

CHUYÊN ĐỀ JAVA

SWING

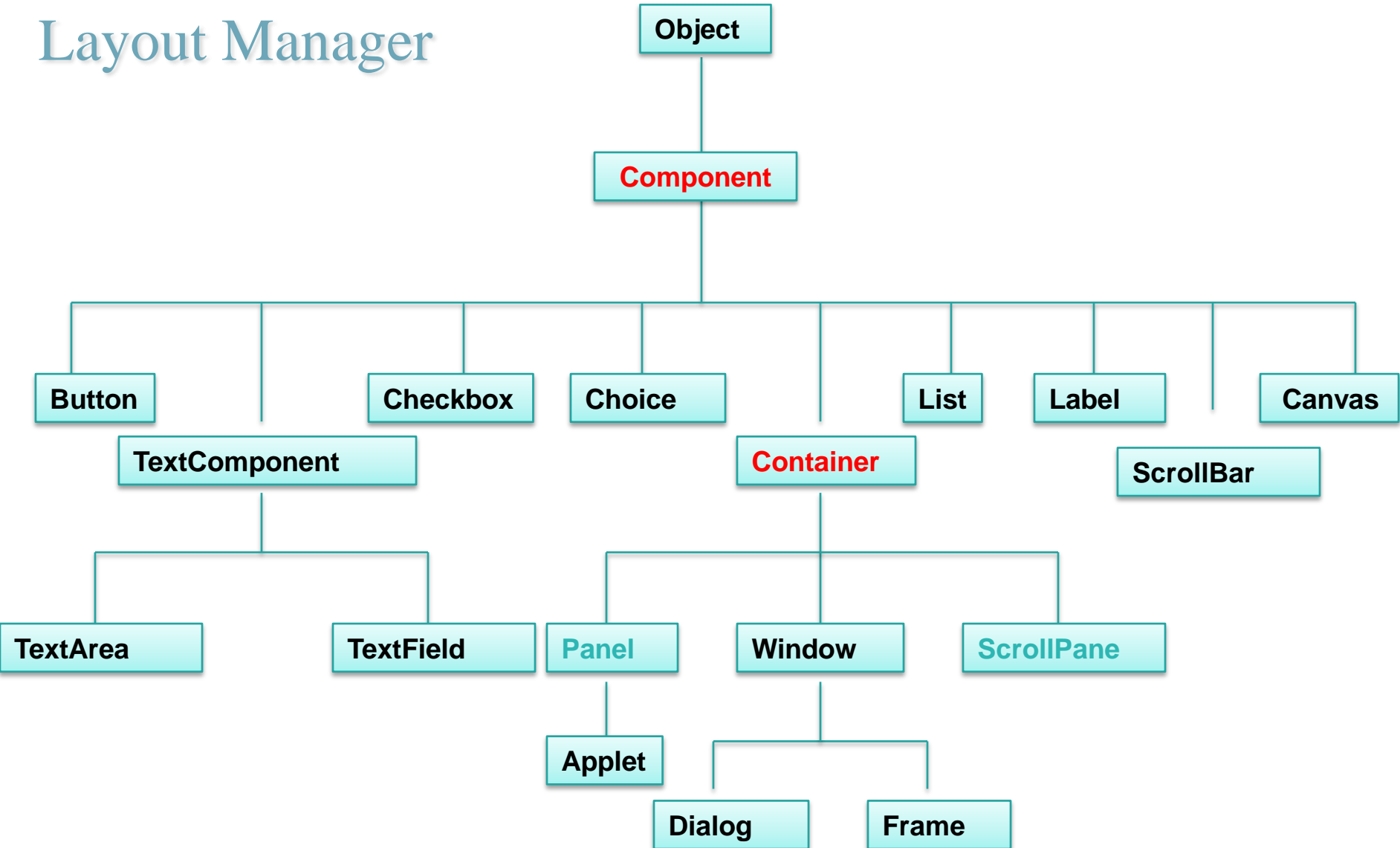
Layout Manager

Nguyễn Hoàng Anh – nhanh@fit.hcmuns.edu.vn

Nội dung

- **Flow Layout**
- **Border Layout**
- **Card Layout**
- **Grid Layout**
- **GridBag Layout**
- **Box Layout**
- **Group Layout**

Layout Manager



Layout Manager

- Một **Container** là một **Component** có thể chứa **các Component khác**.
- Mỗi **Container** có một đối tượng **Layout Manager**
- Layout Manager là một đối tượng quyết định **cách sắp xếp vị trí của** các **Component** bên trong một **Container**.
- Các Layout Manager “**implements**” từ **interface LayoutManager** hoặc **LayoutManager2**.

Layout Manager

- **Flow Layout**
- **Border Layout**
- **Card Layout**
- **Grid Layout**
- **GridBag Layout**
- **Box Layout**
- **Group Layout**

Layout Manager

- Mỗi **Container** có một đối tượng **Layout Manager** mặc định, nhưng hoàn toàn có thể gán cho **Container** một đối tượng **Layout Manger** theo ý muốn.
- Mỗi loại **Layout Manager** có các nguyên tắc riêng cho việc bố trí các **Component** bên trong một **Container**.
- Một **Layout Manager** chịu trách nhiệm **bố trí** các **Component** được thêm vào **Container** và khi **Container** thay đổi kích thước
- Sử dụng phương thức **setLayout (LayoutManager mng)** của **Container** để thay đổi cách bố trí các **Component** bên trong nó.

Flow Layout

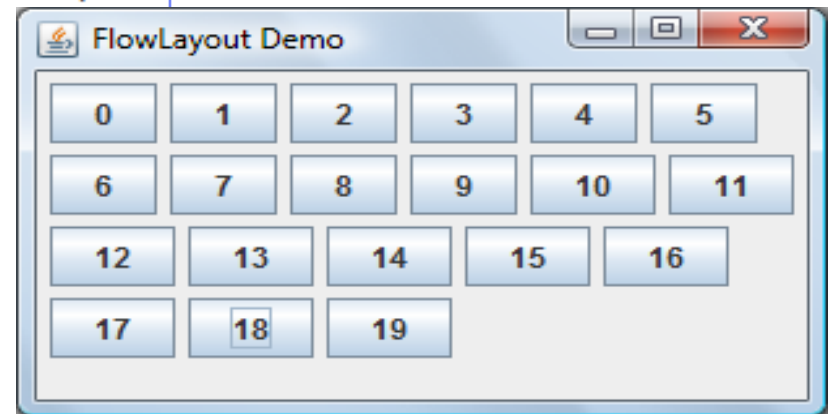
- Flow Layout bố trí các Component trong Container theo dòng, từ trái sang phải theo thứ tự thêm vào.
- Tạo dòng mới khi kích thước dòng còn lại không đủ chứa Component thêm vào.
- Flow Layout bố trí vị trí các Component phụ thuộc vào kích thước của Container.
- Mỗi dòng của các Component được window mặc định canh giữa theo chiều ngang . Có thể điều chỉnh canh trái hoặc phải

Flow Layout

- `FlowLayout ()`
 - `Align: FlowLayout.CENTER`
 - `vgap: 5px, hgap: 5px`
- `FlowLayout (int align)`
 - `align:`
 - `FlowLayout.CENTER` : canh giữa
 - `FlowLayout.LEFT;` : Canh trái
 - `FlowLayout.RIGHT;` : Canh phải
- `FlowLayout(int align, int vgap, int hgap)`

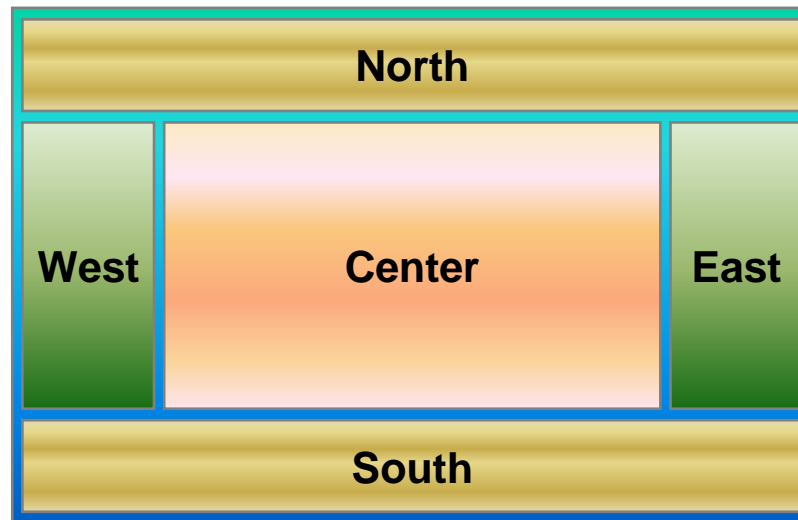
Flow Layout

```
public class FlowLayoutDemo extends javax.swing.JFrame {  
    /** Creates new form FlowLayoutDemo */  
    private JButton [] buttons;  
    public FlowLayoutDemo() {  
        initComponents();  
        this.buttons=new JButton[20];  
        for (int i=0; i<20; i++){  
            this.buttons[i]=new JButton(String.valueOf(i));  
            this.add(this.buttons[i]);  
        }  
        FlowLayout layout=new FlowLayout(FlowLayout.LEFT);  
        this.setLayout(layout);  
    }  
}
```



Border Layout

- **Border Layout** bố trí các **Component** bên trong **Container** theo 5 vùng: "**North**", "**South**", "**East**", "**West**", "**Center**".



Border Layout

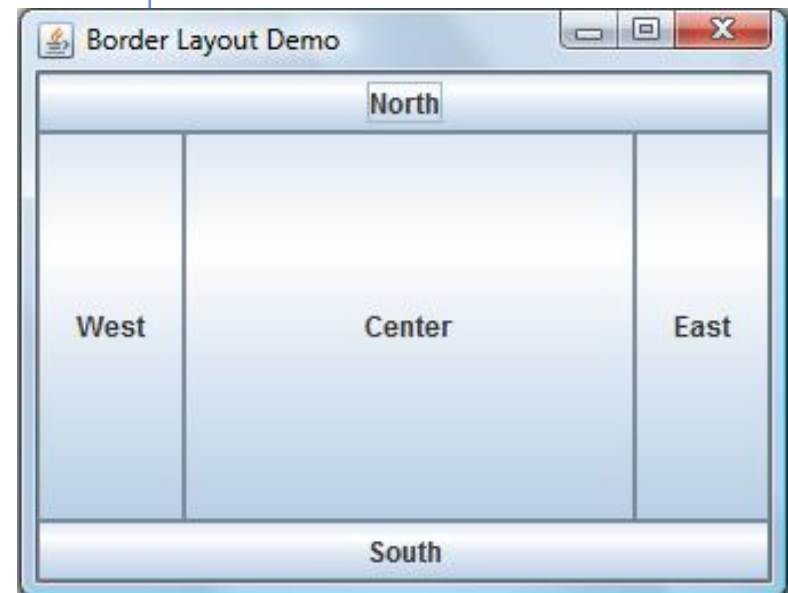
- Mỗi vùng chứa một **Component** (Cũng có thể là JPanel).
- Khi Container thay đổi kích thước các vùng bên trong sẽ thay đổi kích thước theo.
- **BorderLayout.NORTH** , **BorderLayout.EAST**
- **BorderLayout.WEST** , **BorderLayout.CENTER**
- **BorderLayout.SOUTH**
- **BorderLayout()**
- **BorderLayout(int hgap, int vgap)**

Border Layout

```
public class BorderLayoutDemo extends javax.swing.JFrame {
```

```
    /** Creates new form BorderLayoutDemo */
```

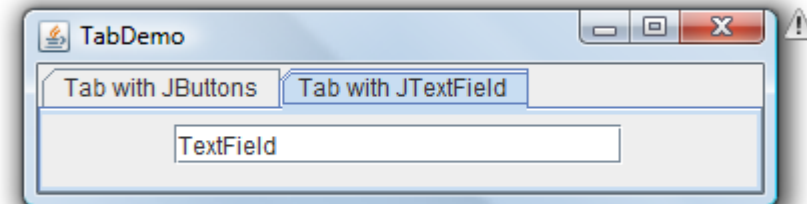
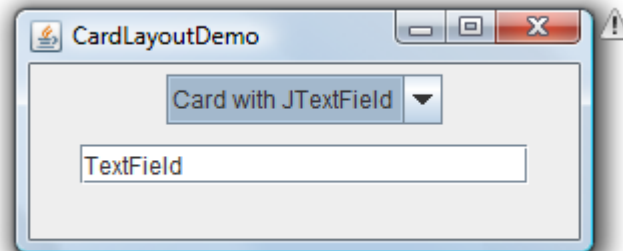
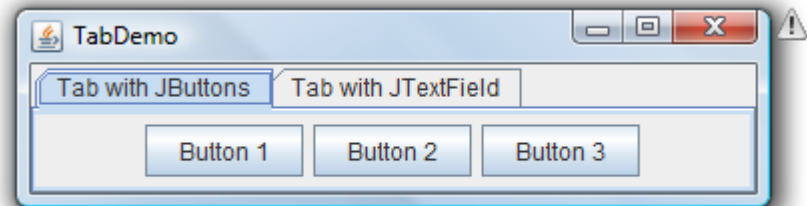
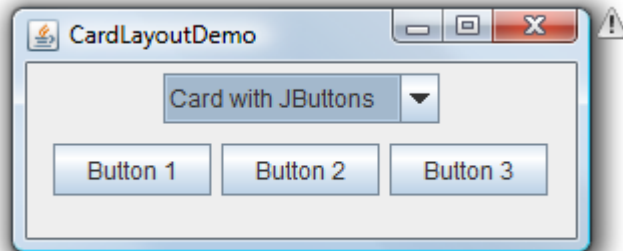
```
    public BorderLayoutDemo() {  
        initComponents();  
        BorderLayout layout = new BorderLayout();  
        this.setLayout(layout);  
        JButton north = new JButton("North");  
        JButton south = new JButton("South");  
        JButton west = new JButton("West");  
        JButton east = new JButton("East");  
        JButton center = new JButton("Center");  
        this.add(north, BorderLayout.NORTH);  
        this.add(east, BorderLayout.EAST);  
        this.add(west, BorderLayout.WEST);  
        this.add(center, BorderLayout.CENTER);  
        this.add(south, BorderLayout.SOUTH);  
    }
```



Card Layout

- **Card Layout** quản lý nhiều **Card** cùng một không gian hiển thị
- **Card Layout** giúp quản lý hai hay nhiều Component (thường là **Jpanel**) để chia sẻ cùng một không gian hiển thị.
- Chỉ duy nhất **top Card** được hiển thị.
- Mỗi “**Card**” có thể sử dụng **Layout Manager** riêng.
- **Card** nào cũng có thể là **top Card**
- Có thể sử dụng **JTabbedPane** để thay cho **Card Layout**

Card Layout



Grid Layout

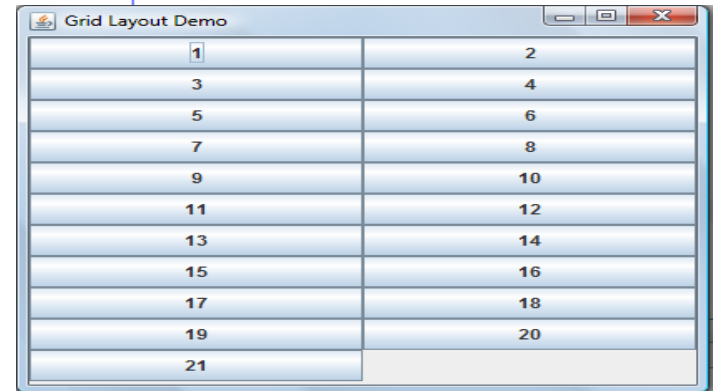
- **Grid Layout** bố trí các **Component** của **Container** vào bên trong một **Grid** với các **Row** và **Column**.
- Mỗi **Component** được bố trí trong 1 **Cell** của lưới.
- Tất cả các **Cell** có cùng kích thước bên trong **Grid**.
- Các **Component** được thêm vào **Container** theo thứ tự từ **trái sang phải**, từ **trên xuống dưới** (mặc định).
- Kích thước của mỗi **Cell** được xác định bởi kích thước của **Container**.

Grid Layout

- `GridLayout()`
 - **rows: 1**
 - **1 Component / 1 Column**
- `GridLayout(int rows, int cols)`
- `GridLayout(int rows, int cols, int hgap, int vgap)`

Grid Layout

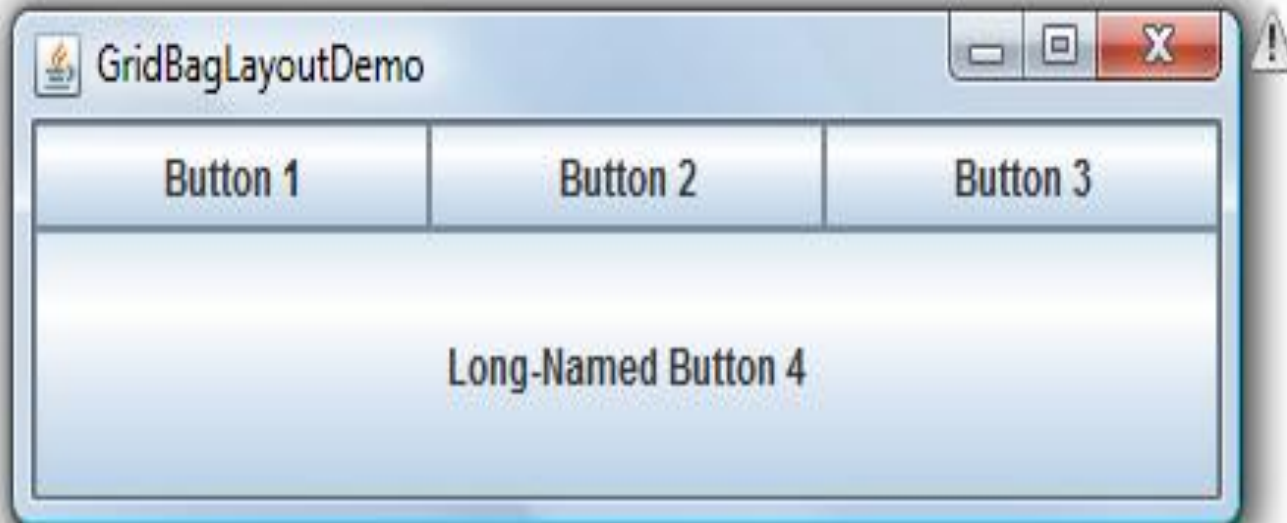
```
public class GridLayoutDemo extends javax.swing.JFrame {  
  
    /** Creates new form GridLayoutDemo */  
    private JButton [] buttons;  
    public GridLayoutDemo() {  
        initComponents();  
        buttons=new JButton[21];  
        for (int i=0; i<buttons.length; i++){  
            this.buttons[i]=new JButton(String.valueOf(i+1));  
            this.add(this.buttons[i]);  
        }  
        GridLayout layout=new GridLayout(11, 2);  
        this.setLayout(layout);  
    }  
}
```



GridBag Layout

- GridBag Layout bố trí các Component trong một Grid với các Row và Column.
- Mỗi Component bên trong Grid được RowSpan và ColumnSpan (giống table HTML)
- Width và Height của các Row/Column có thể khác nhau.
- GridBag Layout là một Layout Manager rất linh động cho việc bố trí các Component bên trong Container theo dạng Grid.
- GridBag Layout là một trong các Layout Manager **thường sử dụng nhất** mà Java Platform cung cấp.

GridBag Layout

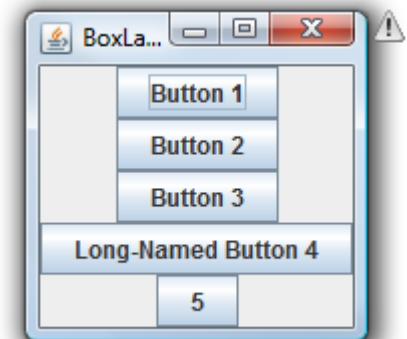


Box Layout

- Box Layout bố trí các Component bên trong Container theo 1 dòng theo trục X, hoặc là trục Y.
- **BoxLayout(Container container, int align)**
 - container: chứa các Component
 - axis:
 - **BoxLayout.X_AXIS** : Trục X
 - **BoxLayout.Y_AXIS** : Trục Y

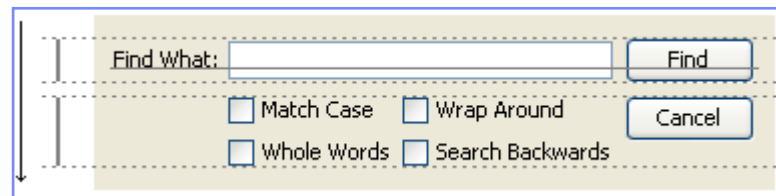
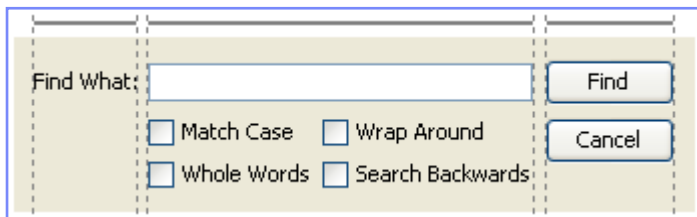
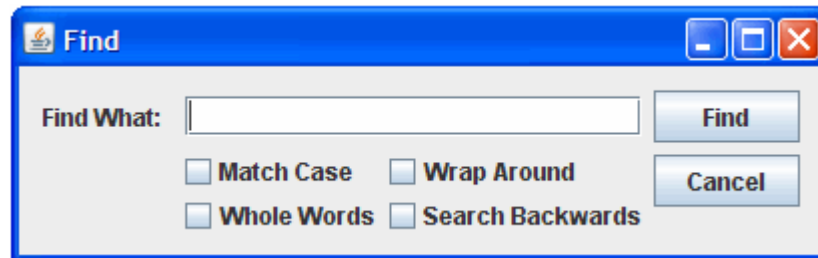
Box Layout

```
public class BoxLayoutDemo extends javax.swing.JFrame {  
    /** Creates new form BoxLayoutDemo */  
    private JButton[] buttons;  
    public BoxLayoutDemo() {  
        initComponents();  
    }  
    private static void addAButton(String text, Container container) {  
        JButton button = new JButton(text);  
        button.setAlignmentX(Component.CENTER_ALIGNMENT);  
        container.add(button);  
    }  
    public static void addComponentsToPane(Container pane) {  
        pane.setLayout(new BoxLayout(pane, BoxLayout.Y_AXIS));  
        addAButton("Button 1", pane);  
        addAButton("Button 2", pane);  
        addAButton("Button 3", pane);  
        addAButton("Long-Named Button 4", pane);  
        addAButton("5", pane);  
    }  
}
```



Group Layout

- Group Layout bố trí các Component bên trong Container theo chiều ngang và chiều dọc.
- Sự bố trí được thực hiện theo mỗi chiều riêng lẻ.



HỎI VÀ ĐÁP

DEMO

Tham khảo

- <http://java.sun.com/docs/books/tutorial/uiswing/layout/>