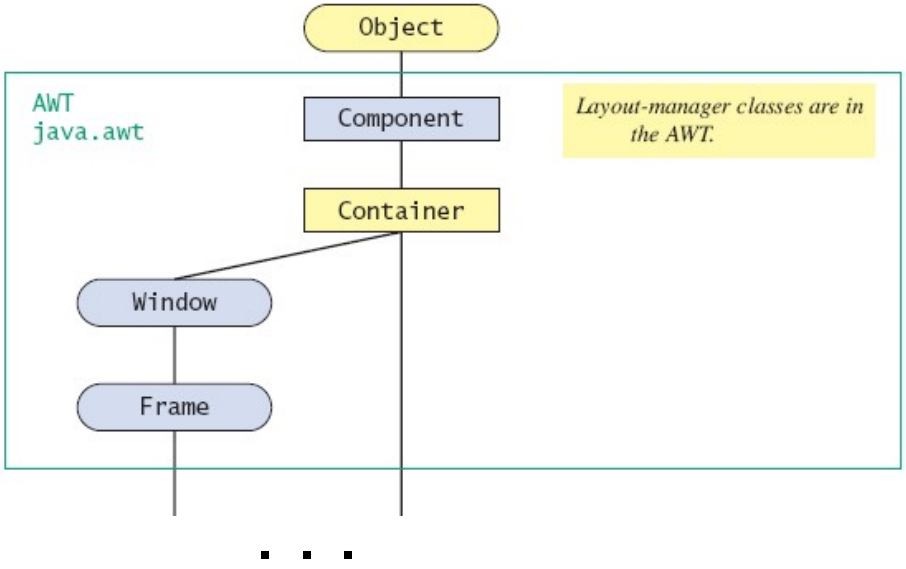
13.4 Container Classes

- A container class
  - can have components added to it.
  - Every Swing container class has an `add` method.
- Some commonly used container classes
  - `JPanel`
  - `Container`
  - Content pane of a `JFrame`

Class `Container`

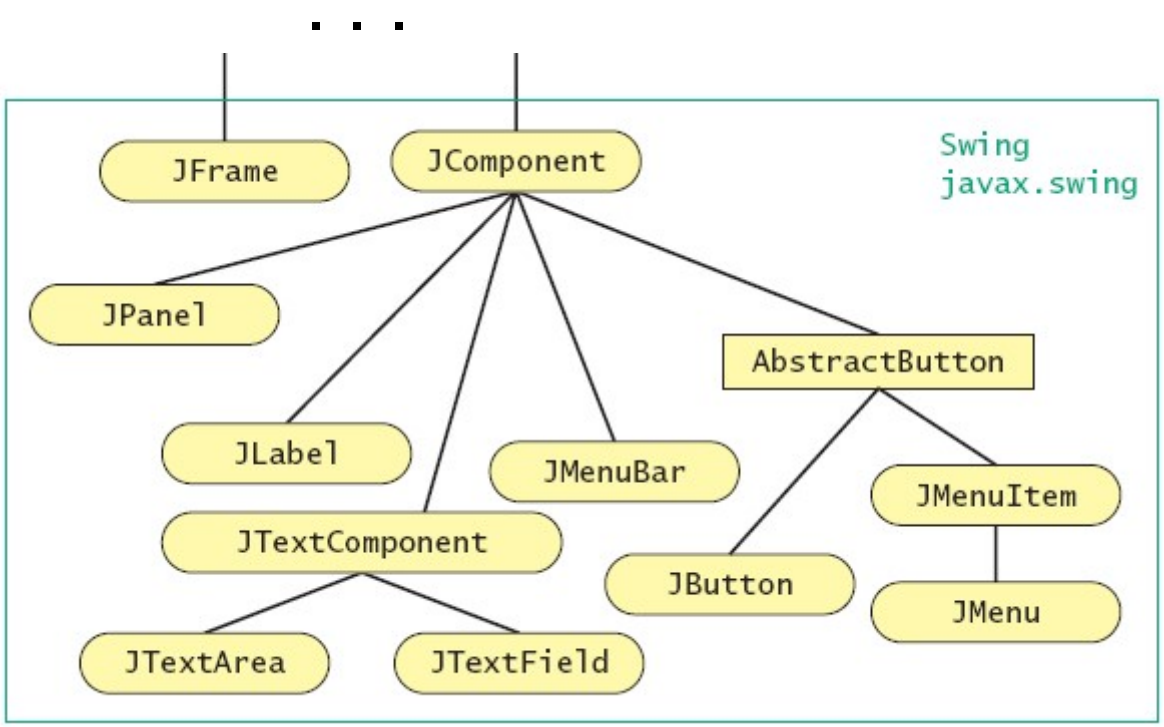
- A container class is a descendant of the class `Container`
- A component class is a descendant of the class `JComponent`

Figure 13.10 hierarchy of swing classes



Class `Container`

- Figure 13.10 hierarchy of swing classes, ctd.



Listing 13.9 Putting the Buttons in a Panel - PanelDemo.java

```
// Listing 13.9 Putting the Buttons in a Panel
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;

/**
 Simple demonstration of putting buttons in a panel.
*/
public class PanelDemo extends JFrame implements ActionListener
{
   public static final int WIDTH = 300;
   public static final int HEIGHT = 200;

   public static void main(String[] args)
   {
      PanelDemo guiWithPanel = new PanelDemo( );
      guiWithPanel.setVisible(true);
   }
```

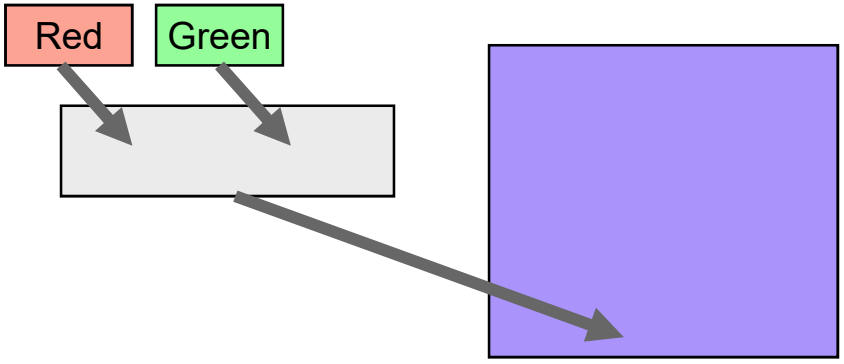
```
public PanelDemo( )
 {
    setSize(WIDTH, HEIGHT);
    addWindowListener(new WindowDestroyer( ));
    setTitle("Panel Demonstration");
    Container contentPane = getContentPane( );
    contentPane.setBackground(Color.BLUE);
    contentPane.setLayout(new BorderLayout( ));

    JPanel buttonPanel = new JPanel( );
    buttonPanel.setBackground(Color.WHITE);

    buttonPanel.setLayout(new FlowLayout( ));

    JButton stopButton = new JButton("Red");
    stopButton.setBackground(Color.RED);
    stopButton.addActionListener(this);
    buttonPanel.add(stopButton);

    JButton goButton = new JButton("Green");
    goButton.setBackground(Color.GREEN);
    goButton.addActionListener(this);
    buttonPanel.add(goButton);

    contentPane.add(buttonPanel, BorderLayout.SOUTH);
 }
```

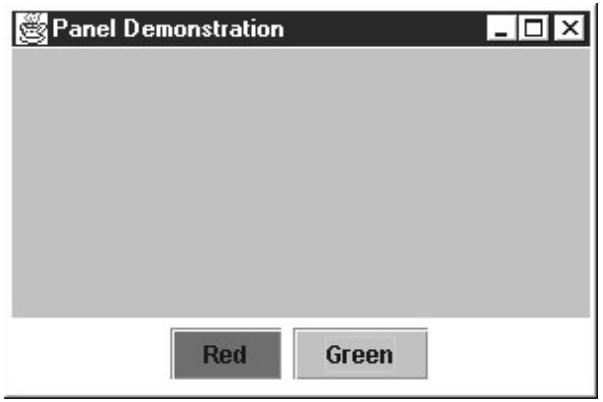
```
public void actionPerformed(ActionEvent e)
  {
    Container contentPane = getContentPane( );

    if (e.getActionCommand( ).equals("Red"))
        contentPane.setBackground(Color.RED);
     else if (e.getActionCommand( ).equals("Green"))
        contentPane.setBackground(Color.GREEN);
     else
        System.out.println("Error in button interface.");
  }
}
```

JPanel

- Used for hierarchical organization of GUIs:
  - » A panel can contain other components
  - » A panel can be added to another container

```
JPanel buttonPanel = new JPanel();
buttonPanel.setLayout(new FlowLayout());
buttonPanel.add(stopButton);
buttonPanel.add(goButton);
contentPane.add(buttonPanel, BorderLayout.SOUTH);
```





Content Pane of a **`JFrame`**

- Components are added to the content pane of a `JFrame` rather than directly to the `JFrame`
- The method `getContentPane` returns a reference to the content pane, which is treated as type `Container`

```
Container contentPane = getContentPane();
JLabel label = new JLabel("blue");
contentPane.add(label);
```

- For containers other than `JFrame` used in this book, `getContentPane` is not used

12.13Hierarchy of Swing Classes