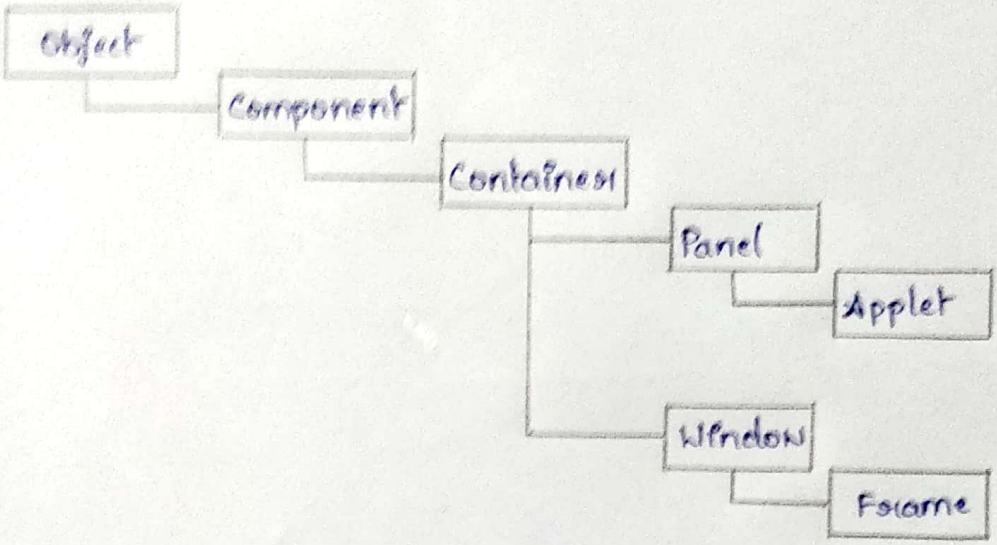
OODP ASSIGNMENT

Done By,
Harikrishnan.V
CS 4A
Roll No: 27

Abstract Window Toolkit - AWT

- The AWT contains numerous classes & methods that allow us to create & manage windows.
- The AWT classes are contained in the java.awt package.

Window Fundamentals



The 2 most common windows are:

- Those derived from Panel, which is used by applets.
- Those derived from Window, which is used by Frame to create a window.

Working with Frames

- Frame() → Creates a standard window with title & borders.
- WindowFrame (String title) → Creates a window with title.
- void setSize (int newWidth, int newHeight) → To set the dimensions of the window.
- void setSize(Dimension newSize) → Dimension object is passed. It contains width & height fields.

- Dimension getSize() → To obtain the current size of a window.
- void setVisible (boolean visibleFlag) → To make a created window visible.
- void setTitle (String newTitle) → To change window title.
- Closing a Frame Window → call setVisible (false) followed by windowClosing().

- Example: To Create a Frame

```
import java.awt.*;
public class Main {
    public static void main (String[] args) {
        Frame f = new Frame ("Tutorialspoint");
        int width = 300;
        int height = 300;
        f.setTitle("FRAME EXAMPLE");
        f.setSize( width, height);
        f.setVisible (true);
    }
}
```

Graphics Class

- Java's Graphics class include methods for drawing many different types of shapes, from simple lines to polygons to text in variety of fonts & colors.

- Methods of Graphics Class

→ drawString(): Displays a text string

→ drawLine(): Draws a straight line.
→ drawArc(): Draws a hollow arc.
→ drawRect(): Draws a hollow rectangle.
→ drawOval(): Draws a hollow Oval.
→ drawRoundRect(): Draws a hollow rectangle with rounded corners.
→ drawPolygon(): Draws a hollow polygon.
→ fillArc(): Draws a filled arc.
→ fillPolygon(): Draws a filled polygon.
→ fillRect(): Draws a filled rectangle.
→ fillRoundRect(): Draws a filled rectangle with rounded corners.
→ getColor(): Retrieves the current drawing color.
→ getFont(): Retrieves the currently used font.
→ setColor(): Sets the drawing color.
→ setFont(): Sets the font.

Color and Font

- **Color Class**

The following are three constructors for Color class:

(i) Color(int red, int green, int blue)
(ii) Color(int rgbvalue)
(iii) Color(float red, float green, float blue)

- **Color Methods**

Color getColor() returns the current color.
setColor(Color newcolor) changes the foreground color.

- set the background color & foreground color using the following methods:

  void setBackground(Color newColor)
  void setForeground(Color newColor)

- We can obtain the current settings for the background & foreground colors by calling getBackground() & getForeground().

  Color.getBackground()
  Color.getForeground()

- Font Class

  Font(String fontName, int fontStyle, int pointStyle)
  Where fontName specifies the name of the desired font.

  The style of the font is given by fontStyle. fontStyle consists of one or more of these three constants: Font.PLAIN, Font.BOLD, Font.ITALIC.

  The size of the font is specified by pointSize.

- Font Methods

  Font getFont() → Obtain information of currently selected font.
  String getName() → Get the name of current font.
  int getSize() → Get the size of current font.
  int getStyle() → Get the style of current font.
  setFont(Font f) → Set the new font.

AWT Controls

- Controls are components that allow a user to interact with the application in various ways.

- The AWT supports the following types of controls:

→ Labels
→ Push buttons
→ Check boxes
→ Check box groups (Radio buttons)
→ Choice lists
→ Lists
→ Scroll bars
→ Text field
→ Text area

- **Adding Controls**

→ Create an instance of the desired control.
→ Add it to a window by calling add()

Component add(Component compObj)

compObj is an instance of the control that we want to add.

- **Removing Controls**

→ Call remove()

void remove(Component obj)

obj is a reference to the control that we want to remove.

→ Remove all controls by calling removeAll().

• Responding to Controls

→ Labels are passive controls. Except for labels, all controls generate events when they are accessed by the user.

→ The program simply implements the appropriate interface and then registers an event listener for each control that we need to monitor.

Labels

- Labels are passive controls that do not support any interaction with the user.

Push Buttons

- A push button contains a label that generates an event when it is pressed.

Check Boxes

- It consists of a small box that can either contain a check mark or not.

Check Box Group

- Used to create a set of mutually exclusive check boxes in which one & only one check box in the group can be checked at any one time.

Choice Lists

- The Choice class is used to create a pop-up list of items from which the user may choose.

Lists

- It provides a compact, multiple-choice, scrolling selection list.

Scroll Bars

- Scroll bars are used to select continuous values between a specified minimum & maximum.

Text Field

- Text fields allow the user to enter strings & to edit the text using the arrow keys, cut & paste keys & mouse selections.

Text Area

- Text Area is a multiline editor.