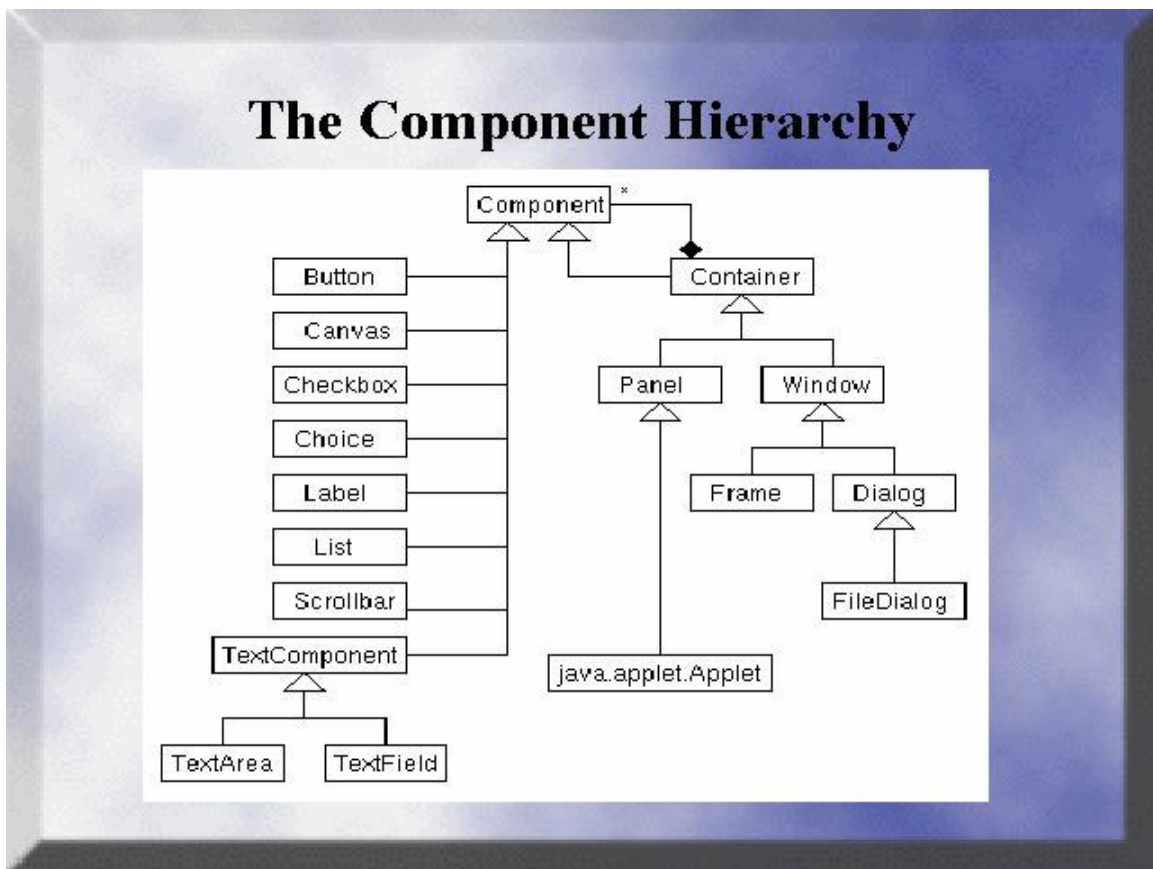


Introduction to AWT programming

Definition

A abbreviated term for the concept of “Abstract Window Toolkit”.
The class library that provides the standard API for building GUIs for Java programs. The AWT includes imaging tools, data transfer classes, GUI components, containers for GUI components, an event system for handling user and system events among parts of the AWT, and layout managers for managing the size and position of GUI components in platform-independent designs.

Class Hierarchy



Understanding the Frame interface

```
import java.awt.*;  
  
public class FirstFrame extends Frame  
{  
    public static void main(String[] args)  
    {  
        FirstFrame f = new FirstFrame();  
        //needs a variable of type class  
        //to perform interface functions  
  
        f.setSize(500,500);  
        //sets the size of the frame to width,height  
  
        f.setLocation(256,192);  
        //sets the location of the frame from X,Y  
  
        f.show();  
        //needs a specific show function to display frame  
    }  
}
```

Understanding the paint() method

One of the important AWT methods is `paint()`. The job of this method is to draw the graphical elements displayed in the application -- which is just a rectangular area in a window. The `paint()` method in the `Application` class doesn't draw anything at all, so `paint()` is one of those methods that the programmer can override in a subclass. The definition of this method must have the form:

```
public void paint(Graphics g) {
    // draw everything here..
}
```

A complete example

```
import java.awt.*;

public class FirstFrame extends Frame
{
    public static void main(String[] args)
    {
        FirstFrame f = new FirstFrame();
        f.setSize(500,500);

        f.setBackground(new Color(0,123,123));
        f.setForeground(new Color(0,0,0));

        f.setLocation(256,192);
        f.show();
    }

    //Built-In class for writing custom messages on the frame
    public void paint(Graphics g)
    {
        Font f = new Font("Times New Roman",Font.BOLD,16);
        g.setFont(f);
        g.setColor(Color.green);
        g.drawString("Not a hello world program",75,100);
        g.setColor(Color.red);
        g.drawString("Not a hello world program",75,100);
    }
}
```

Built in routines to notice

drawString(string,xPos,yPos)

setFont(fontName,fontAttrib,fontSize)

setSize(width,height)

Class Activity 1

Write a Java program that displays a screen like the one shown below.

NOTE: Notice that the output has 2 different kinds of font types.

