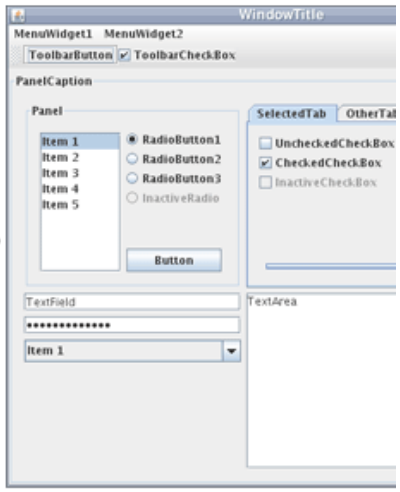
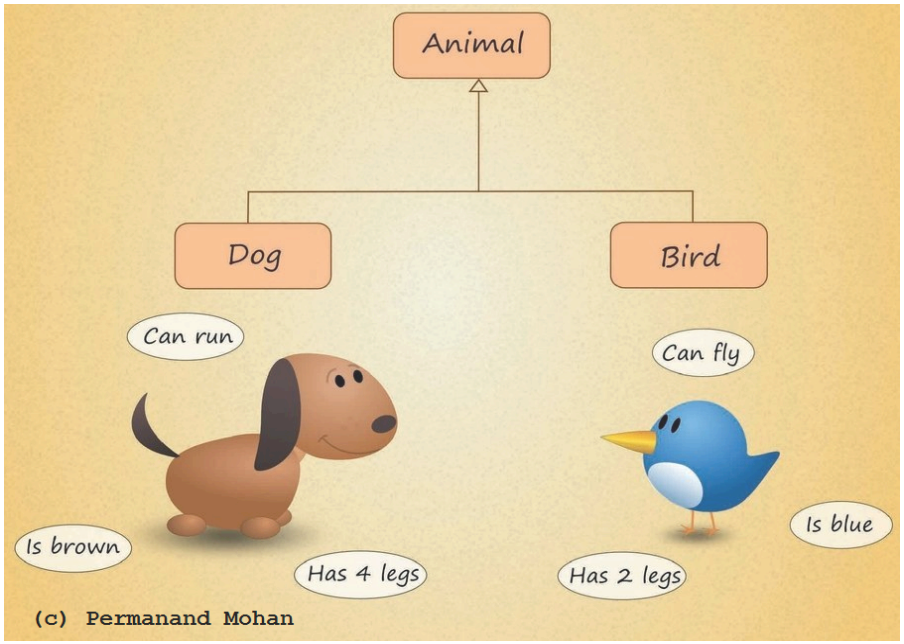




Chapter 5

Lập trình giao diện đồ họa

CT176 – LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG

Mục tiêu

Chương này nhằm giới thiệu
cách thức xây dựng giao diện đồ họa trong Java

Nội dung

- Giới thiệu
- Tạo 1 ứng dụng với giao diện đồ họa
- Các lớp vật chứa
- Các thành phần giao diện Swing
- Sắp xếp bố cục
- Xử lý sự kiện
- Trình đơn, thanh công cụ
- Mô hình MVC

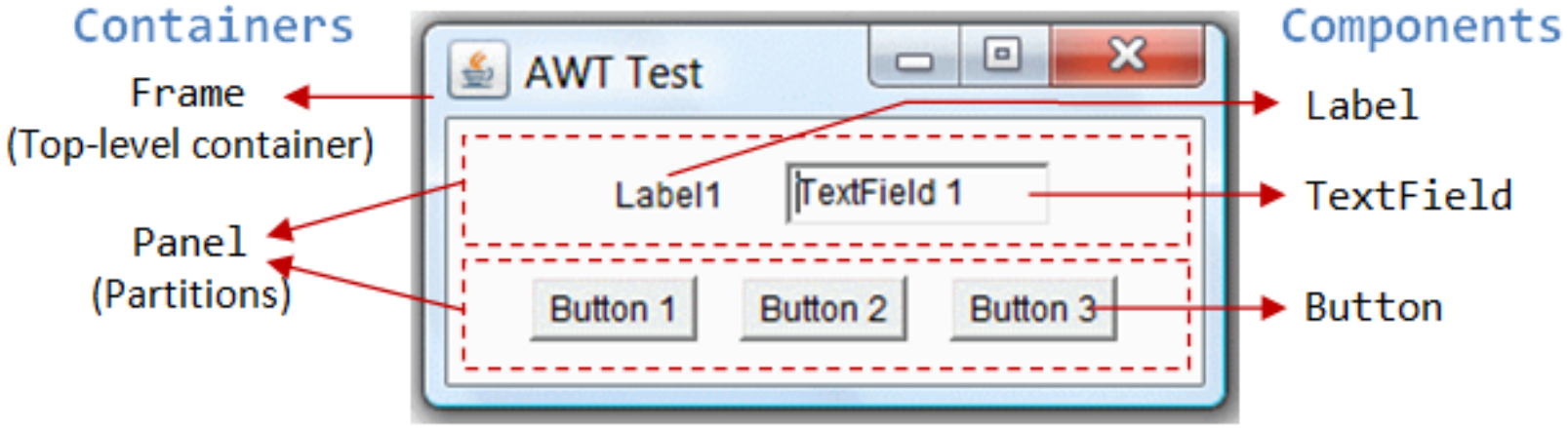
Giới thiệu

- Java cung cấp 2 bộ thư viện hàm dùng cho việc xây dựng giao diện đồ họa là: AWT và SWING.
- **Abstract Window Toolkit** (AWT)
  - Giới thiệu từ JDK 1.0, bao gồm 12 gói
  - 2 gói thường dùng là java.awt và java.awt.event
  - Cung cấp giao diện **phụ thuộc** vào nền GUI của hệ điều hành.
  - Các thành phần được gọi là ***heavyweight*** components.
- **Swing**
  - Nâng cấp của AWT, được giới thiệu từ JDK 1.2
  - Bao gồm 18 gói (cho đến JDK 1.7)
  - Là 1 phần trong JFC (Java Foundation Classes)
  - Giao diện thuần của Java => **độc lập** với nền GUI của hệ điều hành.
  - Các thành phần được gọi là ***lightweight*** components

AWT

- Gói **java.awt** bao gồm các lớp:
  - Thành phần GUI (Button, TextField, and Label, …)
  - Vật chứa GUI (Frame, Panel, Dialog, ScrollPane, …)
  - Sắp xếp bố cục (FlowLayout, BorderLayout, GridLayout, …)
  - Tùy chọn (Graphics, Color, Font, …)
- Gói **java.awt.event** bao gồm các lớp
  - Sự kiện (ActionEvent, MouseEvent, KeyEvent, WindowEvent)
  - Lắng nghe sự kiện (ActionListener, MouseListener, KeyListener, WindowListener, …)
  - Các lớp Adapter (MouseAdapter, KeyAdapter, and WindowAdapter)
- Swing có sử dụng lại 1 số thành phần trong AWT.

Vật chứa (**Container**) và thành phần (**Component**)



- **Component** là các thành phần cơ bản trong GUI.
- **Container** sẽ giữ các component bên trong theo cách sắp xếp bố cục (Layout) cho trước.
- Container có thể giữ các container khác bên trong.
- Không nên "trộn" chung các thành phần AWT và Swing vì các thành phần AWT sẽ được hiển thị trên các thành phần Swing.

Tạo 1 ứng dụng với giao diện đồ họa

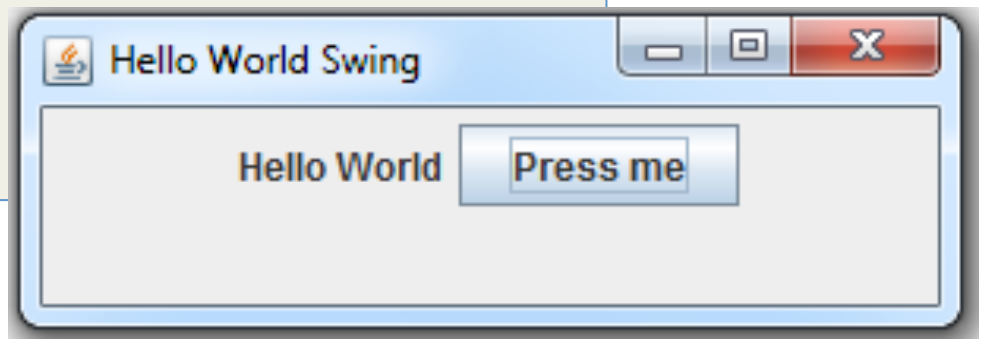
1. Import các gói:
   - java.awt
   - javax.swing
2. Xây dựng container cấp cao (top-level)
3. Chọn cách sắp xếp bố cục
4. Thêm các thành phần giao diện vào container
5. Cài đặt quản lý (lắng nghe, xử lý) các sự kiện
6. Hiển thị container

Ví dụ

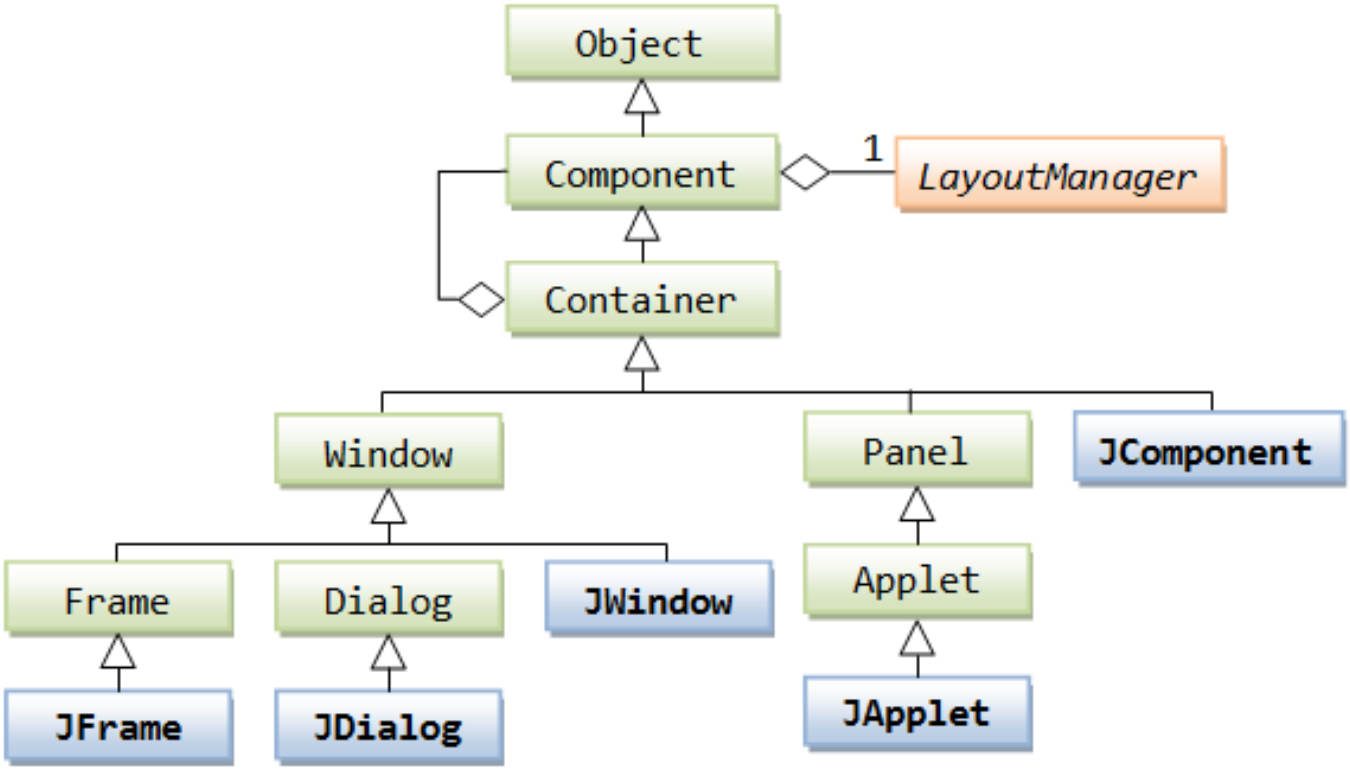
```
import javax.swing.*;
import java.awt.*;
public class HelloWorldSwing {
    public static void main(String[] args) {
        JFrame f = new JFrame("Hello World Swing");
        f.setLayout(new FlowLayout());
        JLabel la1 = new JLabel("Hello World");
        JButton but1 = new JButton("Press me");
        f.getContentPane().add(la1);
        f.getContentPane().add(but1);
        f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        //f.pack();
        f.setSize(300,100);
        f.setVisible(true);
    }
}
```

CT sẽ kết thúc khi đóng cửa sổ

Tự điều chỉnh kích thước cửa sổ cho vừa đủ các thành phần bên trong



Các lớp vật chứa Swing



Các lớp vật chứa (Container)

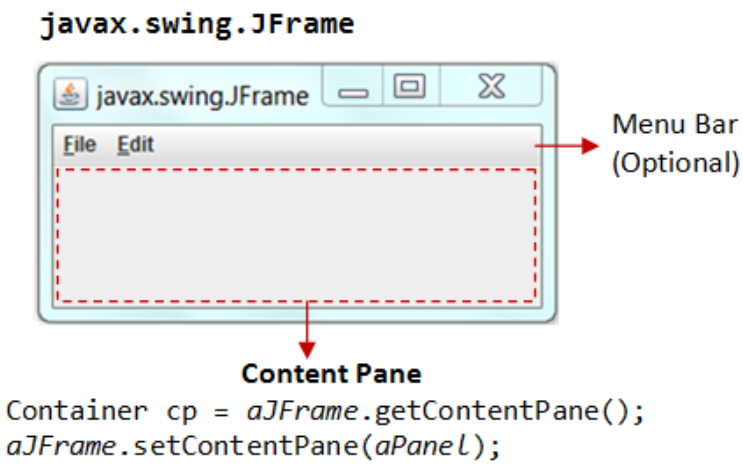
- Lớp vật chứa cấp cao (top-level): có 3 loại chính
  - JFrame
  - JDialog
  - Japplet
- Lớp vật chứa thứ cấp (secondary)
  - Sử dụng để nhóm và sắp xếp bố cục các thành phần
  - Chẳng hạn như: JPanel

JFrame

- Là cửa sổ có khung, icon, tiêu đề, các nút điều khiển (thu nhỏ, phóng to, đóng cửa sổ)
- Có thanh menu (tùy chọn), ô nội dung

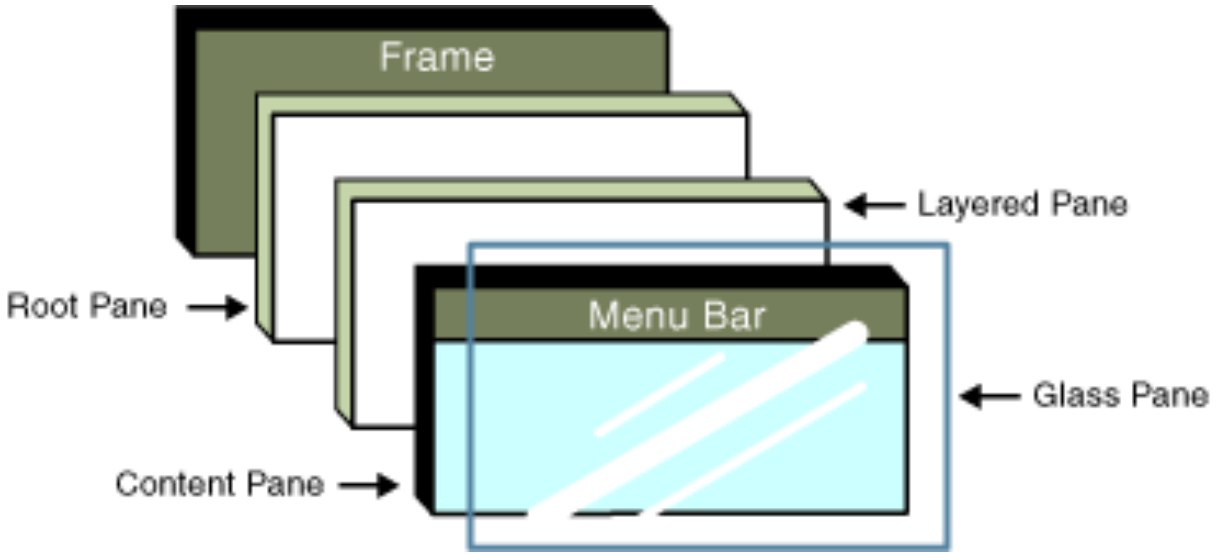
```
public class TestSetContentPane extends JFrame {
   // Constructor
   public TestSetContentPane() {
      // The "main" JPanel holds all the GUI components
      JPanel mainPanel = new JPanel(new FlowLayout());
      mainPanel.add(new JLabel("Hello, world!"));
      mainPanel.add(new JButton("Button"));

      // Set the content-pane of this JFrame
      this.setContentPane(mainPanel);
      ......
   }
   ........
}
```



JFrame

- Có thể được tạo ra bởi nhiều tầng (layer)
- Các tầng có thể chứa nhiều thành phần và được thêm vào JFrame
- Các tầng được sử dụng để tùy biến hiển thị của cửa sổ.



JFrame

- Một số các hàm quan trọng
  - `JFrame()`
  - `void setIconImage(Image)`
  - `void setTitle(String)`
  - `void setSize(int, int)`
  - `void setLocation(int, int)`
  - `void setContentPane(Container)`
  - `void setJMenuBar(JMenuBar)`
  - `protected void setRootPane(JRootPane root);`
  - `public JRootPane getRootPane();`
  - `public Container getContentPane();`
  - `public void setLayeredPane(JLayeredPane layered);`
  - `public JLayeredPane getLayeredPane();`
  - `public void setGlassPane(Component glass);`
  - `public Component getGlassPane();`
  - `JFrame(String)`
  - `Image getIconImage()`
  - `String getTitle()`
  - `Dimension getSize()`

Ví dụ JFrame

```
import javax.swing.*;

public class SimpleFrame {
        public static void main (String[] args) {
                JFrame frame = new JFrame();
                frame.setVisible (true);
        }
}
```

Ví dụ JFrame

```
import javax.swing.JFrame;
public class JFrameExample extends JFrame {
    public JFrameExample() {
        setTitle("Simple example");
        setSize(300, 200);
        setLocationRelativeTo(null);
        setDefaultCloseOperation(EXIT_ON_CLOSE);
    }
    public static void main(String[] args) {
        JFrameExample ex = new JFrameExample();
        ex.setVisible(true);
    }
}
```

Frame sẽ xuất hiện ở giữa màn hình desktop

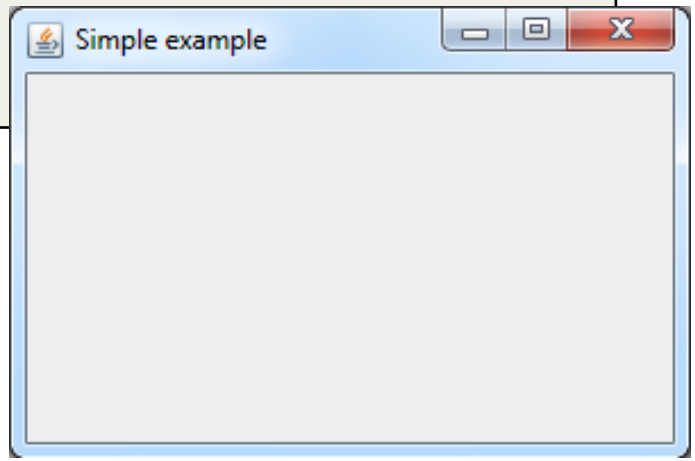


Table 14.3 Useful Properties That Are Specific to `JFrames`

Property	Type	Description	Methods
default Close operation	`int`	What should happen when the frame is closed; choices include: • `JFrame.DO_NOTHING_ON_CLOSE` (don't do anything) • `JFrame.HIDE_ON_CLOSE` (hide the frame) • `JFrame.DISPOSE_ON_CLOSE` (hide and destroy the frame so that it cannot be shown again) • `JFrame.EXIT_ON_CLOSE` (exit the program)	`getDefaultCloseOperation`, `setDefaultCloseOperation(int)`
icon image	`Image`	The icon that appears in the title bar and Start menu or Dock	`getIconImage`, `setIconImage(Image)`
layout	`LayoutManager`	An object that controls the positions and sizes of the components inside this frame	`getLayout`, `setLayout(LayoutManager)`
resizable	`boolean`	Whether or not the frame allows itself to be resized	`isResizable`, `setResizable(boolean)`
title	`String`	The text that appears in the frame's title bar	`getTitle`, `setTitle(String)`

Table 14.4 Useful Properties of All Components (Including `JFrames`)

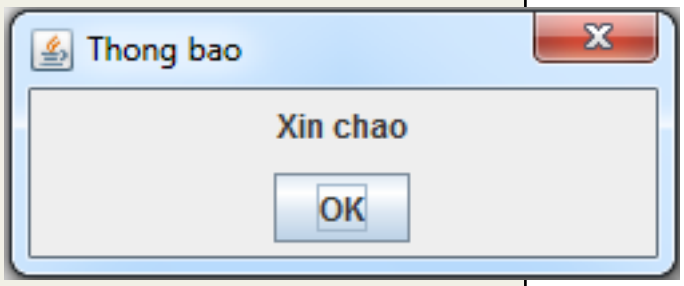
Property	Type	Description	Methods
background	`Color`	Background color	`getBackground`, `setBackground(Color)`
enabled	`boolean`	Whether the component can be interacted with	`isEnabled`, `setEnabled(boolean)`
focusable	`boolean`	Whether the keyboard can send input to the component	`isFocusable`, `setFocusable(boolean)`
font	`Font`	Font used to write text	`getFont`, `setFont(Font)`
foreground	`Color`	Foreground color	`getForeground`, `setForeground(Color)`
location	`Point`	(*x*, *y*) coordinate of component's top-left corner	`getLocation`, `setLocation(Point)`
size	`Dimension`	Current width and height of the component	`getSize`, `setSize(Dimension)`
preferred size	`Dimension`	"Preferred" width and height of the component; the size it should be to make it appear naturally on the screen (used with layout managers, seen later)	`getPreferredSize`, `setPreferredSize(Dimension)`
visible	`boolean`	Whether the component can be seen on the screen	`isVisible`, `setVisible(boolean)`

JDialog

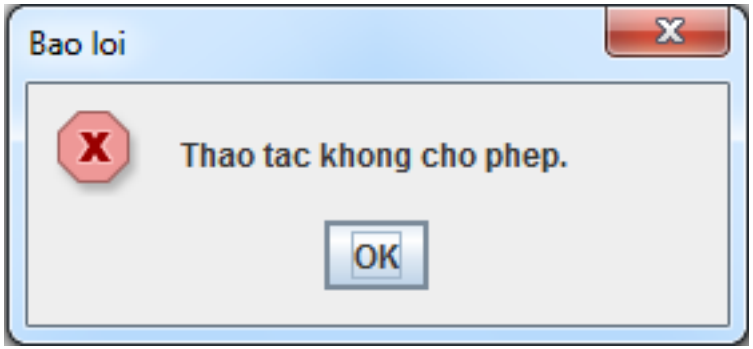
- Đơn giản và bị giới hạn hơn JFrame
- Có khung viền và thanh tiêu đề
- Được dùng để thể hiện thông báo ngắn ra màn hình.
- Dialog có thể thuộc dạng "modal": khi hiển thị dialog thì khóa truy xuất của người dùng đến các cửa sổ khác.
- Dialog có thể thuộc dạng non-modal: sử dụng trực tiếp lớp JDialog.
- JOptionPane sẽ tạo ra dialog dạng modal.

Ví dụ về JDialog

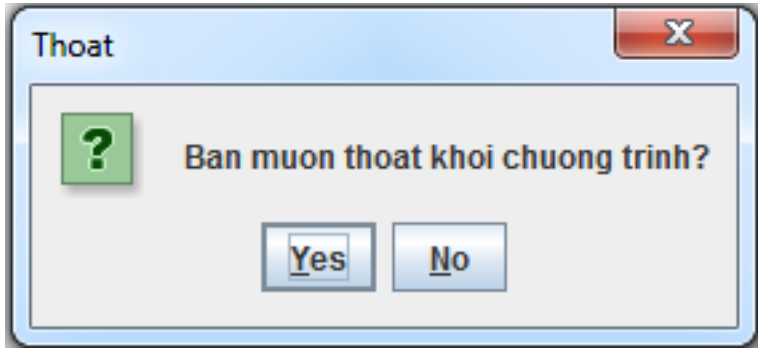
```
import java.awt.*;
import javax.swing.*;
public class AboutDialog extends JDialog {
  public AboutDialog(JFrame parent, String title, String message) {
    super(parent, title, true);
    setSize(250,100);
    setLocationRelativeTo(null);
    JPanel messagePane = new JPanel();
    messagePane.add(new JLabel(message));
    getContentPane().add(messagePane);
    JPanel buttonPane = new JPanel();
    JButton button = new JButton("OK");
    buttonPane.add(button);
    getContentPane().add(buttonPane, BorderLayout.SOUTH);
    setDefaultCloseOperation(DISPOSE_ON_CLOSE);
    setVisible(true);
  }
  public static void main(String[] a) {
    AboutDialog dlg = new AboutDialog(new JFrame(),
                                      "Thong bao", "Xin chao");
  }
}
```



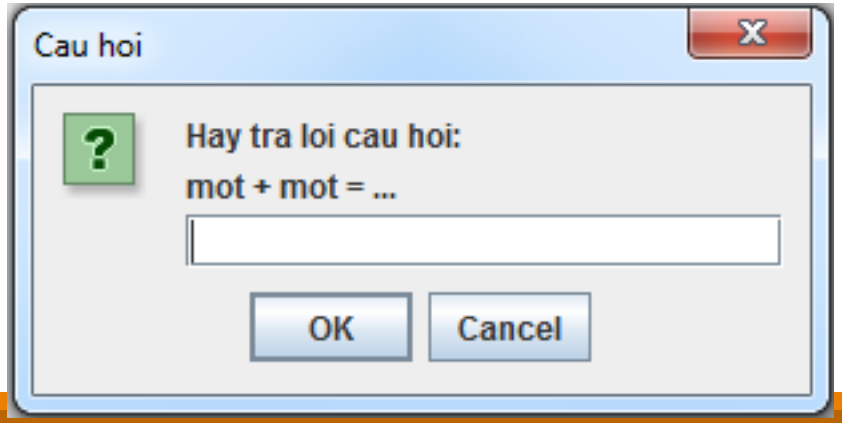
JDialog và JOptionPane




```
JOptionPane.showMessageDialog(null,
"Thao tac khong cho phep.","Bao loi",
JOptionPane.ERROR_MESSAGE);
```




```
int n = JOptionPane.showConfirmDialog
(this, "Ban muon thoat khoi chuong trinh?",
"Thoat", JOptionPane.YES_NO_OPTION);
```

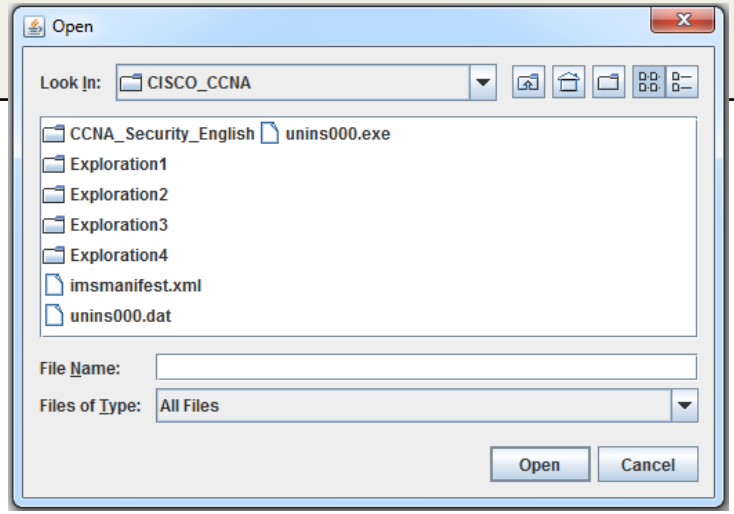



```
String str = JOptionPane.showInputDialog
(this, "Hay tra loi cau hoi:\n
        mot + mot = ...", "Cau hoi",
 JOptionPane.QUESTION_MESSAGE);
```

Lớp vật chứa cấp cao khác

- JFileChooser

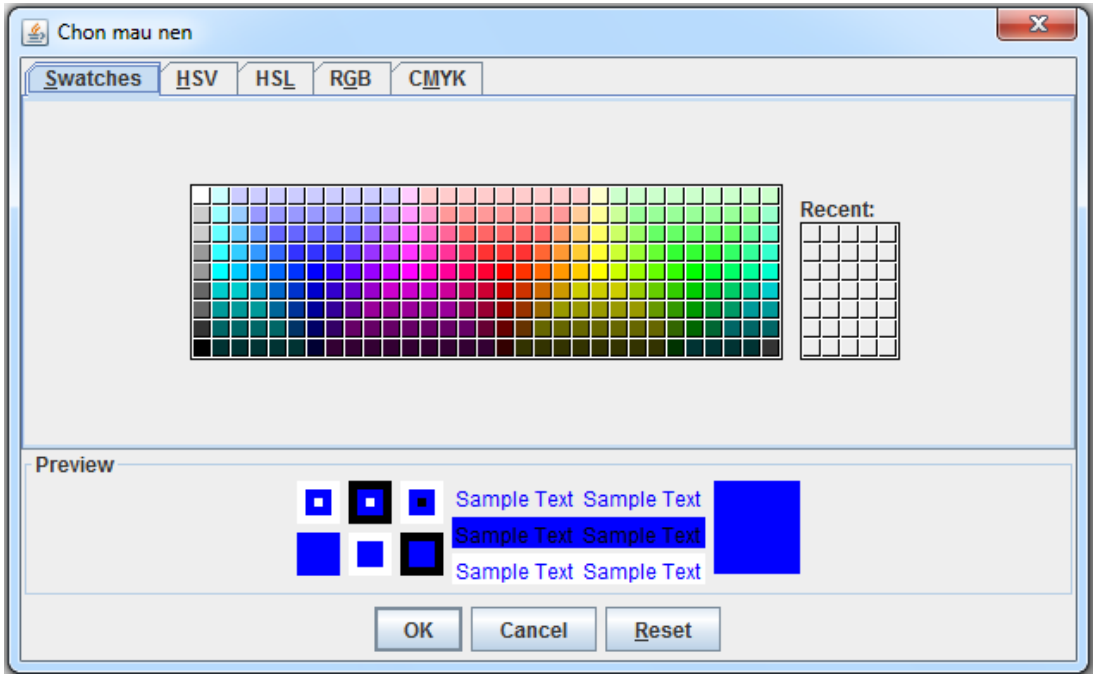
```
JFileChooser fileChooser = new JFileChooser();
fileChooser.setCurrentDirectory(new File("C:\\Cisco_CCNA"));
int result = fileChooser.showOpenDialog(this);
if (result == JFileChooser.APPROVE_OPTION) {
    File selectedFile = fileChooser.getSelectedFile();
    System.out.println("Ban da chon file: " +
selectedFile.getAbsolutePath());
}
```



Lớp vật chứa cấp cao khác

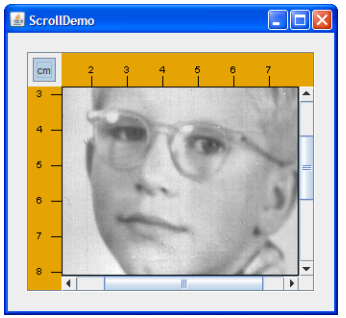
- JColorChooser

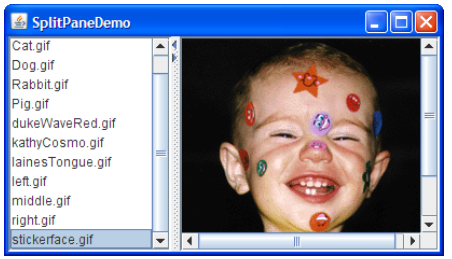
```
Color initcolor=Color.BLUE;
Color color=JColorChooser.showDialog(this,"Chon mau nen",initcolor);
frame.setBackground(color);
```

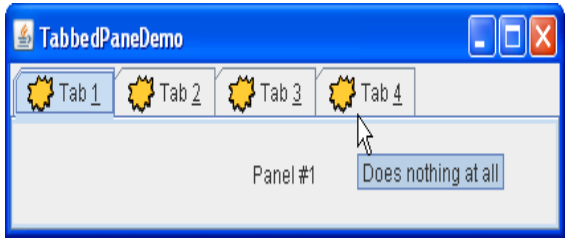


Lớp vật chứa bên trong

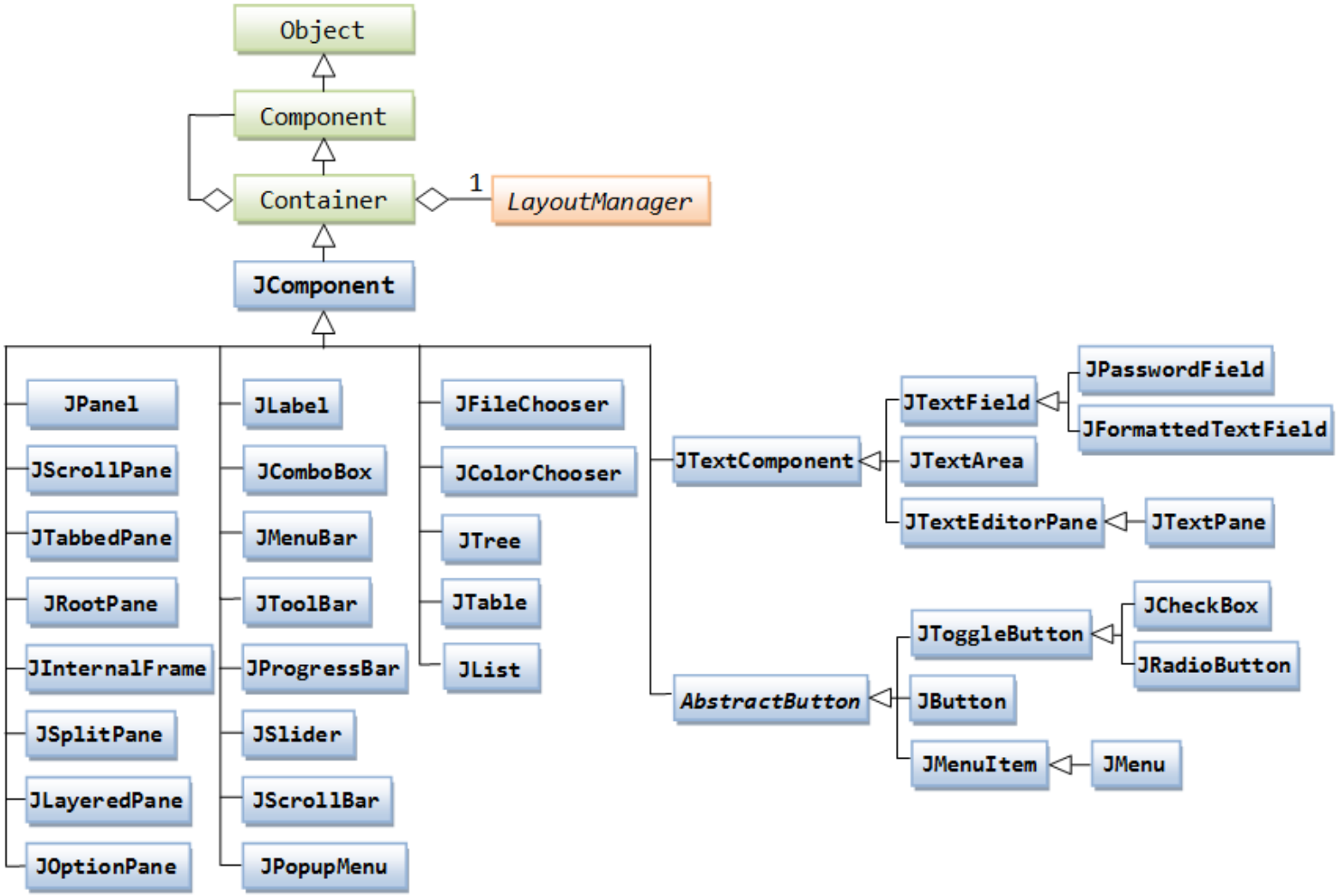
- Không phải các lớp vật chứa cấp cao
- Chứa các thành phần khác bên trong.
- Ví dụ:
  - JScrollPane
  - JSpitPane
  - JTabbedPane
  - JToolbar





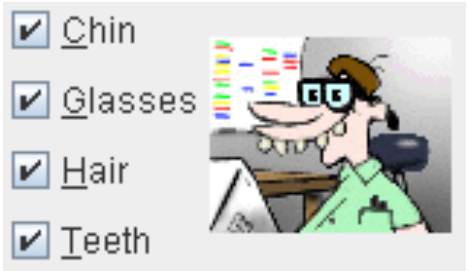


Các thành phần giao diện Swing



Các thành phần giao diện Swing

JLabel

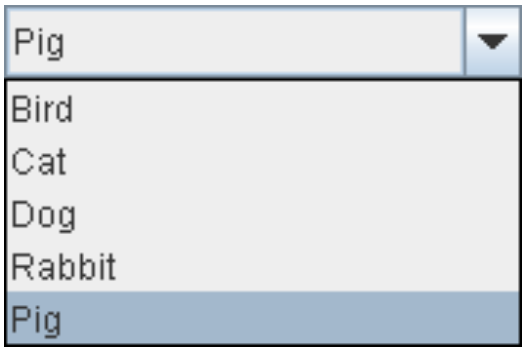


JCheckBox

JRadioButton

JTextField

JPasswordField



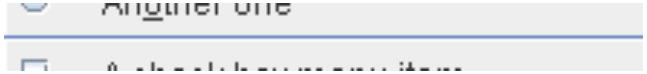
JComboBox

JList

Các thành phần giao diện Swing (tt)

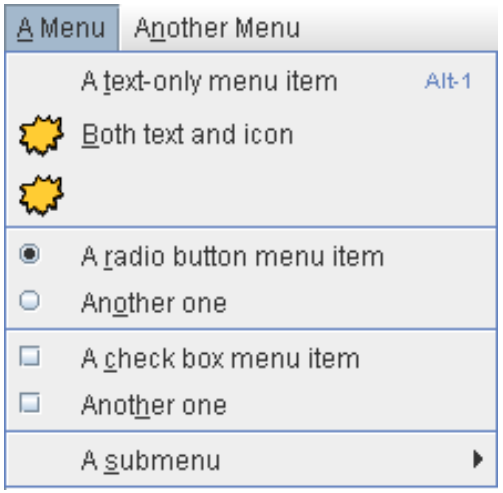
JButton

JProgressBar

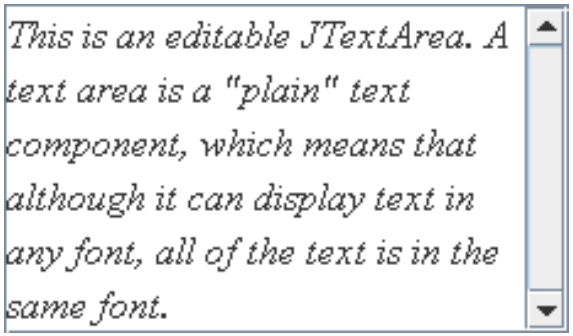


JSeparator

JSpinner



JMenu



JTextArea

JToolTip

JComponent

- Đa số các thành phần giao diện đều hỗ trợ:
  - Text và icon
  - Tạo phím tắt (shortcut)
  - Tool tips
  - Look and feel: giao diện hiển thị như trong hệ điều hành
- Một số hàm chung:
  - `public void setBackground(Color bgColor)`
  - `public void setForeground(Color fgcolor)`
  - `public void setFont(Font font)`
  - `public void setBorder(Border border)`
  - `public void setPreferredSize(Dimension dim)`
  - `public void setOpaque(boolean isOpaque)`
  - `public void setToolTipText(String toolTipMsg)`

JLabel

- Tạo 1 nhãn
  - `JLabel label1 = new JLabel("This is a basic label");`
  - `JLabel label2 = new JLabel(new ImageIcon("images/attention.jpg"));`
  - `JLabel label3 = new JLabel("A label with icon and text”, new ImageIcon("images/attention.jpg"),SwingConstants.CENTER);`
- Thêm 1 nhãn vào 1 container
  - `frame.add(label1);`
  - `dialog.add(label2);`
  - `panel.add(label3);`
- Hiệu chỉnh 1 nhãn
  - `label1.setFont(new java.awt.Font("Arial", Font.ITALIC, 16));`
  - `label2.setOpaque(true); // Có hiển thị màu nền`
  - `label.setForeground(Color.BLUE);`

JTextField

- Tạo 1 đối tượng TextField
  - `JTextField textField1 = new JTextField(”Day la text");`
  - `JTextField textField2 = new JTextField(20);`
  - `JTextField textField3 = new JTextField(”Day la text", 30);`
- Thêm 1 đối tượng TextField vào 1 container
  - `frame.add(textField1);`
  - `dialog.add(textField1);`
- Lấy và gán giá trị của TextField
  - `String content = textField2.getText();`
  - `textField2.setText(”Bo mon Mang MT & TT");`
- Gán Tooltip cho TextField
  - `textField2.setToolTipText(”Dien vao ten bo mon");`
- Gán Focus
  - `textField3.requestFocusInWindow();`

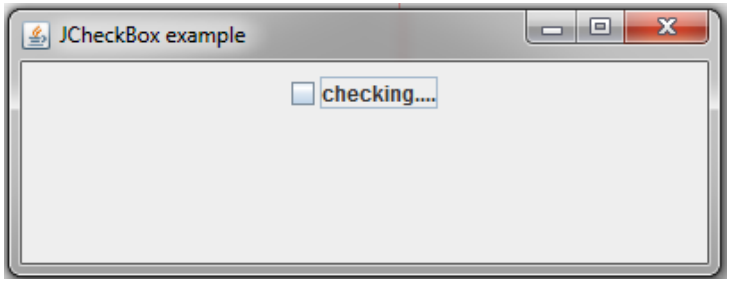
JButton

- Khởi tạo 1 nút bấm
  - `JButton button1 = new JButton("Edit");`
  - `JButton button2 = new JButton(new ImageIcon(”stop.gif"));`
  - `JButton button3 = new JButton("Start", new ImageIcon("images/start.gif"));`
- Thêm 1 nút bấm vào 1 container
  - `frame.add(button1);`
  - `dialog.add(button2);`
  - `panel.add(button3);`

- Hiệu chỉnh
  - `button1.setBackground(Color.YELLOW);`
- Cài đặt 1 phím nóng cho nút bấm
  - `button1.setMnemonic(KeyEvent.VK_E);  // Phim Alt+E`
- Cài đặt nút bấm mặc nhiên
  - `getRootPane().setDefaultButton(button1);`

JCheckBox

- Tạo 1 checkbox
  - `JCheckBox checkbox1 = new JCheckBox();`
  - `JCheckBox checkbox2 = new JCheckBox("Save Password");`
  - `JCheckBox checkbox3 = new JCheckBox("Save", true);`
- Thêm 1 checkbox vào 1 container
  - `frame.add(checkbox1);`
  - `panel.add(checkbox2);`
- Gán và lấy giá trị của checkbox
  - `checkbox.setSelected(true);`
  - `if (checkbox.isSelected()) { … }`



JRadioButton

- Tạo 1 RadioButton

```
JRadioButton optionLinux = new JRadioButton("Linux");
JRadioButton optionWin = new JRadioButton("Windows", true);
```

- Nhóm các RadioButton lại

```
ButtonGroup group = new ButtonGroup();
group.add(optionLinux);
group.add(optionWin);
```

- Thêm RadioButton vào 1 container

```
theFrame.add(optionLinux);
theFrame.add(optionWin);
```

- Gán và lấy giá trị lựa chọn của RadioButton

```
boolean isLinuxSelected = optionLinux.isSelected();
optionWin.setSelected(true);
```

JList

- Tạo kiểu danh sách: có thể chọn DefaultListModel

```
DefaultListModel<String> listModel = new
                                    DefaultListModel<>();
```

- Thêm các thành phần vào ListModel

```
listModel.addElement(”BM Mang MT & TT");
listModel.addElement(”BM CNTT");
```

- Tạo 1 JList sử dụng kiểu danh sách đã tạo phía trên

```
JList<String> dsbm= new JList<>(listModel);
```

- Thêm thanh trượt (Scroll) vào frame

```
add(new JScrollPane(dsbm));
```

- Lấy giá trị lựa chọn 1 thành phần trong list

```
String selectedValuesList = dsbm.getSelectedValuesList();
```

Ví dụ về JList

```
import javax.swing.*;
public class JListExample extends JFrame {
    private JList<String> dsbm;
    public JListExample() {
        DefaultListModel<String> listModel = new DefaultListModel<>();
        listModel.addElement("BM Mang may tinh & TT");
        listModel.addElement("BM Cong Nghe Thong Tin");
        listModel.addElement("BM He Thong Thong Tin");
        listModel.addElement("BM Khoa Hoc May Tinh");
        listModel.addElement("BM Cong Nghe Phan Mem");
        listModel.addElement("BM Tin Hoc Ung Dung");
        dsbm = new JList<>(listModel);
        this.add(dsbm);
        this.add(new JScrollPane(dsbm));
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        this.setTitle("Khoa Cong Nghe Thong Tin & TT");
        this.setSize(300,200);
    }
    public static void main(String[] args) {
                    JListExample jle = new JListExample();
                    jle.setVisible(true);jle.setLocationRelativeTo(null);
    }
}
```

JTextArea

- Tương tự như JTextField nhưng có thể có nhiều dòng.

```
textArea = new JTextArea(5, 20);
```

- Có thể cho phép người dùng chỉnh sửa nội dung

```
textArea.setEditable(false);
```

- Có thể thêm các thanh cuộn

```
JScrollPane scrollPane = new JScrollPane(textArea);
```

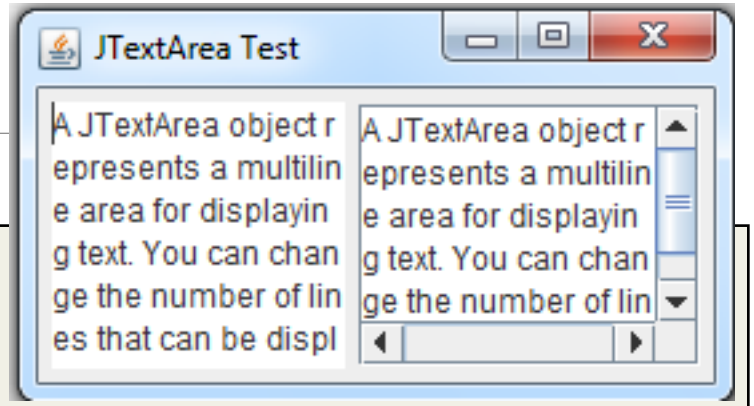
- Có thể nối thêm text vào nội dung của TextArea

```
textArea.append(text + newline);
```

- Có thể cài đặt nội dung text sẽ tự động xuống hàng khi vượt quá chiều dài của ô.

```
textArea.setLineWrap(true);
```

Ví dụ về JTextArea




```
public class JTextAreaTest {
  public static void main(String[] args) {
    JFrame frame = new JFrame("JTextArea Test");
    frame.setLayout(new FlowLayout());
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    String text = "A JTextArea object represents a multiline area for displaying
text. ” + "You can change the number of lines that can be displayed at a time, "
        + "as well as the number of columns. You can wrap lines and words too. "
        + "You can also put your JTextArea in a JScrollPane to make it scrollable.";
    JTextArea textAreal = new JTextArea(text, 5, 10);
    textAreal.setPreferredSize(new Dimension(100, 100));
    JTextArea textArea2 = new JTextArea(text, 5, 10);
    textArea2.setPreferredSize(new Dimension(100, 100));
    JScrollPane scrollPane = new JScrollPane(textArea2,
                                             JScrollPane.VERTICAL_SCROLLBAR_ALWAYS,
                                             JScrollPane.HORIZONTAL_SCROLLBAR_ALWAYS);
    textAreal.setLineWrap(true);  textArea2.setLineWrap(true);
    frame.add(textAreal); frame.add(scrollPane);
    frame.pack();
    frame.setVisible(true);
  }
}
```

Frame sẽ resize vừa đủ cho các thành phần