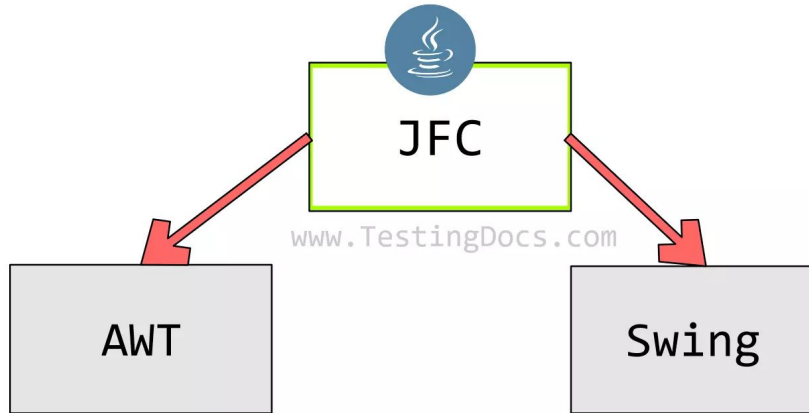


GUI & Event Handling

Ratna Mufidah, S.Kom., M.Kom.

Graphical User Interface (GUI) in Java



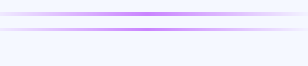
- The **Java Foundation Classes (JFC)** consists of a set of **application programming interfaces (APIs)**.
- **JFC** provides support for developing **functional graphical user interfaces (GUIs)**.



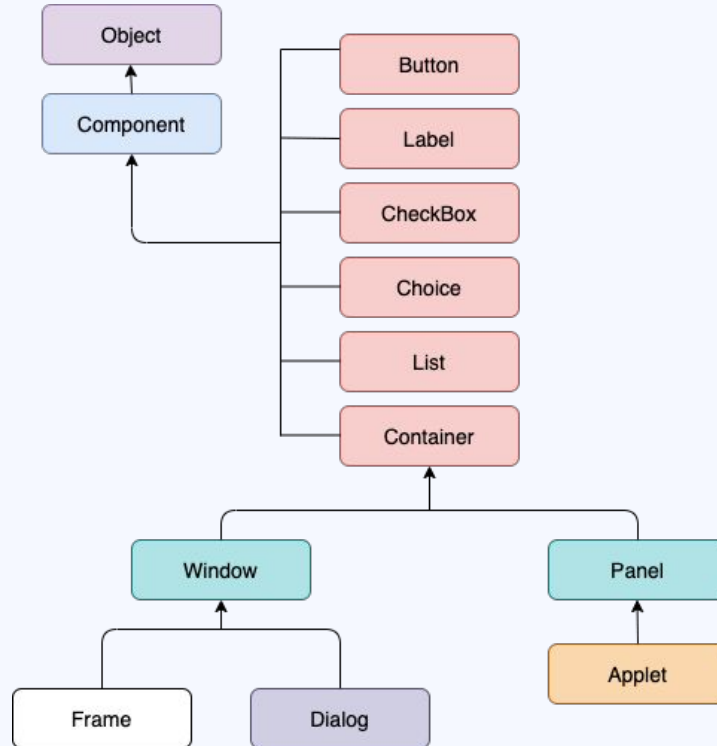
Java AWT

AWT stands for **Abstract window toolkit** is an **Application programming interface (API)** for creating **Graphical User Interface (GUI)** in Java.

Java AWT components are **platform-dependent** i.e. components are displayed according to the view of operating system. **AWT is heavy weight** i.e. its components are using the resources of underlying operating system (OS).



Java AWT



<https://www.javatpoint.com/awt-program-in-java>


Java AWT

- Package: **java.awt**
- **Components**: All the elements like the button, text fields, scroll bars, etc.
- **Container**: a component in AWT that can contain another components like buttons, textfields, labels etc.
The classes that extends Container class are known as container such as **Frame**, **Dialog** and **Panel**.



Java AWT

Container Types

- **Window:** the container that have no borders and menu bars. You must use frame, dialog or another window for creating a window.
 - **Panel:** the container that doesn't contain title bar, border or menu bar. It is generic container for holding the components. It can have other components like button, text field etc.
- 

Java AWT

Container Types

- **Frame**: the container that contain title bar and border and can have menu bars. It can have other components like button, text field, scrollbar etc.
- **Dialog**: the container represents a top level window with a border and a title used to take some form of input from the user. Frame and Dialog both inherits Window class. Frame has maximize and minimize buttons but Dialog doesn't have.

Java AWT

AwtTest1.java

```
1 import java.awt.*;
2
3 public class AwtTest1 {
4     public AwtTest1()
5     {
6         Frame f = new Frame();
7         Button btn=new Button("Button Test");
8         btn.setBounds(80, 80, 100, 50);
9         f.add(btn);           //adding a new Button.
10        f.setSize(300, 250); //setting size.
11        f.setTitle("AWT Test"); //setting title.
12        f.setLayout(null);    //set default layout for frame.
13        f.setVisible(true);   //set frame visibility true.
14    }
15
16    public static void main(String[] args) {
17        // TODO Auto-generated method stub
18        //creating a frame.
19        AwtTest1 awt = new AwtTest1();
20    }
21 }
```


Java AWT

AwtTest2.java

```
1 import java.awt.*;
3
4 public class AwtTest2 extends WindowAdapter {
5     Frame f;
6
7     public AwtTest2()
8     {
9         f = new Frame();
10        f.addWindowListener (this);
11        Button btn=new Button("Button Test");
12        btn.setBounds(80, 80, 100, 50);
13        f.add(btn);           //adding a new Button.
14        f.setSize(300, 250); //setting size.
15        f.setTitle("AWT Test"); //setting title.
16        f.setLayout(null);    //set default layout for frame.
17        f.setVisible(true);   //set frame visibility true.
18
19    }
20
21    public void windowClosing (WindowEvent e) {
22        f.dispose();
23    }
```

```
24
25 public static void main(String[] args) {
26     // TODO Auto-generated method stub
27     //creating a frame.
28     AwtTest2 awt = new AwtTest2();
29 }
30 }
```



Java Event Handling

Event → Changing the state of an object (click on button, dragging mouse etc)

Package → **java.awt.event**

Event handling → **event** and **listener**



Java Event Handling

Event Classes	Listener Interfaces
ActionEvent	ActionListener
MouseEvent	MouseListener and MouseMotionListener
MouseWheelEvent	MouseWheelListener
KeyEvent	KeyListener
ItemEvent	ItemListener
TextEvent	TextListener

Java Event Handling


Event Classes	Listener Interfaces
AdjustmentEvent	AdjustmentListener
WindowEvent	WindowListener
ComponentEvent	ComponentListener
ContainerEvent	ContainerListener
FocusEvent	FocusListener



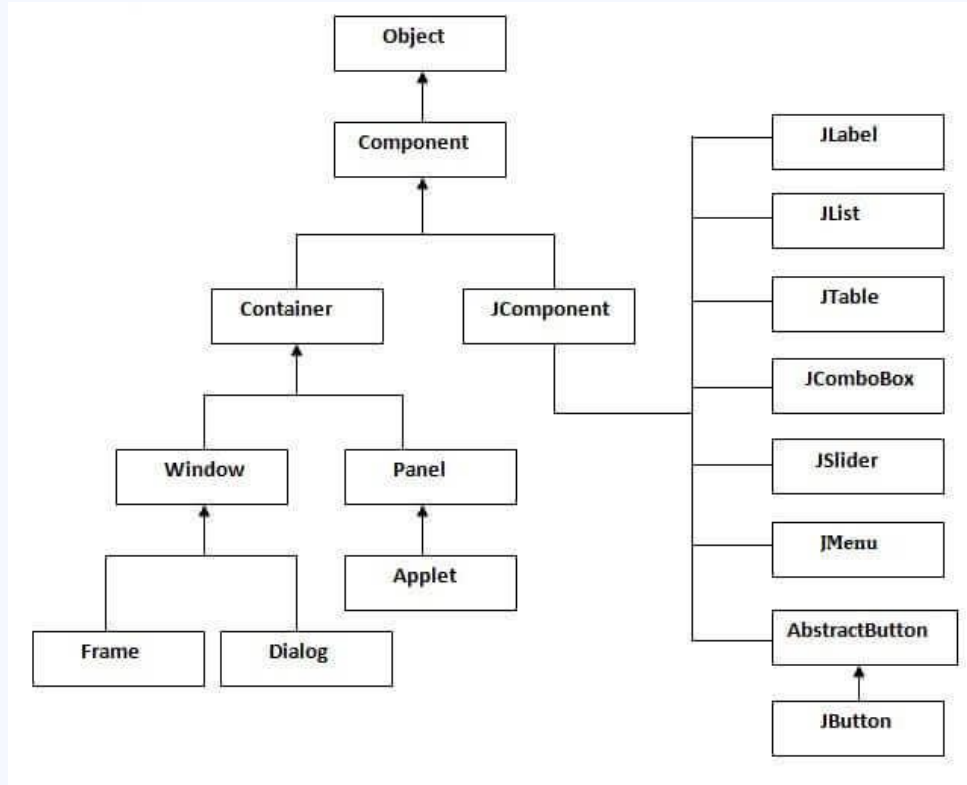
Java Swing

Swing is a **Java Foundation Classes (JFC)** library and an **extension of the Abstract Window Toolkit (AWT)**.

Swing offers **much-improved functionality over AWT**, new components, expanded components features, and excellent event handling with drag-and-drop support.



Java Swing



Java AWT **VS** Java Swing

Java AWT	Java Swing
AWT components are platform-dependent .	Java swing components are platform-independent .
AWT components are heavyweight .	Swing components are lightweight .
AWT doesn't support pluggable look and feel .	Swing supports pluggable look and feel .

Java AWT **VS** Java Swing



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



Thanks !

Ada pertanyaan?

CREDITS: This presentation template was created by [Slidesgo](#), and includes icons by [Flaticon](#), and infographics & images by [Freepik](#)



References

<https://www.geeksforgeeks.org/>

<https://www.w3schools.com/>

<https://www.javatpoint.com/>

<https://www.codejava.net/>

<https://javadevwannabe.blogspot.com/>

