SOT
SCHOOL OF TECHNOLOGY

EVENT HANDLING AND GUI IN JAVA

Presented by:

Dr. Shivangi K. Surati

Assistant Professor,
Department of Computer Science and Engineering,
School of Technology,
Pandit Deendayal Energy University

AWT classes

- Abstract Window Toolkit (AWT)- defines a basic set of controls, windows and dialog boxes
- Limited graphical interface
- Uses platform-specific equivalents or peers for execution
- Look and feel of the components are defined by platform, not Java
- Heavyweight because they use native code resources
- No longer widely used

Problems in AWT

- Native peers-
  - Variations between OS, a component might look, or even act, differently on different platforms
  - May not support philosophy of Java- Write once, run anywhere
  - Look and feel of each component is fixed

Swing in Java

- part of Java Foundation Classes (JFC)- a set of GUI components for Desktop applications
- used to create window-based applications
- built on the top of AWT (Abstract Windowing Toolkit) API and entirely written in java
- provides platform-independent and lightweight components
- provides classes for java swing API such as JButton, JTextField, JTextArea, JRadioButton, JCheckbox, JMenu, JColorChooser etc.

Swing in Java…

- It does not replace AWT, only eliminates limitations
- Key features:
  - Swing components are lightweight
    - Written entirely in Java
    - No mapping with platform-specific peers
    - More efficient and flexible
  - Swing supports a Pluggable Look and Feel (PLAF)
    - Look and Feel components are under control of Swing
    - Possible to change the way that a component is rendered
    - “Plug in” a new component without any side effects

MVC connection

The components implicitly contains three parts:

- Model: The state information associated with the component- checkbox checked or unchecked
- View: How the component is displayed on the screen
- Controller: How the component reacts to the user- click the ckeckbox-change the model to reflect user's choice

- Swing uses a modified version of MVC- combines view and controller into a single logical entity – UI delegate
- Swings follows Model-Delegate architecture or Separable Model architecture

AWT vs 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**.
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)	Swing **follows MVC**.

Hierarchy of Java Swing classes

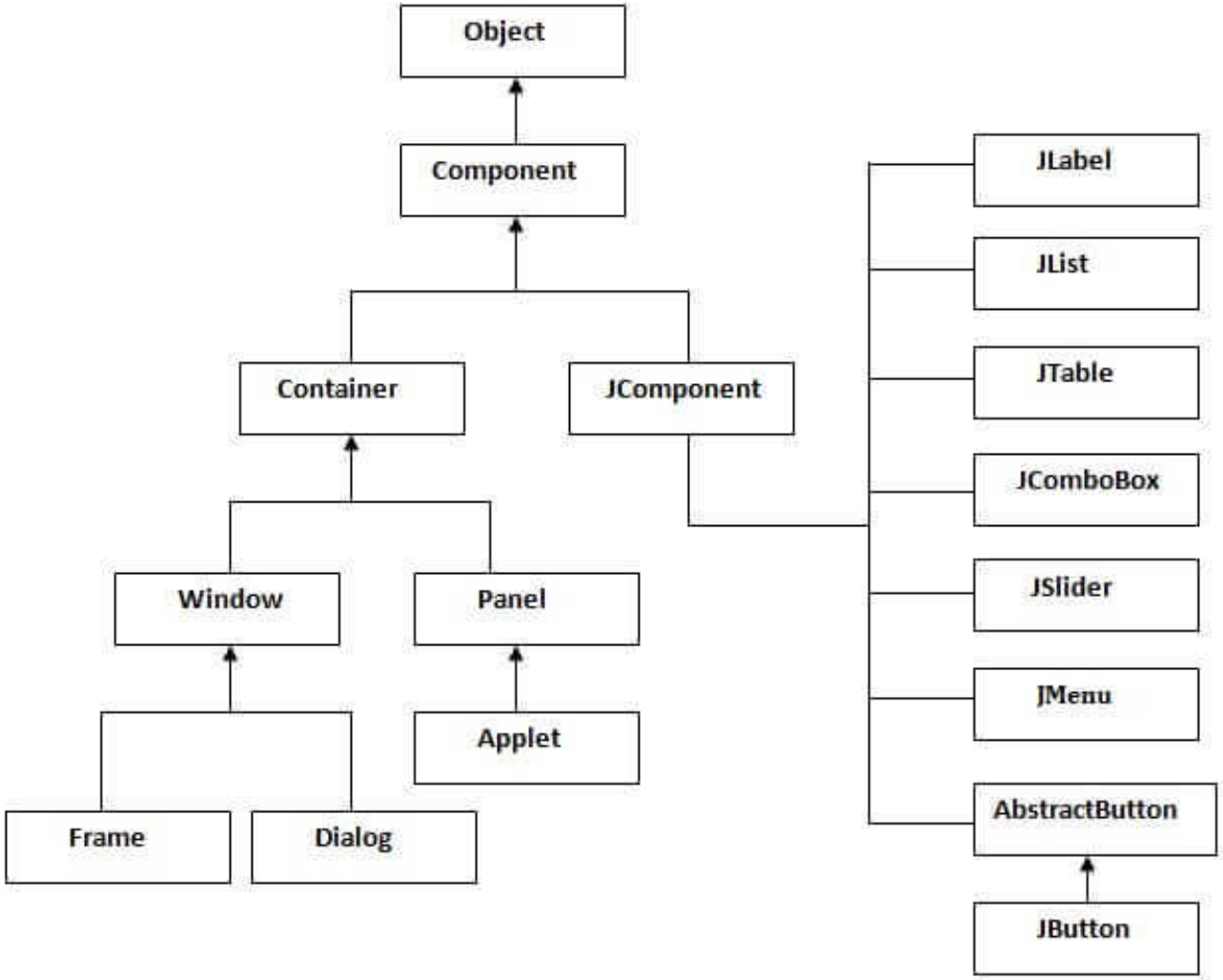

Image source: https://www.javatpoint.com/java-swing

Components and Containers

- Component- an independent visual control- button, slider
- Container- holds a group of components

- For component to be displayed, it must be within a container
- All Swing GUIs have at least one container
- Containers are components, they can also hold other containers
- Top-level containers: Jframe, Japplet, Jwindow, JDialog

Methods of Component class

Method	Description
public void add(Component c)	add a component on another component.
public void setSize(int width, int height)	sets size of the component.
public void setLayout(LayoutManager m)	sets the layout manager for the component.
public void setVisible(boolean b)	sets the visibility of the component. It is by default false.

Top-Level Container Panes

- Each top level container defines a set of panes
- At top level of hierarchy- JRrootPane- manages the other panes
- The other panes that comprise the root pane are-
  - Glass Pane: top-level panel, covers all other panes, transparent instance of JPanel
  - Content Pane: Mostly application interact with this pane, add visual components
  - Layered Pane: Instance of JLayeredPane, gives a depth value to components

Java Swing Examples

- There are two ways to create a frame:
  - By creating the object of Frame class (association)
  - By extending Frame class (inheritance)

- Different Layouts:
  - BorderLayout, CardLayout, FlowLayout, GridLayout, GridBagLayout

Event Handling

- Response to user input
- Handles the events generated by those interactions
- For example, click on button, dragging mouse
- Uses delegation event model approach
- Example programs
  - Register the component with the Listener -Many classes to register the component with the Listener
  - put the event handling code into one of the following places- within class, other class and anonymous class

Event classes and Listener interfaces

Event Classes	Listener Interfaces
ActionEvent	ActionListener
MouseEvent	MouseListener and MouseMotionListener
MouseWheelEvent	MouseWheelListener
KeyEvent	KeyListener
ItemEvent	ItemListener
TextEvent	TextListener
AdjustmentEvent	AdjustmentListener
WindowEvent	WindowListener
ComponentEvent	ComponentListener
ContainerEvent	ContainerListener
FocusEvent	FocusListener