Java swing tutorials

Java swing is part of JFC (Java foundation classes ) and abstraction layer on AWT to create window based application.

Unlike AWT, Java Swing is platform independent and light weight component.

Java AWT Java Swing

1) AWT components are platform-dependent. Java swing components are platform-independent.

2) AWT components are heavyweight. Swing components are lightweight.

3) AWT doesn't support pluggable look and feel. Swing supports pluggable look and feel.

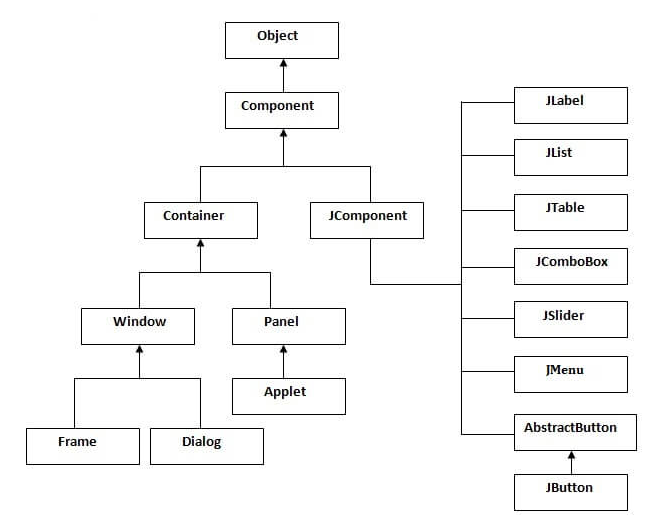
4) AWT provides less components than Swing. Swing provides more powerful components such as tables, lists, scrollpanes, colorchooser, tabbedpane etc.

5) AWT doesn't follows MVC(Model View Controller) where model represents data, view represents presentation and controller acts as an interface between model and view. Swing follows MVC.

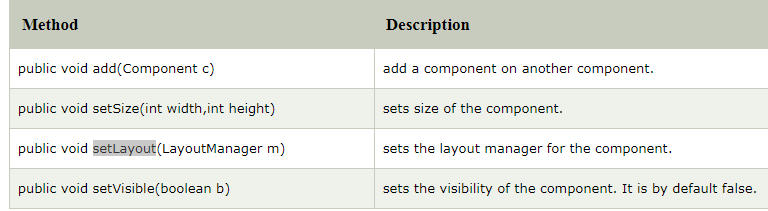
1.)What is JFC?

Java foundation classes are set of GUI components which simplify to create desktop based apploication.

2.Hierarchy of Java swing classes?



3.Commonly used methods of Component class?



4.)There are two ways to create a swing class

a.By extending the class from JFrame class(Inheritance)

b.by creaeting an object of JFrame class(association)

Simple example of Swing using association where we are creating one button and adding it in Frame classs.

**import** javax.swing.\*;

**public** **class** FirstSwingExample {

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

JFrame f=**new** JFrame();//creating instance of JFrame

JButton b=**new** JButton("click");//creating instance of JButton

b.setBounds(130,100,100, 40);//x axis, y axis, width, height

f.add(b);//adding button in JFrame

f.setSize(400,500);//400 width and 500 height

f.setLayout(**null**);//using no layout managers

f.setVisible(**true**);//making the frame visible

}

}

Simple example of swing using inheritance

**import** javax.swing.\*;

**public** **class** Simple2 **extends** JFrame{//inheriting JFrame

JFrame f;

Simple2(){

JButton b=**new** JButton("click");//create button

b.setBounds(130,100,100, 40);

add(b);//adding button on frame

setSize(400,500);

setLayout(**null**);

setVisible(**true**);

}

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

**new** Simple2();

}}

### 5.) Java LayoutManagers

The LayoutManagers are used to arrange components in a particular manner. LayoutManager is an interface that is implemented by all the classes of layout managers. There are following classes that represents the layout managers:

1. java.awt.BorderLayout—Region wise
2. java.awt.FlowLayout—in a line ,one after another
3. java.awt.GridLayout-- arrange the component in Rectangular Grid
4. java.awt.CardLayout-- only one component is visible at a time
5. java.awt.GridBagLayout
6. GridBagLayout class is used to align components vertically, horizontally or along their baseline
7. javax.swing.BoxLayout-- vertically or horizontally
8. javax.swing.GroupLayout
9. javax.swing.ScrollPaneLayout

Default layout manager of window of frame is BoderLayout which arrange the components five region wise.

Default layout manager of Panelor Applet is Flowlayout.

JavaFX is advance of Swing.

JavaFX has rich GUI library to develop Rich internet application .The application developed in JavaFX can run any devices such as mobile desktop,TV ,tablet.

6.) What Is An Event In Swing?

Ans:-Changing an state of an object is called Event.

7.)what is Event handling?

Event handling is a part of program created to act in a response to specific event.

8. Which Method Of The Component Class Is Used To Set The Position And Size Of A Component?

Answer : setBounds

9.How to develop a program for event handling in swing?

Ans-To give the functionality or life to GUI interface in application ,we should remember the things given below.

1.Register the component with listener

2.Event Object

3.Implement the listener

4.Override the method

1.For Each component ,there is method to register or deregister

For example

|  |  |  |
| --- | --- | --- |
| Active Component | Registered Method | Un-Registered method |
| Label | addActionaListener | removeActionaListener |
| Button | addActionaListener | removeActionaListener |
| TextArea | addTextListener | removeTextListener |
| TextField | addTextListener | removeTextListener |
| CheckBox | addItemListener | removeItemListener |
| ScrollBar | addAdjustmentListener | removeAdjustmentListerner |

Button—ActionEvent

ActionListener

actionPerformed

addActionListener

removeActionListener(Action Listener)

CheckBox—ItemEvent

ItemListener

itemStateChanged(Item Event)

addItemListener(ItemListener)

removeItemListener(ItemListener)