Lập trình hướng đối tượng

Biên soạn: Nguyễn Thị Tuyết Hải

Email: tuyethai@ptithcm.edu.vn

Chapter 10: Graphics Programming

Introducing Swing

Creating a Frame

Positioning a Frame

Displaying Information in a Component

Working with 2D Shapes

10.1 Introducing Swing

Abstract Window Toolkit (AWT) -> Swing

The resulting program could run on any of these platforms, with the “look-and-feel” of the target platform.

- Swing has a rich and convenient set of user interface elements.
- Swing has few dependencies on the underlying platform; it is therefore less prone to platform-specific bugs.
- Swing gives a consistent user experience across platforms.

10.2 Creating a Frame

- A top-level window is called a frame in Java: JFrame
- A window that is not contained inside another window

Event dispatch thread

```
EventQueue.invokeLater(() -> {
        statements
});
```

A frame is hidden when the user closes it, but the program does not terminate.

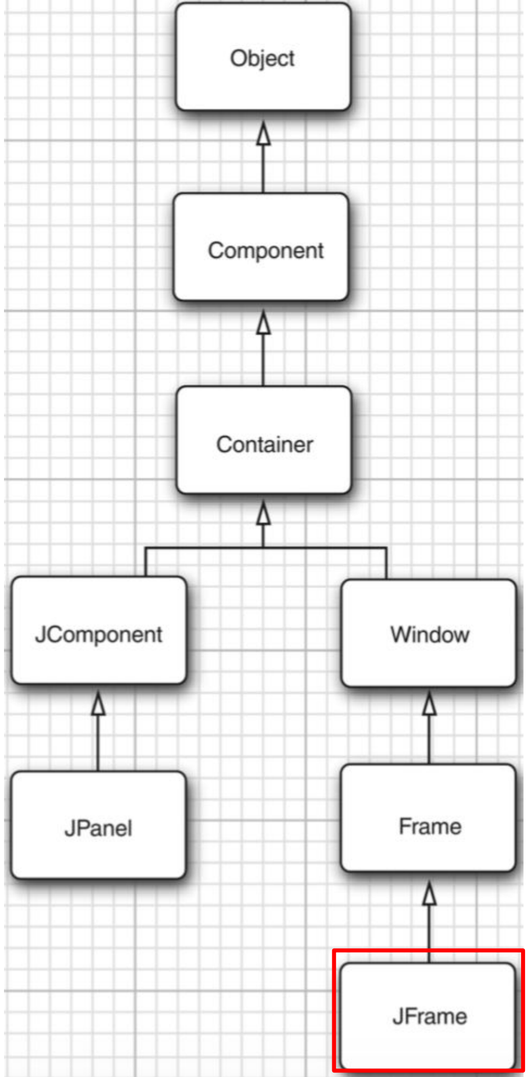
Frames start their life invisible.

```
import java.awt.*;
import javax.swing.*;

class SimpleFrame extends JFrame {
  private static final int DEFAULT_WIDTH = 300;
  private static final int DEFAULT_HEIGHT = 200;

  public SimpleFrame(){
    setSize(DEFAULT_WIDTH, DEFAULT_HEIGHT);
  }
}

public class SimpleFrameTest{
  public static void main(String[] args){
    EventQueue.invokeLater(() ->
      {
        SimpleFrame frame = new SimpleFrame();
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE
        frame.setVisible(true);
      });
  }
}
```

10.3 Positioning a Frame

Most of the methods for working with the size and position of a frame come from the various superclasses of JFrame;

- setLocation, setBounds
- setIconImage
- setTitle
- setResizable



10.3 Positioning a Frame

10.3.1 Frame Properties

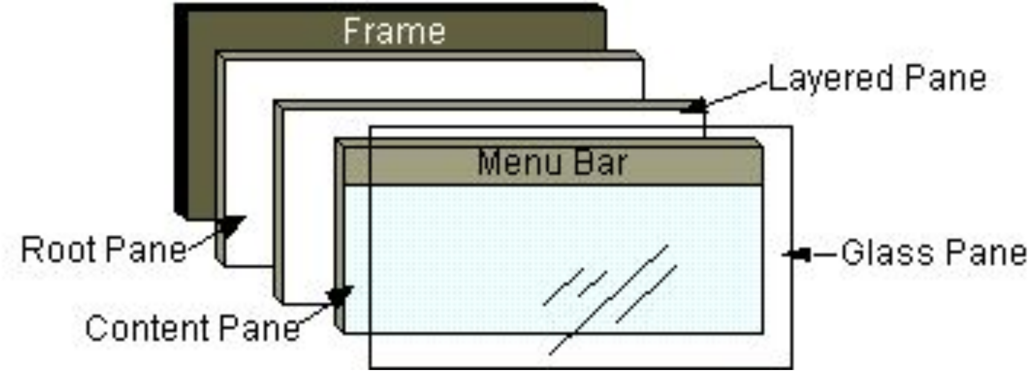
- get the resolution of the user’s screen
- write code that resizes the frames accordingly

```
class SizedFrame extends JFrame
{
  public SizedFrame()
  {
    // get screen dimensions

    Toolkit kit = Toolkit.getDefaultToolkit();
    Dimension screenSize = kit.getScreenSize();
    int screenHeight = screenSize.height;
    int screenWidth = screenSize.width;

    // set frame width, height and let platform pick screen location

    setSize(screenWidth / 2, screenHeight / 2);
    setLocationByPlatform(true);
  }
}
```

10.4 Displaying Information in a Component

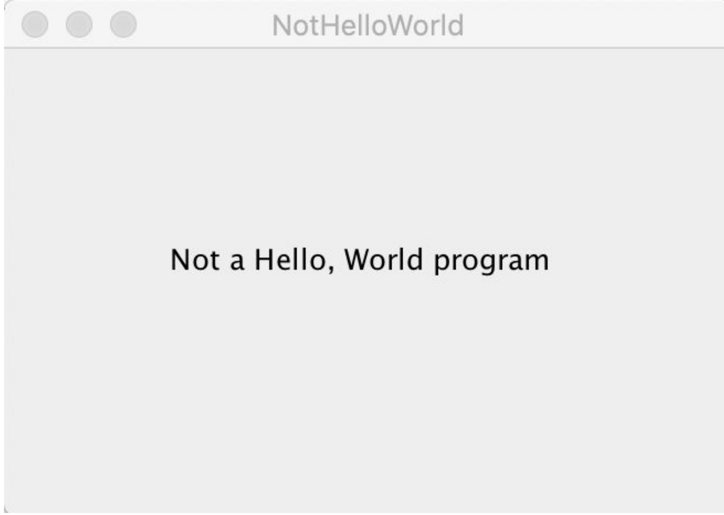
- frames are really designed to be containers for components
- four panes are layered in a JFrame.
  - the root pane: manages other pane
  - layered pane,
  - glass pane
  - content pane
- designing a frame, you add components into the content pane

```
Container contentPane = frame.getContentPane();
Component c = . . .;
contentPane.add(c);
```



10.4 Displaying Information in a Component

```
public class NotHelloWorld {
          public static void main(String[] args) {
                    EventQueue.invokeLater(() -> {
                              JFrame frame = new NotHelloWorldFrame();
                              frame.setTitle("NotHelloWorld");
                                        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
                              frame.setVisible(true);
                    });
          }
}

class NotHelloWorldFrame extends JFrame {
          public NotHelloWorldFrame() {
                    add(new NotHelloWorldComponent());
                    pack();
          }
}

class NotHelloWorldComponent extends JComponent {
          public static final int MESSAGE_X = 75; public static final int MESSAGE_Y = 100;
          private static final int DEFAULT_WIDTH = 300; private static final int DEFAULT_HEIGHT = 200;
          public void paintComponent(Graphics g) {
                    g.drawString("Not a Hello, World program", MESSAGE_X, MESSAGE_Y);
          }
          public Dimension getPreferredSize() {
                    return new Dimension(DEFAULT_WIDTH, DEFAULT_HEIGHT);
          }
}
```



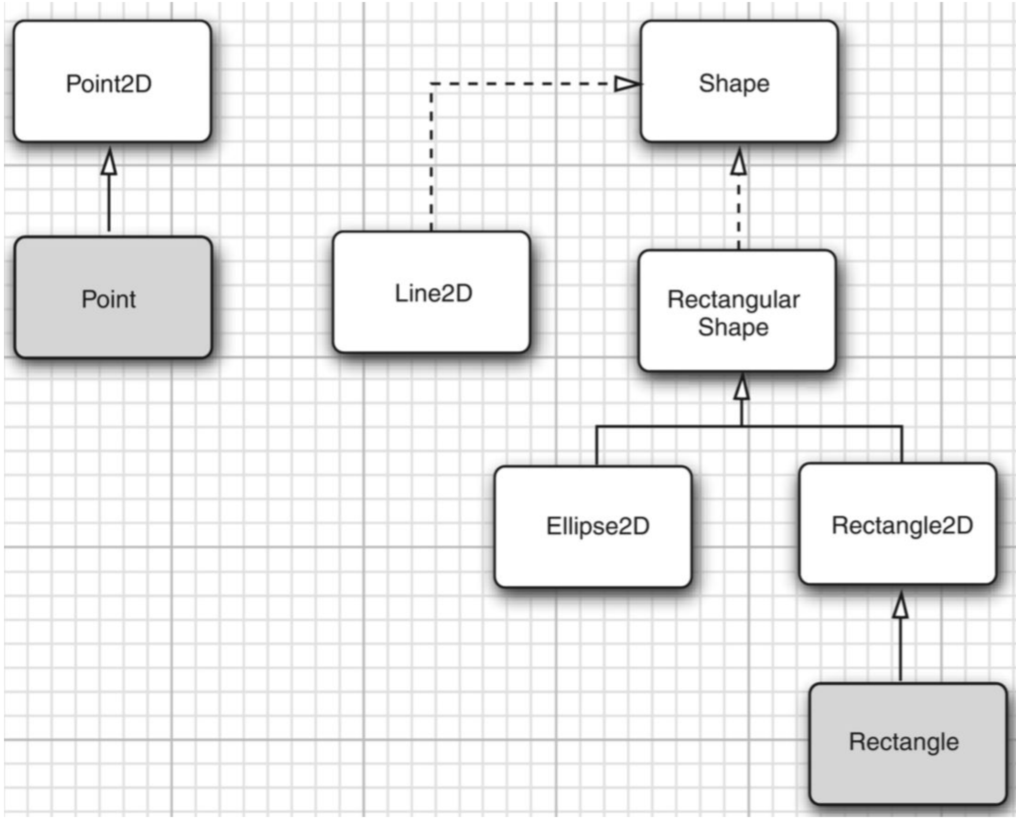
Not a Hello, World program

10.5 Working with 2D Shapes

Classes represent lines, rectangles, and ellipses

- Line2D
- Rectangle2D
- Ellipse2D

These classes implement the Shape interface.


Point2D
Shape
Point
Line2D
Rectangular Shape
Ellipse2D
Rectangle2D
Rectangle