

# Swing Layout Manager

Nguyễn Văn Khiết

cuu duong than cong . com

# Nội dung

- Giới thiệu
- Các layout manager

cuu duong than cong . com

cuu duong than cong . com

# Giới thiệu

- Layout là cách thức sắp xếp các component trên một container, làm cho các component xuất hiện với vị trí như mong muốn.
- Khi kích thước của container thay đổi, thì kích thước và vị trí tương đối của các component trong container cũng thay đổi theo.

# Các layout manager

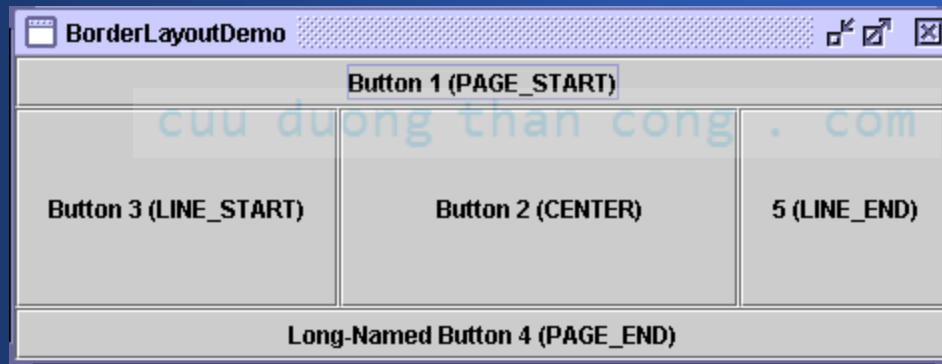
- BorderLayout
- BoxLayout
- CardLayout
- FlowLayout
- GridLayout
- GridBagLayout
- SpringLayout
- GroupLayout
- Null Layout

# FlowLayout

- Flow layout sẽ đặt các component theo thứ tự
  - Từ trái sang phải
  - Khi hết chỗ thì đưa các component vào phía bên dưới và lại bắt đầu từ trái sang phải.
- Kích thước của các component khi đặt vào container được xác định bằng preferred size.

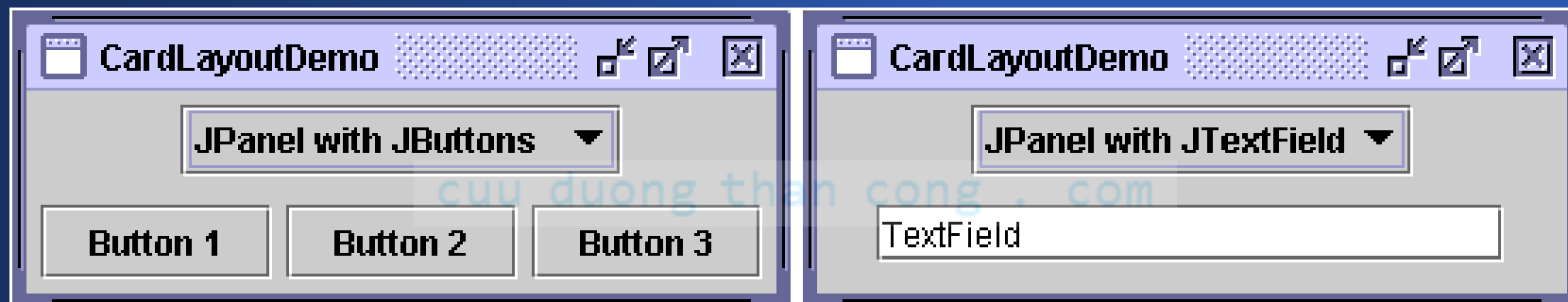
# BorderLayout

- BorderLayout là layout manager default của các đối tượng JPanel.
- Border layout sẽ đặt các component vào trong các vùng sau của container : top, bottom, left, right, center.



# CardLayout

- CardLayout là một dạng bố cục cho phép ta có thể đặt nhiều component khác nhau trên cùng một vùng không gian tại các thời điểm khác nhau.



# CardLayout

- Các hàm cung cấp bởi CardLayout
  - void first(Container);
  - void next(Container);
  - void previous(Container);
  - void last(Container);
  - void show(Container, String);



# GridLayout

- GridLayout trình bày các component lên container dưới dạng lưới
- Container sẽ được chia thành lưới các ô dưới dạng nhiều dòng nhiều cột.
  - Các ô có kích thước bằng nhau
  - Mỗi component sẽ được hiển thị trên 1 ô trong lưới

# GridLayout

- Cách khởi tạo GridLayout
  - public GridLayout(int *rows*, int *columns*)
  - public GridLayout(int *rows*, int *columns*,  
int *horizontalGap*, int *verticalGap*)

cuu duong than cong . com

# GridBagLayout

- GridBagLayout đặt các component trong một lưới các dòng và cột,
  - cho phép các component được đặt trên nhiều ô thuộc nhiều dòng và cột.
  - các dòng và các cột trong lưới các ô có thể có kích thước khác nhau
- Khi thay đổi kích thước của container, GridBagLayout sẽ thay đổi kích thước của từng ô và thay đổi cách thức đặt các component trên container.

# GridBagLayout

- Chương trình phải xác định kích thước và các đặc tính của component bằng cách đưa ra các ràng buộc cho mỗi component. Các ràng buộc này sẽ có ảnh hưởng đến cách thức hiển thị của component trong container.
- Để xác định các ràng buộc, ta phải tạo ra đối tượng GridBagConstraints

# GridBagLayout

- Các thuộc tính của GridBagConstraints
  - **gridx, gridy**
  - **gridwidth, gridheight**
  - **fill**
  - **ipadx, ipady**
  - **insets**
  - **anchor**
  - **weightx, weighty**

# BoxLayout

- BoxLayout là layout mới trong swing
- BoxLayout sẽ đặt lần lượt các component từ trái sang phải trên 1 dòng hoặc từ trên xuống dưới trên 1 cột.

# BoxLayout

- Khi đặt các component vào container, BoxLayout dựa vào các thông tin minimum size, prefer size và maximum size.
- Khi BoxLayout đặt các component từ trên xuống dưới, nó sẽ cố đặt các component với kích thước chiều cao xác định trong preferred size.
- Vùng không gian trống sẽ đặt ở phía bên dưới của container

# BoxLayout

- X Alignment.
  - Giá trị của các X Alignment này sẽ từ 0.0 tới 1.0.
    - 0.0 ứng với Component.LEFT\_ALIGNMENT
    - 0.5 ứng với Component.CENTER\_ALIGNMENT
    - 1.0 ứng với Component.RIGHT\_ALIGNMENT.

cuu duong than cong . com



# BoxLayout

- Các Invisible Component

- Rigid area

- HorizontalGlue

- VerticalGlue

- Filler

-

# SpringLayout

- SpringLayout hoạt động dựa trên việc định nghĩa mối liên hệ giữa cạnh của các component. Ví dụ, ta có thể định nghĩa cạnh bên trái của một component cách đúng 5 pixel so với cạnh phải của một component khác.
- SpringLayout sẽ tạo kích thước của các component (chiều cao và chiều rộng) trong khoảng `minimum size` và `maximum size`
- Ràng buộc Spring áp đặt cho một component thông qua đối tượng `SpringLayout.Constraints`

# GroupLayout

- Là một Layout Manager phát triển cho các GUI builder.
- GroupLayout làm việc với hai chiều : horizontal và vertical riêng lẻ.
- Được đưa vào kể từ Java SE 6.

# GroupLayout

- Tổ chức của GroupLayout
  - Sequential : các component được sắp xếp tuần tự theo 1 chiều
  - Parallel : Các component được sắp xếp trên cùng 1 vùng không gian (của chiều đang làm việc)
    - Vertical : baseline, top, bottom-
    - Horizontal : left, right, center

# GroupLayout

- Một group có thể nằm bên trong một group khác.
- Gap : vùng không gian không nhìn thấy giữa các component.
- Dùng phương thức `addComponent` để đưa các component vào container.