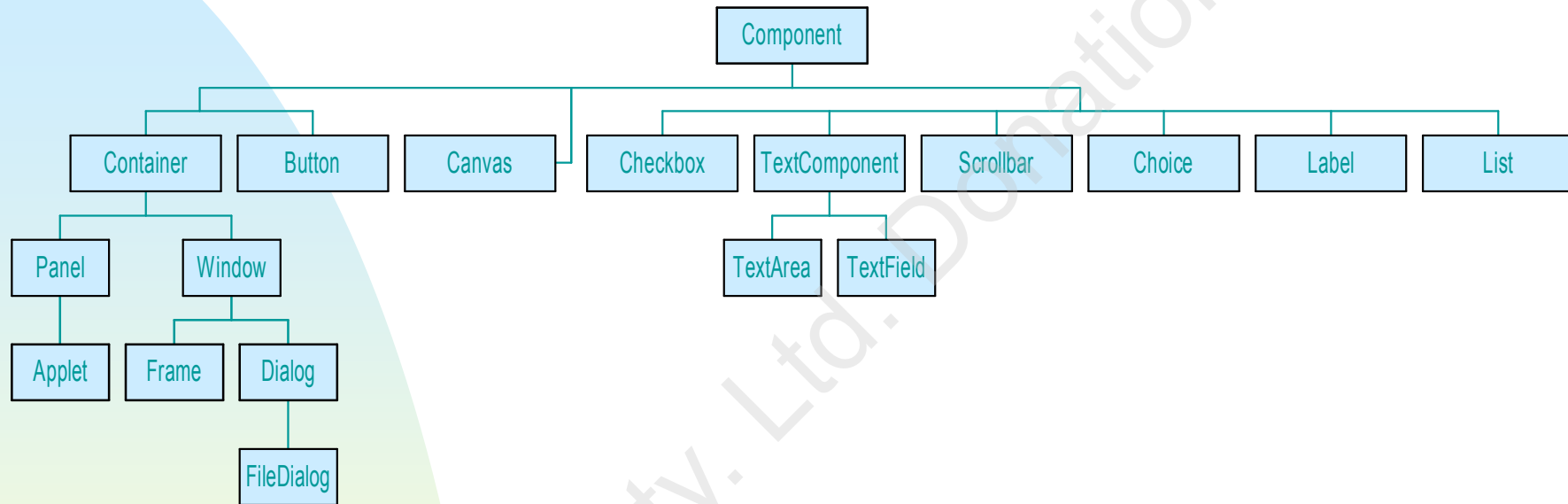


More AWT

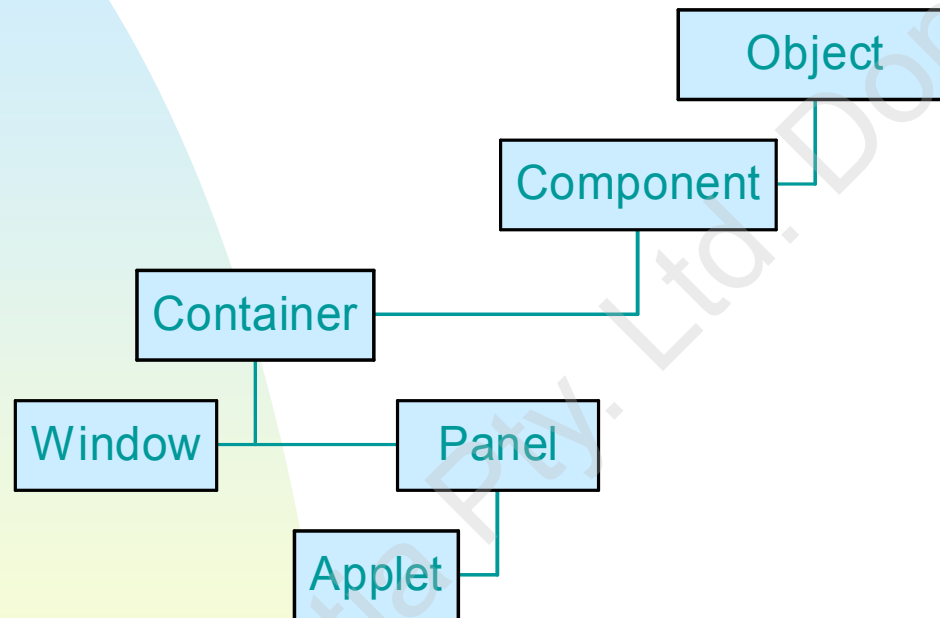
- Standard GUI components



- ◆ reminder: an Applet is already a component
 - ☞ implicitly knows a lot about GUIs

More AWT

- Components
 - ◆ reminder: an Applet is already a component



- ◆ implicitly knows a lot about GUIs

More AWT

- Coordinate system
 - ◆ each component has it's own coordinate system

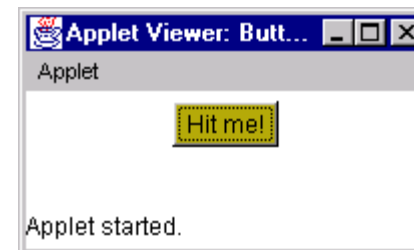


More AWT

■ Component: Button

```
import java.awt.*; import java.awt.event.*;

public class ButtonTest extends java.applet.Applet
{
    public void init ()
    {
        Button theButton = new Button ("Hit me!");
        theButton.addActionListener
        (
            new ActionListener ()
            {
                public void actionPerformed (ActionEvent ae)
                {
                    Color c = new Color ((float) Math.random (),
                                           (float) Math.random (), (float) Math.random ());
                    ((Button) ae.getSource ()).setBackground (c);
                }
            }
        );
        add (theButton);
    }
}
```



Sunday, July 05, 2009

More AWT

■ Component: Checkbox

◆ 2 types

☞ non-exclusive, n selected at a time

```
add (new Checkbox ("I'm an option"));
```

☞ exclusive, 1 at a time (“radio buttons”)

- belong to a CheckboxGroup

```
Checkbox hickory,  
        dickory,  
        dock;
```

```
CheckboxGroup myGroup = new CheckboxGroup ();  
add (hickory = new Checkbox ("Hickory", myGroup, true));    // 'on'  
add (dickory = new Checkbox ("Dickory", myGroup, false));  
add (dock = new Checkbox ("Dock", myGroup, false));
```

More AWT

- Component: Choice

- ◆ select 1 item from a pop-up

```
Choice theChoice = new Choice ();  
theChoice.addItem ("Item 1");  
theChoice.addItem ("Item 2");  
add (theChoice);
```

- Component: List

- ◆ allow user to select n items from a list

```
List l = new List (3, true); // 3 lines visible  
l.addItem ("Item 1");      // multiple selections allowed  
l.addItem ("Item 2");  
l.addItem ("Item 3");  
l.addItem ("Item 4");  
l.addItem ("Item 5");  
add (l);
```

More AWT

- Component: Label
 - ◆ doesn't generate events
 - ◆ single line of text *only*
 - ◆ various alignments
- Component: TextField
 - ◆ a line of optionally editable text
 - ◆ supports echo character
 - ☞ passwords, etc.
- Component: TextArea
 - ◆ mini text editor
 - ☞ scrollbars, editing, selections, etc.

More AWT

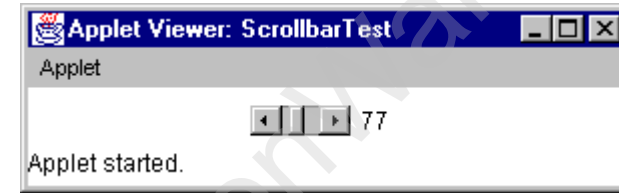
■ Component: Scrollbar

```
Scrollbar scroll = new Scrollbar (Scrollbar.VERTICAL);  
scroll.setValues (0, 8, 0, 255 + 8);  
scroll.setBlockIncrement (16);
```



“Note that the maximum value above, 255, is the maximum value for the scroll bar’s bubble. The actual width of the scroll bar’s track is 255 + 8. When the scroll bar is set to its maximum value, the left side of the bubble is at 255, and the right side is at 255 + 8.”

More AWT



```
import java.awt.*;
import java.awt.event.*;

public class ScrollbarTest extends java.applet.Applet
{
    private Label lab;

    public void init ()
    {
        Scrollbar sb = new Scrollbar (Scrollbar.HORIZONTAL);
        sb.setValues (0, 8, 0, 255 + 8);
        sb.setBlockIncrement (16);
        sb.addAdjustmentListener
        (
            new AdjustmentListener ()
            {
                public void adjustmentValueChanged (AdjustmentEvent e)
                {
                    lab.setText (e.getValue () + "");
                }
            }
        );
        add (sb);

        lab = new Label (sb.getValue () + "");
        add (lab);
    }
}
```

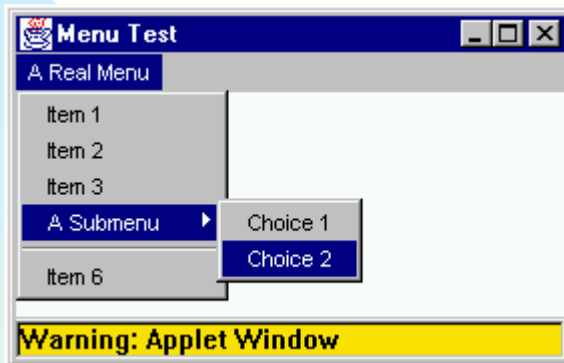
Sunday, July 05, 2009

More AWT

- Component: Canvas
 - ◆ draw / paint surface
 - ◆ responds to mouse movement and key presses
 - ◆ must be extended to be truly useful
 - ☞ paint method does the drawing you want to do

More AWT

- Component: Menu & Component: MenuBar



MenuItems generate
ActionEvents when selected

```
public Frame initFrame ()
{
    Frame theFrame = new Frame ("Menu Test");

    MenuBar theMenubar = new MenuBar ();
    Menu theMenu = new Menu ("A Real Menu");
    theMenu.add (new MenuItem ("Item 1"));
    theMenu.add (new MenuItem ("Item 2"));
    theMenu.add (new MenuItem ("Item 3"));
    Menu subMenu = new Menu ("A Submenu");
    subMenu.add (new MenuItem ("Choice 1"));
    subMenu.add (new MenuItem ("Choice 2"));
    theMenu.add (subMenu);
    theMenu.add (new MenuItem ("-"));
    theMenu.add (new MenuItem ("Item 6"));
    theMenubar.add (theMenu);
    theFrame.setMenuBar (theMenubar);

    [snip...]

    return (theFrame);
}
```

More AWT

- Container: Panel
 - ◆ superclass of Applet
 - ◆ borderless
 - ◆ separate:
 - ☞ clipping area
 - ☞ coordinate system
 - ☞ layout manager
 - ◆ used to
 - ☞ group components
 - ☞ reserve space in a layout

More AWT

- Container: Window

- ◆ used to create separate window

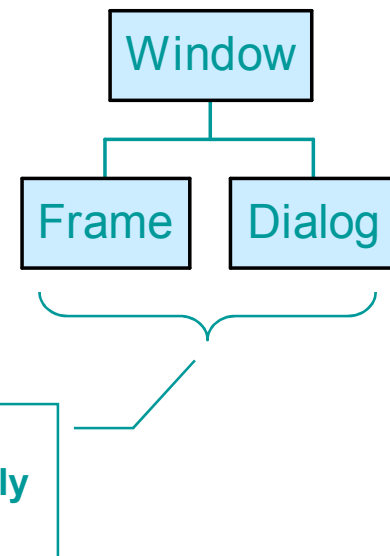
- ☞ not within main applet screen real estate

- ◆ no

- ☞ border / decorations

- ☞ menus

- ◆ not often directly used



More AWT

■ Container: Frame

◆ 'standard' window

☞ decorations

- optionally resizable

☞ menus

☞ etc.

◆ usually need to subclass to do any 'real' work

```
Frame f = new Frame ("My Frame");  
f.setSize (new Dimension (200, 300));  
f.setLocation (50, 50);  
f.setVisible (true);  
f.setResizable (true);
```

■ Container: Dialog

◆ can be modal or non-modal

☞ FileDialog subclass

- only useful for applications, of course

More AWT

- Layout managers

- ◆ portable computing needs to account for differences between platforms

- ☞ washing machine

- ☞ network computer

- ☞ 63" HDTV

- ◆ different fonts, coordinate system, resolution, etc.

- ◆ need to produce 'appropriate' UI regardless

- ◆ each container has an associated layout manager

- ☞ encapsulates and enforces layout policies

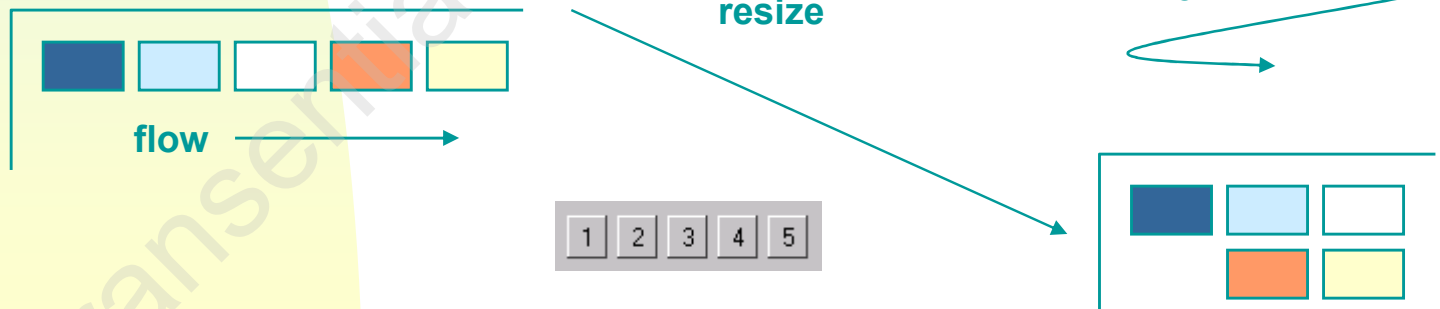
- ☞ when container is (re)sized, manager (re)arranges contained components

More AWT

■ FlowLayout

◆ default layout manager for Panels (Applets)

```
public class TestDefaultLayout extends Applet
{
    public void init ()
    {
        add (new Button ("1"));
        add (new Button ("2"));
        add (new Button ("3"));
        add (new Button ("4"));
        add (new Button ("5"));
    }
}
```

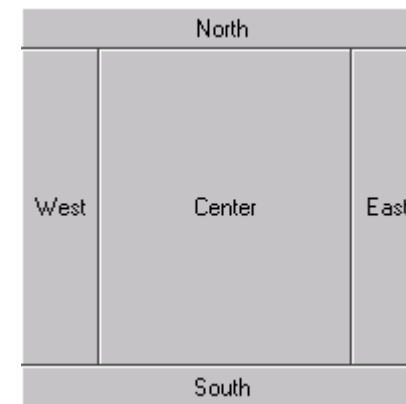


More AWT

■ BorderLayout

- ◆ default for Windows (incl. Frames & Dialogs)
- ◆ aligns components around the edge of a container
 - ☞ 4 cardinal points
 - ☞ CENTER takes up slack

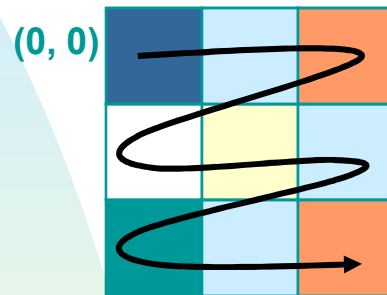
```
public class TestBorderLayout extends Applet
{
    public void init ()
    {
        setLayout (new BorderLayout ());
        add (BorderLayout.NORTH, new Button ("North"));
        add (BorderLayout.SOUTH, new Button ("South"));
        add (BorderLayout.EAST, new Button ("East"));
        add (BorderLayout.WEST, new Button ("West"));
        add (BorderLayout.CENTER, new Button ("Center"));
    }
}
```



More AWT

■ GridLayout

- ◆ arranged in a grid



evenly sized
according to
contents

1	2	3
4	5	6
7	8	9

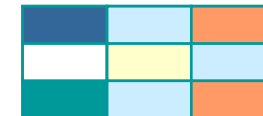
```
setLayout (new GridLayout (3, 3));  
add (new Button ("1"));  
add (new Button ("2"));  
...
```

- ◆ establish component's

- ☞ preferredSize
- ☞ minimumSize
- ☞ maximumSize

Dimension

resize



More AWT

■ GridBagLayout

◆ “the mother of all layout managers”

☞ “incredibly complex”

☞ “We recommend that you set the weights at 100. Then run the program and see how the layout looks. If you want to **tweak** the sizes of the columns or rows, adjust the weights”

☞ “...it’s just too painful and ridiculous to come up with an example for this...”

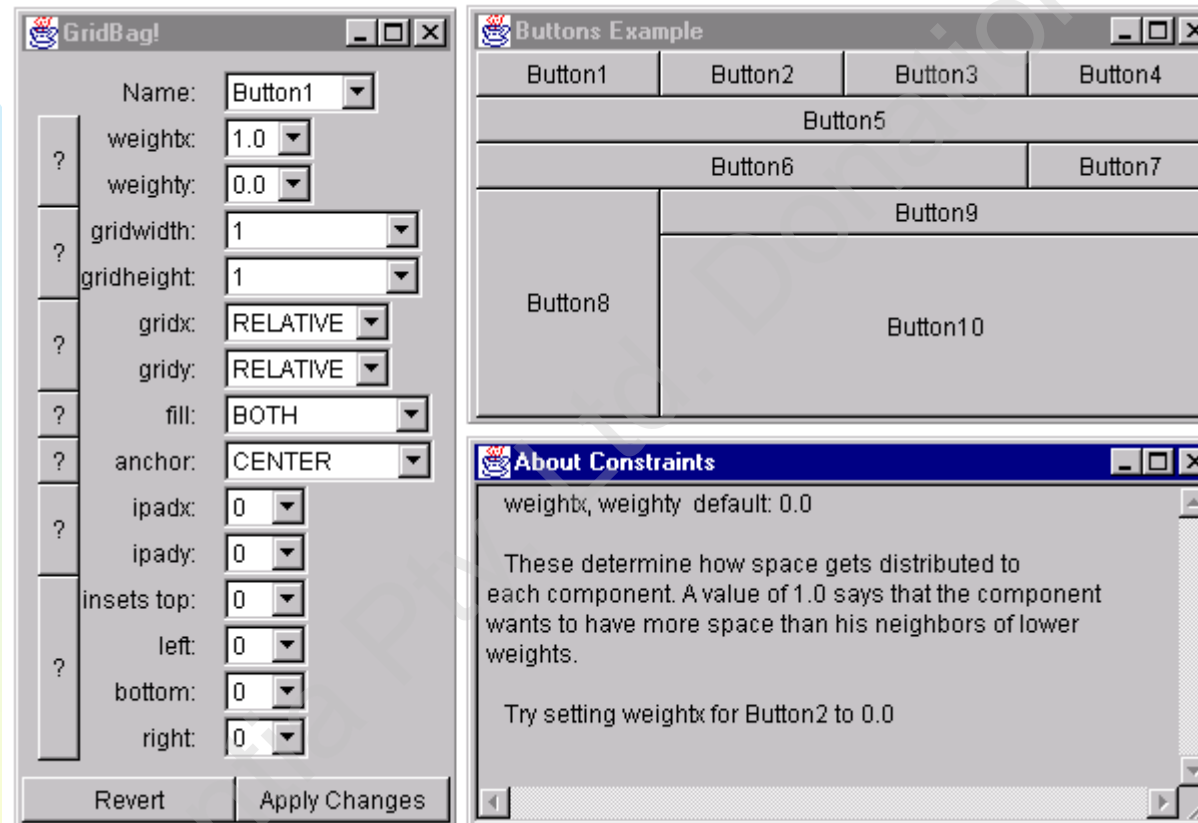
◆ most flexible

☞ only manager that allows components to span cells (& thus be unequal sizes)

☞ constraint based

- associated GridBagConstraints class

More AWT



More AWT

■ ...but if you *must*...

```
public LabelledSlider (String prefix, int val, int min,
                      int max, ChangeListener listener)
{
    this.prefix = prefix;
    this.listener = listener;

    setLayout (new GridBagLayout ());
    GridBagConstraints gbc = new GridBagConstraints ();
    slider = new JSlider (JSlider.HORIZONTAL, min, max, val);
    slider.addChangeListener (this);
    slider.setMajorTickSpacing (max / 4);
    slider.setMinorTickSpacing (max / 8);
    slider.setPaintTicks (true);
    slider.setPaintLabels (true);

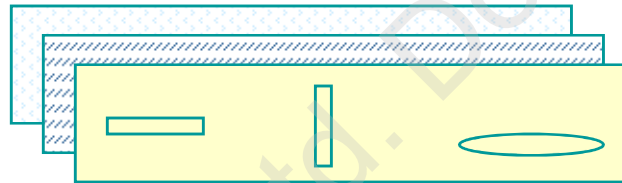
    gbc.weightx = 1;
    gbc.gridwidth = 1;
    add (this.label = new JLabel (prefix + slider.getValue ()), gbc);
    gbc.weightx = 5;
    gbc.gridwidth = 10;
    gbc.fill = GridBagConstraints.HORIZONTAL;
    gbc.anchor = GridBagConstraints.WEST;
    add (slider, gbc);
}
```

More AWT

- CardLayout

- ◆ layers components on top of each other

- ☞ only topmost visible at any time



- ◆ cycle through the cards

- ☞ first

- ☞ last

- ☞ next

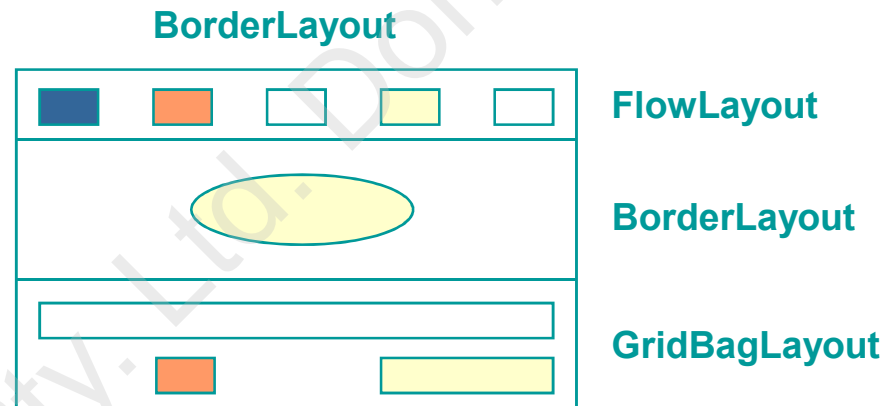
- ☞ previous

- ☞ show

Sunday, July 05, 2009

More AWT

- *Must* use multiple layout managers to achieve complex effect
 - ◆ also need numerous ‘filler’ panels



- Can use *no* layout manager/can “roll you own”
 - ◆ frowned upon...
 - ◆ (possibly) bye bye portability!
 - ◆ hello (more) aspirin, whiskey, etc.!