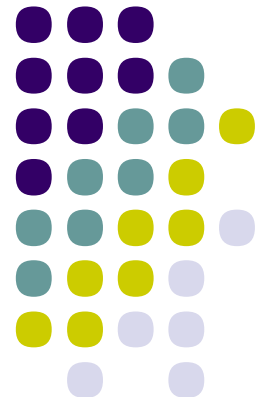


# GUI Layout Manager

---

Nguyễn Thị Tú Mi  
Email: [nttmi@hcmuaf.edu.vn](mailto:nttmi@hcmuaf.edu.vn)





# Quản lý bố cục

- Cung cấp để sắp xếp các thành phần GUI
- Cung cấp những khả năng cách trình bày cơ bản
- Xử lý các chi tiết bố cục
- Lập trình viên có thể tập trung vào “vẽ ngoài” cơ bản
- Giao tiếp **LayoutManager**

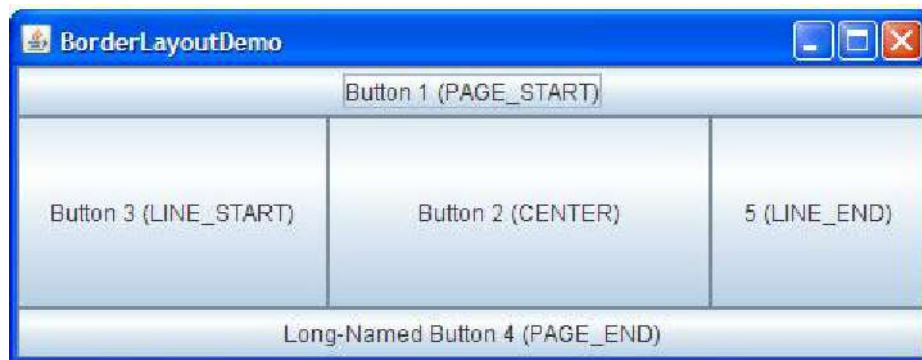


# BorderLayout

- **BorderLayout:**

- Mặc định cho khung nội dung của JFrames (và các Window khác) và JApplets.
- Sắp xếp các thành phần vào 5 vùng:
  - NORTH (đỉnh container)
  - SOUTH (đáy container)
  - EAST (bên trái container)
  - WEST (bên phải container)
  - CENTER (ở giữa container)

# Ví dụ sử dụng BorderLayout





# Ví dụ sử dụng BorderLayout

```
public class BorderLayoutDemo {
    public static boolean RIGHT_TO_LEFT = false;
    public static void addComponentsToPane(Container pane) {
        if (!(pane.getLayout() instanceof BorderLayout)) {
            pane.add(new JLabel("Container doesn't use BorderLayout!"));
            return;
        }
        if (RIGHT_TO_LEFT) {
            pane.setComponentOrientation(java.awt.ComponentOrientation.RIGHT_TO_LEFT);
        }
        JButton button = new JButton("Button 1 (PAGE_START)");
        pane.add(button, BorderLayout.PAGE_START);
        button = new JButton("Button 2 (CENTER)");
        button.setPreferredSize(new Dimension(200, 100));
        pane.add(button, BorderLayout.CENTER);
        button = new JButton("Button 3 (LINE_START)");
        pane.add(button, BorderLayout.LINE_START);
        button = new JButton("Long-Named Button 4 (PAGE_END)");
        pane.add(button, BorderLayout.PAGE_END);
        button = new JButton("5 (LINE_END)");
        pane.add(button, BorderLayout.LINE_END);
    }
}
```



# Ví dụ sử dụng BorderLayout

```
private static void createAndShowGUI() {
    JFrame frame = new JFrame("BorderLayoutDemo");
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    addComponentsToPane(frame.getContentPane());
    frame.pack();
    frame.setVisible(true);
}

public static void main(String[] args) {
    try {
        UIManager.setLookAndFeel("javax.swing.plaf.metal.MetalLookAndFeel");
    } catch (UnsupportedLookAndFeelException ex) {
        ex.printStackTrace();
    } catch (IllegalAccessException ex) {
        ex.printStackTrace();
    } catch (InstantiationException ex) {
        ex.printStackTrace();
    } catch (ClassNotFoundException ex) {
        ex.printStackTrace();
    }
    UIManager.put("swing.boldMetal", Boolean.FALSE);
    javax.swing.SwingUtilities.invokeLater(new Runnable() {
        public void run() {
            createAndShowGUI();
        }
    });
}
```

# BoxLayout



- BoxLayout:
  - Sắp xếp các thành phần theo chiều dọc và chiều ngang

# Ví dụ sử dụng BoxLayout







# Ví dụ sử dụng BorderLayout

```
public class BorderLayoutDemo {  
    public static void addComponentsToPane(Container pane) {  
        pane.setLayout(new BorderLayout(pane, BorderLayout.Y_AXIS));  
        addAButton("Button 1", pane);  
        addAButton("Button 2", pane);  
        addAButton("Button 3", pane);  
        addAButton("Long-Named Button 4", pane);  
        addAButton("5", pane);  
    }  
    private static void addAButton(String text, Container container) {  
        JButton button = new JButton(text);  
        button.setAlignmentX(Component.CENTER_ALIGNMENT);  
        container.add(button);  
    }  
    private static void createAndShowGUI() {  
        JFrame frame = new JFrame("BoxLayoutDemo");  
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
        addComponentsToPane(frame.getContentPane());  
        frame.pack();  
        frame.setVisible(true);  
    }  
  
    public static void main(String[] args) {  
        javax.swing.SwingUtilities.invokeLater(new Runnable() {  
            public void run() {  
                createAndShowGUI();  
            }  
        });  
    }  
}
```

# CardLayout



- Dùng để hiển thị các Component xếp chồng lên nhau

# Ví dụ sử dụng CardLayout





# Ví dụ sử dụng CardLayout

```
public class CardLayoutDemo implements ItemListener {
    JPanel cards; //a panel that uses CardLayout
    final static String BUTTONPANEL = "Card with JButtons";
    final static String TEXTPANEL = "Card with JTextField";
    public void addComponentToPane(Container pane) {
        JPanel comboBoxPane = new JPanel(); //use FlowLayout
        String comboBoxItems[] = { BUTTONPANEL, TEXTPANEL };
        JComboBox cb = new JComboBox(comboBoxItems);
        cb.setEditable(false);
        cb.addItemListener(this);
        comboBoxPane.add(cb);
        JPanel card1 = new JPanel();
        card1.add(new JButton("Button 1"));
        card1.add(new JButton("Button 2"));
        card1.add(new JButton("Button 3"));
        JPanel card2 = new JPanel();
        card2.add(new JTextField("TextField", 20));
        cards = new JPanel(new CardLayout());
        cards.add(card1, BUTTONPANEL);
        cards.add(card2, TEXTPANEL);
        pane.add(comboBoxPane, BorderLayout.PAGE_START);
        pane.add(cards, BorderLayout.CENTER);
    }
    public void itemStateChanged(ItemEvent evt) {
        CardLayout cl = (CardLayout)(cards.getLayout());
        cl.show(cards, (String)evt.getItem());
    }
}
```



# Ví dụ sử dụng CardLayout

```
private static void createAndShowGUI() {
    JFrame frame = new JFrame("CardLayoutDemo");
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    CardLayoutDemo demo = new CardLayoutDemo();
    demo.addComponentToPane(frame.getContentPane());
    frame.pack();
    frame.setVisible(true);
}

public static void main(String[] args) {
    try {
        UIManager.setLookAndFeel("javax.swing.plaf.metal.MetalLookAndFeel");
    } catch (UnsupportedLookAndFeelException ex) {
        ex.printStackTrace();
    } catch (IllegalAccessException ex) {
        ex.printStackTrace();
    } catch (InstantiationException ex) {
        ex.printStackTrace();
    } catch (ClassNotFoundException ex) {
        ex.printStackTrace();
    }
    UIManager.put("swing.boldMetal", Boolean.FALSE);
    javax.swing.SwingUtilities.invokeLater(new Runnable() {
        public void run() {
            createAndShowGUI();
        }
    });
}
```



# FlowLayout

- **FlowLayout:**

- Bộ quản lý bố cục cơ bản nhất
- Mặc định cho `Java.awt.Applet`, `Java.awt.Panel` và `javax.swing.JPanel`.
- Đặt các thành phần theo tuần tự (trái qua phải) theo thứ tự khi chúng được thêm.
- Cũng có thể chỉ rõ thứ tự của các thành phần bởi việc sử dụng phương thức `add()` Container, với các đối số là một thành phần và một số nguyên chỉ số.

# Ví dụ sử dụng FlowLayout







# Ví dụ FlowLayout

```
public class FlowLayoutDemo extends JFrame{
    JRadioButton RtoLbutton;
    JRadioButton LtoRbutton;
    FlowLayout experimentLayout = new FlowLayout();
    final String RtoL = "Right to left";
    final String LtoR = "Left to right";
    JButton applyButton = new JButton("Apply component orientation");
    public FlowLayoutDemo(String name) {
        super(name);
    }
    public void addComponentsToPane(final Container pane) {
        final JPanel compsToExperiment = new JPanel();
        compsToExperiment.setLayout(experimentLayout);
        experimentLayout.setAlignment(FlowLayout.TRAILING);
        JPanel controls = new JPanel();
        controls.setLayout(new FlowLayout());
        LtoRbutton = new JRadioButton(LtoR);
        LtoRbutton.setActionCommand(LtoR);
        LtoRbutton.setSelected(true);
        RtoLbutton = new JRadioButton(RtoL);
        RtoLbutton.setActionCommand(RtoL);
        compsToExperiment.add(new JButton("Button 1"));
        compsToExperiment.add(new JButton("Button 2"));
        compsToExperiment.add(new JButton("Button 3"));
        compsToExperiment.add(new JButton("Long-Named Button 4"));
        compsToExperiment.add(new JButton("5"));
```





# Ví dụ FlowLayout

```
compsToExperiment.setComponentOrientation(  
    ComponentOrientation.LEFT_TO_RIGHT);  
final ButtonGroup group = new ButtonGroup();  
group.add(LtoRbutton);  
group.add(RtoLbutton);  
controls.add(LtoRbutton);  
controls.add(RtoLbutton);  
controls.add(applyButton);  
applyButton.addActionListener(new ActionListener() {  
    public void actionPerformed(ActionEvent e) {  
        String command = group.getSelection().getActionCommand();  
        if (command.equals("Left to right")) {  
            compsToExperiment.setComponentOrientation(  
                ComponentOrientation.LEFT_TO_RIGHT);  
        } else {  
            compsToExperiment.setComponentOrientation(  
                ComponentOrientation.RIGHT_TO_LEFT);  
        }  
        compsToExperiment.validate();  
        compsToExperiment.repaint();  
    }  
});  
pane.add(compsToExperiment, BorderLayout.CENTER);  
pane.add(controls, BorderLayout.SOUTH);  
}
```



# Ví dụ FlowLayout

```
private static void createAndShowGUI() {
    FlowLayoutDemo frame = new FlowLayoutDemo("FlowLayoutDemo");
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frame.addComponentsToPane(frame.getContentPane());
    frame.pack();
    frame.setVisible(true);
}

public static void main(String[] args) {
    try {
        UIManager.setLookAndFeel("javax.swing.plaf.metal.MetalLookAndFeel");
    } catch (UnsupportedLookAndFeelException ex) {
        ex.printStackTrace();
    } catch (IllegalAccessException ex) {
        ex.printStackTrace();
    } catch (InstantiationException ex) {
        ex.printStackTrace();
    } catch (ClassNotFoundException ex) {
        ex.printStackTrace();
    }
    UIManager.put("swing.boldMetal", Boolean.FALSE);
    javax.swing.SwingUtilities.invokeLater(new Runnable() {
        public void run() {
            createAndShowGUI();
        }
    });
}
```



# GridLayout

- **GridLayout:**

- Sắp xếp các thành phần vào một lưới gồm các hàng và các cột.
- Các thành phần được bổ sung bắt đầu tại ô trên-trái
  - Tiến hành trái-quả-phải cho đến khi hàng đầy
- Các ô trong khung lưới có cùng kích thước

# Ví dụ sử dụng GridLayout





# Ví dụ sử dụng GridLayout

```
public class GridLayoutDemo extends JFrame {
    static final String gapList[] = {"0", "10", "15", "20"};
    final static int maxGap = 20;
    JComboBox horGapComboBox;
    JComboBox verGapComboBox;
    JButton applyButton = new JButton("Apply gaps");
    GridLayout experimentLayout = new GridLayout(0,2);
    public GridLayoutDemo(String name) {
        super(name);
        setResizable(false);
    }
    public void initGaps() {
        horGapComboBox = new JComboBox(gapList);
        verGapComboBox = new JComboBox(gapList);
    }
    public void addComponentsToPane(final Container pane) {
        initGaps();
        final JPanel compsToExperiment = new JPanel();
        compsToExperiment.setLayout(experimentLayout);
        JPanel controls = new JPanel();
        controls.setLayout(new GridLayout(2,3));
        JButton b = new JButton("Just fake button");
        Dimension buttonSize = b.getPreferredSize();
        compsToExperiment.setPreferredSize(new Dimension((int) (buttonSize.getWidth() * 2.5)+maxGap,
            (int) (buttonSize.getHeight() * 3.5)+maxGap * 2));
        compsToExperiment.add(new JButton("Button 1"));
        compsToExperiment.add(new JButton("Button 2"));
        compsToExperiment.add(new JButton("Button 3"));
        compsToExperiment.add(new JButton("Long-Named Button 4"));
        compsToExperiment.add(new JButton("5"));
    }
}
```





# Ví dụ sử dụng GridLayout

```
controls.add(new Label("Horizontal gap:"));
controls.add(new Label("Vertical gap:"));
controls.add(new Label(" "));
controls.add(horGapComboBox);
controls.add(verGapComboBox);
controls.add(applyButton);
applyButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        String horGap = (String) horGapComboBox.getSelectedItem();
        String verGap = (String) verGapComboBox.getSelectedItem();
        experimentLayout.setHgap(Integer.parseInt(horGap));
        experimentLayout.setVgap(Integer.parseInt(verGap));
        experimentLayout.layoutContainer(compsToExperiment);
    }
});
pane.add(compsToExperiment, BorderLayout.NORTH);
pane.add(new JSeparator(), BorderLayout.CENTER);
pane.add(controls, BorderLayout.SOUTH);
}

private static void createAndShowGUI() {
    //Create and set up the window.
    GridLayoutDemo frame = new GridLayoutDemo("GridLayoutDemo");
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    //Set up the content pane.
    frame.addComponentsToPane(frame.getContentPane());
    //Display the window.
    frame.pack();
    frame.setVisible(true);
}
```



# Ví dụ sử dụng GridLayout

```
}  
  
public static void main(String[] args) {  
    /* Use an appropriate Look and Feel */  
    try {  
        //UIManager.setLookAndFeel("com.sun.java.swing.plaf.windows.WindowsLookAndFeel");  
        UIManager.setLookAndFeel("javax.swing.plaf.metal.MetalLookAndFeel");  
    } catch (UnsupportedLookAndFeelException ex) {  
        ex.printStackTrace();  
    } catch (IllegalAccessException ex) {  
        ex.printStackTrace();  
    } catch (InstantiationException ex) {  
        ex.printStackTrace();  
    } catch (ClassNotFoundException ex) {  
        ex.printStackTrace();  
    }  
    /* Turn off metal's use of bold fonts */  
    UIManager.put("swing.boldMetal", Boolean.FALSE);  
  
    //Schedule a job for the event dispatch thread:  
    //creating and showing this application's GUI.  
    javax.swing.SwingUtilities.invokeLater(new Runnable() {  
        public void run() {  
            createAndShowGUI();  
        }  
    });  
}
```



# GridBag Layout

- Chia component thành lưới các ô nhưng cho phép tùy chỉnh độ lớn hiển thị của các component trên layout



# Ví dụ sử dụng GridBagLayout





# Ví dụ sử dụng GridBagLayout

```
public class GridBagLayoutDemo {  
    final static boolean shouldFill = true;  
    final static boolean shouldWeightX = true;  
    final static boolean RIGHT_TO_LEFT = false;  
    public static void addComponentsToPane(Container pane) {  
        if (RIGHT_TO_LEFT)  
            pane.setComponentOrientation(ComponentOrientation.RIGHT_TO_LEFT);  
        JButton button;  
        pane.setLayout(new GridBagLayout());  
        GridBagConstraints c = new GridBagConstraints();  
  
        if (shouldFill)  
            c.fill = GridBagConstraints.HORIZONTAL;  
        button = new JButton("Button 1");  
  
        if (shouldWeightX)  
            c.weightx = 0.5;  
        c.fill = GridBagConstraints.HORIZONTAL;  
        c.gridx = 0;  
        c.gridy = 0;  
        pane.add(button, c);  
  
        button = new JButton("Button 2");  
        c.fill = GridBagConstraints.HORIZONTAL;  
        c.weightx = 0.5;  
        c.gridx = 1;  
        c.gridy = 0;  
        pane.add(button, c);  
    }  
}
```



# Ví dụ sử dụng GridBagLayout

```
button = new JButton("Button 3");
c.fill = GridBagConstraints.HORIZONTAL;
c.weightx = 0.5;
c.gridx = 2;
c.gridy = 0;
pane.add(button, c);

button = new JButton("Long-Named Button 4");
c.fill = GridBagConstraints.HORIZONTAL;
c.ipady = 40;           //make this component tall
c.weightx = 0.0;
c.gridwidth = 3;
c.gridx = 0;
c.gridy = 1;
pane.add(button, c);

button = new JButton("5");
c.fill = GridBagConstraints.HORIZONTAL;
c.ipady = 0;           //reset to default
c.weighty = 1.0;       //request any extra vertical space
c.anchor = GridBagConstraints.PAGE_END; //bottom of space
c.insets = new Insets(10,0,0,0); //top padding
c.gridx = 1;           //aligned with button 2
c.gridwidth = 2;       //2 columns wide
c.gridy = 2;           //third row
pane.add(button, c);
}
```



# Ví dụ sử dụng GridBagLayout

```
private static void createAndShowGUI() {
    //Create and set up the window.
    JFrame frame = new JFrame("GridBagLayoutDemo");
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

    //Set up the content pane.
    addComponentsToPane(frame.getContentPane());

    //Display the window.
    frame.pack();
    frame.setVisible(true);
}

public static void main(String[] args) {
    //Schedule a job for the event-dispatching thread:
    //creating and showing this application's GUI.
    javax.swing.SwingUtilities.invokeLater(new Runnable() {
        public void run() {
            createAndShowGUI();
        }
    });
}
```



# Các Layout manager khác

- GroupLayout
- SpringLayout

# Summary

