenuItem = new JMenuItem("Cut");
    cutMenuItem.setActionCommand("Cut");

    JMenuItem copyMenuItem = new JMenuItem("Copy");
    copyMenuItem.setActionCommand("Copy");

    JMenuItem pasteMenuItem = new JMenuItem("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 JCheckBoxMenuItem showWindowMenu =

```

```

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

final JRadioButtonMenuItem showLinksMenu =
    new JRadioButtonMenuItem("Show Links", true);
showLinksMenu.addItemListener(new ItemListener() {
    public void itemStateChanged(ItemEvent e) {
        if(menuBar.getMenu(3)!= null){
            menuBar.remove(linkMenu);
            mainFrame.repaint();
        }else{
            menuBar.add(linkMenu);
            mainFrame.repaint();
        }
    }
});

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

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

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

```

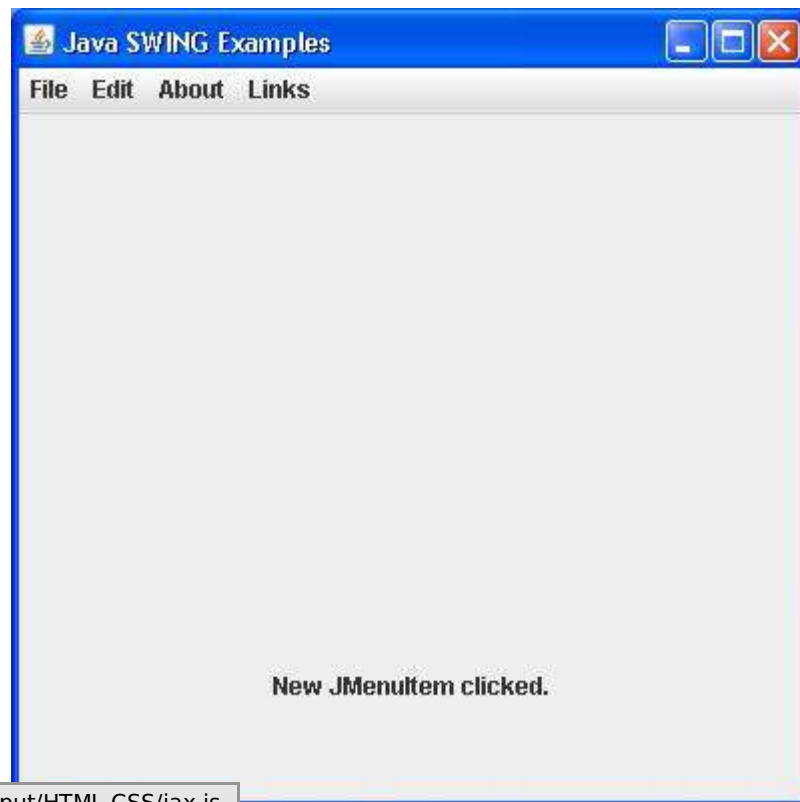
Compile the program using command prompt. Go to **D:/ > SWING** and type the following command.

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

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

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

Verify the following output. *Click on File Menu. Select any menu item.*



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