Lecture 14

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.

- 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.

AWT vs Swing

- AWT components are **platform-dependent**. Java swing components are **platform-independent**.

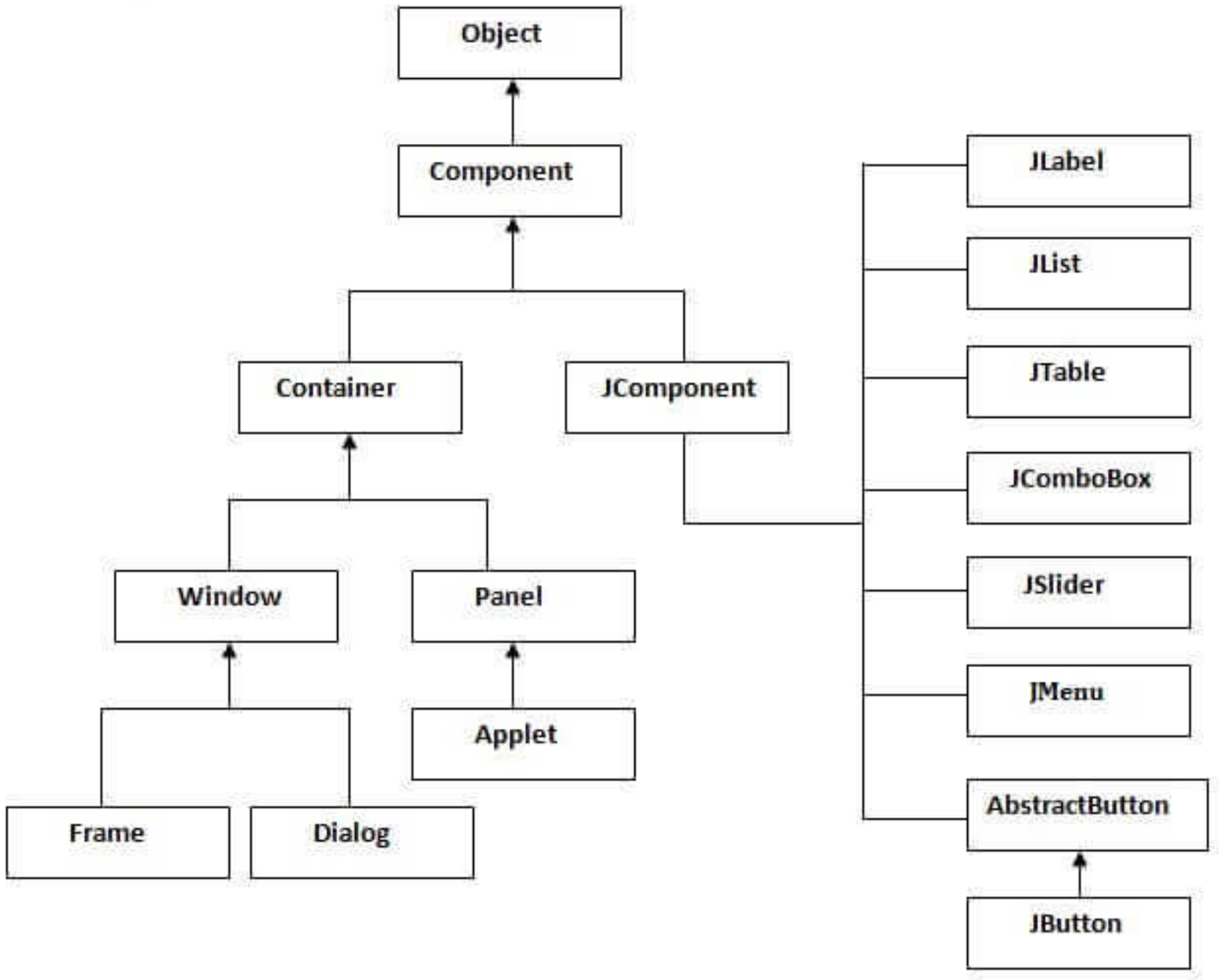
- AWT components are **heavyweight**. Swing components are **lightweight**.

- AWT doesn't support pluggable look and feel. Swing supports pluggable look and feel.

- 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.

Hierarchy of Swing classes



JButton, JLabel, JTextField

- The JButton class is used to create a labeled button that has platform independent implementation. The application result in some action when the button is pushed. It inherits AbstractButton class.

- The object of JLabel class is a component for placing text in a container. It is used to display a single line of read only text. The text can be changed by an application but a user cannot edit it directly. It inherits JComponent class.

- The object of a JTextField class is a text component that allows the editing of a single line text. It inherits JTextComponent class.

JTextArea, JPasswordField

- The object of a JTextArea class is a multi line region that displays text. It allows the editing of multiple line text. It inherits JTextComponent class.

- The object of a JPasswordField class is a text component specialized for password entry. It allows the editing of a single line of text. It inherits JTextField class.

JCheckBox, JRadioButton

- The **JCheckBox** class is used to create a checkbox. It is used to turn an option on (true) or off (false). Clicking on a CheckBox changes its state from "on" to "off" or from "off" to "on ".It inherits JToggleButton class.

- The **JRadioButton** class is used to create a radio button. It is used to choose one option from multiple options. It is widely used in exam systems or quiz. It should be added in ButtonGroup to select one radio button only.

JComboBox, JTable

- The object of Choice class is used to show popup menu of choices. Choice selected by user is shown on the top of a menu. It inherits JComponent class.

- The JTable class is used to display data in tabular form. It is composed of rows and columns.

JList, JOptionPane

- The object of JList class represents a list of text items. The list of text items can be set up so that the user can choose either one item or multiple items. It inherits JComponent class.

- The JOptionPane class is used to provide standard dialog boxes such as message dialog box, confirm dialog box and input dialog box. These dialog boxes are used to display information or get input from the user. The JOptionPane class inherits JComponent class.

JScrollBar, JMenuBar, JMenu, JMenuItem

- The object of JScrollbar class is used to add horizontal and vertical scrollbar. It is an implementation of a scrollbar. It inherits JComponent class.
- The JMenuBar class is used to display menubar on the window or frame. It may have several menus. The object of JMenu class is a pull down menu component which is displayed from the menu bar. It inherits the JMenuItem class. The object of JMenuItem class adds a simple labeled menu item. The items used in a menu must belong to the JMenuItem or any of its subclass.

JPopupMenu, JCheckBoxMenuItem

- PopupMenu can be dynamically popped up at specific position within a component. It inherits the JComponent class.
- JCheckBoxMenuItem class represents checkbox which can be included on a menu . A CheckBoxMenuItem can have text or a graphic icon or both, associated with it. MenuItem can be selected or deselected. MenuItems can be configured and controlled by actions.

JSeparator, JProgressBar

- The object of JSeparator class is used to provide a general purpose component for implementing divider lines. It is used to draw a line to separate widgets in a Layout. It inherits JComponent class.

- The JProgressBar class is used to display the progress of the task. It inherits JComponent class.

JTree, JColorChooser

- The JTree class is used to display the tree structured data or hierarchical data. JTree is a complex component. It has a 'root node' at the top most which is a parent for all nodes in the tree. It inherits JComponent class.

- The JColorChooser class is used to create a color chooser dialog box so that user can select any color. It inherits JComponent class.

JSlider, JTabbedPane

- The JTabbedPane class is used to switch between a group of components by clicking on a tab with a given title or icon. It inherits JComponent class.

- The Java JSlider class is used to create the slider. By using JSlider, a user can select a value from a specific range.

JSpinner, JDialog

- The object of JSpinner class is a single line input field that allows the user to select a number or an object value from an ordered sequence.

- The JDialog control represents a top level window with a border and a title used to take some form of input from the user. It inherits the Dialog class. Unlike JFrame, it doesn't have maximize and minimize buttons.

JPanel, JFileChooser

- The JPanel is a simplest container class. It provides space in which an application can attach any other component. It inherits the JComponents class. It doesn't have title bar.

- The object of JFileChooser class represents a dialog window from which the user can select file. It inherits JComponent class.

JToggleButton, JToolBar

- JToggleButton is used to create toggle button, it is two-states button to switch on or off.
- JToolBar container allows us to group other components, usually buttons with icons in a row or column. JToolBar provides a component which is useful for displaying commonly used actions or controls.

JViewport, JFrame

- The JViewport class is used to implement scrolling. JViewport is designed to support both logical scrolling and pixel-based scrolling. The viewport's child, called the view, is scrolled by calling the JViewport.setViewPosition() method.

- The javax.swing.JFrame class is a type of container which inherits the java.awt.Frame class. JFrame works like the main window where components like labels, buttons, textfields are added to create a GUI.

JComponent, JLayeredPane

- The JComponent class is the base class of all Swing components except top-level containers. Swing components whose names begin with "J" are descendants of the JComponent class. For example, JButton, JScrollPane, JPanel, JTable etc. But, JFrame and JDialog don't inherit JComponent class because they are the child of top-level containers. The JComponent class extends the Container class which itself extends Component. The Container class has support for adding components to the container.

- The JLayeredPane class is used to add depth to swing container. It is used to provide a third dimension for positioning component and divide the depth-range into several different layers.

JDesktopPane, JEditorPane

- The JDesktopPane class, can be used to create "multi-document" applications. A multi-document application can have many windows included in it. We do it by making the contentPane in the main window as an instance of the JDesktopPane class or a subclass. Internal windows add instances of JInternalFrame to the JdesktopPane instance. The internal windows are the instances of JInternalFrame or its subclasses.

- JEditorPane class is used to create a simple text editor window. This class has setContentType() and setText() methods.

JScrollPane, JSplitPane

- A JscrollPane is used to make scrollable view of a component. When screen size is limited, we use a scroll pane to display a large component or a component whose size can change dynamically.

- JSplitPane is used to divide two components. The two components are divided based on the look and feel implementation, and they can be resized by the user. If the minimum size of the two components is greater than the size of the split pane, the divider will not allow you to resize it. The two components in a split pane can be aligned left to right using JSplitPane.HORIZONTAL_SPLIT, or top to bottom using JSplitPane.VERTICAL_SPLIT. When the user is resizing the components the minimum size of the components is used to determine the maximum/minimum position the components can be set to.

JTextPane, JRootPane

- JTextPane is a subclass of JEditorPane class. JTextPane is used for styled document with embedded images and components. It is text component that can be marked up with attributes that are represented graphically. JTextPane uses a DefaultStyledDocument as its default model.

- JRootPane is a lightweight container used behind the scenes by JFrame, JDialog, JWindow, JApplet, and JInternalFrame.

Displaying graphics

1. **public abstract void drawString(String str, int x, int y):** is used to draw the specified string.
2. **public void drawRect(int x, int y, int width, int height):** draws a rectangle with the specified width and height.
3. **public abstract void fillRect(int x, int y, int width, int height):** is used to fill rectangle with the default color and specified width and height.
4. **public abstract void drawOval(int x, int y, int width, int height):** is used to draw oval with the specified width and height.
5. **public abstract void fillOval(int x, int y, int width, int height):** is used to fill oval with the default color and specified width and height.
6. **public abstract void drawLine(int x1, int y1, int x2, int y2):** is used to draw line between the points(x1, y1) and (x2, y2).
7. **public abstract boolean drawImage(Image img, int x, int y, ImageObserver observer):** is used draw the specified image.
8. **public abstract void drawArc(int x, int y, int width, int height, int startAngle, int arcAngle):** is used draw a circular or elliptical arc.
9. **public abstract void fillArc(int x, int y, int width, int height, int startAngle, int arcAngle):** is used to fill a circular or elliptical arc.
10. **public abstract void setColor(Color c):** is used to set the graphics current color to the specified color.
11. **public abstract void setFont(Font font):** is used to set the graphics current font to the specified font.

image in swing

- For displaying image, we can use the method drawImage() of Graphics class.

- public abstract boolean drawImage(Image img, int x, int y, ImageObserver observer): is used draw the specified image.