

GUI in Java

A graphical user interface (GUI) application in Java provides a visual way for users to interact with software through graphical elements like windows, buttons, and menus. Basically Java offer two framework for GUI application:

1. AWT
2. Swing

GUI Application are the combination of :

Container : Window, Frame, Panel

Component : Button, Textbox, Dropdown, TextArea, Radio Button, CheckBox, ScrollBar,

All the container and components are in the form of classes in the AWT and Swing package.

AWT :

The Abstract Window Toolkit (AWT) is Java's original GUI toolkit, providing a set of platform-dependent components for building user interfaces. It is part of the Java Foundation Classes (JFC). AWT relies on the underlying operating system to create and manage UI elements, which means that AWT applications have a native look and feel. However, this also makes them platform-dependent, as the appearance and behavior of components can vary across different operating systems.

Package : java.awt

Swing:

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.

Package : Javax.swing

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

Difference between AWT and Swing

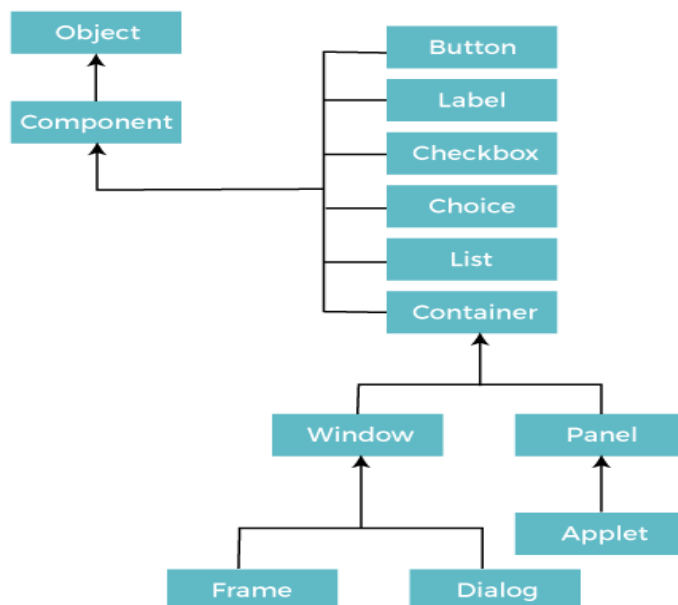
There are many differences between java awt and swing that are given below.

No.	Java AWT	Java Swing
1)	AWT components are platform-dependent . : In every OS AWT components and Container looks different.	Java swing components are platform-independent .
2)	AWT components are heavyweight .	Swing components are lightweight .
3)	AWT doesn't support pluggable look and1 feel .	Swing supports pluggable look and feel .
4)	AWT provides fewer components than Swing.	Swing provides more powerful components such as tables, lists, scrollpanes, colorchooser, tabbedpane etc.
5)	AWT doesn't follow MVC (Model View Controller) where model represents data, view represents presentation and controller acts as an interface between model and view.	Swing follows MVC .

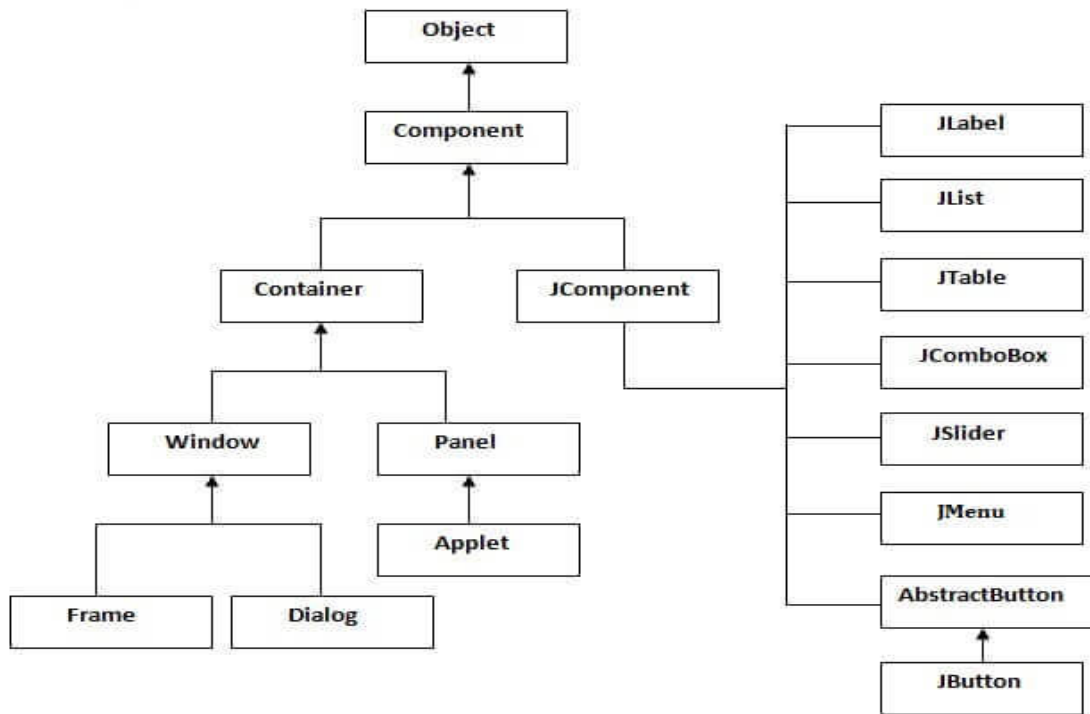
Java Foundation Classes(JFC)

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

AWT component :



Swing Component



JFrame, JPanel

Steps to create GUI Form in Swing

1. Import `javax.swing.*` package
2. Create object of the container class like `JFrame`
e.g. `JFrame f=new JFrame("Title");`
3. Create objects of various component which need to add in container
e.g. `JButton b=new JButton("click");`
4. Set Bounds of the all component. Bounds decides the location of component in the container. Use method :
`setBounds( x, y, width, height)`
e.g. `txt.setBounds(180,120, 120,50);`
5. Add all the component in the container
e.g. `objectOfFrame.add(objectOfComponent)`
6. Set the size of Container
e.g `objectOfFrame.setSize(width,height)`
7. Set the layout of container
e.g. `objectOfFrame.setLayout(null)`
8. Set the visibility true for container
e.g. `objectOfFrame.setVisible(true);`

AWT Example :

```
import java.awt.*;

public class AWT_Example {

    public static void main(String[] args)

    {

        Frame frame = new Frame("Basic Program");

        Label label = new Label("Hello World!");

        label.setBounds(50,50,50,50);

        frame.add(label);

        frame.setSize(300, 300);

        frame.setVisible(true);

    }

}
```

Swing Example

```
import javax.swing.*;

public class FirstSwing
{
    public static void main(String[] args)
    {
        JFrame f=new JFrame(); //creating instance of JFrame

        JButton b=new JButton("click");           //creating instance of JButton
        JTextField txt = new JTextField();
        b.setBounds(130,100,100, 40); //x axis, y axis, width, height
        txt.setBounds(180,120, 120,50);

        f.add(b); //adding button in JFrame
        f.add(txt);
```

```
f.setSize(400,500);           //400 width and 500 height
f.setLayout(null);
f.setVisible(true);           //making the frame visible
}
}
```

Example 2:

```
import javax.swing.*;
```

```
public class LoginForm {
```

```
    public static void main(String args[])
    {
        JFrame jf = new JFrame();
```

```
        JTextField jt1 = new JTextField();
        JTextField jt2 = new JTextField();
        JButton jb = new JButton("Click");
        jb.setBounds(120, 200, 100, 40);
        jt1.setBounds(120,90,100,50);
        jt2.setBounds(120,140,100,50);
```

```
        jf.add(jb);
        jf.add(jt1);
        jf.add(jt2);
        jf.setSize(400,400);
        jf.setLayout(null);
        jf.setVisible(true);
    }
```

```
}
```