

# NGÔN NGỮ LẬP TRÌNH JAVA

## BUỔI 10 LẬP TRÌNH GIAO DIỆN VỚI AWT, SWING



GVGD: ThS. Lê Thanh Trọng

# NỘI DUNG

- 1. Giới thiệu Java GUI**
- 2. Container**
- 3. Nguyên tắc xây dựng GUI**
- 4. Layout**
- 5. Swing component**

- 1. Giới thiệu Java GUI**
2. Container
3. Nguyên tắc xây dựng GUI
4. Layout
5. Swing component

# Java Graphical User Interface (GUI)

- ❖ Gói *AWT* được sử dụng trong các phiên bản trước jdk1.2
- ❖ *Swing* được đưa vào trong phiên bản jdk1.2 và có nhiều chức năng hơn AWT
- ❖ *Swing components* thường có tên bắt đầu với 'J'
  - AWT có lớp *Panel*
  - Swing có lớp tương ứng là *JPanel*

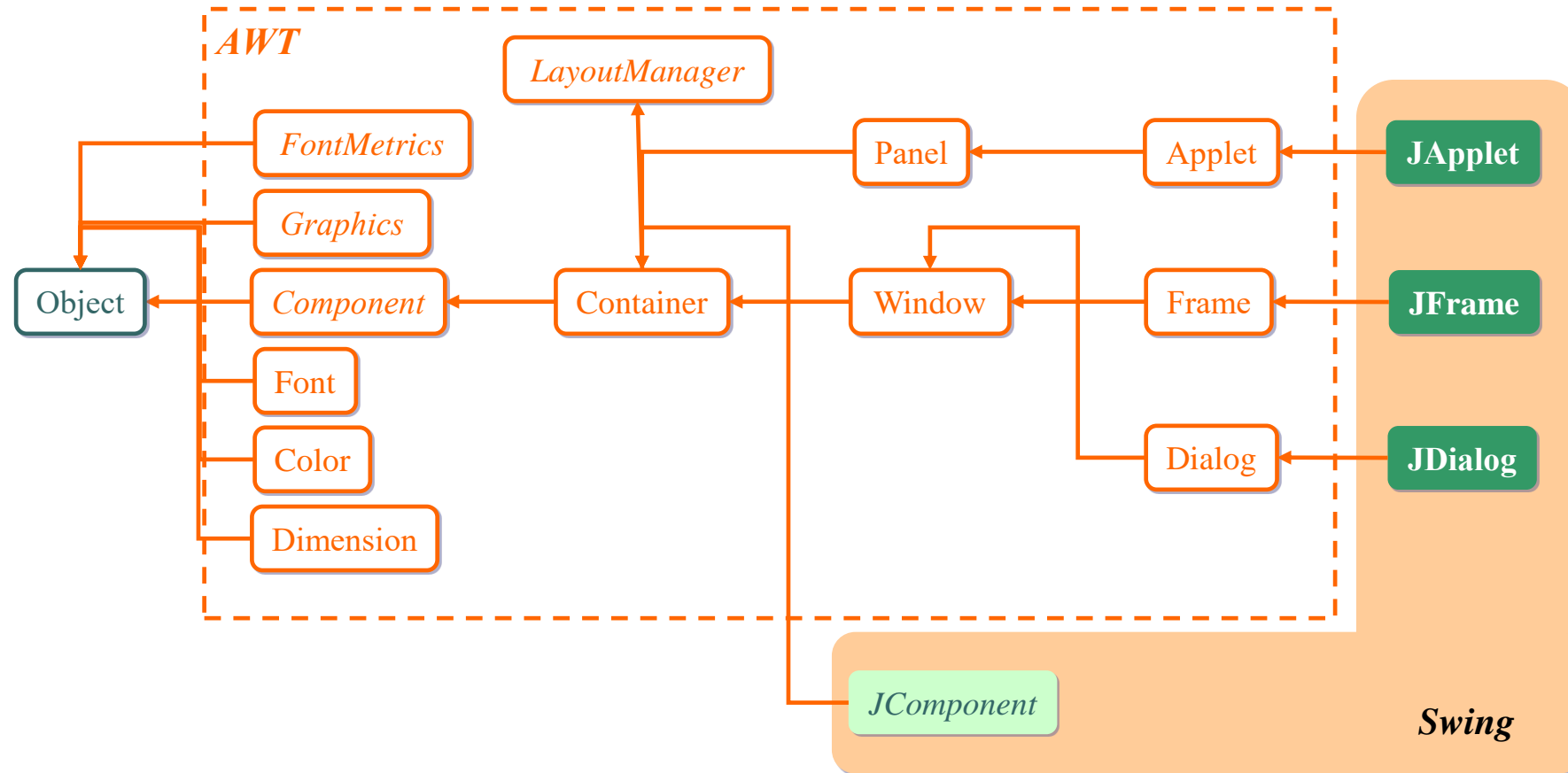
# Java GUI

- ❖ **AWT** (Abstract Windows Toolkits)
  - `import java.awt.*;`
- ❖ **Swing** (Java Foundation Classes Package)
  - `import javax.swing.*;`
  - `import java.awt.*;`
  - `import java.awt.event.*;`
- ❖ Tất cả Swing component có tên bắt đầu với J...
- ❖ Gói Swing là 1 thành phần của Java Foundation Classes (JFC cung cấp một tập các chức năng giúp xây dựng các ứng dụng GUI)

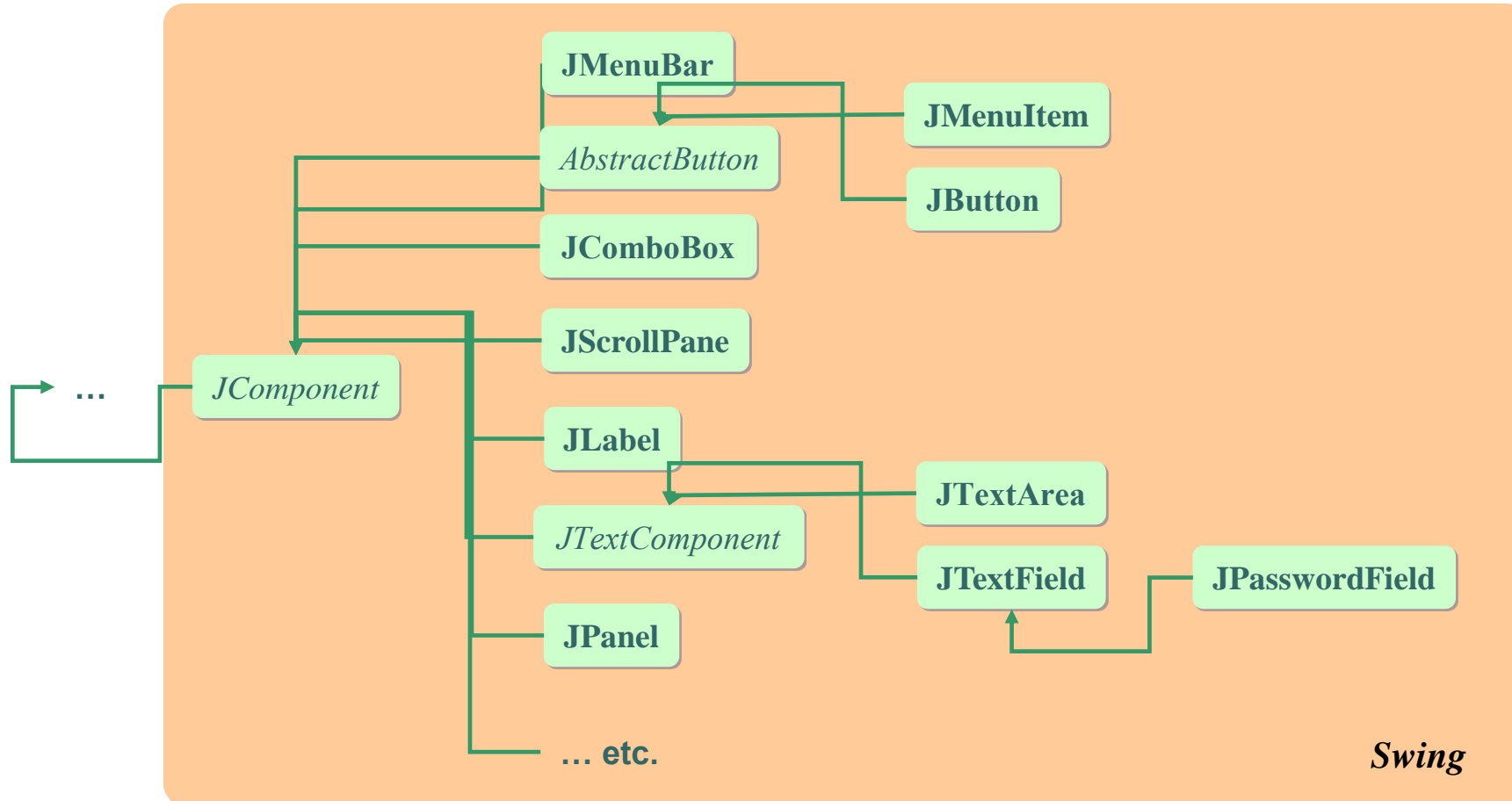
# Java GUI hierarchy

- ❖ Swing Components có nhiều mức khác nhau
- ❖ Các mức này qui định các Component trong 1 ứng dụng GUI kết hợp với nhau
- ❖ Cách kết hợp này gọi là *Swing Containment Hierarchy*

# Java GUI hierarchy



# Java GUI hierarchy





# NỘI DUNG

1. Giới thiệu Java GUI
- 2. Container**
3. Nguyên tắc xây dựng GUI
4. Layout
5. Swing component

# Container

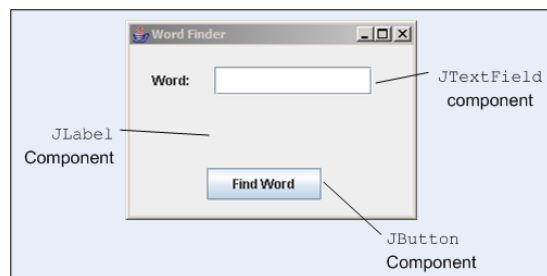
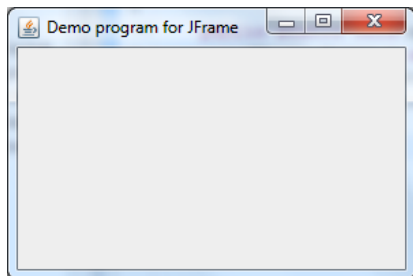
- ❖ Là thành phần mà có thể chứa các thành phần khác, có thể vẽ và tô màu.
  - Frame/JFrame, Panel/JPanel, Dialog/JDialog, ScrollPane/JScrollPane, ...
- ❖ Gắn component vào khung chứa
  - *containerObj.add(compObj);*
- ❖ Lấy thông tin của component
  - *objectName.get...();*
- ❖ Gán thông tin cho component
  - *objectName.set...();*

# NỘI DUNG

1. Giới thiệu Java GUI
2. Container
- 3. Nguyên tắc xây dựng GUI**
4. Layout
5. Swing component

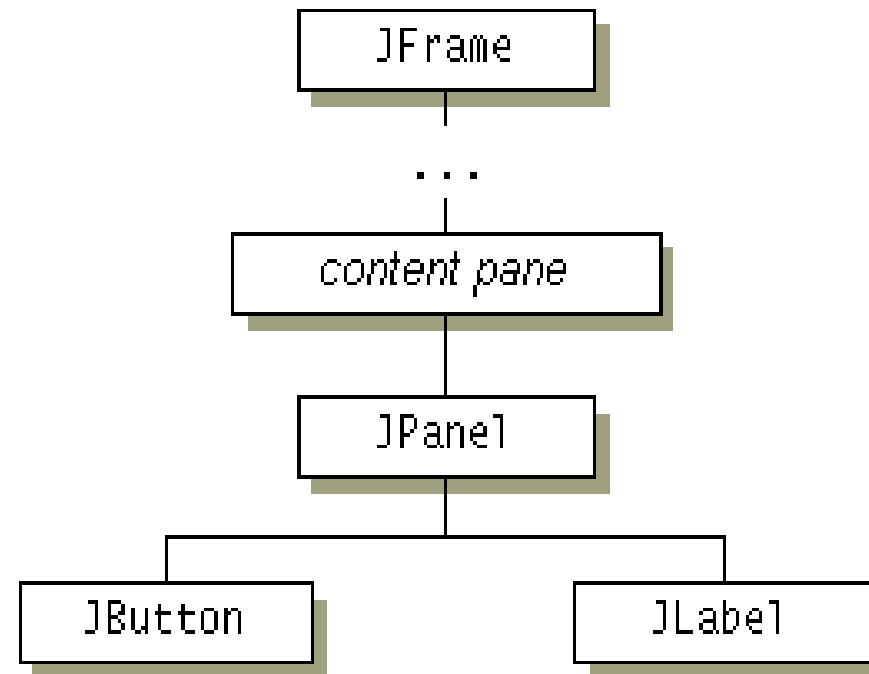
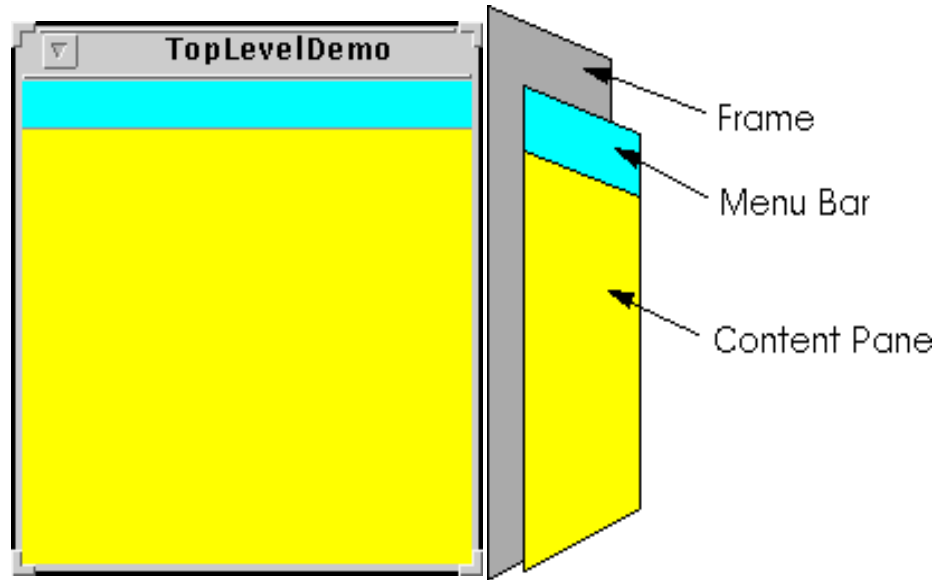
# Nguyên tắc xây dựng GUI

1. Lựa chọn 1 container: Frame/JFrame, Window/JWindow, Dialog/JDialog, ...
2. Tạo các điều khiển: (buttons, text areas..)
3. Đưa các điều khiển vào vùng chứa
4. Sắp xếp các điều khiển(layout)
5. Thêm các xử lý sự kiện (Listeners)



# Containment Hierarchy

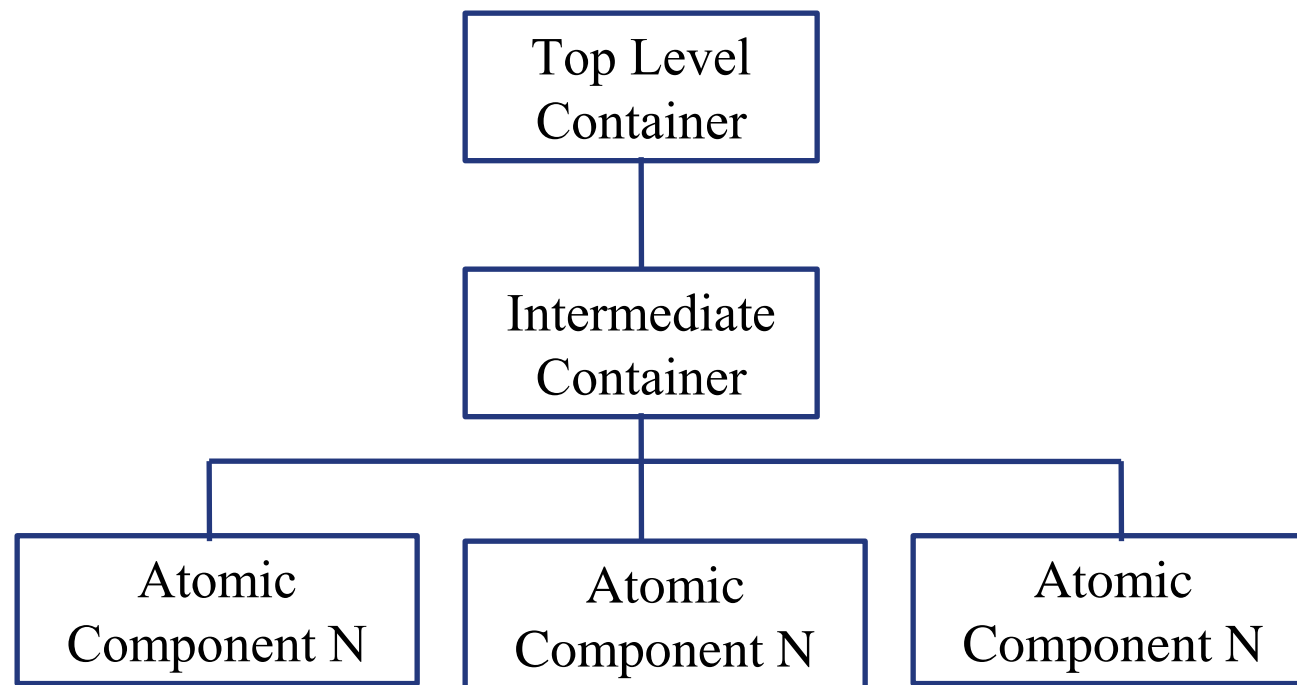
❖ Top-level container: JFrame, JDialog, JApplet



# Containment Hierarchy

## ❖ Top Level Containers

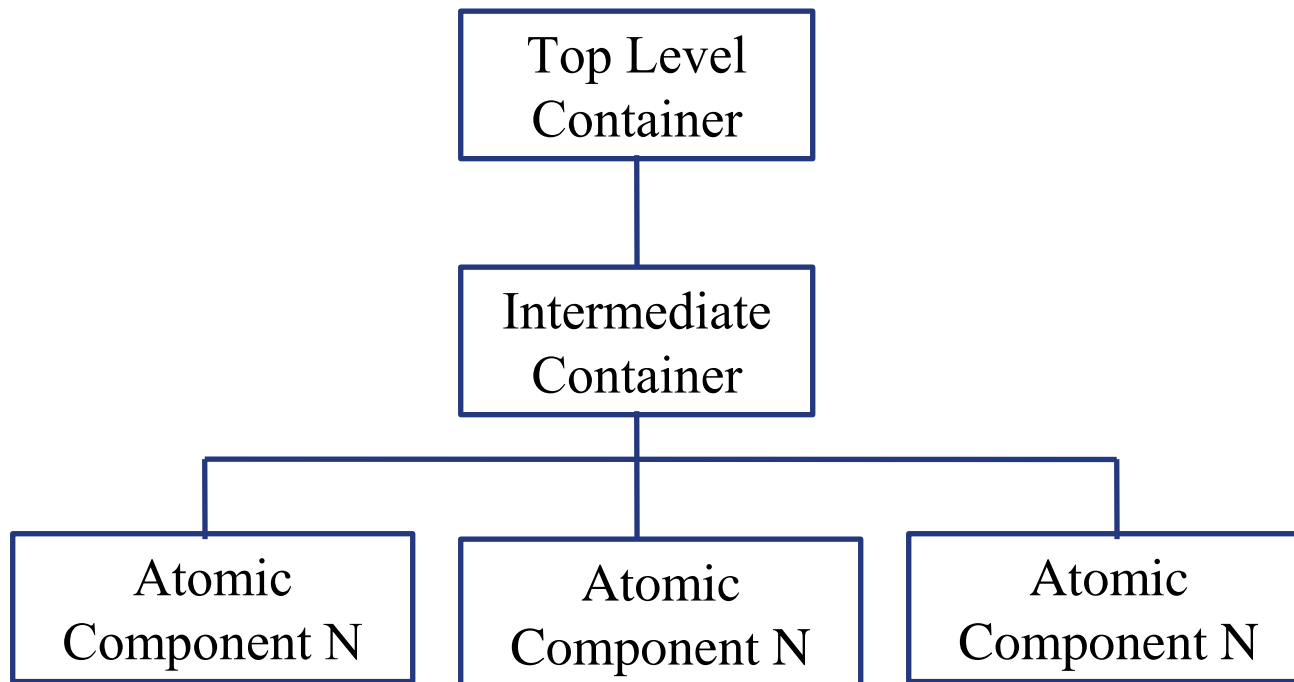
- Là nơi để hiển thị các Component khác
- Ví dụ:
  - JFrame
  - JDialog
  - JApplet



# Containment Hierarchy

## ❖ Intermediate Containers

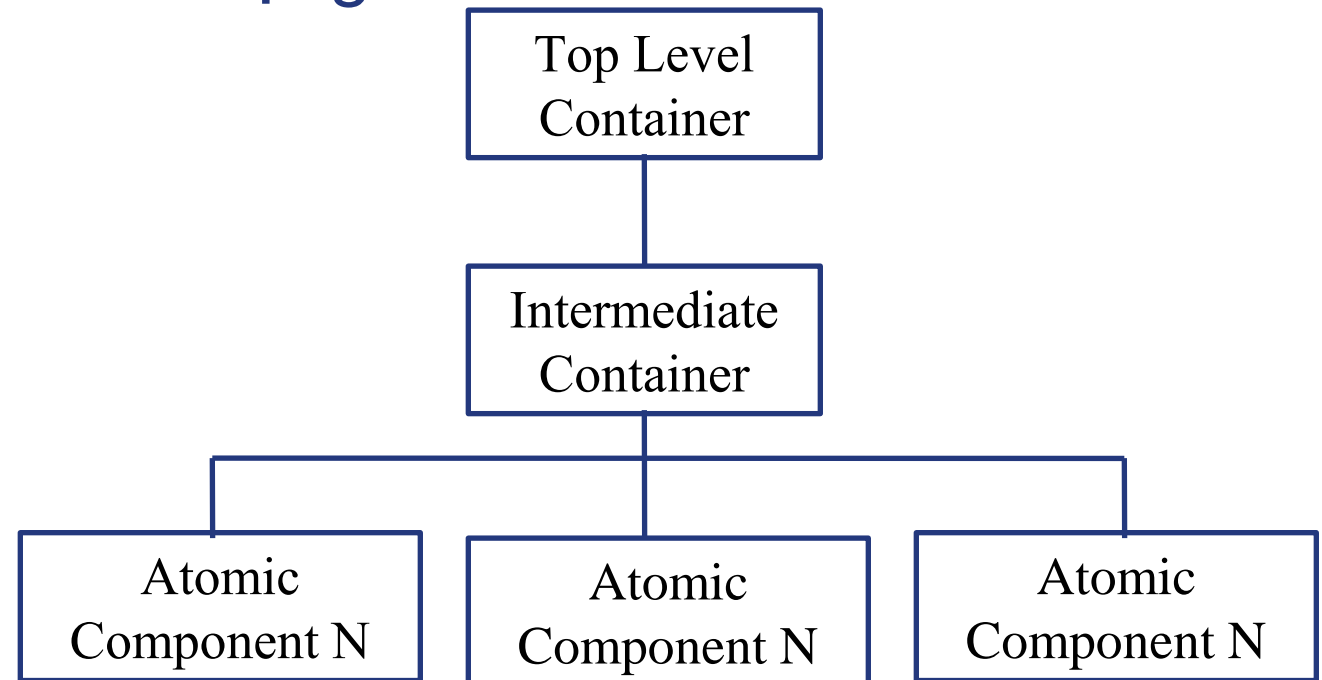
- Dùng để xác định vị trí của các Atomic Components
- Hoặc cung cấp cơ chế để tương tác với khung chứa
- Ví dụ:
  - JPanel
  - JScrollPane
  - JTabbedPane



# Containment Hierarchy

## ❖ Atomic Components

- Không chứa các Components khác
- Dùng để cung cấp thông tin đến người sử dụng
- Hoặc lấy thông tin từ người sử dụng
- Ví dụ:
  - JButton
  - JLabel
  - JComboBox
  - JTextField
  - JTable





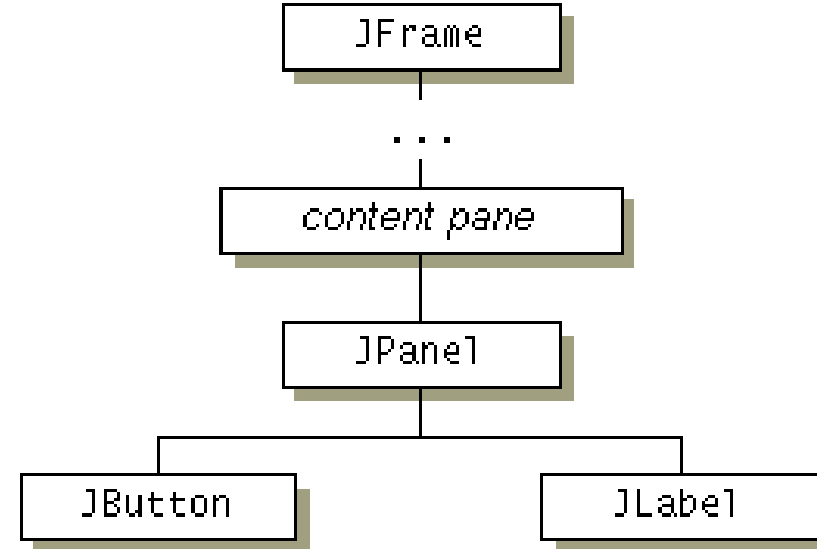
# Containment Hierarchy

- ❖ Mọi top level container có 1 intermediate container gọi là content pane
- ❖ Các atomic component phải gắn vào content pane
- ❖ Ví dụ:

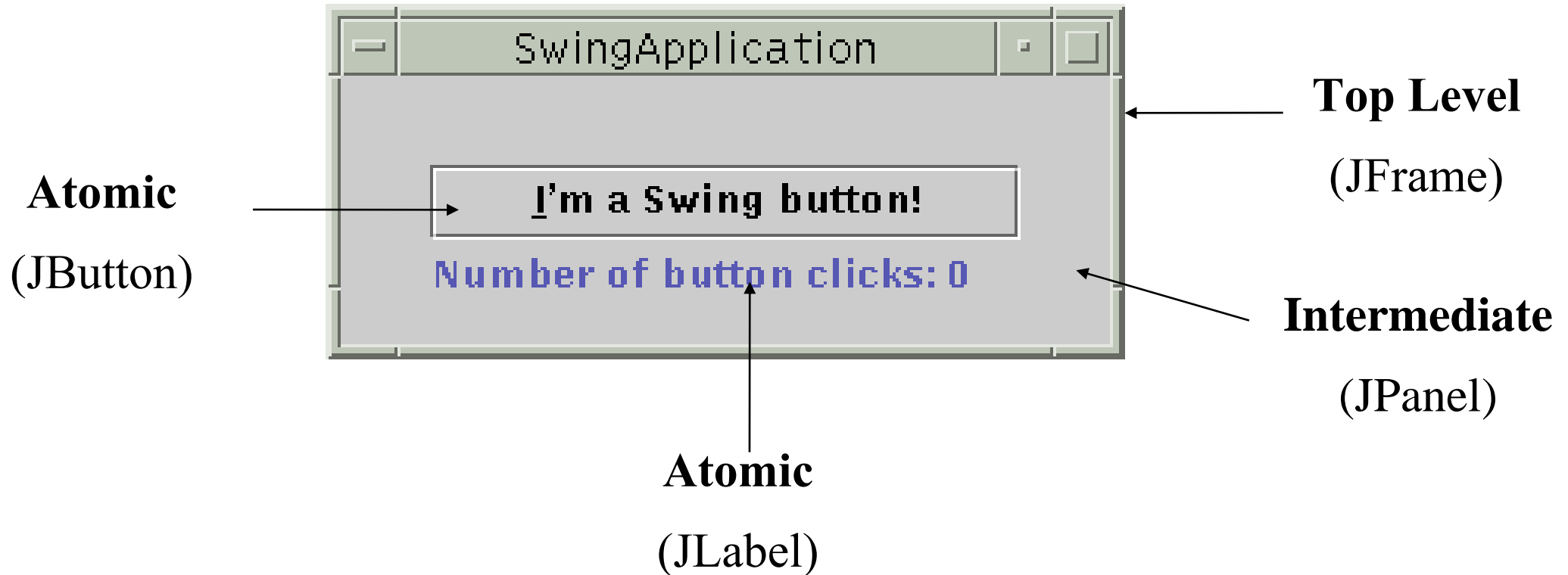
*JFrame frame = new JFrame(...);*

*JPanel pane = new JPanel();*

*frame.getContentPane().add(pane);*



# Containment Hierarchy



# JFrame

`javax.swing`

## Class JFrame

[`java.lang.Object`](#)

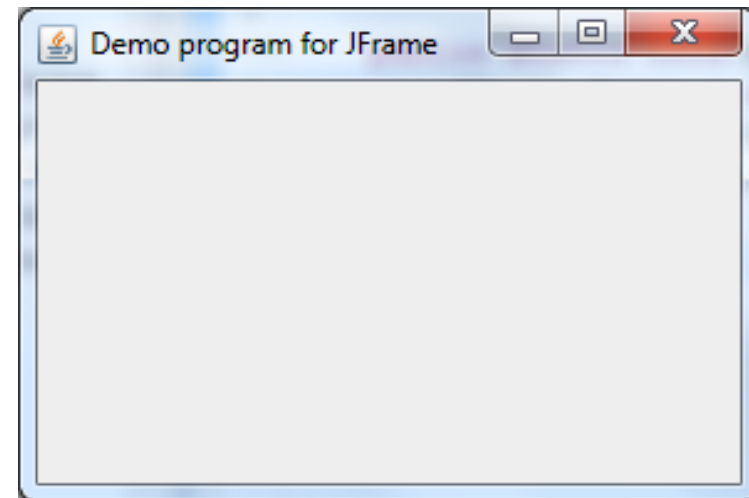
└ [`java.awt.Component`](#)

└ [`java.awt.Container`](#)

└ [`java.awt.Window`](#)

└ [`java.awt.Frame`](#)

└ `javax.swing.JFrame`



All Implemented Interfaces:

[`Accessible`](#), [`ImageObserver`](#), [`MenuContainer`](#), [`RootPaneContainer`](#), [`Serializable`](#), [`WindowConstants`](#)

# JFrame

```
// Thiet lap che do look and feel
JFrame.setDefaultLookAndFeelDecorated(true);

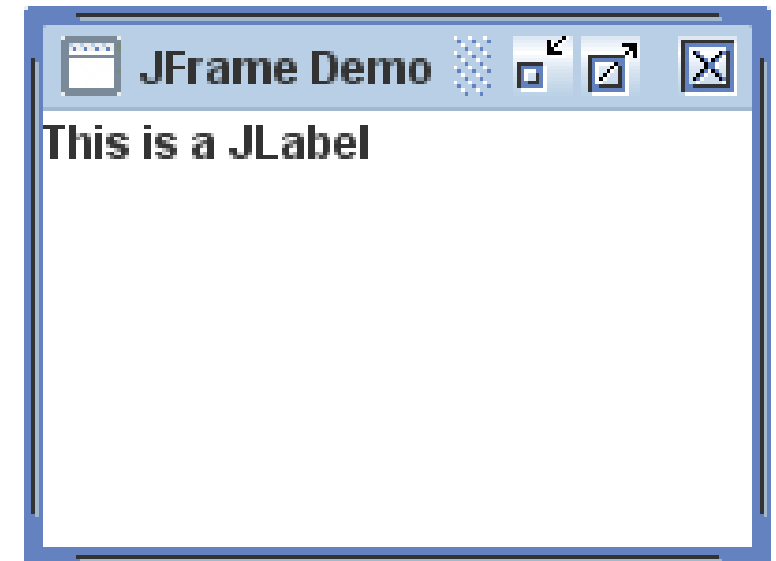
// Tao 1 doi tuong JFrame
JFrame frame = new JFrame("JFrame Demo");

// Xu ly khi dong frame
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

// Tao 1 component va them vao JFrame
JLabel label = new JLabel("This is a JLabel");
frame.getContentPane().add(label, BorderLayout.NORTH);

// Dinh cac thuoc tinh cho frame
frame.setBounds(50, 50, 200, 150);
frame.getContentPane().setBackground(Color.WHITE);

// Hien thi frame
frame.setVisible(true);
```



# NỘI DUNG

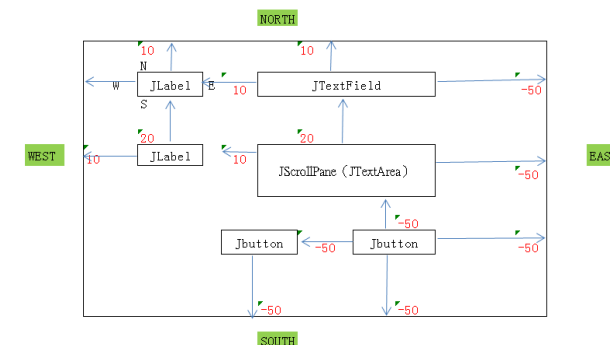
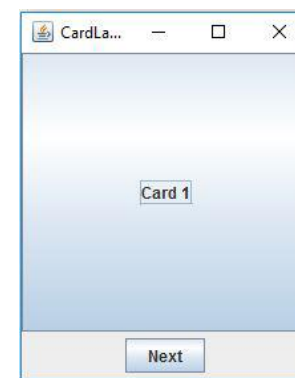
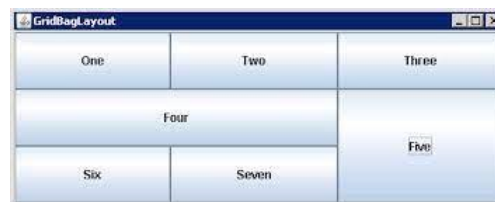
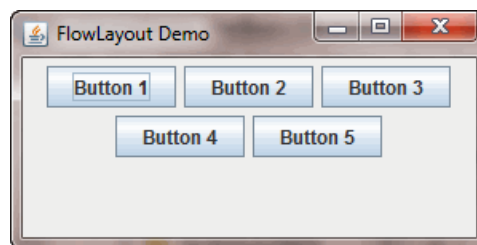
1. Giới thiệu Java GUI
2. Container
3. Nguyên tắc xây dựng GUI
4. **Layout**
5. Swing component

# Khái niệm Layout Manager

❖ Là việc bố trí các thành phần trên container

❖ Các loại layout khác nhau:

- Flow Layout
- Border Layout
- Grid Layout
- GridBag Layout
- Null Layout
- CardLayout
- SpringLayout



❖ Trình quản lý layout được thiết lập bằng cách gọi phương thức *setLayout( )*

# FlowLayout

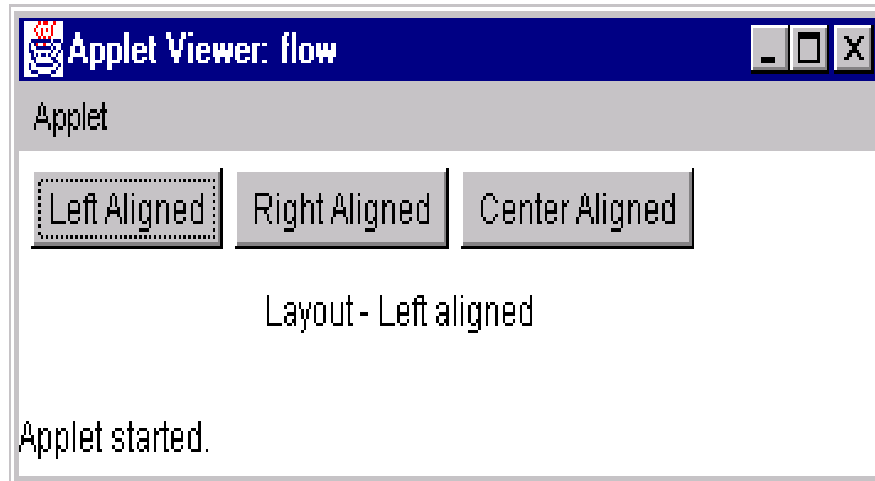
- ❖ Là trình quản lý layout mặc định cho các applet và các panel
- ❖ Với FlowLayout các thành phần sẽ được sắp xếp từ góc trái trên đến góc phải dưới của màn hình
- ❖ Constructor:

*FlowLayout layout = new FlowLayout();*

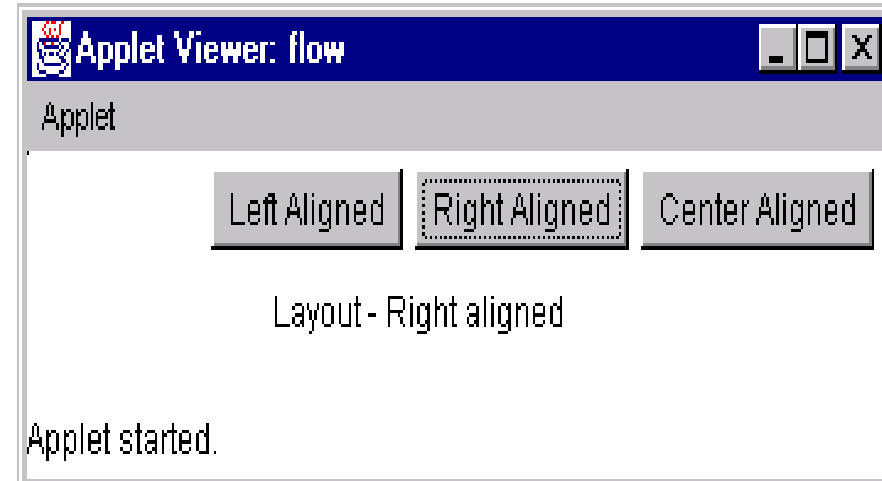
*FlowLayout layout = new FlowLayout(FlowLayout.RIGHT);*

*// Canh lề bên phải*

# FlowLayout



Flow Layout – Canh trái



Flow Layout – Canh phải



# BorderLayout

- ❖ Là layout mặc định cho Window, Frame và Dialog
- ❖ Trình quản lý này có thể sắp xếp đến 5 thành phần trong container: NORTH, EAST, SOUTH, WEST và CENTER
- ❖ Ví dụ: Để thêm một thành phần vào vùng North của container

```
Button b1= new Button("North Button");
```

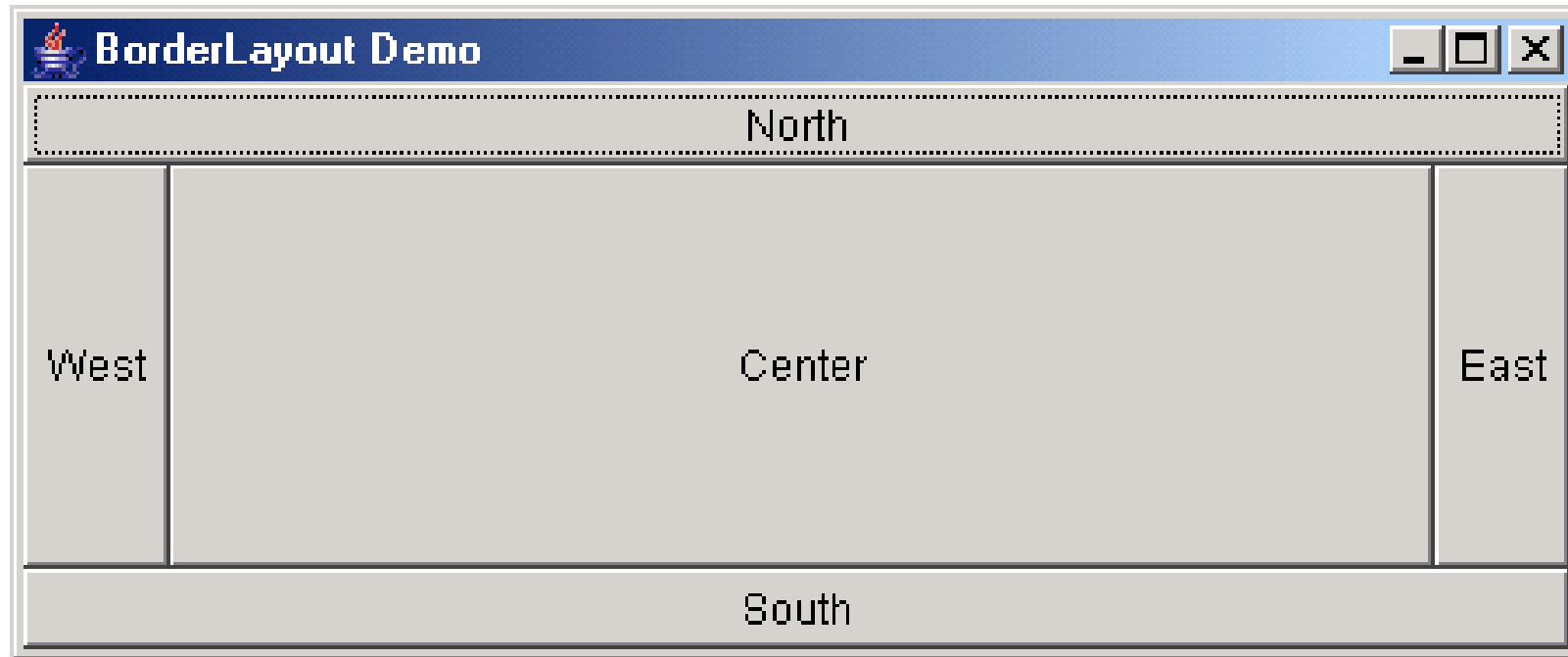
```
setLayout(new BorderLayout( ));
```

```
add(b1, BorderLayout.NORTH);
```

# BorderLayout



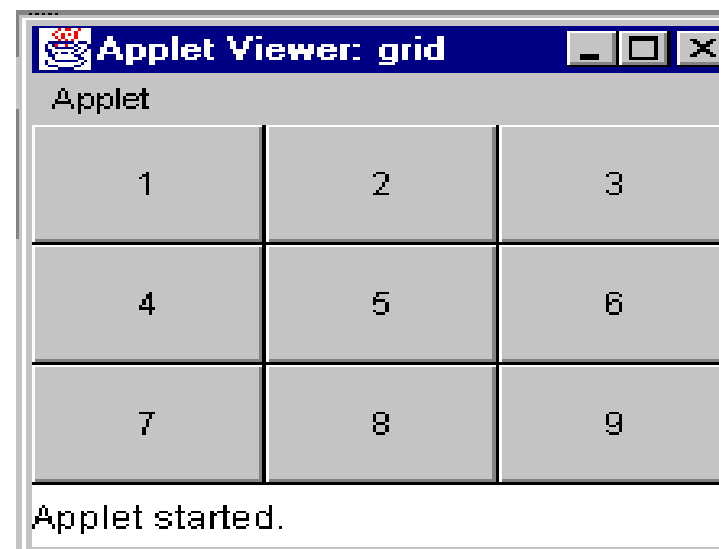
**UIT**  
TRƯỜNG ĐẠI HỌC  
CÔNG NGHỆ THÔNG TIN



# GridLayout

- ❖ Hỗ trợ việc chia container thành một lưới
- ❖ Các thành phần được bố trí trong các dòng và cột
- ❖ Một ô lưới nên chứa ít nhất một thành phần
- ❖ Kiểu layout này được sử dụng khi tất cả các thành phần có cùng kích thước

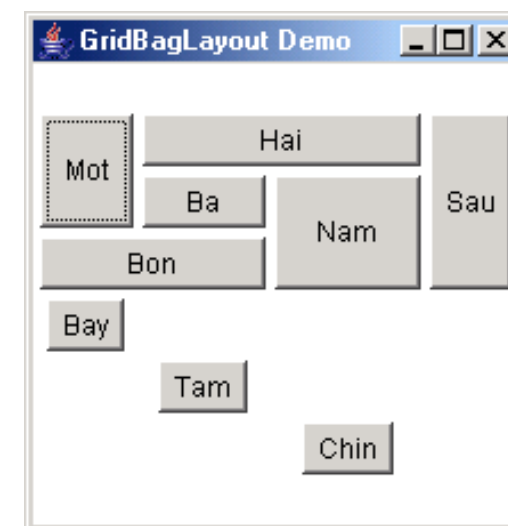
```
GridLayout layout = new  
    GridLayout(3, 3);  
containerObj.setLayout(layout);
```



# GridBagLayout

- ❖ Bố trí các thành phần một cách chính xác
- ❖ Các thành phần không cần có cùng kích thước
- ❖ Các thành phần được sắp xếp trong một lưới chứa các dòng và các cột
- ❖ Thứ tự đặt các thành phần không tuân theo hướng từ *trái-sang-phải* và *trên-xuống-dưới*
- ❖ Constructor

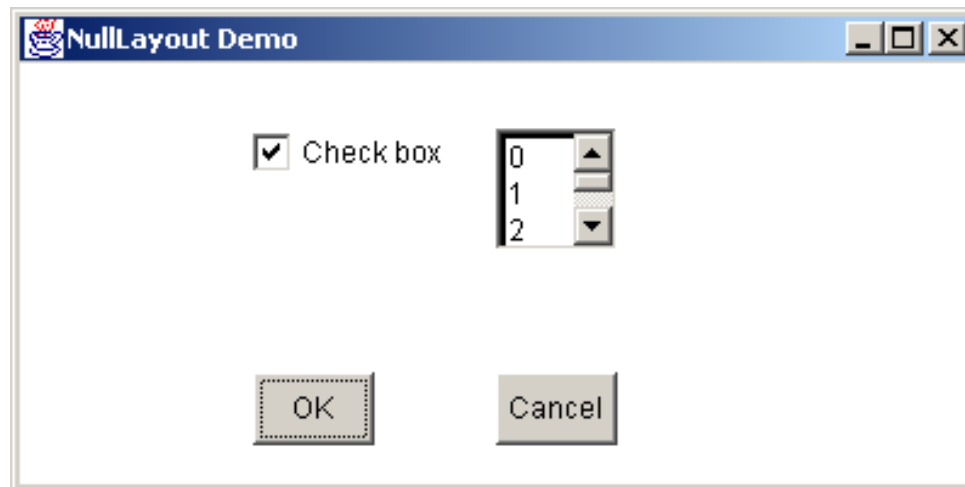
*GridBagLayout gb = new GridBagLayout( );*



# NullLayout

- ❖ Tự do trong việc định vị trí và kích thước của các components

```
Frame fr = new Frame("NullLayout Demo");  
fr.setLayout(null);
```



# NỘI DUNG

1. Giới thiệu Java GUI
2. Container
3. Nguyên tắc xây dựng GUI
4. Layout
5. **Swing component**

# Swing Component

`javax.swing`

Class JComponent

[java.lang.Object](#)

└ [java.awt.Component](#)

└ [java.awt.Container](#)

└ `javax.swing.JComponent`

**All Implemented Interfaces:**

[ImageObserver](#), [MenuContainer](#), [Serializable](#)

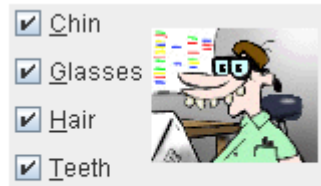
**Direct Known Subclasses:**

[AbstractButton](#), [BasicInternalFrameTitlePane](#), [Box](#), [Box.Filler](#), [JColorChooser](#), [JComboBox](#), [JFileChooser](#),  
[JInternalFrame](#), [JInternalFrame.JDesktopIcon](#), [JLabel](#), [JLayeredPane](#), [JList](#), [JMenuBar](#), [JOptionPane](#), [JPanel](#),  
[JPopupMenu](#), [JProgressBar](#), [JRootPane](#), [JScrollBar](#), [JScrollPane](#), [JSeparator](#), [JSlider](#), [JSpinner](#), [JSplitPane](#),  
[JTabbedPane](#), [JTable](#), [JTableHeader](#), [JTextComponent](#), [JToolBar](#), [JToolTip](#), [JTree](#), [JViewport](#)

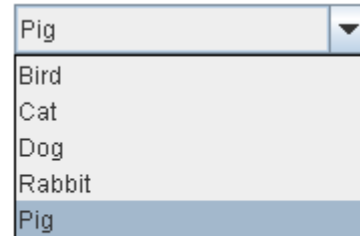
# Swing Component



JButton



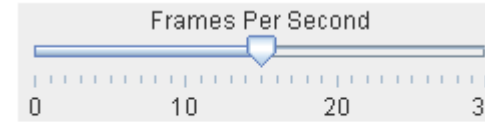
JCheckBox



JComboBox



JList



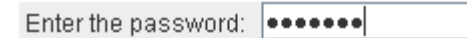
JSlider



JTextField



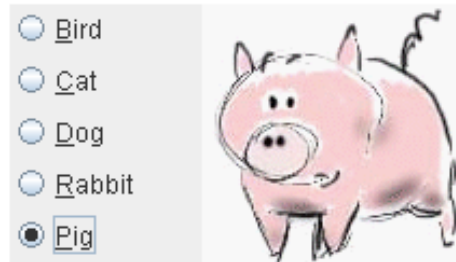
JTree



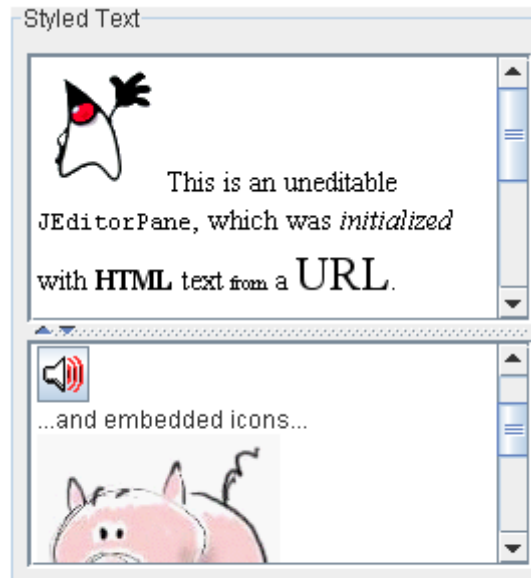
JPasswordField



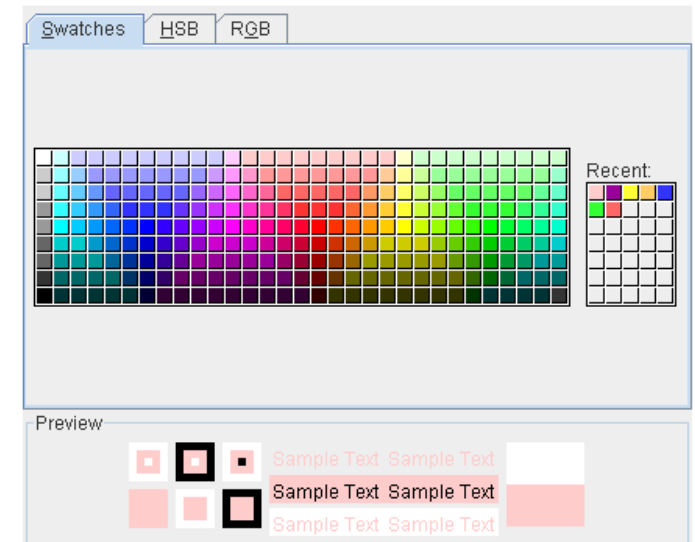
JSpinner



JRadioButton



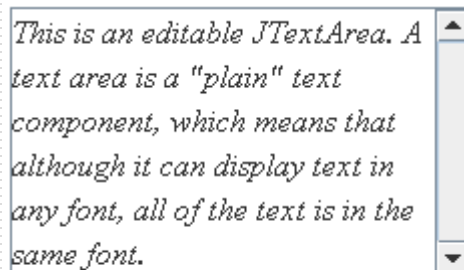
JEditorPane and JTextPane



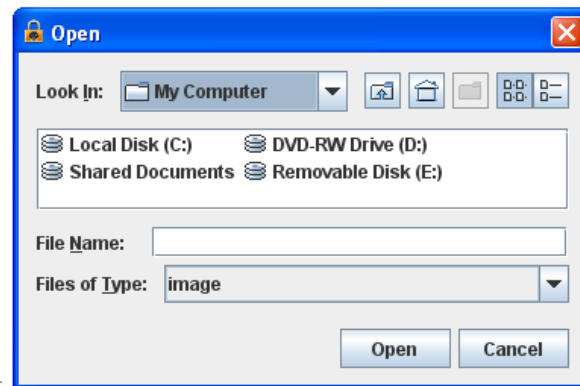
JColorChooser



JMenu



JTextArea



JFileChooser



# Swing Component

❖ Phương thức dùng để gắn components:

*objectName.add(... );*

❖ Phương thức dùng để lấy thuộc tính:

*objectName.getxxx( );*

❖ Phương thức dùng để gán thuộc tính:

*objectName.setxxx( );*

# Tóm tắt bài học

- ❖ AWT, Swing hỗ trợ xây dựng ứng GUI trong Java
- ❖ Swing Components thường có tên bắt đầu với 'J'
- ❖ Swing Components có nhiều mức khác nhau, các mức này qui định các Component trong 1 ứng dụng GUI kết hợp với nhau (*Object -> Component -> Container -> Window -> Frame -> JFrame*)
- ❖ Container là thành phần mà có thể chứa các thành phần khác, có thể vẽ và tô màu (*Frame/JFrame, Panel/JPanel, Dialog/JDialog, ScrollPane/JScrollPane*)

# Tóm tắt bài học

## ❖ Các bước xây dựng GUI:

- Lựa chọn 1 container: Frame/JFrame, Window/JWindow, Dialog/JDialog,
- Tạo các điều khiển: (buttons, text areas..)
- Đưa các điều khiển vào vùng chứa
- Sắp xếp các điều khiển(layout)
- Thêm các xử lý sự kiện

## ❖ Layout là việc bố trí các thành phần trên container, các loại layout khác nhau: Flow Layout, Border Layout, Grid Layout, GridBag Layout, Null Layout, CardLayout, SpringLayout

## ❖ Có nhiều component trong cửa sổ như: label, textfield, button, checkbox, combobox, list, tree, table, menu...

## Bài tập

1. Viết chương trình hiển thị hộp thoại cho phép người dùng nhập vào tên và cửa sổ chính xuất ra dòng "Xin chào" + tên
2. Viết chương trình xử lý khi ấn phím cửa sổ hiện ra phím vừa ấn là phím gì?
3. Viết chương trình gồm 2 text area và một button. Khi ấn button cho phép copy text được chọn từ text area 1 sang text area 2
4. Viết chương trình gồm 3 nút nằm trên 1 hàng và nằm giữa cửa sổ: trái, phải, giữa. Khi ấn nút bất kì thì cả 3 nút được canh lề tương ứng với nút được chọn

## Bài tập

5. Viết chương trình gồm 5 nút (trái, phải, trên, dưới, giữa) nằm năm phía của cửa sổ (BorderLayout). Khi người dùng ấn một nút thì nút đó ẩn và các nút còn lại được hiển thị.
6. Viết chương trình demo với Jlist chứa các thể loại sách (văn học, khoa học, lịch sử, công nghệ, thể thao, văn hóa). Khi người dùng ấn "Summit" thì cho biết những thể loại mà người dùng đã chọn.
7. Viết chương trình dùng hộp thoại màu đổi màu của cửa sổ
8. Viết chương trình dùng hộp thoại file cho chọn file hình ảnh và đổi icon của một JLabel

## Bài tập

9. Viết chương trình gồm một textfield và 4 radiobutton (thường, bôi đậm, in nghiêng, bôi đậm và in nghiêng). Người dùng chọn radiobutton nào thì style textfield thay đổi tương ứng
10. Cách vẽ các hình/fill chữ nhật, oval, đường cong, đa giác
11. Vẽ 20 ngôi sao năm cánh, mỗi ngôi sao có màu ngẫu nhiên xoay thành vòng tròn

