CS202: IT Workshop
Java

GUI

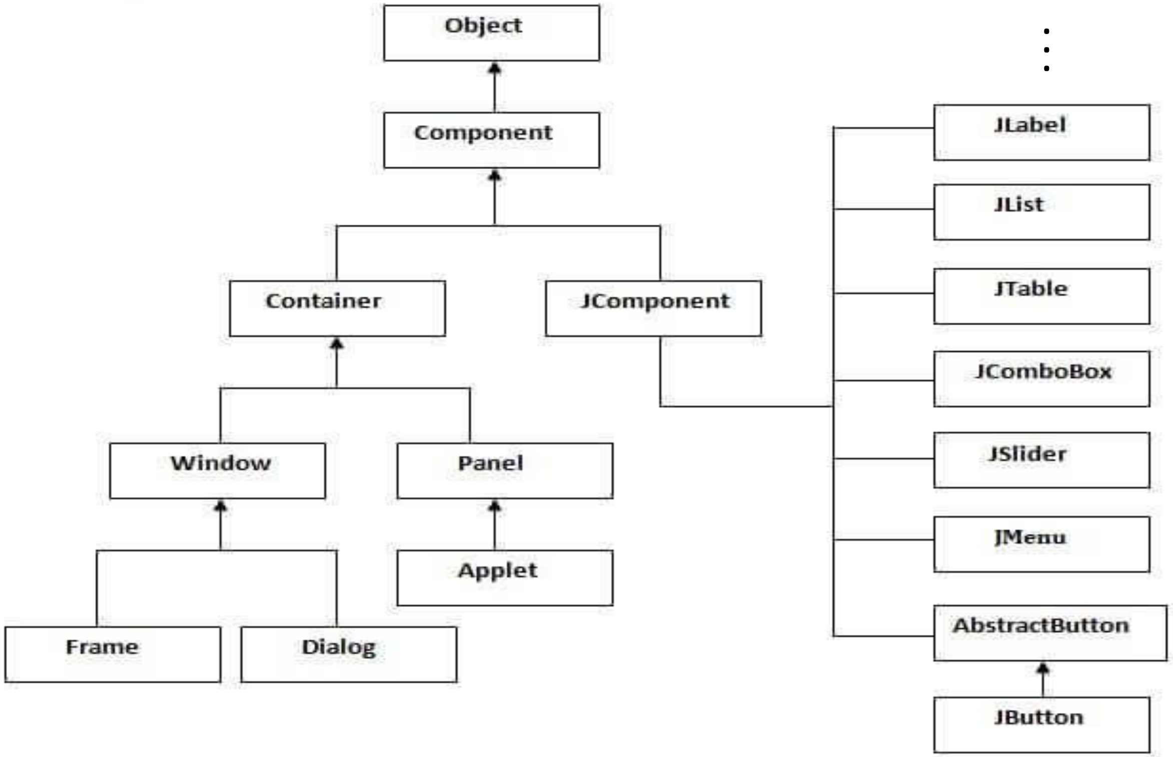
Ref:

1. Herb Schildt: Java The Complete Reference, 8/e Tata Mcgraw Hill Education.
2. Internet

Graphical User Interface in Java

- 3 sets of Java APIs for graphics programming:
  - **AWT (Abstract Windowing Toolkit):** basic one (java.awt)
  - **Swing**: built on AWT (javax.swing)
  - **JavaFX**: introduced recently in JDK 8 (javafx)
- Support from other organizations are also available: (Eclipse's Standard Widget Toolkit (SWT), Google Web Toolkit (GWT), Java bindings for OpenGL (JOGL) etc.)

Class hierarchy of swing



- ❑ There are two types of GUI elements:
  - ○ **Component**: elementary GUI entities (e.g. Button, Label, etc.)
  - ○ **Container**: used to hold components in a specific layout
- ❑ Components are to be added to the container for their appearance
- ❑ AWT also follows similar class hierarchy

Creating GUI

- We can create a frame by
  - Creating an instance of JFrame
  - Extending JFrame and then creating an object of the subclass

```
JFrame f=new JFrame("Welcome");
```

```
class SampleGUI extends JFrame {
          . . .
          new SampleGUI ("Welcome");
          . . .
```

A window is created with a title bar having Java icon, max-min-close buttons and title Welcome

- A frame supports a number of methods; e.g. we can set the displaying position of the window using

```
f.setLocation(200, 300);
```

Top-Left corner is considered to be (0, 0)

Displayed at position x=200 and y=300

Adding components to Container

- ❑ We need to create a component and then we can add

```
JButton b = new JButton("click");

f.add(b);
```

Name of the button is set as click.

It is added to the frame f

- ❑ We may also set various parameters of a component

```
b.setBounds(30,10,100, 40);
```

Button is placed at position (30,10) within the container with width 100 and height 40

```
b.setEnabled(false);
```

This will disable the button

```
b.setBackground(Color.green);
```

Changes the color of the button

Code (SampleGUI.java, SampleGUIExtendJFrame.java): Screen share

Adding components to Container

- We may add other components similarly

```
JTextField tf = new JTextField();
tf.setBounds(130,170,200, 50);
f.add(tf);

JTextArea ta = new JTextArea();
ta.setBounds(30,270,200, 100);
f.add(ta);
```

Creates a text field
e.g. to enter name in a form

Creates a text area
User may write inside that

- If we need to display text on GUI, we may use label

```
JLabel l1 = new JLabel("First name");
```

Code (SampleGUI2.java): Screen share

Panel and Layout manager

- ❑ **Panel** is the simplest container without title bar
- ❑ We may add a group of components in a panel and position the panel on the window

```
JPanel p1 = new JPanel();
p1.setBounds(70, 200, 200, 50);
p1.add(b1); p1.add(b2);
f.add(p1);
```

Parameters can be set for the entire panel

- ❑ If we need to add multiple components in a container, we can place them nicely using **Layout manager**

```
JPanel p1 = new Jpanel ( new GridLayout(2,2) );
p1.setBounds(70, 200, 200, 50);
p1.add(b1); p1.add(b2); p1.add(b3); p1.add(b4);
f.add(p1);
```

GridLayout displays components in a two-dimensional grid

Other layouts are also available

Code (SampleGUI2.java): Screen share

Do we want our GUI to interact?

Yes, GUI should do some work

No, It is already done!

Events in GUI

- User needs to interact with GUI
- Events are generated when user interacts with GUI
- A source generates an event → It is sent to one or more listeners → listener processes the event
  (i.e. a listener waits until it receives an event)
- Activities that can generate events are pressing a button, entering a character via the keyboard, clicking the mouse, etc.
- A source must register listeners in order for the listeners to receive notifications

```
b.addActionListener(this);
```

Here, b is JButton object

Some action will be performed upon pressing the button

Code (IPFinder.java**): Screen share**

Handling GUI events

- ❑ Listeners are created by implementing one or more of the interfaces
- ❑ Each listener invokes some methods in response to events

```
class IPFinder extends JFrame implements ActionListener {
        . . .

        public void actionPerformed(ActionEvent e) {
                . . .
                }
        . . .
}
```

Code (IPFinder.java, GUIMenu.java): Screen share

Other commonly used events

Event class	Description
FocusEvent	When a component gains or loses keyboard focus.
ItemEvent	When a check box or list item is clicked; also when a choice selection is made or a checkable menu item is selected or deselected.
MouseEvent	When the mouse is dragged, moved, clicked, pressed, or released; also generated when the mouse enters or exits a component.
MouseWheelEvent	When the mouse wheel is moved.
TextEvent	Generated when the value of a text area or text field is changed.

Questions?