Module-4

GUI programming with Java

Java provides two packages (API) to develop GUI programs.

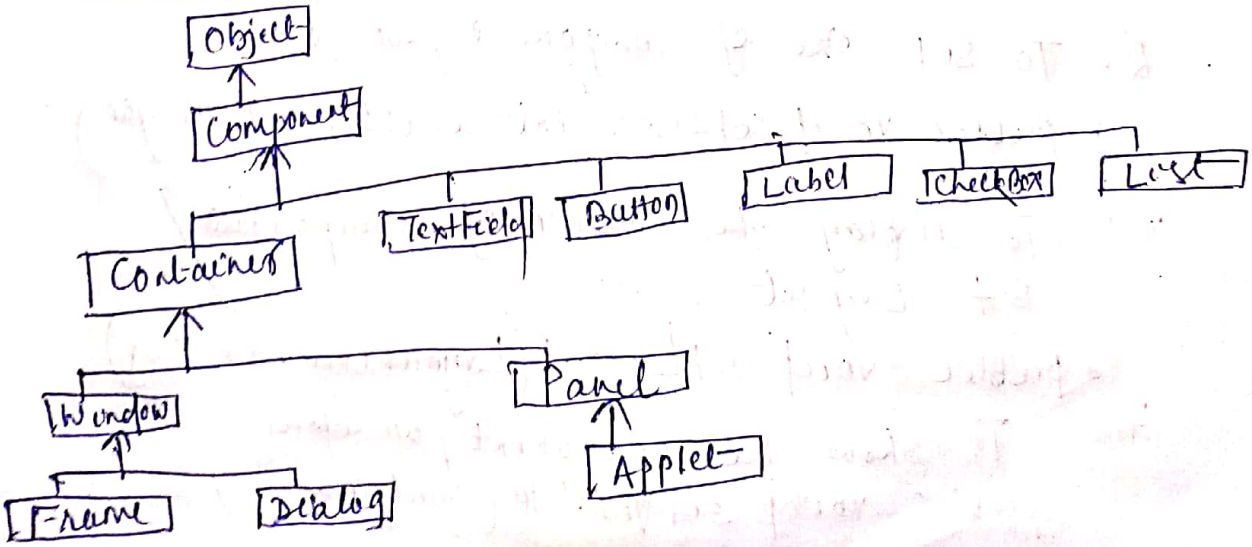
1. awt package (Abstract window Toolkit)
2. swing package

AWT

It is an API to develop GUI or window-based applications in java.

- Java AWT components are platform-dependent ie components are displayed according to the view of operating system.
- AWT is heavy weight- ie components use the resources of operating system.
- The java.awt package provides classes for AWT api such as TextField, Label, Button, RadioButton, CheckBox, Choice, TextArea etc.

AWT Hierarchy



Container

The Container is a component that can contain another component like Button, TextField, Label etc. The classes which extends (inherit) container class are also containers like Frame, Applet etc.

Window

It is also a container which has no border and menubar.

Panel:

It is also a container without having title bar and menubar.
It can have other components like Button, TextField etc.

Frame

It is a container which can have title bar and menubar. It can contain other components Button, TextField etc.

Methods of Component class.

1. To add a component to a container
   - public void add(Component object)
2. To set size of component/window.
   - public void setSize(int width, int height)
3. To display the order of component/ set Layout.
   - public void setLayout(LayoutManager obj)
4. To show the component/window
   public void setVisible(boolean b)

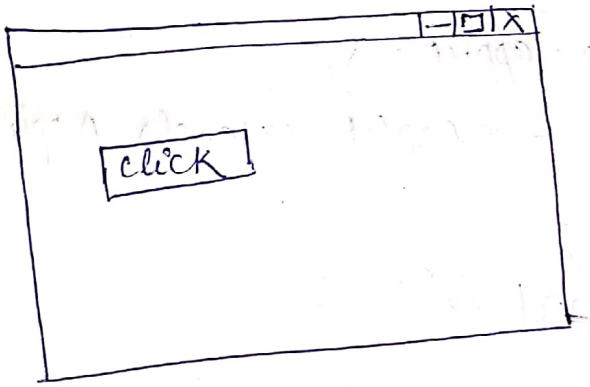
Note
These methods must be called through the component object while designing GUI.

How to create a GUI ? (In two ways)

1— By extending or inheriting Frame class.
or
2— By creating the object of Frame class (Association)

1— By extending Frame class.

Exa create a GUI as Follows.



→ click is a button
All the classes belong to awt package. So import—

```
import java.awt.*;

class GUI1 extends Frame
{
  GUI1()
  { Button b = new Button("click");
    setLayout(new FlowLayout());
    add(b);
    setSize(300, 400);
```

```
setVisible(true);
}
public static void main(String[] s)
{ GUI1 G = new GUI1();
}
}
```

Note

Same program can also be created through Applet.

Exa

```
import java.awt.*;
import java.applet.*;
public class GUIApplet extends Applet
{ Button b;
  public void init()
  { b=new Button("Click");
    add(b);
    setLayout(new FlowLayout());
    setVisible(true);
  }
}
/* <applet code="GUIApplet" width=300 height=400>
</applet> */
```

Note setSize() is not called bcoz width & height is already set by applet code.

1st program is java application program which contains main() method and in this program all component methods like add(), setSize() etc are called inside constructor level.

In the 2nd case (Applet) all the methods are called in init() method during initialization of applet. (It does not have main() method.)

2 By creating object of Frame class.

```
import java.awt.*;

class GUIDemo
{
    Button b;

    public static void main(String[] S)
    {
        Frame f = new Frame("My Frame");
        b = new Button("click");
        f.add(b);
        f.setLayout(new FlowLayout());
        f.setSize(300,400);
        f.setVisible(true);
    }
}
```

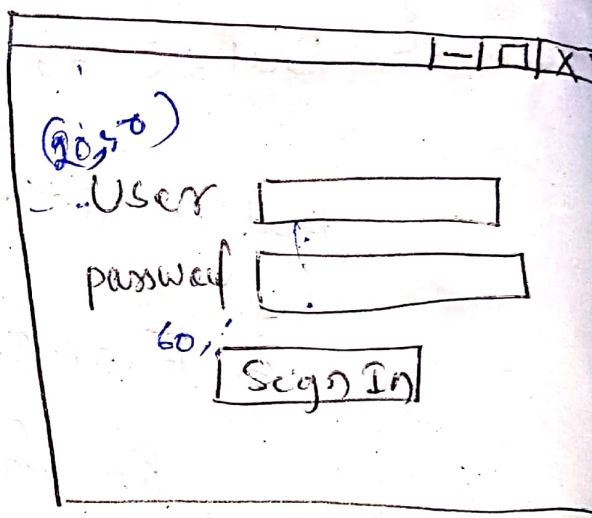
Explain

In the first case GUI1 class is the subclass of Frame class so GUI1 is also a Frame class. So all the component class methods are called directly in the constructor without creating object.

But in the 2nd case since Frame class object is created explicitly in that case all the component methods are called by Frame object f.

→ Whichever comfortable you adopt that way.

Design the log in page.



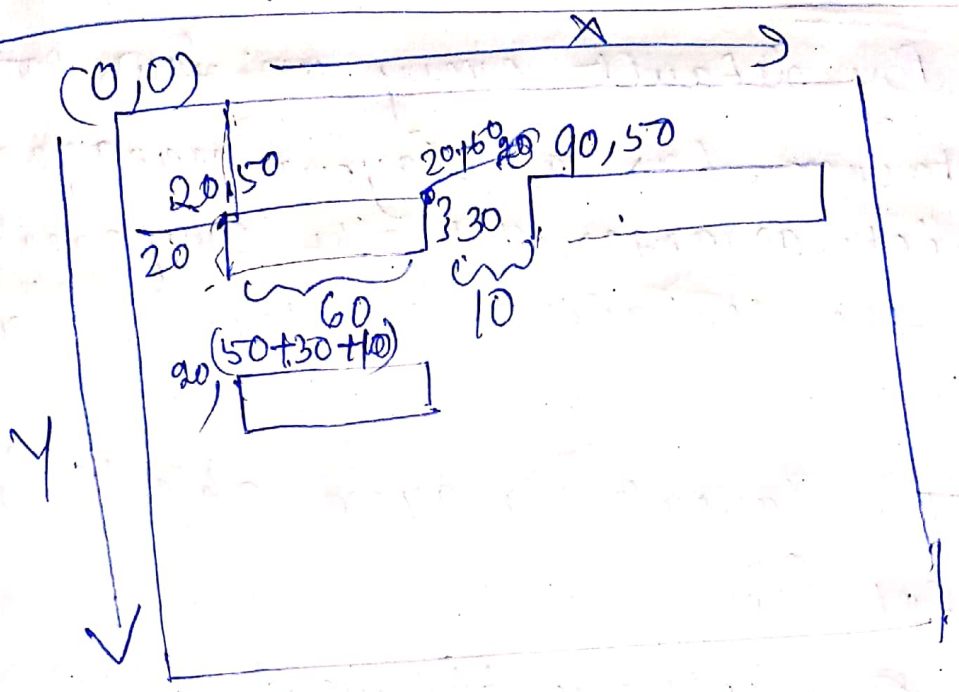
```
import java.awt.*;

class Login extends Frame
{  Label l1, l2;        //only declaration
   TextField t1, t2;
   Button b;

   Frame()
   {  l1 = new Label("User");
      l2 = new Label("password");
      t1 = new TextField(25);
      t2 = new TextField(25);
      b = new Button("SignIn");
// add the component in the order to display
      add(l1);
      add(t1);
      add(l2);
      add(t2);
      add(b1);
```

```
setLayout(null); // no layout Manager is used
// Give coordinate value to each component
    l1.setBounds(20,50,60,30);
    t1.setBounds(90,50,60,30);
    l2.setBounds(20,90,60,30);
    t2.setBounds(90,90,60,30);
    b.setBounds(60,140,40,30);
    setSize(500,500);
    setVisible(true);
}
public static void main(String[] s)
{
    new Login(); // Annonymous object — one time used
}
}
```

How to know the coordinate value



Initial coordinate is (0,0)

& TextField value coordinate

$X = 20 + 60 + 10 = 90$.

60 → width

10 → ~~differe~~ space

Y = same as label.

Accordingly you have to calculate all coordinate value.

setBounds(x, y, w, h)

x coord, y coord, w → width, h → height

Every ~~coordin~~ component has setBounds() method to give a location in window.

Note

— By default every ~~window~~ container has a layout manager which arranges all the components in the container in specific order.

— The layout manager object is set in

setLayout(LayoutManager) method. ie setLayout(new FlowLayout()); left to right

To display the components at a specific coordinate value setBounds() method is used.

For which setLayout() method must be nullify ie setLayout(null);

Swing

- Java swing provides API which is used to develop window based application. or GUI.
- It is a part of JFC (Java Foundation Classes)

* If AWT is there then why swing?

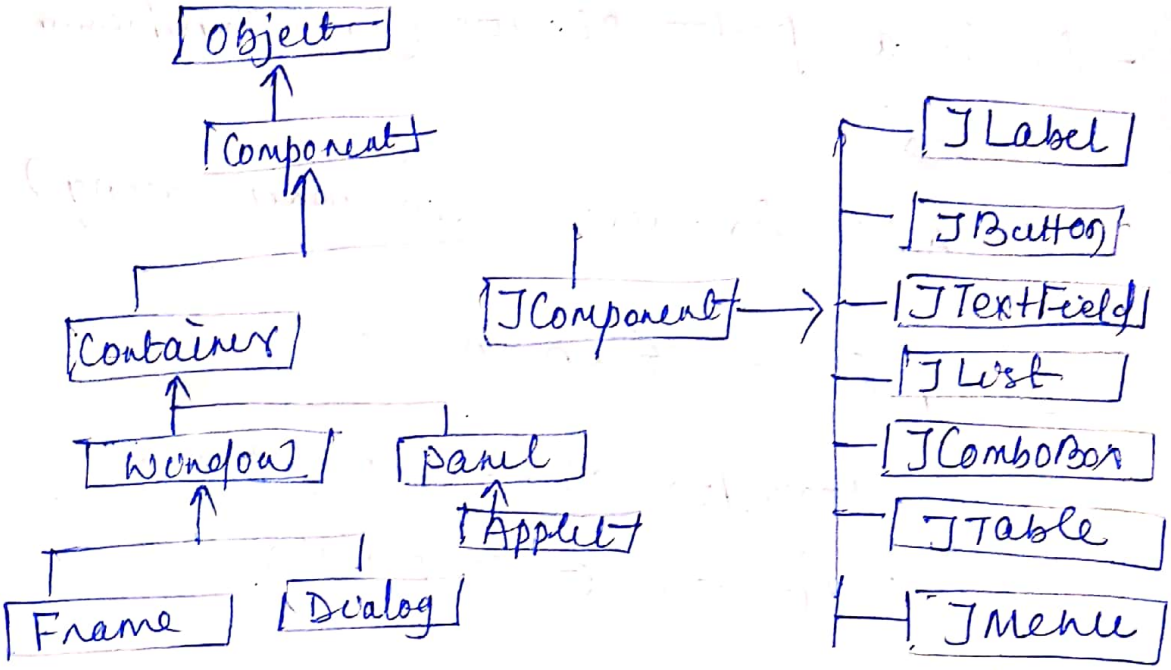
Swing is built on AWT.

Swing provides platform independent and light weight components.

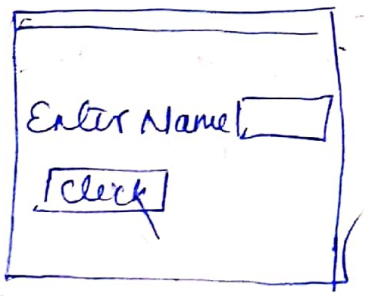
AWT	Swing
1- AWT components are platform dependent	1- Components are platform independent.
2- components are heavy weight	2- light weight.
3- Does not support pluggable look and feel	3 - provides pluggable look and Feel.
4- Less components are there.	4 - More components like table, lists, Scrollpane etc.
5 - Does not follow MVC model.	5 - Follow MVC model.

(Model) M - represents data
(View) V - " presentation
(Controller) C - " interface between model & view

In swing all the components are JComponent as follows



① Design the GUI



By swing also GUI can be created in 2 ways.

1- By extends JFrame class
Or
2 - By creating object of JFrame class

① Note

To use swing import—
javax.swing.*

```
import javax.swing.*;

class swingDemo extends JFrame
{
    JLabel l;
    JTextField t;
    JButton b;
    swingDemo()
    {
        l = new JLabel("Enter Name");
        t = new JTextField(25);
        b = new JButton("click");
    add(l); add(t); add(b);
    setLayout(new FlowLayout());
    setSize(500, 500);
    setVisible(true);
    }
  public static void main(String[] s)
  {
    swingDemo D = new swingDemo();
  }
}
```

size

↓

Arranges components from left to right