Container and Layouts
Visual Design

Object-Oriented Programming

Container

- Used to group *Component* together for display
- No limit to the number of Components
- Container is a subclass of component
  - A container object has access to all the component methods
  - A container object is a component object
  - A container can hold containers
- Parent – children relationship
- Local coordinate system
- Use layoutManager to arrange how its components
- An abstract class
Container – Organization Methods

- To add a component
  - Void add(Component comp)
    - Add the component to the end
  - Void add(Component comp, int index)
    - Add the component at given position
  - Void add(String name, Component comp)
    - For use with some layoutManager
- Force the Container to be laid out
  - Void doLayout()
- Find out about Components
  - Component getComponentAt(int x, int y)
    - Locates the component that contains the x,y position.
  - Component getComponentAt(Point p)
    - Gets the component that contains the specified point.
  - int getComponentCount()
    - Gets the number of components (only top-level) in this panel.

More about Container

- Find the relationship
  - For a component
    - Container getParent()
  - For a container
    - boolean isAncestorOf( Component c) (recursively)
- Remove component from container
  - void remove( Component comp)
    - Removes the specified component from this container.
  - void remove(int index)
    - Removes the component, specified by index, from this container.
  - void removeAll()
    - Removes all the components from this container.
- Search only top-level list
- Act recursively
Panel and Layout

- Panel – A container with choice of arranging Components;
- Class Panel extends Container
  - Panel() -- default “FlowLayout”
  - Panel(LayoutManager layout)
- Layout Methods for Container
  - Void doLayout()
  - LayoutManager getLayout()
  - Dimension getMinimumSize()
  - Dimension getPreferredSize()
  - Void setLayout(LayoutManager layout)

Different Layout

- Layout classes implement LayoutManager interface
- FlowLayout
  - From left to right, from top to bottom
- BorderLayout
  - Five regions: north, south, east, west and center
- GridLayout
  - A number of rows and columns
  - Each component has the same size
- No Layout
  - Do it by ourselves
FlowLayout

- From left to right, from top to bottom
- Fit all components in the visible portion of their Container (If possible)
- Alignments:
  - FlowLayout.LEFT
  - FlowLayout.RIGHT
  - FlowLayout.CENTER
- Constructors:
  - FlowLayout()
  - FlowLayout(int alignment)
  - FlowLayout(int alignment, int hGap, int vGap)

```java
this.setLayout(newFlowLayout(FlowLayout.RIGHT, 20, 10));
```

More About FlowLayout

- Access methods:
  - int getAlignment() - get the Alignment
  - int getVgap() - get vertical gap
  - int getHgap() - get horizontal gap
- Modify things:
  - Void setAlignment()
  - Void setHgap(int h)
  - Void setVgap(int v)
- Default layout for Panel and Applet
BorderLayout

- Five regions specified by:
  - BorderLayout.NORTH
  - BorderLayout.EAST
  - BorderLayout.WEST
  - BorderLayout.SOUTH
  - BorderLayout.CENTER
- North and South get preferred heights, stretched horizontally
- East and West get preferred widths, stretched vertically
- Resize components to fill their allotted spaces.

```java
This.add(BorderLayout.EAST, myButton);
```

More about BorderLayout

- Constructors:
  - BorderLayout()
  - BorderLayout(int hGap, int vGap);
- Access methods:
  - int getHgap();
  - int getVgap();
- Modifier:
  - void setHgap(int h);
  - Void setVgap(int v);
GridLayout

- Specify a number of rows and columns
- Add components in order: from left to right, from top to bottom
- Resize component to fill the cell
- All cells have the same sizes

Constructors

- GridLayout()
- GridLayout(int rows, int columns)
- GridLayout(int rows, int columns, int hgap, int vgap)

Accessors:

- int getColumn(); int getRows()
- int getHgap(); int getVgap();

Modifiers:

- void setColumn(int columns); void setRows(int rows)
- void setHgap(int h); void setVgap(int v);

No Layout

- myContainer. setLayout(null);

Methods to arrange components:

- setSize()
- setBounds()
- setLocation()

Canvas

- To paint on the screen: Canvas()
- Has a paint() method
Class WarningMessage extends Canvas{
    private String myMessage;
    Public WarningMessage(String message) {
        myMessage = message;
        setSize(60, 25);
    }
    Public void paint(Graphics g) {
        g.setColor(Color.red);
        g.setFont(new Font("SansSerif", Font.BOLD, 18));
        g.drawString("!", 10, 20);
        g.drawString("!", 16, 24);
        g.drawString("!", 22, 16);
        g.setColor(Color.black);
        g.setFont(new Font("SansSerif", Font.BOLD+Font.ITALIC, 10));
        g.drawString(myMessage, 32, 18);
    }
}

Windows
- Window - Frame, Dialog
- A window appears as a rectangular area displayed on top of anything already showing
- A naked rectangle without any of ornaments
- Default layout: BorderLayout
- Useful methods:
  - void dispose() – free up all resources
  - void pack() – resize the window to fit the components it contains
  - void show() – make it appear on the screen
Frames

- A Frame is a Window with: title, scroll bars, menu bar, cursor, system widgets (hide, minimize, grow)
- Default size (0, 0); setBounds()
- Initially invisible; show()

Constructors
- Frame()
- Frame(String title)

Resize
- boolean isResizable()
- void setResizable(boolean canResize)

Dialogs

- A Dialog is a Window intended for simple input and output
- Default layout: BorderLayout
- A Dialog
  - must be associated with a Frame known as a parent frame.
  - With Frame, must set size and location of a Dialog
  - Show(), make it appear; dispose() free up resources

Constructors
- Dialog(Frame parent)
- Dialog(Frame parent, String title)
- Dialog(Frame parent, boolean isModal)
- Dialog(Frame parent, String title, boolean isModal)

Useful methods
- boolean isModal(); void setModal(boolean isModal)
- boolean isResizable(); void setResizable()
Summary

- Container – Organization Methods
  - More about Container
- Panel and Layout
  - Different Layout
    - FlowLayout
      - More About FlowLayout
    - BorderLayout
      - More about BorderLayout
    - GridLayout
    - No Layout
- Windows
- Frames
- Dialogs