Swing

Part 1 – Layout Managers

Chapter 11, Core Java, Volume I

Contents

- Layout Managers
- Flow Layout
- Border Layout
- Grid Layout
- Example: Calculator

Layout Managers

- In Swing, you can lay out components in a container:
  - by writing layout code by hand
  - by using drag-and-drop visual GUI builders
  - by choosing a layout manager

- *Layout managers* determine the positions and sizes of components in a container.
  - Flow Layout (default in JPanel)
  - Border Layout (default in JFrame)
  - Grid Layout
  - Grid Bag Layout
  - Box Layout
  - Card Layout
  - Spring Layout
  - Group Layout
- Can achieve simple layouts by nesting panels with border layout, flow layout, grid layout, etc.

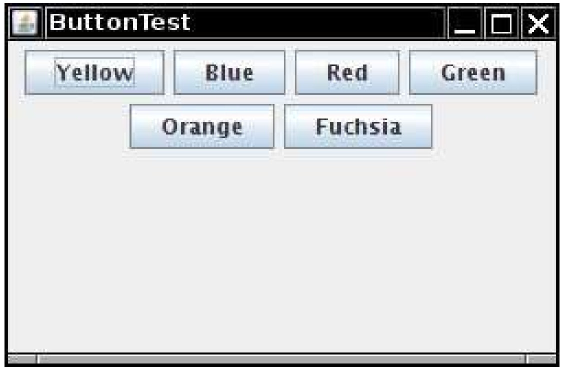
Flow Layout

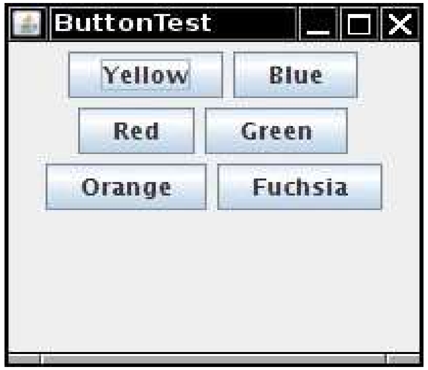
- The FlowLayout manager puts components in a row, sized at their preferred size.
  - a new row is started when there is no more space.
  - defulat layout for JPanel

- Alignment of components in a flow layout
  - FlowLayout.CENTER (default)
  - FlowLayout.LEFT
  - FlowLayout.RIGHT

- Creating and Setting a Layout Manager

```
MyFrame f = new MyFrame();
FlowLayout fl = new FlowLayout(FlowLayout.LEFT);
f.setLayout(fl);
```



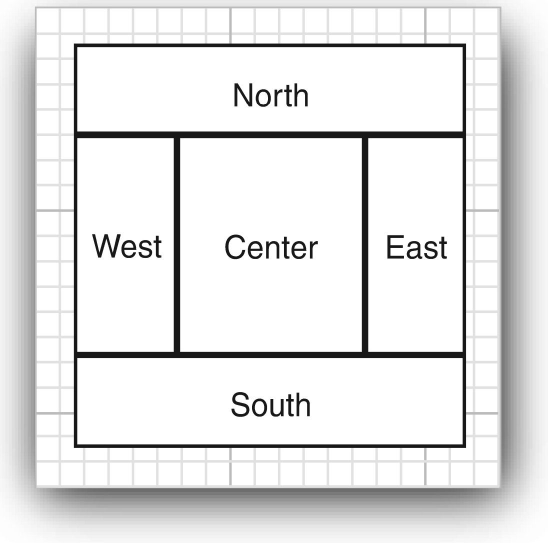


Border Layout

- Default for the content pane of a `JFrame`.
- Five named areas:

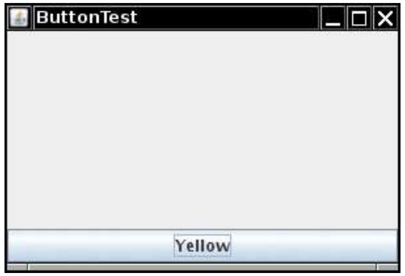
  `frame.add(panel, BorderLayout.SOUTH);`

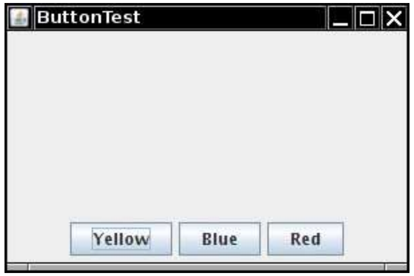
- Unlike the flow layout, the border layout grows all components to fill the available space.



- **Caution**: Don't put a button directly into an area.
  - Put buttons into a `JPanel`, and add the panel.

```
JPanel panel = new JPanel();
panel.add(yellowButton);
panel.add(blueButton);
panel.add(redButton);
frame.add(panel, BorderLayout.SOUTH);
```

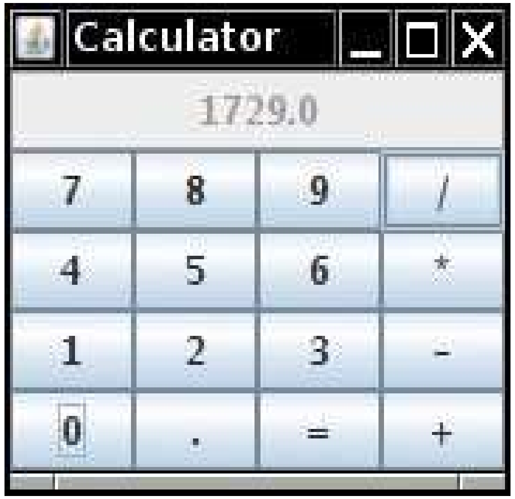




Grid Layout

- Grid layout arranges all components in rows and columns:

```
panel.setLayout(new GridLayout(4, 4));
panel.add(new JButton("1"));
panel.add(new JButton("2"));
```



- All components are given the same size.
- Components grow to fit the entire cell.
- If row or column count is zero, an arbitrary number of components can be added:

```
toolbar.setLayout(new GridLayout(0,2));
```

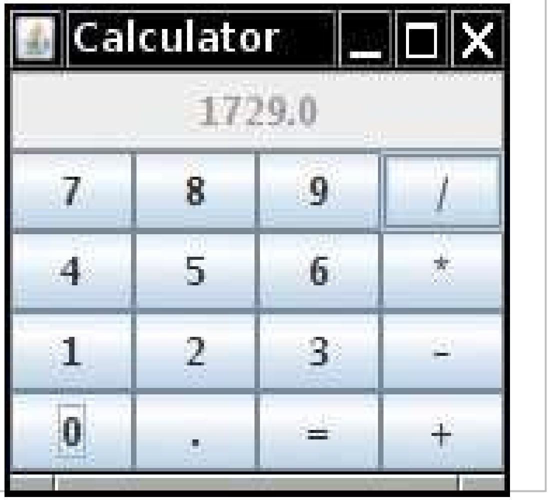
Example: Calculator

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

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

```
import javax.swing.*;

public class CalculatorFrame extends JFrame
{
   public CalculatorFrame()
   {
      add(new CalculatorPanel());
      pack();
   }
}
```



Example: Calculator

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


// A panel with calculator buttons and a result display.


public class CalculatorPanel extends JPanel
{
   private JButton display; // display for input/result
   private JPanel panel;      // panel for grid
   private double result;
   private String lastCommand;
   private boolean start;
```

```
public CalculatorPanel()
{
   setLayout(new BorderLayout());

   result = 0;
   lastCommand = "=";
   start = true;

   // add the display

   display = new JButton("0");
   display.setEnabled(false);
   add(display, BorderLayout.NORTH);

   ActionListener insert = new InsertAction();
   ActionListener command
                             = new CommandAction();
```

Example: Calculator

```
     // add the buttons in a 4 x 4 grid

    panel = new JPanel();
    panel.setLayout(new GridLayout(4, 4));

    addButton("7", insert);
    addButton("8", insert);
    addButton("9", insert);
    addButton("/", command);

    addButton("4", insert);
    addButton("5", insert);
    addButton("6", insert);
    addButton("*", command);
```

```
    addButton("1", insert);
    addButton("2", insert);
    addButton("3", insert);
    addButton("-", command);

    addButton("0", insert);
    addButton(".", insert);
    addButton("=", command);
    addButton("+", command);

    add(panel, BorderLayout.CENTER);
 } // end of constructor
```

Example: Calculator

```
 private void addButton(String label,
                              ActionListener listener)
 {
    JButton button = new JButton(label);
    button.addActionListener(listener);
    panel.add(button);
 }
void calculate(double x)
 {
    if (lastCommand.equals("+")) result += x;
    else if (lastCommand.equals("-")) result -= x;
    else if (lastCommand.equals("*")) result *= x;
    else if (lastCommand.equals("/")) result /= x;
    else if (lastCommand.equals("=")) result = x;
    display.setText("" + result);
 }
```

```
private class InsertAction implements
      ActionListener
{
   public void actionPerformed(ActionEvent event)
   {
      String input = event.getActionCommand();
      if (start)
      {
         display.setText("");
         start = false;
      }
      display.setText(display.getText() + input);
   }
}
```

Example: Calculator

```
private class CommandAction implements
                                   ActionListener
{
   public void actionPerformed(ActionEvent event)
   {
      String command = event.getActionCommand();

      if (start)
      {
         if (command.equals("-"))
         {
            display.setText(command);
            start = false;
         }
         else lastCommand = command;
      }
```

```
         else
         {
            calculate(Double.parseDouble(display.getText()));
            lastCommand = command;
            start = true;
         }
      }
   } // end of actionPerformed()
} // end of CalculatorPanel class
```