Object-Oriented Programming (CS F213)
Labsheet-8

Prof.R Gururaj

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.

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

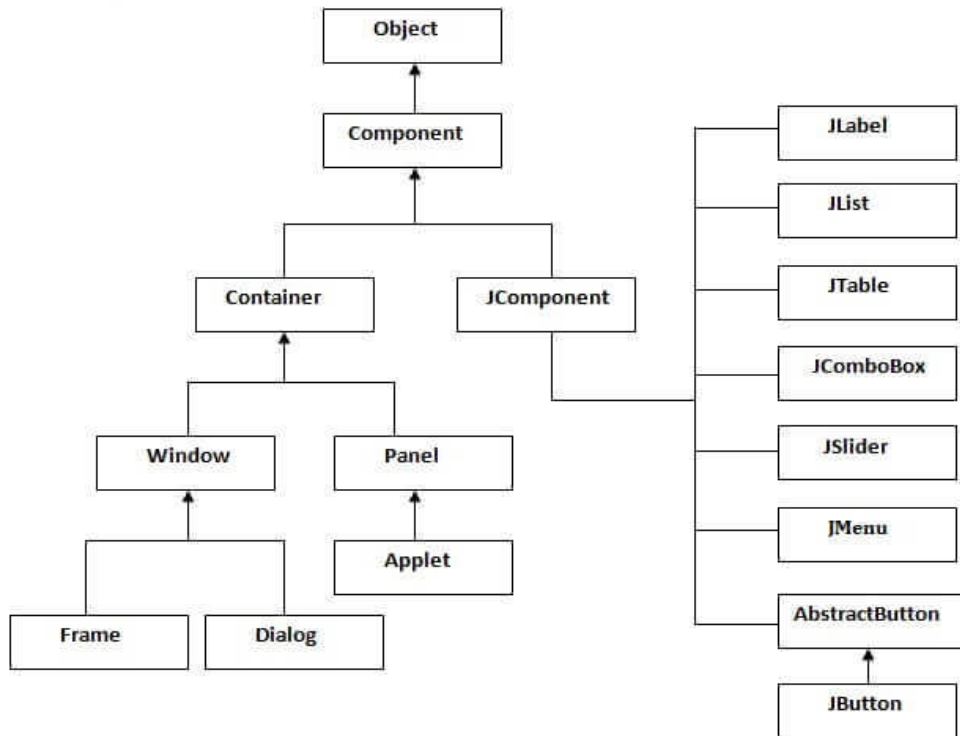Unlike AWT, Java Swing provides platform-independent and lightweight components.

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

Differences between java AWT and swing that are given below.

1.Java swing components are **platform-independent**.

2.Swing components are **lightweight**.

3. Swing provides **more powerful components** such as tables, lists, scrollpanes, colorchooser, tabbedpane etc.

4. Swing **follows MVC**.

where model represents data, view represents presentation and controller acts as an interface between model and view.

Hierarchy of Java Swing classes

```
                        ┌──────────┐
                        │  Object  │
                        └──────────┘
                             ▲
                        ┌──────────┐              ┌──────────┐
                        │Component │              │  JLabel  │
                        └──────────┘              └──────────┘
                             ▲
              ┌──────────────┴──────────────┐      ┌──────────┐
                                                    │  JList   │
         ┌───────────┐            ┌─────────────┐  └──────────┘
         │ Container │            │ JComponent  │
         └───────────┘            └─────────────┘  ┌──────────┐
              ▲                                      │  JTable  │
      ┌───────┴───────┐                             └──────────┘
                                                    ┌───────────┐
  ┌──────────┐    ┌──────────┐                      │JComboBox  │
  │  Window  │    │  Panel   │                      └───────────┘
  └──────────┘    └──────────┘                      ┌──────────┐
       ▲               ▲                            │ JSlider  │
       │          ┌──────────┐                      └──────────┘
       │          │  Applet  │                      ┌──────────┐
       │          └──────────┘                      │  JMenu   │
  ┌────────┬────────┐                               └──────────┘
  │ Frame  │ Dialog │                          ┌──────────────┐
  └────────┴────────┘                          │AbstractButton│
                                               └──────────────┘
                                                      ▲
                                               ┌──────────┐
                                               │ JButton  │
                                               └──────────┘
```

Practice problem-1 // Demonstrating JFrame and JButton

```java
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
        }
}
```

Practice Problem-2

Write a Java program (code snippet) that takes a string as commandline argument that indicates the zipcode of cities. The pattern is- any zipcode starts with first digit between 1-7 and has digit between 0-9 as other 5 digits, and in total 6 digits. Now write a class Demo that has *public static void main()* and checks if the given Zipcode is valid or not using Java regex package. If it is valid print valid zipcode hence invalid zipcode.

Practice problem-3

We plan to implement a *doubly linked list*. Now, write a program with all necessary classes and methods to implement a doubly linked list. Each **Node** in the list has a similar structure with one *int* type member with name *val*, along with necessary pointers-prev and next.. The Linked list will have operations- *void append*(*int val*) that takes an integer representing the node value and creates a node and appends the new node at the end of the list, the method *void insert(int val, int index)* creates a node with given *val* and inserts the node at the index mentioned as the 2nd argument. Note that the index starts from 0 (first node in the list). The last method is *void printList()* will print the values associated with the nodes from the start to end. You are required to design these methods with appropriate signatures to facilitate the given functionality. Also define a class **ListDemo** with *public static void main()* that instantiates the list and appends 26, 40, and inserts 30 at index 1in same order and then prints the list. There is no max cap for the list.
// define a node structure using Node class

Exercise

Exercise-1:
Now write code for insert() method for Doubly Linked List.

Exercise-2
Write a Java program (code snippet) that takes two string as commandline arguments. The first is the string and the second is the delimiter. Now tokenize the string (first argument) using the delimiter (second argument) and print the list of tokens obtained. Use *StringTokenizer*.

******