

Chương 6: Lập trình GUI (tt)



ThS. Trần Văn Hữu

❖ Giới thiệu về Swing

- Những đặc điểm của Swing
- Các thành phần GUI thông dụng của Swing
 - JFrame, JPanel, JScrollPane
 - Border, Border Title
 - JLabel, JTextField, JTextArea, JButton
 - JCheckBox, JRadioButton - ButtonGroup
- Các kỹ thuật gắn sự kiện.



Giới thiệu Swing

❖ AWT (Abstract Windows Toolkit)

- Java 1
- Được gắn với platform xác định
- Thích hợp với việc phát triển các ứng dụng GUI đơn giản.

❖ Swing

- Java 2
- Không gắn với platform cố định
- Mạnh, đa năng, linh hoạt

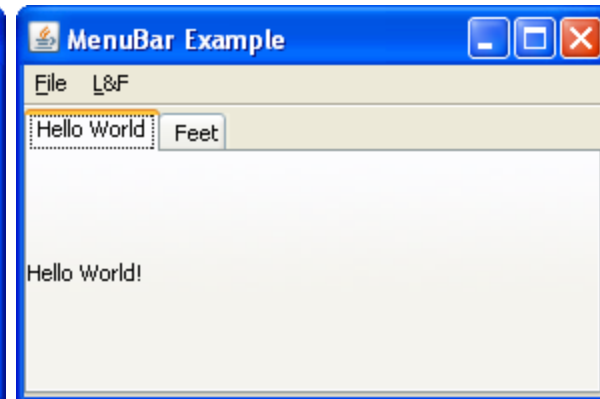


Giới thiệu Swing

- ❖ Là một gói nằm trong thư viện JFC (Java Eoundation Classes), gồm nhiều classes và interfaces hỗ trợ mạnh mẽ cho việc lập trình giao diện đồ họa trên JDK.
- ❖ JFC là được thiết kế để hỗ trợ lập trình viên tạo ra các ứng dụng **enterprise**.

Những đặc điểm của Swing

- ❖ Cảm quan (Look & Feels)
 - Metal (mặc định), Motif và Windows.





Những đặc điểm của Swing

- ❖ Sử dụng kiến trúc **Model - View - Controller (MVC)**.
- ❖ Có nhiều những thành phần mới
 - **Table, Tree, Slider, Progress Bar, Spinner, Internal frame và Text**
- ❖ Tạo **Tooltip** với các component
- ❖ Tạo các sự kiện bàn phím cho các component.



AWT & Swing

❖ Thiết kế UI

- Sử dụng các Jcomponent trong Swing như: JFrame, JPanel, JLabel, JButton, JTextField, ...
- Sử dụng các layout trong AWT như: FlowLayout, BorderLayout, CardLayout, ...
- Sử dụng các lớp trợ giúp trong AWT như: Graphics, Color, Font, FontMetrics,
- Sử dụng các lớp sự kiện trong AWT.

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





JComponent

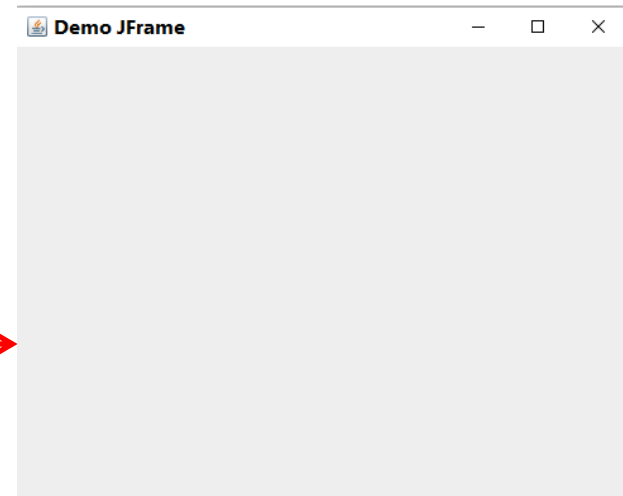
❖ Một số thành phần GUI thường dùng:

- JFrame, JPanel, JScrollPane
- JButton, JLabel
- JTextArea
- JCheckbox
- JRadioButton
- JList
- JComboBox
- ...

JFrame

```
public class MyWindow extends JFrame {
    public MyWindow(String titre) {
        super(titre);
    }
    public void showWindow()
    {
        setSize(new Dimension(500, 400));
        setDefaultCloseOperation(EXIT_ON_CLOSE);
        setLocationRelativeTo(null);
        setVisible(true);
    }
}

public class TestMyWindow {
    public static void main(String[] args) {
        MyWindow f = new MyWindow("Demo JFrame");
        f.showWindow();
    }
}
```





JFrame

```
super(tieuDe);
```

Tạo tiêu đề cho cửa sổ

```
setDefaultCloseOperation(EXIT_ON_CLOSE);
```

Cho phép click vào 'x' để đóng cửa sổ

```
setSize(new Dimension(500, 400));
```

Chỉ định Width = 500 và Height = 300

```
setLocationRelativeTo(null);
```

Hiển thị cửa sổ ở giữa màn hình

```
setVisible(true);
```

Hiển thị cửa sổ

JPanel

```
public class BorderLayoutUI extends JFrame {  
    public BorderLayoutUI(String tieuDe) {  
        super(tieuDe);  
        addControls();  
    }  
    private void addControls() {  
        JPanel pnMain = new JPanel();  
        pnMain.setLayout(new BorderLayout());  
        this.add(pnMain);  
  
        JPanel pnNorth = new JPanel();  
        pnNorth.setBackground(Color.BLUE);  
        pnNorth.setPreferredSize(new Dimension(0, 100));  
        pnMain.add(pnNorth, BorderLayout.NORTH);  
        ...  
    }  
    ...  
}
```



JPanel

```
JPanel pnMain = new JPanel();
```

Tạo mới JPanel

```
pnMain.setLayout(new BorderLayout());
```

Chỉ định dạng hiển thị cho Panel

```
pnNorth.setBackground(Color.BLUE);
```

Chỉ định màu nền cho Panel

```
pnNorth.setPreferredSize(new Dimension(0, 100));
```

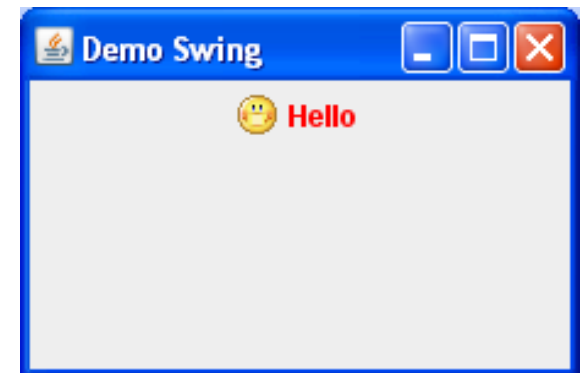
Chỉ định kích thước cho Panel

```
pnMain.add(pnNorth, BorderLayout.NORTH);
```

Thêm pnNorth vào pnMain với ràng buộc hiển thị BorderLayout.NORTH

❖ JLabel

- JLabel (Icon img): Hiển thị icon.
- JLabel (String str): Hiển thị text.
- JLabel (String str, Icon img, int align): hiển thị text và icon.
 - **Alignment:** SwingConstants.*LEFT*, SwingConstants.*RIGHT*, SwingConstants.*CENTER*, ...
- lbl.setToolTipText("ToolTip");
- lbl.setDisplayedMnemonic('e');





JLabel

- Hiển thị văn bản và không cho phép chỉnh sửa

```
JLabel lbl = new JLabel("Đồng hồ");  
lbl.setForeground(Color.RED);
```



Đồng hồ

//Thêm Icon vào Label

```
ImageIcon icon = new ImageIcon("watch.png");  
lbl.setIcon(icon);
```



Đồng hồ



JTextComponent

- ❖ JTextComponent is the root class of all Swing text components.
 - JTextField
 - JTextArea
 - JPasswordField
 - JPasswordField p = new JPasswordField("123");
 - p.setEchoChar('a');

TextField

➤ Hiển thị và nhập dữ liệu

```
JLabel lblTen = new JLabel("Nhập tên: ");  
JTextField txtTen = new JTextField(15);  
pnMain.add(lblTen);  
pnMain.add(txtTen);
```

Nhập tên:

```
//Không cho phép chỉnh sửa  
txtTen.setEditable(false);
```



TextField

```
txtTen.setText("Hello Tèo");
```

Gán giá trị cho JTextField

```
String s = txtTen.getText();
```

Lấy giá trị của JTextField

```
//Chuyển đổi kiểu dữ liệu
```

```
int n = Integer.parseInt(s);
```

```
float f = Float.parseFloat(s);
```

```
double d = Double.parseDouble(s);
```

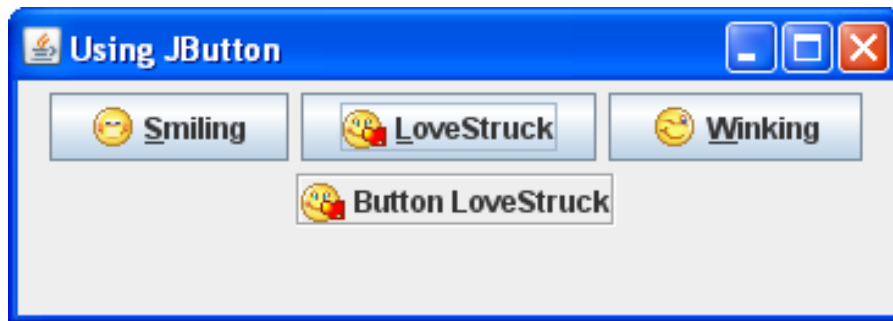
```
txtTen.requestFocus(true);
```

Di chuyển con trỏ văn bản vào JTextField

JButton

❖ JButton

- JButton(), JButton(Icon icon)
- JButton(String text)
- JButton(String text, Icon icon)
- b1.setMnemonic(KeyEvent.VK_F);





JButton

It is very important, attach event to do something that you want.

```
JButton btn=new JButton("Watch");
```

```
btn.setIcon(new ImageIcon("mywatch.png"));
```

```
btn.setMnemonic('W');
```



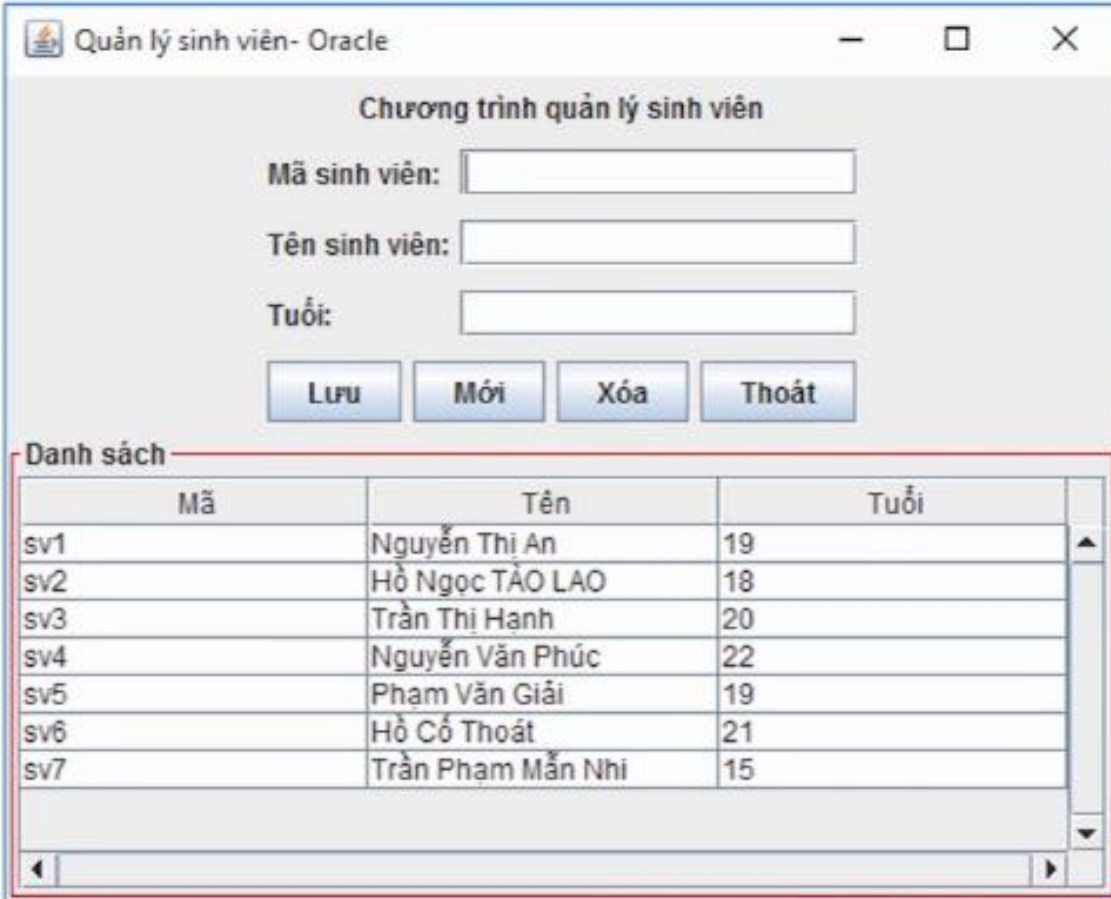
→ Alt+W to call btn command

```
btn.addActionListener(new ActionListener() {  
    public void actionPerformed(ActionEvent arg0) {  
        //do something here → coding here  
    }  
});
```

Add event for this button:

Border và Border Title

❖ Border, Border Title



Quản lý sinh viên- Oracle

Chương trình quản lý sinh viên

Mã sinh viên:

Tên sinh viên:

Tuổi:

Danh sách

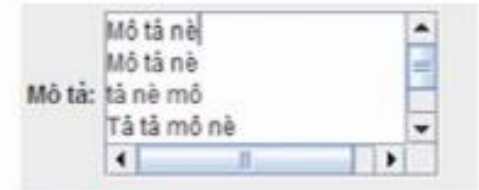
Mã	Tên	Tuổi
sv1	Nguyễn Thị An	19
sv2	Hồ Ngọc TÁO LAO	18
sv3	Trần Thị Hạnh	20
sv4	Nguyễn Văn Phúc	22
sv5	Phạm Văn Giải	19
sv6	Hồ Cổ Thoát	21
sv7	Trần Phạm Mẫn Nhi	15

Border và Border Title

❖ Border, Border Title

```
JPanel pnTable=new JPanel();  
Border border=  
    BorderFactory.createLineBorder(Color.RED);  
TitledBorder borderTitle=  
    BorderFactory.createTitledBorder(  
        border, "Danh sách");  
pnTable.setBorder(borderTitle);
```

JTextArea và JScrollPane



❖ JTextArea, JScrollPane

Input data multi line

```
JLabel lblDes=new JLabel("Mô tả:");  
JTextArea are=new JTextArea(5, 15);  
JScrollPane sc=new JScrollPane(are);  
add(lblDes);add(sc);
```

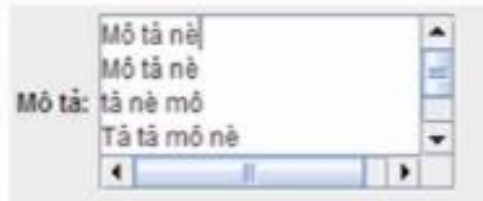
```
JTextArea are=new JTextArea(5, 15);
```

➔ 5 rows, 15 columns

We should use JScrollPane to create Scroll for JTextArea when user input data over limit row or column

JTextArea và JScrollPane

❖ JTextArea, JScrollPane



```
JScrollPane sc=new JScrollPane(are,  
JScrollPane.VERTICAL_SCROLLBAR_ALWAYS,  
JScrollPane.HORIZONTAL_SCROLLBAR_ALWAYS);
```




JTextArea và JScrollPane

```
ImageIcon img=new ImageIcon("baby.jpg");  
JLabel lblImg=new JLabel(img);  
JScrollPane scimg=new JScrollPane(lblImg,  
JScrollPane.VERTICAL_SCROLLBAR_ALWAYS,  
JScrollPane.HORIZONTAL_SCROLLBAR_ALWAYS);  
scimg.setPreferredSize(new Dimension(600, 500));  
add(scimg);
```



- ❖ JCheckBox class has the following constructors:
 - 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

❖ JCheckBox

Make multi choice

Môn học yêu thích:

<input checked="" type="checkbox"/> Java	<input type="checkbox"/> F Sharp
<input type="checkbox"/> C Sharp	<input checked="" type="checkbox"/> Ruby

```
JPanel pnCheck=new JPanel();
pnCheck.setLayout(new GridLayout(2, 2));
Border bor2=BorderFactory
    .createEtchedBorder(Color.BLUE, Color.RED);
TitledBorder titlebor2=
    new TitledBorder(bor2, "Môn học yêu thích:");
pnCheck.setBorder(titlebor2);
JCheckBox chk1=new JCheckBox("Java");
JCheckBox chk2=new JCheckBox("F Sharp");
JCheckBox chk3=new JCheckBox("C Sharp");
JCheckBox chk4=new JCheckBox("Ruby");
pnCheck.add(chk1);pnCheck.add(chk2);
pnCheck.add(chk3);pnCheck.add(chk4);
add(pnCheck);
```

JCheckBox

❖ JCheckBox

Make multi choice

Set grid layout 2 rows and 2 column

```
pnCheck.setLayout(new GridLayout(2, 2));
```

Create JCheckBox:

```
JCheckBox chk1=new JCheckBox("Java");
```

Add chk1 into the pnCheck:

```
pnCheck.add(chk1);
```

Add pnCheck into the Window:

```
add(pnCheck);
```

```
if(chk1.isSelected())  
{  
    //do something  
}
```

Create border with 2 color: Blue, Red:

```
BorderFactory.createEtchedBorder(Color.BLUE, Color.RED);
```



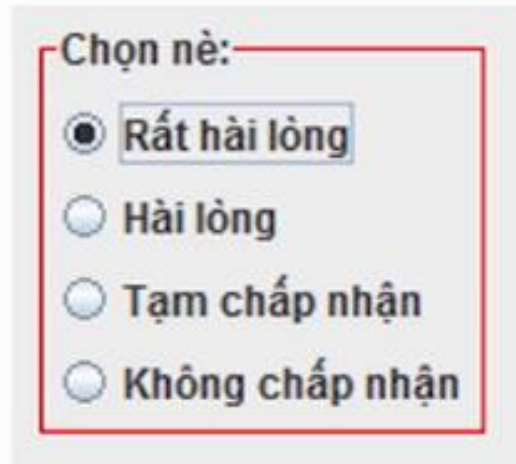
JRadioButton

- ❖ JRadioButton object can be created by using:
 - JRadioButton()
 - JRadioButton(Icon icon)
 - JRadioButton(Icon, boolean selected)
 - JRadioButton(String text)
 - JRadioButton(String text, boolean selected)
 - JRadioButton(String text, Icon icon)
 - JRadioButton(String text, Icon icon, boolean selected)

JRadioButton - ButtonGroup

❖ JRadioButton- ButtonGroup
Make single choice

Must add **JRadioButton** into the **ButtonGroup**



Chọn nè:

- ☒ Rất hài lòng
- ☐ Hài lòng
- ☐ Tạm chấp nhận
- ☐ Không chấp nhận

```
if(rad1.isSelected())  
{  
}
```




JRadioButton - ButtonGroup

❖ JRadioButton- ButtonGroup

```
JPanel pnGroup=new JPanel();
pnGroup.setLayout(new BorderLayout(pnGroup, BorderLayout.Y_AXIS));
Border bor=BorderFactory.createLineBorder(Color.RED);
TitledBorder titlebor=new TitledBorder(bor, "Chọn nè:");
pnGroup.setBorder(titlebor);
JRadioButton rad1=new JRadioButton("Rất hài lòng");
JRadioButton rad2=new JRadioButton("Hài lòng");
JRadioButton rad3=new JRadioButton("Tạm chấp nhận");
JRadioButton rad4=new JRadioButton("Không chấp nhận");
```

```
ButtonGroup group=new ButtonGroup();
group.add(rad1);group.add(rad2);
group.add(rad3);group.add(rad4);
pnGroup.add(rad1);pnGroup.add(rad2);
pnGroup.add(rad3);pnGroup.add(rad4);
add(pnGroup);
```



JRadioButton - ButtonGroup

❖ JRadioButton- ButtonGroup

Create Border with title:

```
Border bor=BorderFactory.createLineBorder(Color.RED);  
TitledBorder titlebor=new TitledBorder  
                                (bor, "Chọn nè:");  
pnGroup.setBorder(titlebor);
```

Define a buttongroup to add all radio:

```
ButtonGroup group=new ButtonGroup();  
group.add(rad1);
```

And add all Radio into the pnGroup:

```
pnGroup.add(rad1);
```

Add pnGroupd into the Window:

```
add(pnGroup);
```



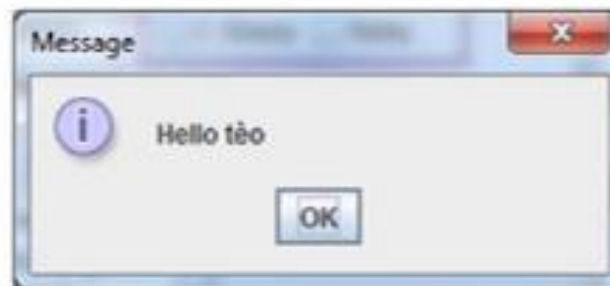

Các kỹ thuật xử lý sự kiện

- ❖ **Inline anonymous listener**
- ❖ **Listener in variable**
- ❖ **Listener class**

Các kỹ thuật xử lý sự kiện

➤ Inline anonymous listener

```
 JButton btn1=new JButton("Say Hello!");  
 btn1.addActionListener(new ActionListener()  
 {  
     public void actionPerformed(ActionEvent arg0)  
     {  
         JOptionPane.showMessageDialog(null, "Hello tèo");  
     }  
 });
```





Các kỹ thuật xử lý sự kiện

➤ Inline anonymous listener

```
btn1.addMouseListener(new MouseListener() {  
    public void mouseReleased(MouseEvent arg0) {  
    }  
    public void mousePressed(MouseEvent arg0) {  
    }  
    public void mouseExited(MouseEvent arg0) {  
    }  
    public void mouseEntered(MouseEvent arg0) {  
    }  
    public void mouseClicked(MouseEvent arg0) {  
    }  
});
```



Các kỹ thuật xử lý sự kiện

➤ Listener in variable

Hold a reference to the Listener in a variable

➔ could share event

```
ActionListener btnClick=new ActionListener()  
{  
    public void actionPerformed(ActionEvent arg0) {  
        JOptionPane.showMessageDialog(null, "Click!");  
    }  
};  
    btn1.addActionListener(btnClick);  
    btn2.addActionListener(btnClick);
```



Các kỹ thuật xử lý sự kiện

➤ Listener class

```
private class MyClick implements ActionListener
{
    @Override
    public void actionPerformed(ActionEvent arg0) {
        // TODO Auto-generated method stub

    }
}
```

```
btn1.addActionListener(new MyClick());
```

Could use class event anywhere

Bài tập 1

- ❖ Xây dựng giao diện chương trình giải phương trình bậc 2 như sau:



The screenshot shows a Windows application window titled "Giải Phương Trình Bậc 1". The window has a standard Windows title bar with minimize, maximize, and close buttons. The main content area has a light gray background. At the top, the title "Giải Phương Trình Bậc 1" is displayed in a large, bold, blue font. Below the title, there are two input fields. The first is labeled "Hệ số a:" and the second is labeled "Hệ số b:". Below these fields are three buttons: "Giải", "Thoát" (with a red power button icon), and "Help". At the bottom, there is a label "Kết quả:" followed by a wide text input field.

Giải Phương Trình Bậc 1

Hệ số a:

Hệ số b:

Kết quả:

Bài tập 2

❖ Xây dựng giao diện chương trình hiển thị thông tin khách hàng như sau:

The image shows two windows from a software application. The first window, titled "Thông tin khách hàng", contains a form for entering customer data. The second window, titled "Message", displays the entered information as a summary.

Thông tin khách hàng

Thông tin:

Nhập tên : Trần Thành Công

Thủ Dầu Một, Bình Dương

Địa chỉ :

Sở thích

☒ Đi bơi

☒ Đi xem phim

Giới tính

☒ Nam

☐ Nữ

OK

Message

i Trần Thành Công
Thủ Dầu Một, Bình Dương
Đi bơi
Đi xem phim
Nam

OK

Hỏi & đáp

