**Java Swing**

**Java Swing** is a part of Java Foundation Classes (JFC) that is *used to create window-based applications*. It is built on the top of AWT (Abstract Windowing Toolkit) API and entirely written in java.

Unlike AWT, Java Swing provides platform-independent and lightweight components.

The javax.swing package provides classes for java swing API such as JButton, JTextField, JTextArea, JRadioButton, JCheckbox, JMenu, JColorChooser etc.

### **What is JFC**

### The Java Foundation Classes (JFC) are a set of GUI components which simplify the development of desktop applications.

### **Hierarchy of Java Swing classes**

### hierarchy of javax swing

### **Commonly used Methods of Component class**

The methods of Component class are widely used in java swing that are given below.

|  |  |
| --- | --- |
| **Method** | **Description** |
| public void add(Component c) | add a component on another component. |
| public void setSize(int width,int height) | sets size of the component. |
| public void setLayout(LayoutManager m) | sets the layout manager for the component. |
| public void setVisible(boolean b) | sets the visibility of the component. It is by default false. |

## **Java Swing Examples**

There are two ways to create a frame:

* By creating the object of Frame class (association)
* By extending Frame class (inheritance)

We can write the code of swing inside the main(), constructor or any other method.

Example

**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 java swing

### **Example of Swing by Association inside constructor**

**import** javax.swing.\*;

**public** **class** Simple

{

JFrame f;

Simple(){

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

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

b.setBounds(130,100,100, 40);

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

}

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

**new** Simple();

}

}

The setBounds(int xaxis, int yaxis, int width, int height)is used in the above example that sets the position of the button.

### **Simple example of Swing by inheritance**

We can also inherit the JFrame class, so there is no need to create the instance of JFrame class explicitly.

*File: Simple2.java*

**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();

}}

* JButton class
* JRadioButton class
* JTextArea class
* JComboBox class
* JTable class
* JColorChooser class
* JProgressBar class
* JSlider class
* Digital Watch
* Graphics in swing
* Displaying image
* Edit menu code for Notepad
* OpenDialog Box
* Notepad
* Puzzle Game
* Pic Puzzle Game
* Tic Tac Toe Game
* BorderLayout
* GridLayout
* FlowLayout
* CardLayout