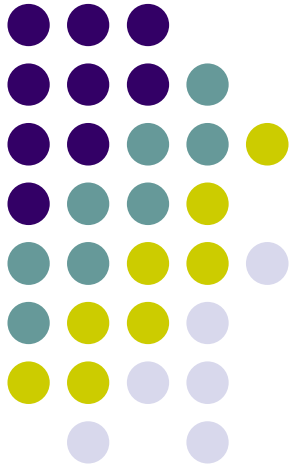
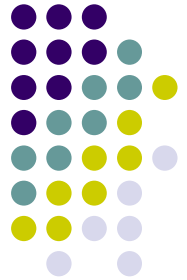
Swing Package

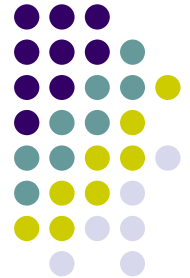


Nội dung



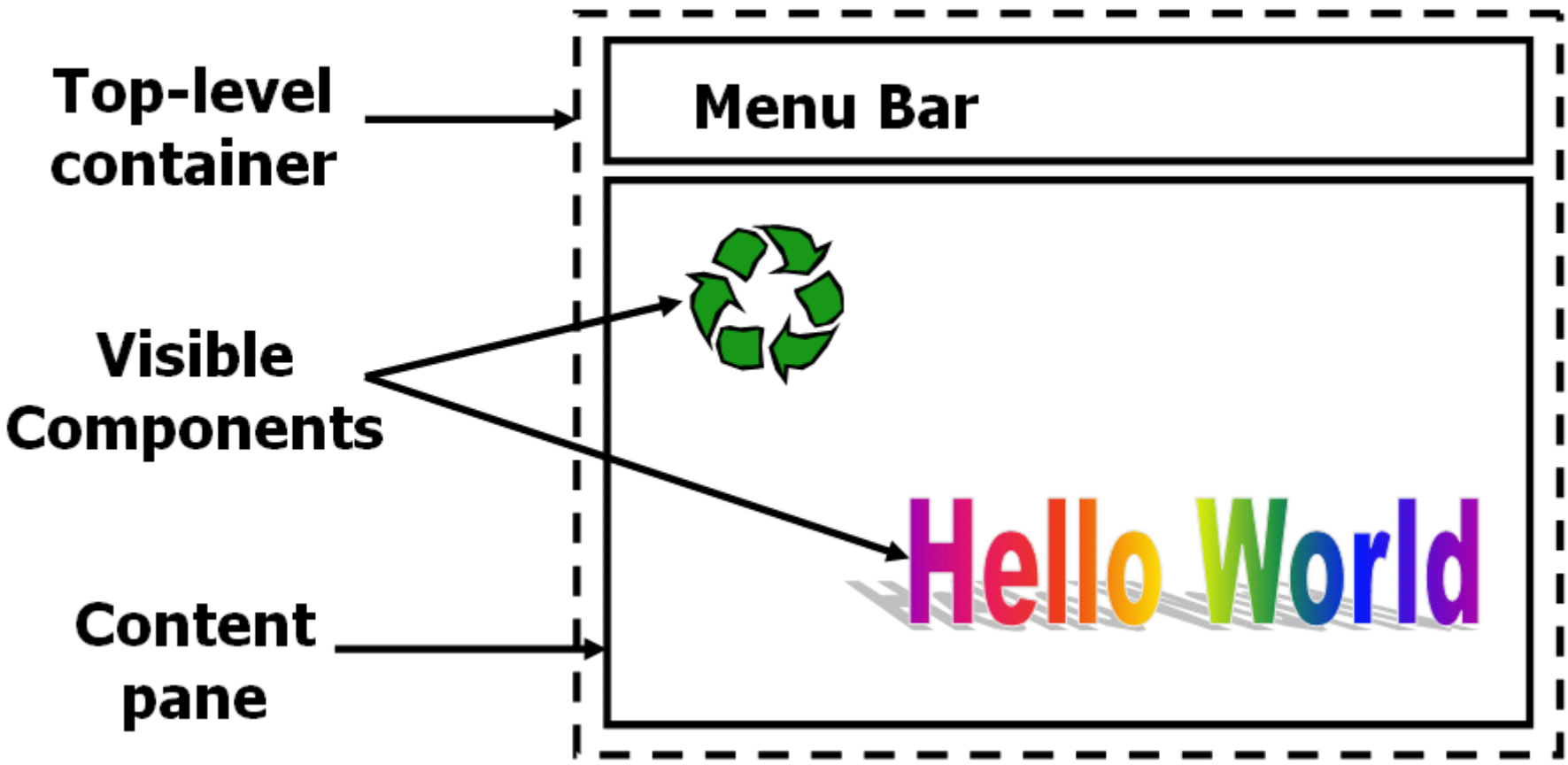
- Giao diện người dùng với Java (Java GUI)
- Các container: JFrame, JPanel, JDialog
- Các component: (JLabel, JTextField, JButton, JComboBox, JCheckBox, JRadioButton
- Layout manager: FlowLayout, GridLayout, BorderLayout, …
- Tạo menu

Giao diện người dùng với Java

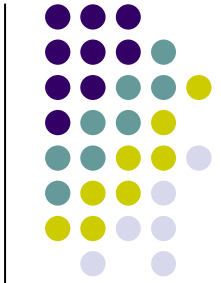


- Cung cấp các công cụ cho phép tạo giao tiếp trực quan và hấp dẫn với người dùng, được biết đến là swing
- Giao diện với người dùng bao gồm 1 cửa sổ chính, và các control được đặt lên trên
- Các thành phần tạo giao diện nằm trong gói javax.swing
- Tên của các lớp này bắt đầu bằng chữ J

Giao diện người dùng với Java (2)



JFrame

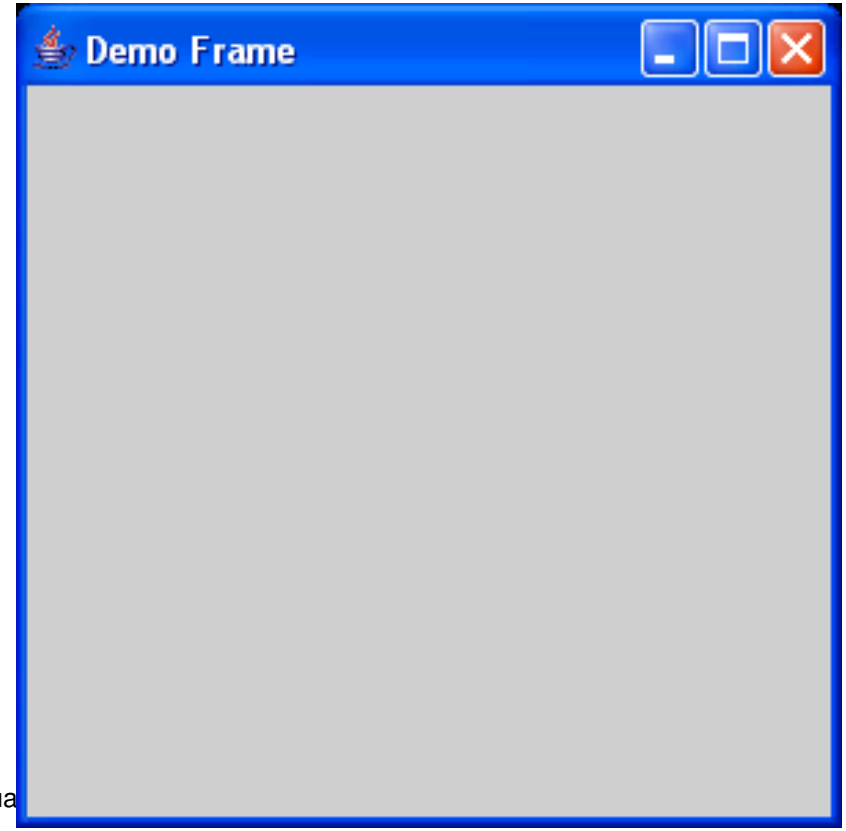


- Đây là cửa sổ chính, dùng để chứa các thành phần giao diện khác. Đóng vai trò là 1 container
- JFrame thường dùng để tạo ra cửa sổ trong chương trình swing
- Hàm dựng
  - JFrame()
  - JFrame(String title)
- Các thành phần đồ họa được đưa vào content pane, không đưa trực tiếp vào đối tượng JFrame
- Ví dụ:
  - frame.getContentPane().add(b);

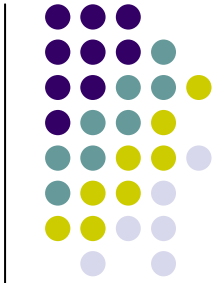
Ví dụ

```
import javax.swing.*;

public class DemoFrm extends JFrame
{
	public DemoFrm()
	{
		super("Demo Frame");
		setSize(300, 300);
		setVisible(true);
	}

	public static void main(String arg[])
	{
		new DemoFrm();
	}
}
```

Kết quả

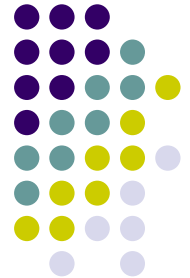


JPanel



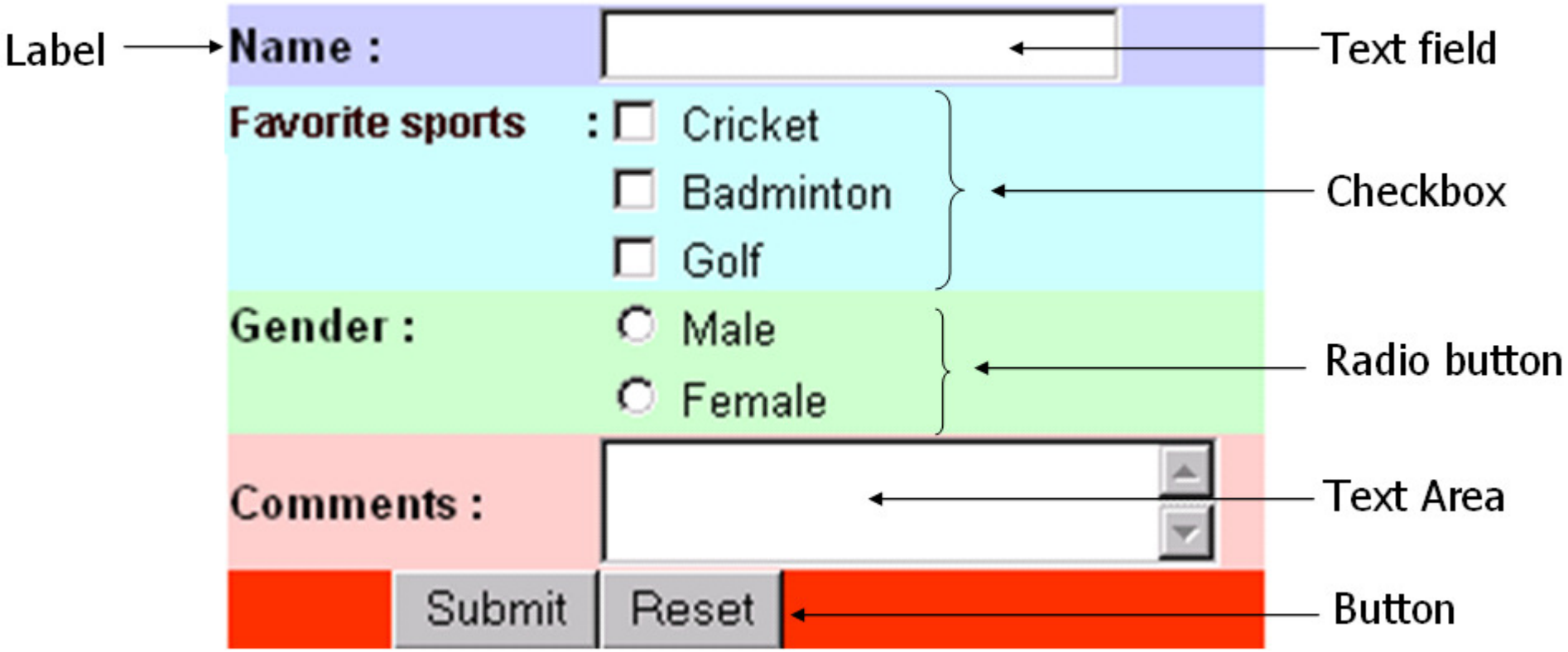
- JPanel là container trung gian dùng để chứa các component khác
- Thường dùng để phân chia các component trong ứng dụng
- Layout mặc định là FlowLayout
- Các hàm dựng:
  - JPanel()
  - JPanel(LayoutManager lm)

Các thành phần giao tiếp người dùng cơ bản

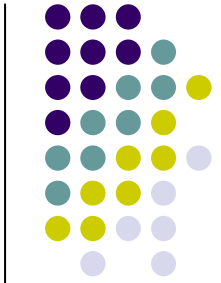


- Form được dùng để thu thập thông tin từ phía người dùng
- Trong khi tạo giao tiếp, component được dùng để cho phép nhập liệu là textfield hoặc textbox
- Để khởi tạo các phần tử, các bước cần phải thực hiện là:
  - Tạo phần tử
  - Thiết lập các thuộc tính (size, color, font, …)
  - Xác định vị trí cho nó
  - Đưa nó vào màn hình

Các loại component

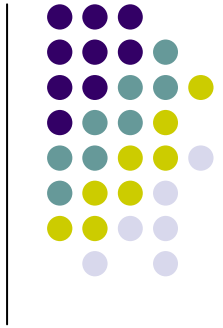


JLabel



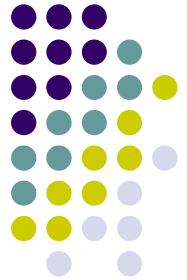
- Được dùng để hiển thị văn bản (text) và hình ảnh (image)
- Tạo hiệu ứng trực quan cho màn hình giao diện
- Các hàm dựng của JLabel:
  - JLabel(Icon img)
  - JLabel(String st)
  - JLabel(String st, Icon img, int align)

Ví dụ

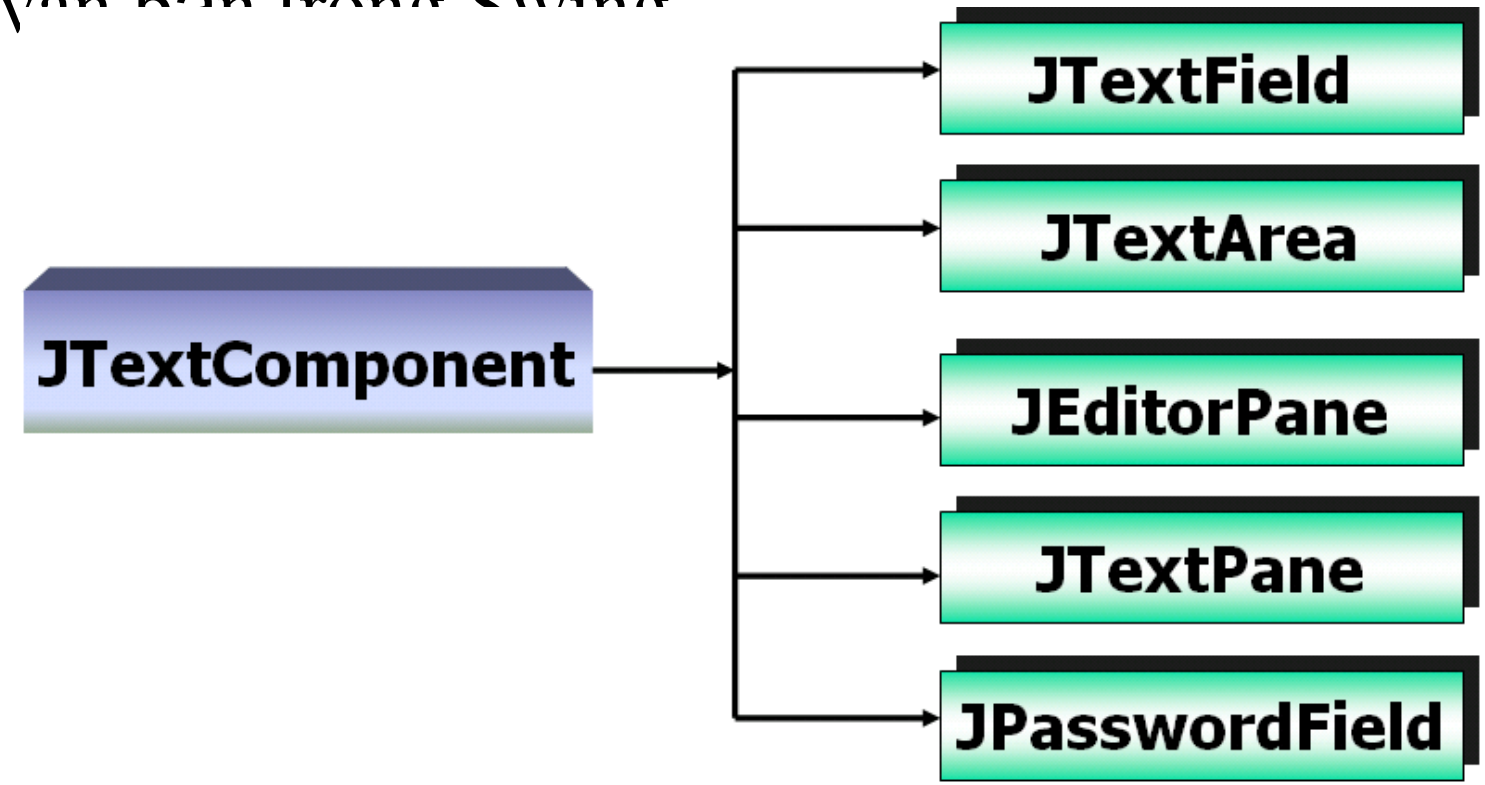


```
import javax.swing.*;
import java.awt.*;
public class DemoFrm extends JFrame
{
	public DemoFrm()
	{
		super("Demo Frame");
		Container c = getContentPane();
		JLabel lblHello = new JLabel("Hello, world");
		c.add(lblHello);
		setSize(300, 300);
		setVisible(true);
	}

	public static void main(String arg[])
	{
		new DemoFrm();
	}
}
```

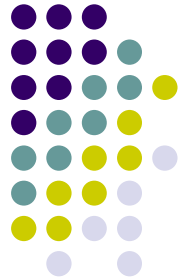
JTextComponent



- Đây là lớp cha của tất cả các lớp hiển thị văn bản trong Swing

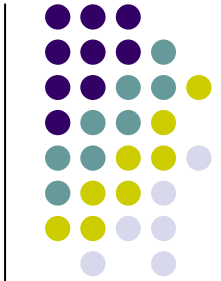


JTextComponent (2)



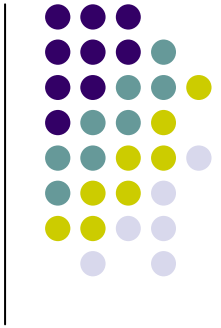
- JTextField cho phép soạn thảo chỉ 1 dòng văn bản
- Các hàm dựng của JTextField:
  - JTextField()
  - JTextField(int columns)
  - JTextField(String text)
  - JTextField(String text, int column)
- Chuỗi hiển thị được xác định trong trường text, và số cột hiển thị được xác định trong trường column

Ví dụ



```
import javax.swing.*;
import java.awt.*;
public class DemoFrm extends JFrame
{
	public DemoFrm()
	{
		super("Demo Frame");
		Container c = getContentPane();
		c.setLayout(new FlowLayout());
		JLabel lblHello = new JLabel("Ho va ten");
		JTextField tfHoten = new JTextField(20);
		c.add(lblHello);
		c.add(tfHoten);
		pack();   // setsize vua đủ
		setVisible(true);
	}

	public static void main(String arg[])
	{
		new DemoFrm();
	}
}
```

JTextArea

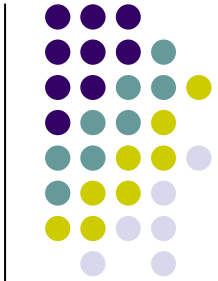


- Đây là component cho phép nhập vào nhiều dòng văn bản
- Có hỗ trợ các thanh cuộn
- Các hàm dựng:
  - JTextArea()
  - JTextArea(int rows, int cols)
  - JTextArea(String text)
  - JTextArea(String text, int rows, int cols)

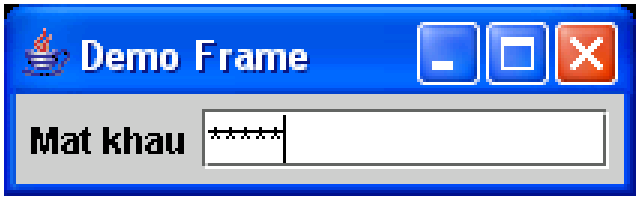
Ví dụ

```
import javax.swing.*;
import java.awt.*;
public class DemoFrm extends JFrame
{
	public DemoFrm()
	{
		super("Demo Frame");
		Container c = getContentPane();
		c.setLayout(new FlowLayout());
		JLabel lblHello = new JLabel("Mo ta");
		JTextArea tfHoten = new JTextArea(5, 15);
		c.add(lblHello);
		c.add(tfHoten);
		pack();
		setVisible(true);
	}

	public static void main(String arg[])
	{
		new DemoFrm();
	}
}
```

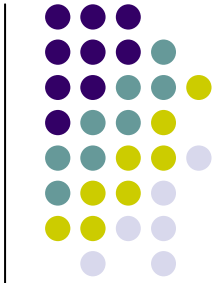
JPasswordField



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

public class DemoFrm extends JFrame
{
	public DemoFrm()
	{
		super("Demo Frame");
		Container c = getContentPane();
		c.setLayout(new FlowLayout());
		JLabel lblHello = new JLabel("Mat khau");
		JPasswordField tfHoten = new JPasswordField(12);
		tfHoten.setEchoChar('*');
		c.add(lblHello);
		c.add(tfHoten);
		pack();
		setVisible(true);
	}

	public static void main(String arg[])
	{
		new DemoFrm();
	}
}
```

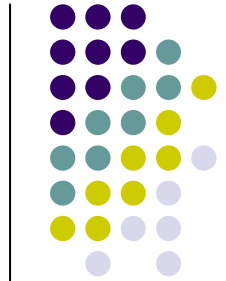


JButton

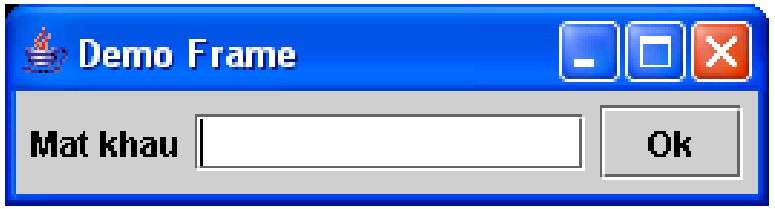


- Thể hiện chức năng nút bấm
- JButton là lớp con của lớp AbstractButton
- Đối tượng JButton bao gồm chuỗi văn bản, hình ảnh và các đường viền
- Các hàm dựng:
  - JButton()
  - JButton(Icon icon)
  - JButton(String text)
  - JButton(String text, Icon icon)
  - JButton(Action a)

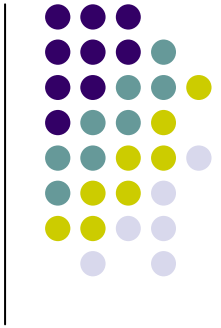
Ví dụ



```
import javax.swing.*;
import java.awt.*;
public class DemoFrm extends JFrame
{
	public DemoFrm()
	{
		super("Demo Frame");
		Container c = getContentPane();
		c.setLayout(new FlowLayout());
		JLabel lblHello = new JLabel("Mat khau");
		JPasswordField tfHoten = new JPasswordField(12);
		tfHoten.setEchoChar('*');
		JButton btnOK = new Button("Ok");
		c.add(lblHello);
		c.add(tfHoten);
		c.add(btnOK);
		pack();
		setVisible(true);
	}

	public static void main(String arg[])
	{
		new DemoFrm();
	}
}
```



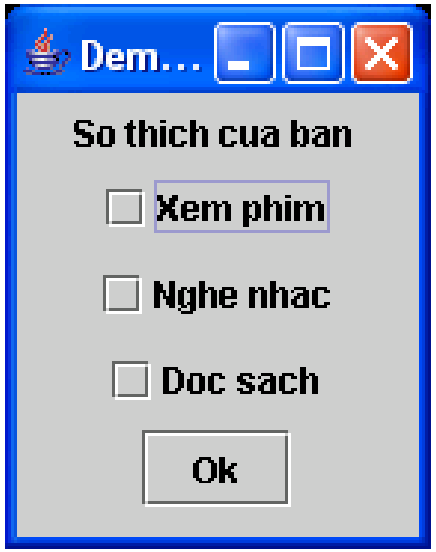
JCheckBox



- Checkbox được dùng để cung cấp cho người dùng khả năng lựa chọn
- Các hàm dựng của lớp JCheckBox:
  - JCheckBox()
  - JCheckBox(Icon icon)
  - JCheckBox(Icon icon, boolean selected)
  - JCheckBox(String text)
  - JCheckBox(String text, boolean selected)
  - JCheckBox(String text, Icon icon)
  - JCheckBox(String text, Icon icon, boolean selected)
  - JCheckBox(Action a)

Ví dụ

```
import javax.swing.*;
import java.awt.*;
public class DemoFrm extends JFrame
{
	public DemoFrm()
	{
		super("Demo Frame");
		Container c = getContentPane();
		c.setLayout(new FlowLayout());
		JLabel lblHello = new JLabel("So thich cua ban ");
		JCheckBox cbPhim = new JCheckBox("Xem phim");
		JCheckBox cbNhac = new JCheckBox("Nghe nhac");
		JCheckBox cbSach = new JCheckBox("Doc sach");
		JButton btnOK = new JButton("Ok");
		c.add(lblHello);
		c.add(cbPhim);
		c.add(cbNhac);
		c.add(cbSach);
		c.add(btnOK);
		pack();
		setVisible(true);
	}
	public static void main(String arg[])
	{
		new DemoFrm();
	}
```

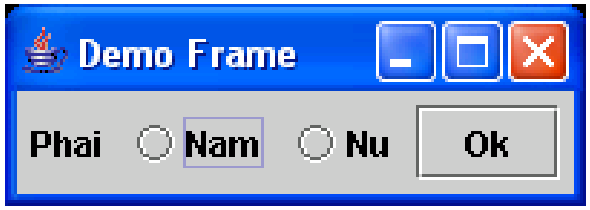


JRadioButton

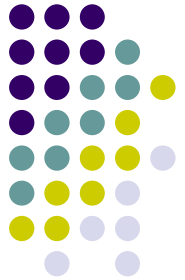
- 1 tập các nút đài cho phép chỉ lựa chọn được 1 tại 1 thời điểm
- Dùng lớp ButtonGroup để tạo ra nhóm
- Các hàm dựng của lớp JRadioButton:
  - JRadioButton()
  - JRadioButton(Icon icon)
  - JRadioButton(Icon icon, boolean selected)
  - JRadioButton(String text)
  - JRadioButton(String text, boolean selected)
  - JRadioButton(String text, Icon icon)
  - JRadioButton(String text, Icon icon, boolean selected)
  - JRadioButton(Action a)

Ví dụ

```
import javax.swing.*;
import java.awt.*;
public class DemoFrm extends JFrame
{
	public DemoFrm()
	{
		super("Demo Frame");
		Container c = getContentPane();
		c.setLayout(new FlowLayout());
		JLabel lblHello = new JLabel("Phai ");
		JRadioButton rbNam = new JRadioButton("Nam");
		JRadioButton rbNu = new JRadioButton("Nu");
		ButtonGroup bg = new ButtonGroup();
		bg.add(rbNam);
		bg.add(rbNu);
		JButton btnOK = new JButton("Ok");
		c.add(lblHello);
		c.add(rbNam);
		c.add(rbNu);
		c.add(btnOK);
		pack();
		setVisible(true);
	}
```

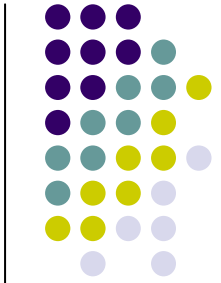


JList



- Khi các thông tin dùng để chọn lựa nhiều, chúng ta có thể dùng 1 danh sách để cho phép việc chọn
- Component JList cho phép sắp xếp dữ liệu hiển thị, có thể phân nhóm
- JList có thể hiển thị chuỗi và icon
- JList không hỗ trợ double click chuột

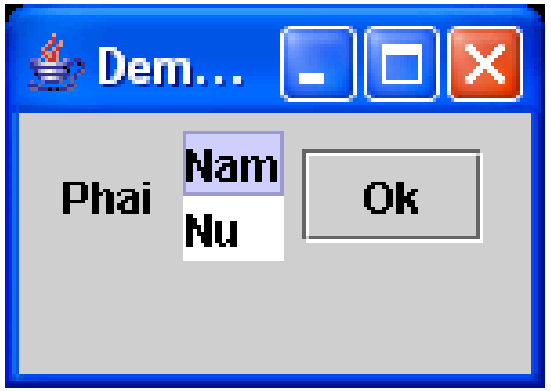
JList (2)



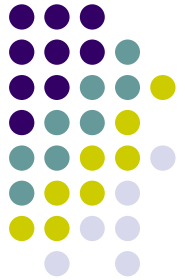
- Các hàm dựng
  - JList()
  - JList(ListModel dataModel)
  - JList(Object []listData)

Ví dụ

```
import javax.swing.*;
import java.awt.*;
public class DemoFrm extends JFrame
{
	public DemoFrm()
	{
		super("Demo Frame");
		Container c = getContentPane();
		c.setLayout(new FlowLayout());
		JLabel lblHello = new JLabel("Phai ");
		String []listData = {"Nam", "Nu"};
		JList list = new JList(listData);
		JButton btnOK = new JButton("Ok");
		c.add(lblHello);
		c.add(list);
		c.add(btnOK);
		pack();
		setVisible(true);
	}

	public static void main(String arg[])
	{
		new DemoFrm();
	}
}
```

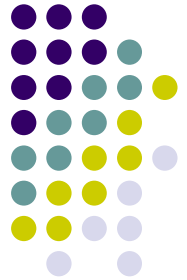


JComboBox



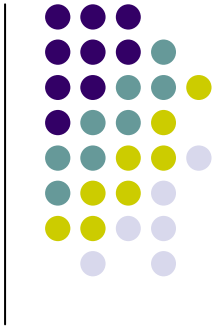
- Là sự kết hợp giữa TextField và Listbox
- Cấu trúc gần giống như JList, các hàm dựng:
  - JComboBox()
  - JComboBox(ComboBoxModel asModel)
  - JComboBox(Object []item)

Bố trí các thành phần bên trong các đối tượng chứa



- Sử dụng Layout Managers
  - FlowLayout
  - BorderLayout
  - GridLayout
- Làm việc không có Layout Manager(Absolute Positioning)

FlowLayout



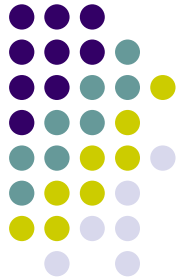
- Là một class cung cấp cách quản lý việc xắp đặt đơn giản các component.
- Lớp FlowLayout có ba cấu trúc:
  - `public FlowLayout()`
  - `public FlowLayout(int alignment)`
  - `public FlowLayout(int alignment, int horizontalGap, int verticalGap)`

`alignment` có các giá trị `FlowLayout.LEFT`, `FlowLayout.CENTER`, hoặc `FlowLayout.RIGHT`. Thông số `horizontalGap` và `verticalGap` xác định số Pixel đặt giữa các thành phần. Giá tri 5 cho mỗi thông số.

FlowLayout

```
import javax.swing.*;
import java.awt.*;
public class FlowLayoutDemo extends JFrame{
   public FlowLayoutDemo() {
      FlowLayout layout  = new FlowLayout();
                  Container c = getContentPane();
                  c.setLayout(layout);
                  c.add(new JButton("Button 1"));
                  c.add(new JButton("Button 2"));
                  c.add(new JButton("Button 3"));
                  c.add(new JButton("Long-Named Button 4"));
                  c.add(new JButton("5"));
   }
   public static void main (String[] args) {
                  FlowLayoutDemo fr = new FlowLayoutDemo();
                  fr.setDefaultCloseOperation(EXIT_ON_CLOSE);
                  fr.pack();
                  fr.setVisible(true);
      }
}
```

BorderLayout



- `add(Component c, BorderLayout layout)`

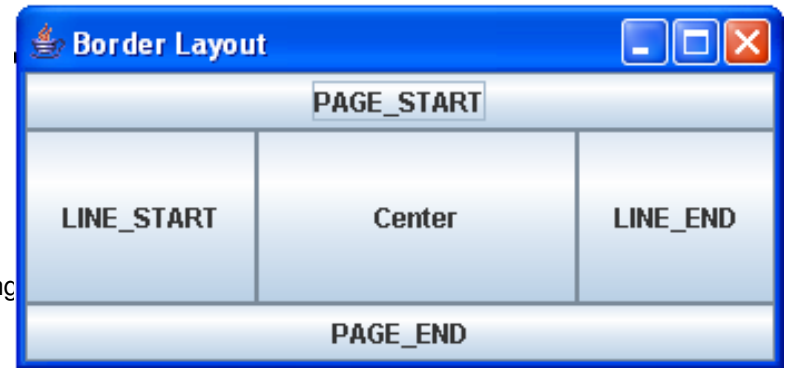
  - c: component thêm vào container.

  - layout: có 5 giá trị
    - PAGE_START
    - PAGE_END
    - LINE_START
    - LINE_END
    - CENTER

  Nếu add() có một thông số thì thành phần đó sẽ không hiển thị.
- Mặc định, BorderLayout không đặt khoảng trống giữa các thành.
- public BorderLayout(int horizontalGap, int verticalGap)

BorderLayout

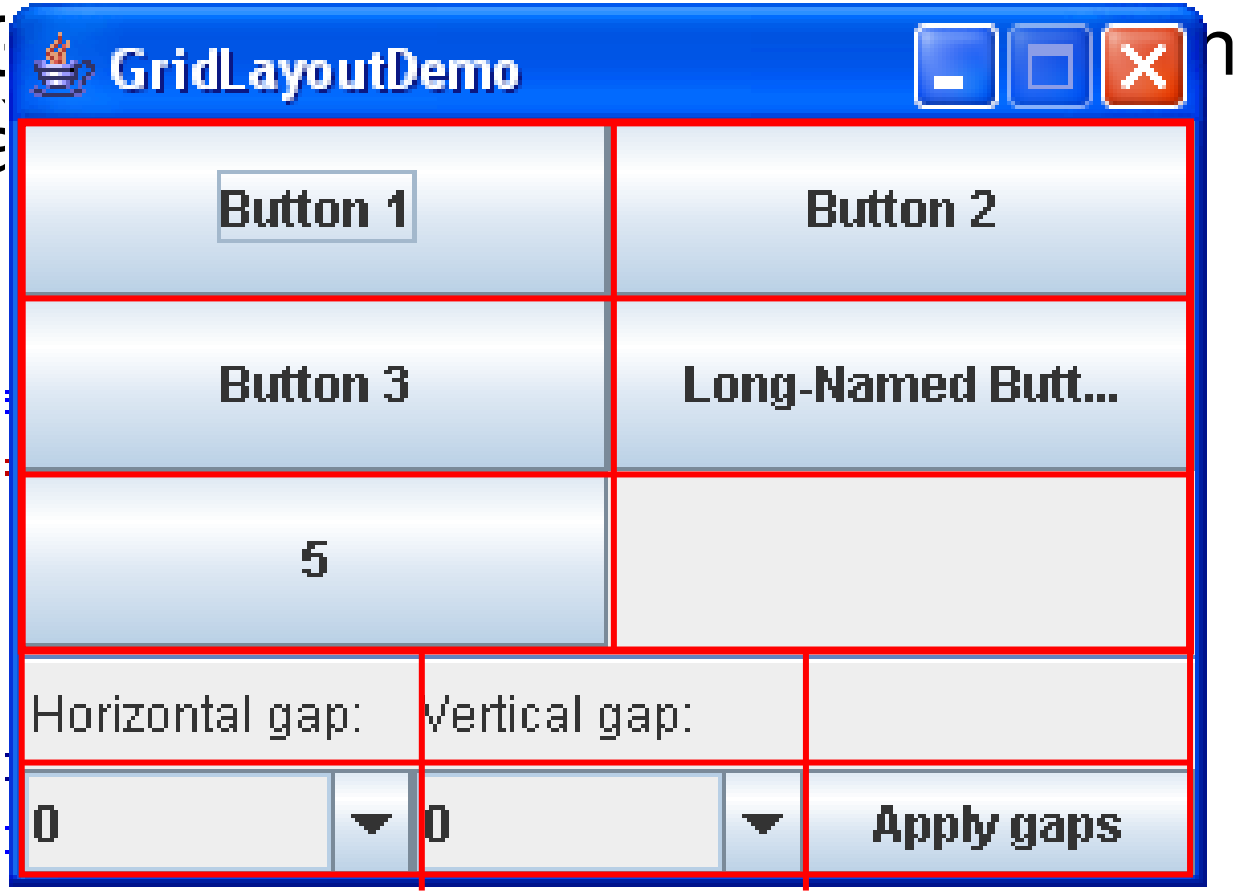
```
import java.awt.*;
import javax.swing.*;
public class BorderLayoutTest {
	public static void main(String args[]) {
		JFrame frame = new JFrame("Border Layout");
		Container contentPane = frame.getContentPane();
		contentPane.setLayout(new BorderLayout());
		contentPane.add(new JButton("PAGE_START"), BorderLayout.PAGE_START);
		contentPane.add(new JButton("PAGE_END"), BorderLayout.PAGE_END);
		contentPane.add(new JButton("LINE_END"), BorderLayout.LINE_END);
		contentPane.add(new JButton("LINE_START"), BorderLayout.LINE_START);
		contentPane.add(new JButton("Center"), BorderLayout.CENTER);
		frame.setSize(300, 200);
		frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
		frame.pack();
		frame.show();
	}
}
```



GridLayout

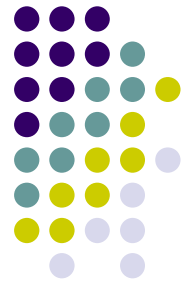
- Lớp G… hiều cột và…

```
public
public
```



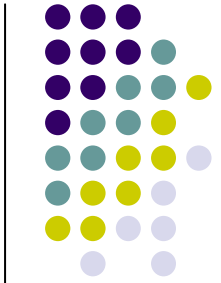
GridLayoutDemo.java

Làm việc không có Layout Manager(Absolute Positioning)



- Bạn có thể đặt vị trí cho các component mà không cần dùng layout manager. Giải pháp này được dùng để xác định hoàn toàn kích thước và vị trí của component.
- Mặc dù có thể làm việc mà không cần Layout Manager, bạn nên dùng Layout Manager nếu có thể. Layout managers dể thay đổi kích thước của Container và điều chỉnh hình dạng của các thành phần phụ thuộc vào Platform.

Ví dụ



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

public class AbsoluteLayoutDemo {
   public static void addComponentsToPane(Container pane) {
      pane.setLayout(null);

      JButton b1 = new JButton("one");
      JButton b2 = new JButton("two");
      JButton b3 = new JButton("three");

      pane.add(b1);
      pane.add(b2);
      pane.add(b3);

      Insets insets = pane.getInsets();
      Dimension size = b1.getPreferredSize();
      b1.setBounds(25 + insets.left, 5 + insets.top,  size.width, size.height);
      size = b2.getPreferredSize();
      b2.setBounds(55 + insets.left, 40 + insets.top, size.width, size.height);
      size = b3.getPreferredSize();
      b3.setBounds(150 + insets.left, 15 + insets.top, size.width + 50, size.height + 20);
   }
```

Ví dụ

```
    private static void createAndShowGUI() {
        JFrame frame = new JFrame("AbsoluteLayoutDemo");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        addComponentsToPane(frame.getContentPane());

        //Size and display the window.
        Insets insets = frame.getInsets();
        frame.setSize(300 + insets.left + insets.right,
                      125 + insets.top + insets.bottom);
        frame.setVisible(true);
    }

    public static void main(String[] args) {
            createAndShowGUI();
    }
}
```