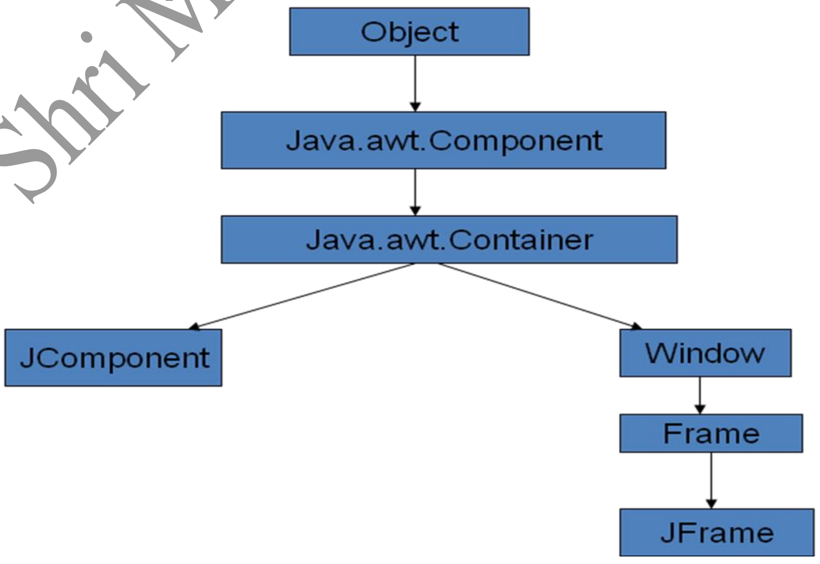
5 GUI with Swing & Event Handling

Swing Introduction

- Java introduced a package called abstract window toolkit and it contain large number of classes and interfaces that are supported.
- The methods of the swing class do nnot use the native methods.
- The methods in the swing class can be used to create screens with the same look and feel of the screens in different platforms.
- The Swing classes are subclass of java.awt.Component and java.awt.Container
- The name of the Swing class are start with letter J.
- The top-level class of Swing is JComponent.
- GUI components like button,label,checkbox text are handled in the JComponent class.



JFrameWindow

- It is a subclass of Frame class.
- When a JFrame is created its size is(0,0) and is invisible.
- A JFrameWindow is created by using the following constructor.
    - JFrame()
    - JFrame(String title)

JApplet

- It extends Applet class.
- JApplet has several functionalities which are not found in Applet class.

JPanel

- It can exactly use the same way as a Panel.
- We can place components to JPanel and then add the JPanel to some container.
- JPanel also act as a replacement for canvas.
- When using JPanel as a drawing area we should first, set the preffered size via setPrefferedSize.
- Secondly, we should use paintComponent for drawing.

JComponent

- It is a subclass of Container.
- This class contain a large number of subclasses which define component like JButton, JLabel, JComboBox etc.
- Which acts as user interface component.

JLabel

- Labels are instances of the JLabel class.
- JLabel is a subclass of JComponent. It can display text and/or icon.
- Some constructors are as follow:
    - JLabel(Icon i)
    - Label(String s)
    - JLabel(String s,Icon I,int align)
- Here, s and I are the string and icon used for the label.The align argument is either LEFT,RIGHT,CENTER.

JTextField

- The swing text field is encapsulated by the JText Component class which extends JComponent.
- JTextField allows you to edit or enter one line of text.
- Its constructors are as follow:
    - JTextField()
    - JTextField(int cols)

    - JTextField(String s, int cols)
    - JTextField(String s)
- Here s is the string to be presented, and cols is the number of columns in the text field.

JTextArea

- It is a subclass of JText Component.
- It displays multiple lines of text.
- There is no scroll to view the text if the text is large, then a JScrollPane has to be created using the text area Component.
    - JTextArea()
    - JTextArea(int row,int column)
    - JTextArea(String text,int row,int column)

JButton

- It provides the functionality of push button.
- It allows an icon, a string with the push button.
    - JButton(Icon i)
    - JButton(int row,int col)
    - JButton(String s,Icon i)

JPasswordField

- It creates the textfield same as the JTextField but the difference is when text is displayed the actual characters are replaced by * characters.
    - JPasswordField()
    - JPasswordField(int size)
    - JPasswordField(String str)
    - JPasswordField(String str,int size)

JCheckBox

- This class provides the functionality of checkbox.
    - JCheckBox(Icon i)
    - JCheckBox(Icon i,boolean state)
    - JCheckBox(string s)
    - JCheckBox(string s,boolean state)
    - JCheckBox(string s,Icon i)
    - JCheckBox(string s,Icon i,boolean state)

JRadioButton

- This class provides the functionality of a radio button, which is concrete implementation of AbstractButton.
    - JRadioButton(Icon i)
    - JRadioButton(Icon I,boolean state)
    - JRadioButton(String s)
    - JRadioButton(String s,boolean state)
    - JRadioButton(String s,Icon i)

- JRadioButton(String s,Icon I,boolean state)

JcomboBox

- It normally displays one entry.
- It can also display a drop down list that allows a user to select a different entry.
- You can also type your selection into the text field.
    - JComboBox()
    - JComboBox(Vector v)

JList

- It create a list of items and allow the user to select one or more items from the list.
- A list is used when the number of items for selection is large.
    - JList()
    - JList(Vector v)

JScrollBar

- It is a visual component that can be used ti bring a section of large area of text or image to view by scrolling.
- The scrollbar are vertical or horizontal.
    - JScrollBar()
    - JScrollBar(int orientation,int scrollpos,int vosible,int min,int max)

JMenuBar

- It holds many menus in its bar.
- A menubar has to be attached to a JFrame window or to a JApplet window.
    - JMenuBar()

JMenu

- It is a container for JMenuItem.
- A menu can be attached with several menuitems.
    - JMenu()
    - JMenu(String str)

JMenuItem

- It is a part of JMenu and displayed in a window.
    - JMenuItem()
    - JMenuITem(Icon i)
    - JMenuItem(String str)

DialogBOx

- They are used to obtain user input.
- They don’t have a menubar.
- When a modal dialog box is active, all input is directed to it until it is closed.

- This means that you can not access other parts of your program until you have closed the dialog box.
    - Dialog(Frame parentWindow,boolean mode)
    - Dialog(Frame parentWindow,string title,boolean mode)
- Here parentwindow is the owner of the dialog box.

FileDialog

- This class extends Dialog and provides a dialog that allows a user to select a file for reading or writing.
- This is very useful for applications that need to save data to a file and later retrieve that data.
- This class defines two consatans FileDialog.LOAD and FileDialog.SAVE
- These determine if the file dialog is being used to select file for reading and writing.
    - FileDialog(Frame Parent)
    - FileDIalog(Frame Parent,String str)
    - FileDialog(Frame Parent,String str,int rw)

Event delegation model

Event

- An event is an object that describes a state change in a source.
- It can be generated as consequence of a person interacting with the elements.
- A person interacting with the elements in a graphical user interface.
- Some of activities that cause events to be generated are pressing a button, entering a character via the keyboard, selecting an item from a list and clicking a mouse.
- Event may also occur that are not directly caused by interaction with user interface.
- For example an event may be generated when a timer expires, a counter exceeds a value, software or hardware failure occurs or an operation is completed.
- A source is an object that generates an event.
- This occurs when the internal state of that object changes in someway.
- Source may generate more than one type of event.
- A source must register listeners in order for the listeners to receive notifications about a specific type event.
- Each type of event has its own registration method.
  - **Syntax**

```
public void addTypeListener(TypeListener l)
```

- Here, type is the name of the event, and l is a reference to the event listener.
- for example, the method that registeres keyboard event listener is called addKeyListener().
- The method that registers a mouse motion listeners is called addMouseMotionListener()
- When an event occurs, all registered listeners are notified and receive a copy of event object.
- This is known as multicasting event.
- A source must also provide a method that allows a listener to unregister an interest in a specific type of event.

  Syntax

```
Public void removeTypeListener(TypeListener l)
```

- Here, type is an object that notified when an event listener.
- For example, to remove a keyboard listener, you should call removeKeyListener().

Source generating events

Button	generates action events when the button is pressed
Checkbox	generatres item events when the check box is selected or deselected.
Choice	generates item events when the choice is changed.
List	generates action events when an item is double clicked or when an item is selected or deselected
MenuItem	generates action events when a menu item is selected or deselected
Scrollbar	generates adjustment events when the scroll

	bar is manipulated.
TextComponent	generate text events when the user enters a character.
Window	generates window events when a window is activatedeactivated,opened or quit.

AWT Event Packages

- The AWTEvent class is a subclass of EventObject and is part of the java.awt package.
- AWTEvent is an abstract class.
- Following are the methods of event class.

1)ActionEvent

- it is generated when a button is pressed, a list item double-clicked or a menu item is selected.
- it has the three constructor which are as follow:
  ActionEvent(Object src,int type,String cmd)
  ActionEvent(Object src, int type, String cmd, int modifier)
  ActionEvent(Object src, int type, String cmd, long when, int modifiers)
- Here src is referenced to the object that generated this event.the type of the event is specified by type and its command string cmd.
- The modifiers indicate which modifier key is pressed when an event is generated.

2)AdjustmentEvent

- it is generated by a scrollbar.
- There are five types of adjustment events.
- The AdjustmentEvent class defines integer constant that can be used to identify them.
- the constructor is as follow:
  AdjustmentEvent(Adjustable src, int id, int type, int data)
- the constant are as follow:

BLOCK_DECREMENT	The user clicked inside the scrollbar to decrease its value.
BLOCK_INCREMENT	The user clicked inside the scrollbar to increase to value.
TRACK	The slider was dragged
UNIT_DECREMENT	The button at the end of the scrollbar was clicked to decrease its value.
UNIT_INCREMENT	The button at the end of scrollbar was clicked to increase its value.

3)ComponentEvent

- it is generated when the size, position or visibility of a component is changed.
- It defines the integer constant which are as follow:

COMPONENT_HIDDEN	The component was hidden
COMPONENT_MOVED	The component was moved
COMPONENT_RESIZED	The component was resized
COMPONENT_SHOWN	The component became visible.

ContainerEvent

- It is a subclass of ComponentEvent.
- It is generated when a component is added or removed from a container.

4)FocusEvent

- it is generated when a component gains or loses input focus.
- it is identified by the integer constants FOCUS_GAINED and FOCUS_LOST.

5)InputEvent

- it is the subclass of a component event and is the super class for KeyEvent and MouseEvent.

6)ItemEvent

- it is generate when a checkbox or list item is clicked or when a menu item is selected or deselected.

7)KeyEvent

- It is generated when keyboard input occurs.
- There are 3 types of key events.

  1)KEY_PRESSED
  2)KEY_RELEASED
  3)KEY_TYPED

8)MouseEvent

- It is subclass of InputEvent.
- There are 8 types of MouseEvents.

  MOUSE_CLICKED
  MOUSE_MOVED
  MOUSE_DRAGGED
  MOUSE_ENTERED
  MOUSE_PRESSED
  MOUSE_RELEASED
  MOUSE_EXITED
  MOUSE_WHEEL

9)WindowEvent

- It is a subclass of ComponentEvent.
- There are 10 types of window events.

  WINDOW_ACTIVATED
  WINDOW_CLOSED
  WINDOW_CLOSING
  WINDOW_DEACTIVATED
  WINDOW_DEICONIFIED
  WINDOW_GAINED_FOCUS

WINDOW_ICONIFIED
WINDOW_LOST_FOCUS
WINDOW_OPENED
WINDOW_STATE_CHANGED

Swing Event Package

- it contains the following classes.

Class	Description
AbstractButton	Abstract super class for swing buttons.
ButtonGroup	Encapsulates a mutually exclusive set of buttons.
ImageIcon	Encapsulates an icon
JApplet	The swing version of Applet
JButton	The swing push button class.
JCheckBox	The Swing check box class.
JComboBox	The Swing combo box class.
JLabel	The swing Label class
JRadioButton	The Swing radio button class.
JScrollPane	The Swing scroll pane class.
JTable	The Swing table class.
JTextField	The Swing Text Field class.

Listener Interfaces

- This interface defined by the java.awt.event package.
- The interface and its description is as follow:

Interface	Description
ActionListener	Defines method to receive action events.
AdjustmentListener	Defines method to receive adjustment event.
ComponentListener	Defines four method to recognize when a component is hidden,moved,resized or shown.
FocusListener	Defines two methods to recognize when a component gains or loses its focus.
ItemListener	Defines one method to recognize when the state of an item change.
KeyListener	Defines three method to recognize when a key is pressed,released or typed
MouseListener	Defines five methods when the mouse is clicked, ennteres a component,is pressed or is released,exit.

MouseMotionListener	It defines two methods to recognize when the mouse is dragged or moved.
MouseWheelListener	It defines one method to recognize when the mouse wheel is moved.
TextListener	Defines one method to recognize when a text value changes.
WindowFocusListener	Defines two methods to recognize when a window gains or loses input focus.
WindowListener	Defines seven methods to recognize when a window is activated ,closed, deactivated,deiconified,iconified,opened or quit.

AWT controls

- The java GUI classes are contained in the java.awt package.
- AWT package provides full support for user interface, components like labels, buttons, checkboxes, textfields, scrollbars, menus etc.

1)Labels

- A Label is a string that appears on a graphical user interface.
- these class defines three constructors.

  1)Label()
  2)Label(String str)
  3)Label(String str, int align)
- Here, str is the text for the label.

2)Buttons

- A Button is a component that appears as a push button.
- A button is a component that contain a label and that generates an event when it is pressed.
- this class defines two constructor.

  1)Button()
  2)Button(String str)
- Here str is the text for the button.

3)Checkbox

- A Checkbox is a component that combines a label and a small box.
- it is a control that is used to turn an option on or off.
- the constructor are as follow:

  1)Checkbox()
  2)Checkbox(String str)
  3)Checkbox(String str,boolean state)
  4)Checkbox(String str,boolean state,CheckboxGroup grp)
- The first form creates a checkbox whose label is blank.
- In the second form str is the text for the check box.
- In the third form if state is true, a check mark appears in the box.
- The fourth form used to group several check boxes together.

- The parameter grp is referenced to the check box group.

4)CheckboxGroup

- A CheckboxGroup is a set of check boxes.
- in this group only one check box in the group can be checked at one time.
- These check box group are often called radio buttons.
- For, this type of Checkbox you must first define the group to which they will belong and then specify that group whenyou construct the checkbox.

```
CheckboxGroup()
```

5)Choice

- A choice is a component that provides a list of menu. When user clicks on a choice, it displays whole list of choice and then a selection can be made.
- it can have only one item selected at a time. Choice shows the currently selected item.
- Choice define the only one constructor:

```
Choice()
```

6)List

- A List is a component that allows a user to select one or more items. If the list is large then it will automatically provide scroll bars so that user may see the entire list.
- An item event is generated when a user selects or deselects one of the item.
- It defines the following constructor:

```
1)List()
2)List(int rows)
3)List(int rows,boolean multiple)
```

- Here, in the second form rows is the number of items that are visible to user.
- In the third form if multiple is true , a user may select multiple entries in a list.
- Otherwise only one item may be selected.

7)Scrollbar

- Scrollbars are used to select any int values between a specified minimum and maximum.
- A scroll bar contains a slider that may be dragged to continuously vary its value.
- it defines the following constructor:

```
Scrollbar()
Scrollbar(int orientation)
Scrollbar(int orientation,int value,int width,int min,int max)
```

8)TextField

- A textField allows a user to enter one line of text.
- TextField allow the user to enter strings and to edit the text using the arrow keys, cut and paste keys, and mouse selection.
- It defines the following constructor:

```
TextField()
TextField(String str)
TextField(int cols)
TextField(String str, int cols)
```

- Here, str is the text for the field. The argument cols indicate the width of a field in character.

9)TextArea

- A textArea allows a user to enter multiple lines of text. Sometimes a single line of text input is not enough for a given task.
- To handle these situation, the AWT includes a simple multiline editor called TextArea.
- It has the following constructor:

```
TextArea()
TextArea(String str)
TextArea(int rows,int cols)
TextArea(String str,int rows,int cols)
TextArea(String str,int rows,int cols,int scrollbars)
```

- Here, str is the text for the area. The rows and cols arguments indicate the number of rows and columns in the component.

10)Canvas

- A canvas provides a rectangular area on which we can draw. This is valuable because we can create graphs and other diagrams on the canvas by using methods of the Graphics class.
- The canvas class extends component class.
- it defines the following constructor.

```
Canvas()
```

11)Menubars and Menus

- A window can have a menubar associated with it.A menu bar displays a list of top level menu choice.
- Each choice is associate wiuth a drop down menu. This concept is implemented by the following classes:

  Menubar, Menu and MenuItem
- In short, a menu object contains a list of MenuItem objects.
- Each menu object contains a list of MenuItem objects. Each MenuItem object represents something that can be selected by the user.
- Menu is a superclass of MenuItem. We can also create a hierarchy of nested submenus.
- It is also possible to include checkable MenuItem.

Adaptor class

- The adaptor classes are very special classes that are used to make event handling.
- There are listener interfaces that have many methods for event handling.
- By implementing an interface we have to implement all the methods of that interface.
- By sometimes we need only one or some methods of the interface.
- In that case we use adaptor class.

- For example, the MouseListener interface has five methods mouseClicked(), mouseEntered(), mouseExited(), mousePressed() and mouseReleased().
- If in your program you just need two events but you implement five methods of interface.
- So you can use the adaptor class.
- The adaptor class contain an empty implementation for each method of the event listener interface.

Listener interface	**Adaptor classes**
ComponentListener	ComponentAdapter
ContainerListener	ContainerAdapter
FocusListener	FocusAdapter
KeyListener	KeyAdapter
MouseListener	MouseAdapter
MouseMotionListener	MouseMotionAdapter
WindowListener	WindowAdapter