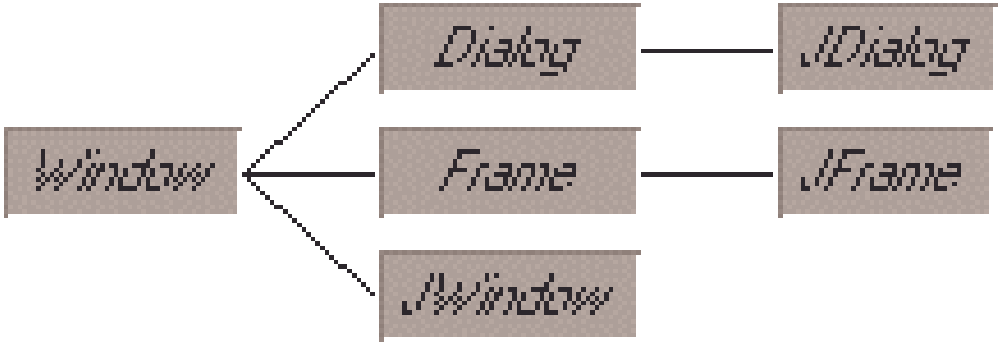
Part 13

Java Programming Language
Mr.Rungrote Phonkam
rungrote@it.kmitl.ac.th

Contents

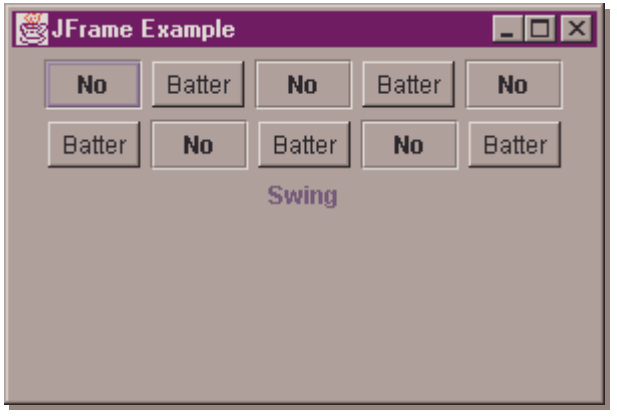
1. Swing Container



1.1 Swing Container(JFrame)

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

public class FrameTester {
  public static void main (String args[]) {
    JFrame f = new JFrame ("JFrame Example");
    Container c = f.getContentPane();
    c.setLayout (new FlowLayout());
    for (int i = 0; i < 5; i++) {
      c.add (new JButton ("No"));
      c.add (new Button ("Batter"));
    }
    c.add (new JLabel ("Swing"));
    f.setSize (300, 200);
    f.show();
  }
}
```



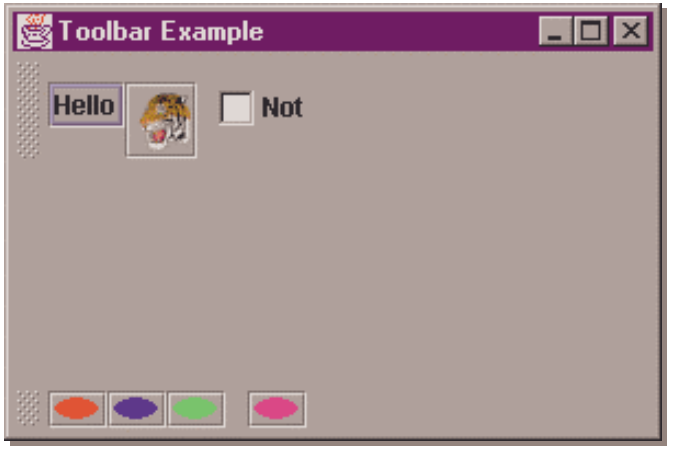
1.2 Swing Container(Toolbars)

```
public class ToolbarPanel extends JPanel {
  ToolbarPanel() {
    setLayout (new BorderLayout());
    JToolBar toolbar = new JToolBar();
    JButton myButton = new JButton("Hello");
    toolbar.add(myButton);
    Icon tigerIcon = new ImageIcon("SmallTiger.gif");
    myButton = new JButton(tigerIcon);
    toolbar.add(myButton);    toolbar.addSeparator();
    toolbar.add (new Checkbox ("Not"));
    add (toolbar, BorderLayout.NORTH);
    toolbar = new JToolBar();    Icon icon = new AnOvalIcon(Color.red);
    myButton = new JButton(icon);
    toolbar.add(myButton);
    icon = new AnOvalIcon(Color.blue);    myButton = new JButton(icon);
    toolbar.add(myButton);
    icon = new AnOvalIcon(Color.green);    myButton = new JButton(icon);
    toolbar.add(myButton);    toolbar.addSeparator();
    icon = new AnOvalIcon(Color.magenta);
    myButton = new JButton(icon);
    toolbar.add(myButton);    add (toolbar, BorderLayout.SOUTH);
  }
```

1.2 Swing Container(Toolbars)

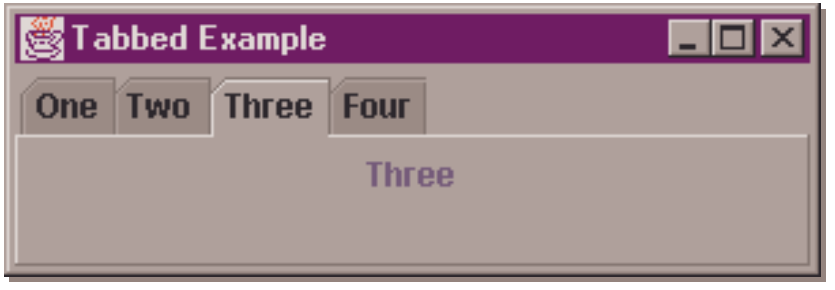
```
class AnOvalIcon implements Icon {
    Color color;
    public AnOvalIcon (Color c) {
      color = c;
    }
    public void paintIcon (Component c, Graphics g, int x, int y) {
      g.setColor(color);
      g.fillOval (x, y, getIconWidth(), getIconHeight());
    }
    public int getIconWidth() {
      return 20;
    }
    public int getIconHeight() {
      return 10;
    }
  }
}
```

1.2 Swing Container(Toolbars)



.3 Swing Container(JTabbedPane)

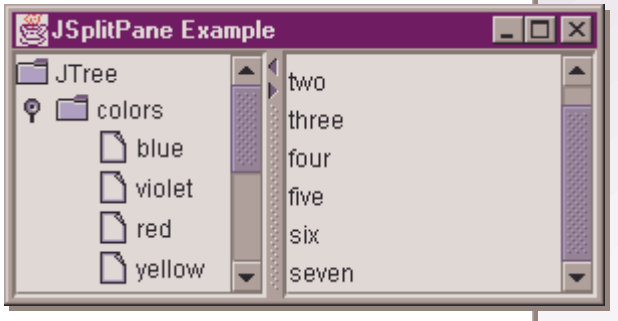
```
public class TabbedPanel extends JPanel {
  String tabs[] = {"One", "Two", "Three", "Four"};
  public JTabbedPane tabbedPane = new JTabbedPane();
  public TabbedPanel() {
    setLayout (new BorderLayout());
    for (int i=0;i<tabs.length;i++)
      tabbedPane.addTab (tabs[i], null, createPane (tabs[i]));
    tabbedPane.setSelectedIndex(0);
    add (tabbedPane, BorderLayout.CENTER);
  }
  JPanel createPane(String s) {
    JPanel p = new JPanel();
    p.add(new JLabel(s));
    return p;
  }
}
```

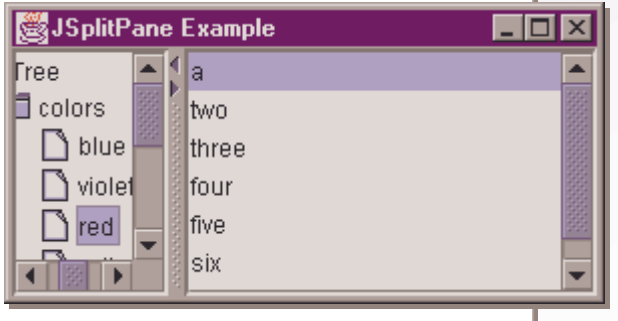


1.4 Swing Container(JSplitPane)

```
public class JSplitPanel extends JPanel {

  public JSplitPanel() {
    setLayout(new BorderLayout());
    JTree tree = new JTree();
    String[] items = {"a", "two", "three", "four", "five", "six", "seven"};
    JList list = new JList(items);
    JScrollPane left = new JScrollPane(tree);
    JScrollPane right = new JScrollPane(list);
    left.setMinimumSize(new Dimension(0,0));
    right.setMinimumSize(new Dimension(0,0));
    JSplitPane pane = new JSplitPane(
      JSplitPane.HORIZONTAL_SPLIT, left, right);
    pane.setDividerLocation(0.5);
    pane.setOneTouchExpandable(true);
    add(pane, BorderLayout.CENTER);
  }
}
```





2 Layout Managers

AWT Supported 5 Layout

- FlowLayout
- BorderLayout
- GridLayout
- CardLayout
- GridBagLayout

2.1 FlowLayout

- วาง Component ในแนวนอน
- เมื่อจำนวน component เต็มพื้นที่ในแนวนอน การวางจะเริ่มต้นที่แถวใหม่ด้านล่าง
- ไม่สามารถระบุตำแหน่งที่แน่ชัดของ Component ได้ เนื่องจากเวลาใช้งานจะมีการขยับตัวของ Component

2.1 FlowLayout

- วาง Component ในแนวนอน
- เมื่อจำนวน component เต็มพื้นที่ในแนวนอน การวางจะเริ่มต้นที่แถวใหม่ด้านล่าง
- ไม่สามารถระบุตำแหน่งที่แน่ชัดของ Component ได้ เนื่องจากเวลาใช้งานจะมีการขยับตัวของ Component

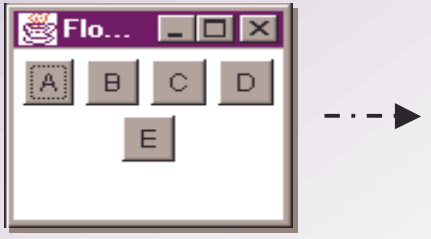
เมธรอดในการวาง Component

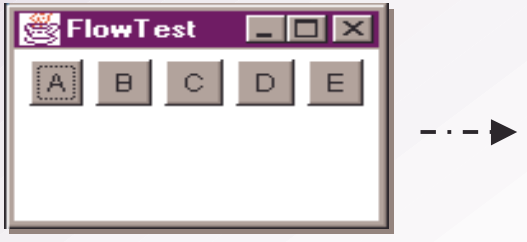
public Component add(Component comp)

public Component add(Component comp, int index)

2.1 FlowLayout

```
public class FlowTest {
    public static void main(String[] args) {
      Frame f = new Frame("FlowTest");
      f.setLayout(new FlowLayout());
      f.add(new Button("A"));
      f.add(new Button("B"));
      f.add(new Button("C"));
      f.add(new Button("D"));
      f.add(new Button("E"));
      f.setVisible(true);
    }
  }
```

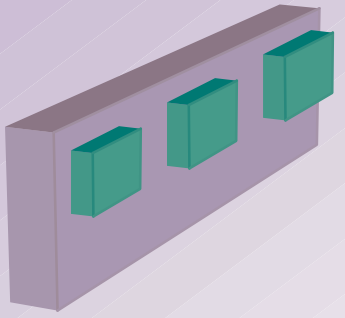




2.1 FlowLayout

การบังคับการขยับ Component ด้วย FlowLayout ซ้อน

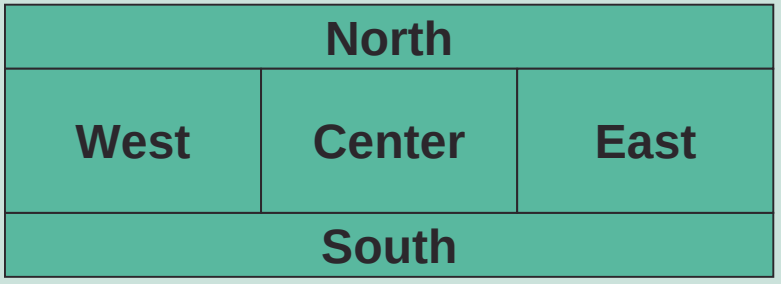
```
Panel p1 = new Panel(new FlowLayout());
Panel p2 = new Panel(new FlowLayout());
p2.add(new Button("A"));
p2.add(new Button("B"));
p2.add(new Button("C"));
p2.add(new Button("D"));
p2.add(new Button("E"));
p1.add(p2);
```



2.2 BorderLayout

- วาง Component ตามแนวทิศ

  North, West, East, South and Center



- ถ้าส่วนไหนไม่มีการวาง Component จะถูกตัดออกไป

เมธรอดในการวาง Component

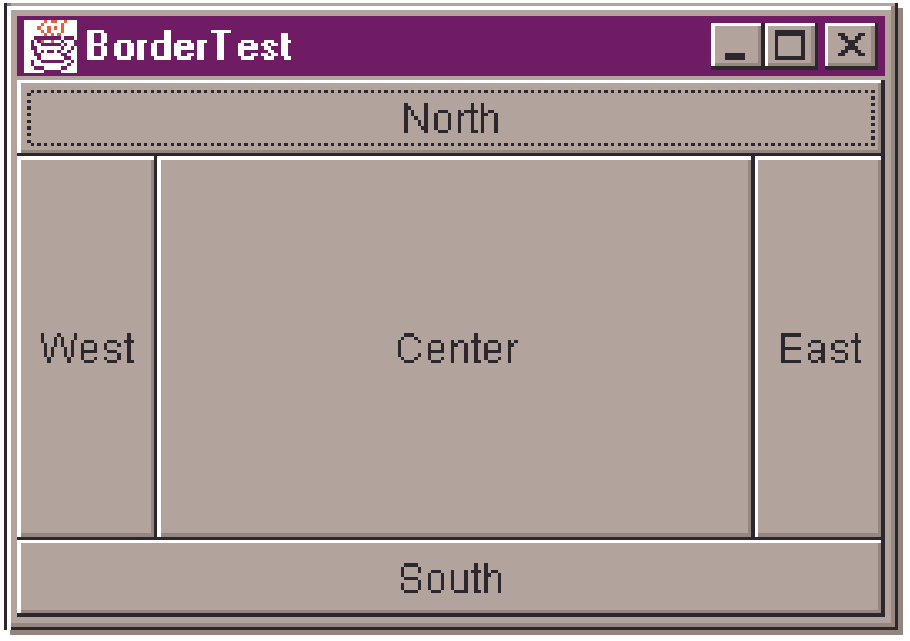
public void add(String constraint, Component component)
public void add(Component component, Object constraint)

2.2 BorderLayout

```
Frame f = new Frame("orderTest");
 f.setLayout(new BorderLayout());
 f.add(new Button("North"),  BorderLayout.NORTH);
 f.add(new Button("South"),  BorderLayout.SOUTH);
 f.add(new Button("East"),   BorderLayout.EAST);
 f.add(new Button("West"),   BorderLayout.WEST);
 f.add(new Button("Center"), BorderLayout.CENTER);
```

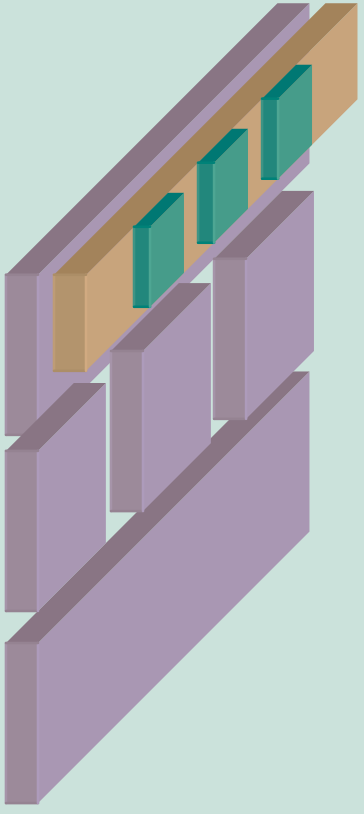
```
Frame f = new Frame("BorderTest");
 f.setLayout(new BorderLayout());
 f.add("North",  new Button("North"));
 f.add("South",  new Button("South"));
 f.add("East",   new Button("East"));
 f.add("West",   new Button("West"));
 f.add("Center", new Button("Center"));
```

2.2 BorderLayout



2.2 BorderLayout

BorderLayout & FlowLayout



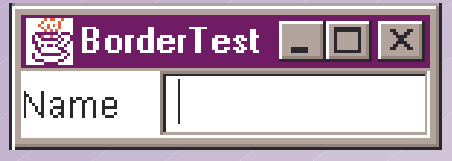
- BorderLayout
  - North
    - FlowLayout
      - Component1, 2, ...
  - West
  - Center
  - East
  - South

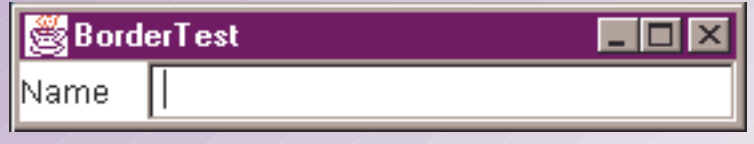
2.2 BorderLayout

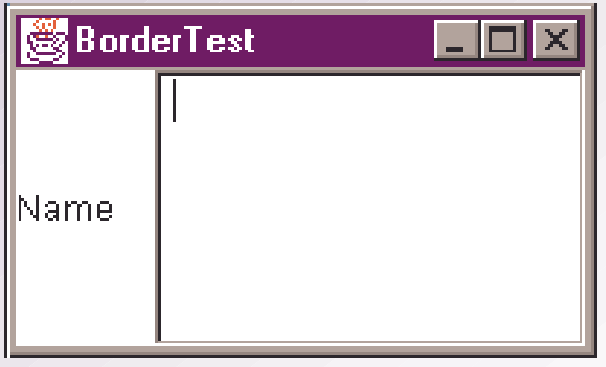
```
Panel flow = new Panel(new FlowLayout());
Panel border = new Panel(new BorderLayout());
flow.add(new Button("A")); flow.add(new Button("B"));
flow.add(new Button("C"));
flow.add(new Button("D"));
flow.add(new Button("E"));
border.add(flow, BorderLayout.NORTH);
```

2.2 BorderLayout

```
Panel p = new Panel(new BorderLayout());
Label nameLabel = new Label("Name:");
TextField entry = new TextField(); p.add(nameLabel, BorderLayout.WEST);
p.add(entry, BorderLayout.CENTER);
```



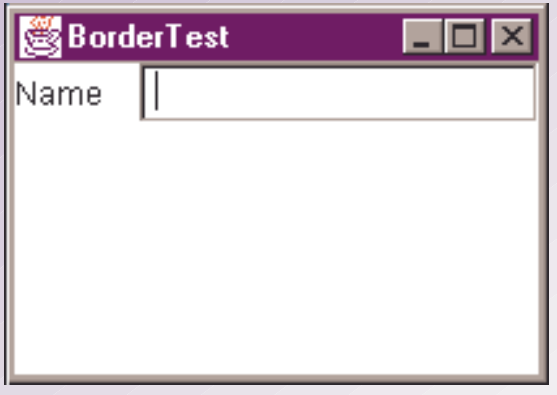




2.2 BorderLayout

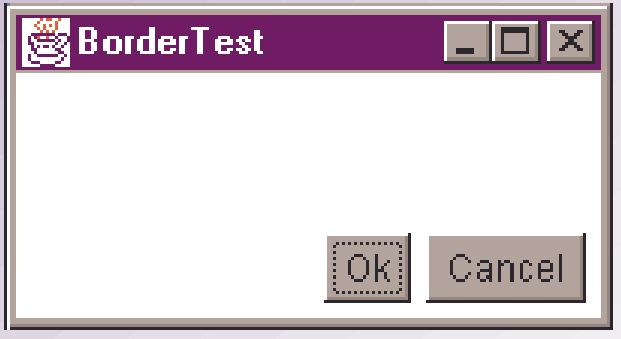
```
Panel p = new Panel(new BorderLayout());
Label nameLabel = new Label("Name:");
TextField entry = new TextField(); p.add(nameLabel, BorderLayout.WEST);
p.add(entry, BorderLayout.CENTER);
Panel p2 = new Panel(new BorderLayout());

p2.add(p, BorderLayout.NORTH);
```



2.2 BorderLayout

```
Frame f = new Frame("BorderTest");
Panel p = new Panel(new FlowLayout(FlowLayout.RIGHT));
f.setLayout(new BorderLayout());
p.add(new Button("Ok"));
p.add(new Button("Cancel"));
f.add(p, BorderLayout.SOUTH);
```



2.3 GridLayout

- วาง Component ตามช่องแนวนอนและแนวตั้ง
- สามารถระบุได้ว่าต้องการกี่ส่วนในแนวนนอน และกี่ส่วนในแนวตั้ง

GridLayout(hSize, vSize, *hGap, vGap*)

hSize	คือจำนวนช่องแนวนอน (Row)
vSize	คือจำนวนช่องแนวตั้ง (Column)
hGap	ระยะห่างแนวนอน
vGap	ระยะห่างแนวตั้ง

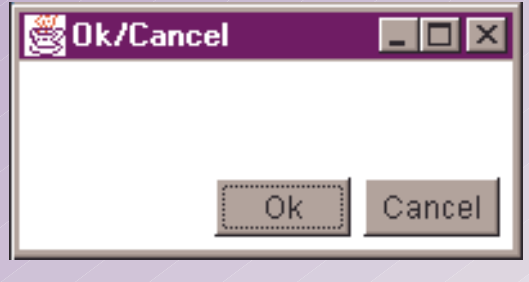
- ขนาดของแต่ละส่วนมีขนาดเท่ากัน

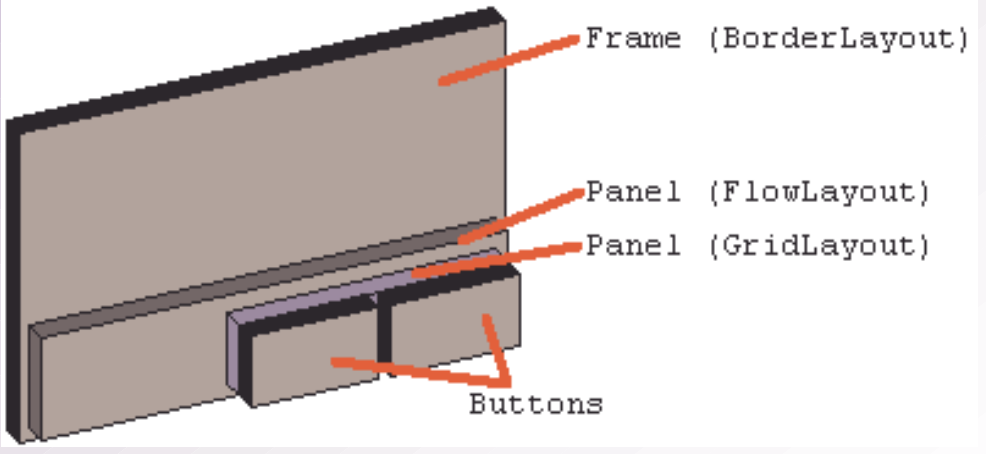
2.3 GridLayout

```
Frame f = new Frame("Grid Test");
f.setLayout(new GridLayout(3,4));
for (int x = 1; x < 13; x++)
f.add(new Button(""""+x));
```

2.3 GridLayout

```
Frame f = new Frame("Ok/Cancel");
f.setLayout(new BorderLayout());
Panel p = new Panel();
p.setLayout(new FlowLayout(FlowLayout.RIGHT));
Panel p2 = new Panel()
p2.setLayout(new GridLayout(1,0,5,5));
p2.add(new Button("Ok"));
p2.add(new Button("Cancel"));
p.add(p2, BorderLayout.EAST);
f.add(p, BorderLayout.SOUTH);
```





2.4 CardLayout

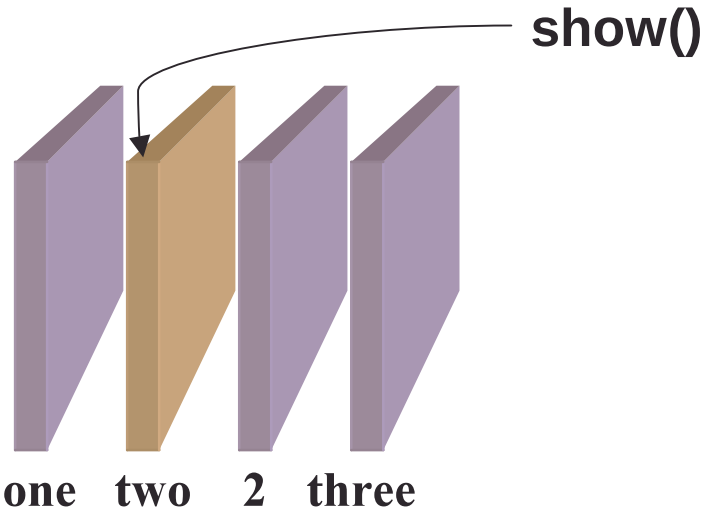
- แสดง Component หนึ่งตัวในช่วงเวลา
- การวาง Component ใช้เมธรอดใดๆเหล่านี้

  public void add(Component component, String key);
  public void add(String key, Component component);
  public void add(String key, Component component, int index);

- การแสดง Component ใช้เมธรอด show
- การสลับการแสดง Component ใช้เมธรอด next, previous

2.4 CardLayout

```
Panel p = new Panel(new CardLayout());
p.add("one", new Button ("the first component"));
p.add(new Button ("the second component"), "two");
p.add("three", new Button ("the third component"));
p.add("between two and three", new Button ( "the fourth component"), 2);
CardLayout l = (CardLayout)p.getLayout();
l.show(p, "two");
```

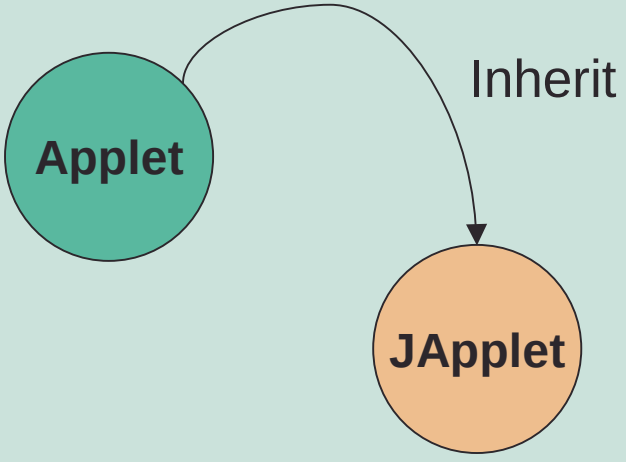


2.5 GridBagLayout

- ซับซ้อนในการใช้งาน
- ซับซ้อนในการดูแล และจัดการ
- ยังมี Bug อยู่ใน JDK 1.2
- ใช้หลักการของ state ในการจัดการ Component ต่างๆ

3 Swing in Applet

- JApplet สืบทอดคลาสมาจาก Applet
- JApplet มีการจัดการเลย์เอาท์แบบ BorderLayout ในขณะที่ Applet เป็นแบบ FlowLayout



3 Swing in Applet

```
public class AppTester extends JApplet {
 public void init () {
   Container c = getContentPane();
   JButton jb = new JButton ("Default");
   c.add (jb, BorderLayout.WEST);
   jb = new JButton ("LayoutManager");
   c.add (jb, BorderLayout.CENTER);
   jb = new JButton ("is");
   c.add (jb, BorderLayout.EAST);
   jb = new JButton ("BorderLayout: " +
     (c.getLayout() instanceof BorderLayout));
   c.add (jb, BorderLayout.SOUTH);
 }
}
```

Practice

จงพิจารณาลักษณะของการวางปุ่มกดบนเครื่องคิดเลข และให้เขียนโปรแกรมเพื่อสร้างอินเตอร์เฟสที่มีลักษณะคล้ายกับเครื่องคิดเลข