

LẬP TRÌNH JAVA

**CHƯƠNG 5: LẬP TRÌNH GIAO DIỆN
ĐỒ HỌA**

LẬP TRÌNH JAVA

Tiết 1: Tổng quan lập trình giao diện

Giới thiệu về JFC

- ✓ JFC (Java Foundation Class, thuộc gói javax.swing)
 - Một bộ các lớp cho phép tạo GUI và chấp nhận các nhập liệu của người dùng thông qua bàn phím và chuột.
 - Cung cấp các thành phần khác nhau để tạo GUI hiệu quả.
- ✓ Các thành phần JFC:
 - Vật chứa (**Container**)
 - Thành phần (**Component**)
 - Trình quản lý cách trình bày (**Layout manager**)
 - Đồ họa (**Graphic**) và các tính năng vẽ (**draw**)



Giới thiệu về JFC

✓ Container:

- Vùng chứa các thành phần giao diện.
- Ké thừa từ lớp Container.
- Là một cửa sổ.

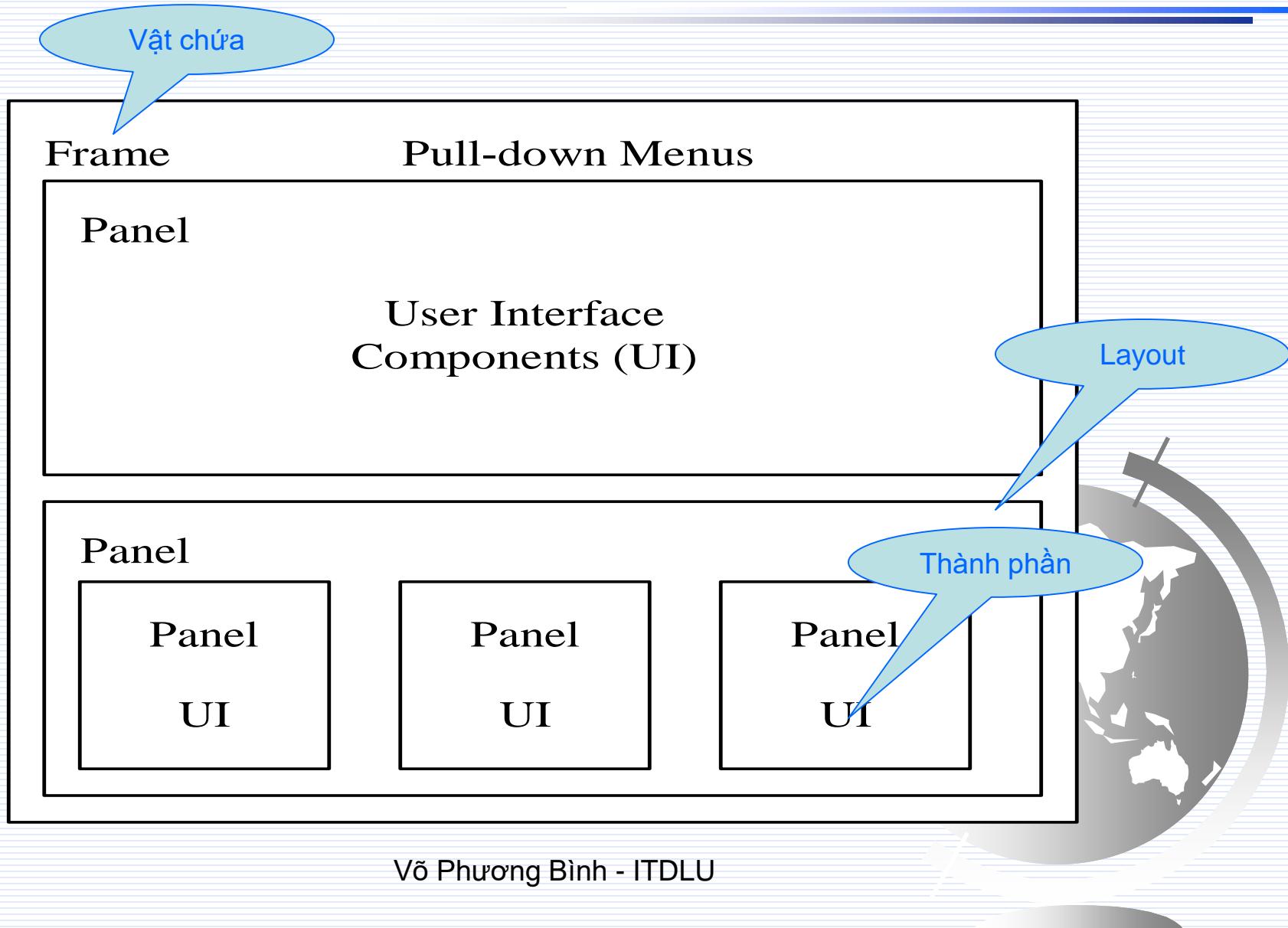
✓ Lớp Container nằm trong gói javax.swing

✓ Frame và Panel là các Container thường được sử dụng.

- Frame là cửa sổ độc lập.
- Panel là vùng nằm trong cửa sổ khác.

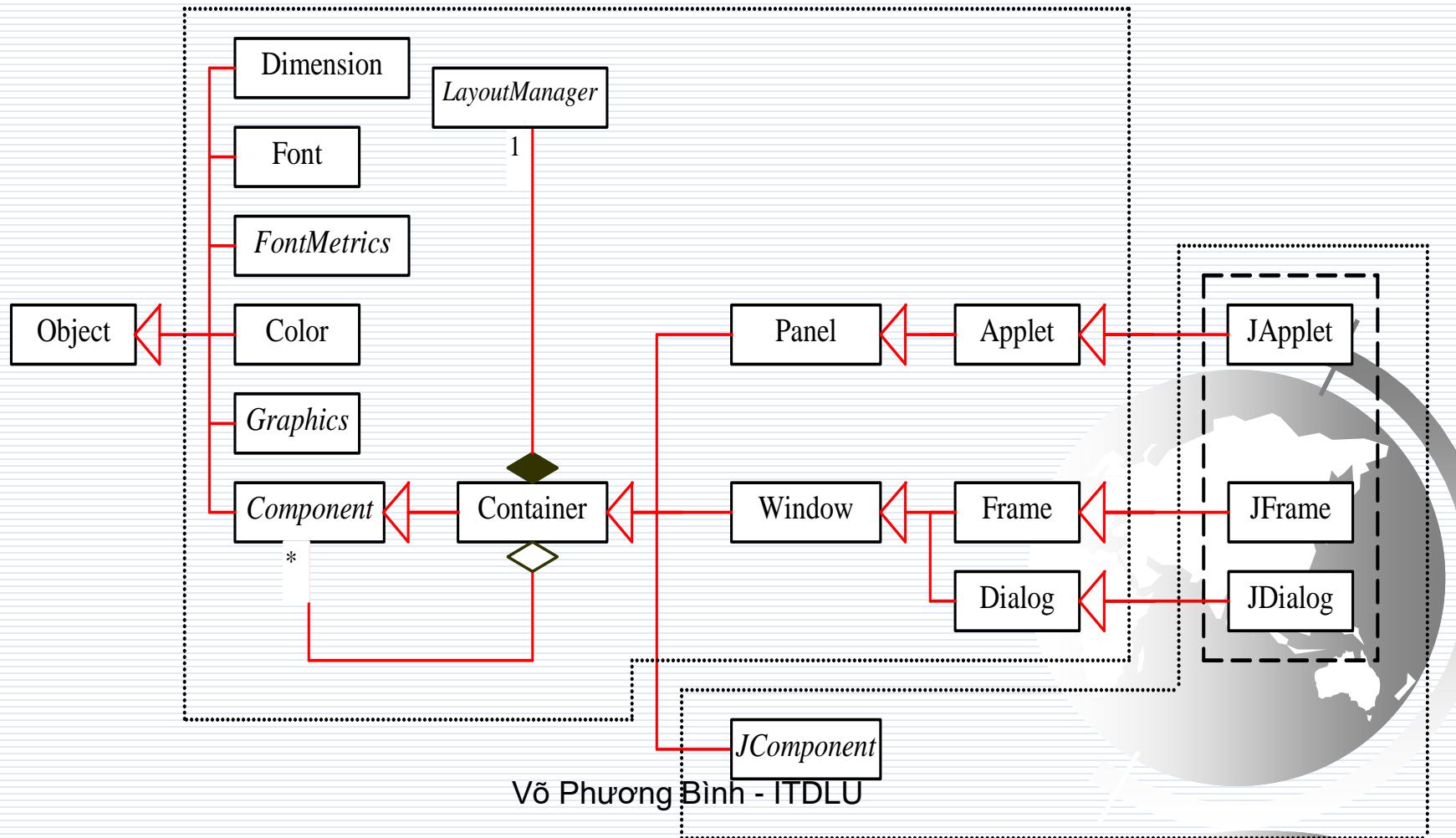


Giới thiệu về JFC

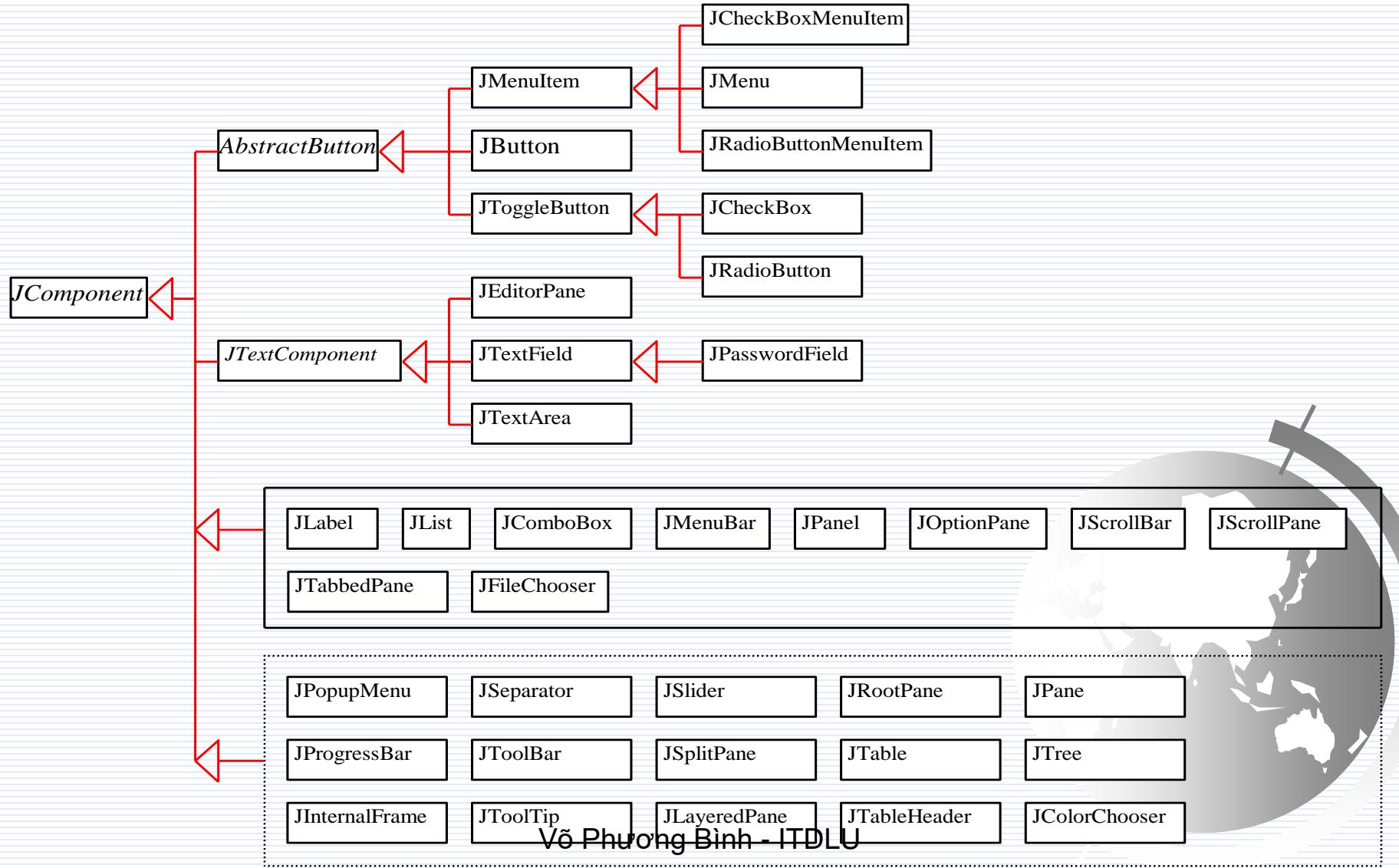


Giới thiệu về JFC

- ✓ Gói Swing (`javax.swing`) chứa các lớp, giao diện và các gói khác:



Giới thiệu về JFC



Giới thiệu về JFC

```
1. import javax.swing.*;  
2. class FrameDemo extends JFrame{  
3.     public FrameDemo(String title){  
4.         super(title);  
5.     }  
6.     public static void main(String args[]){  
7.         FrameDemo f=new FrameDemo("My Frame");  
8.         f.setSize(300,200);  
9.         f.setVisible(true);  
10.    }  
11. }
```



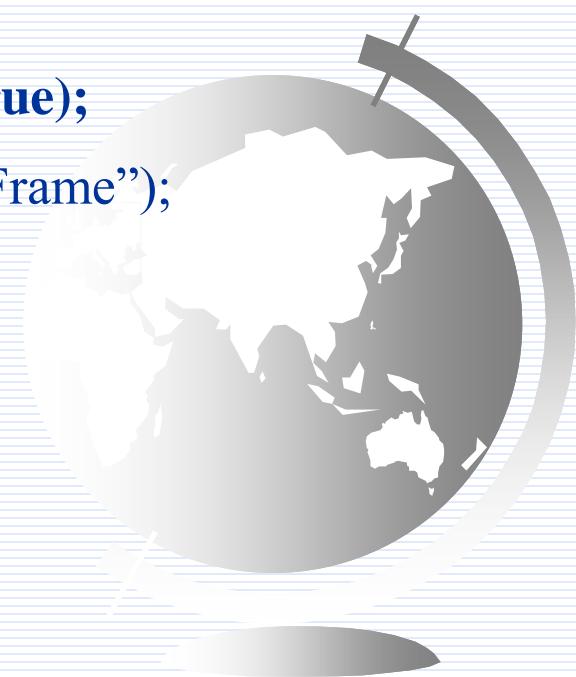
Giới thiệu về JFC



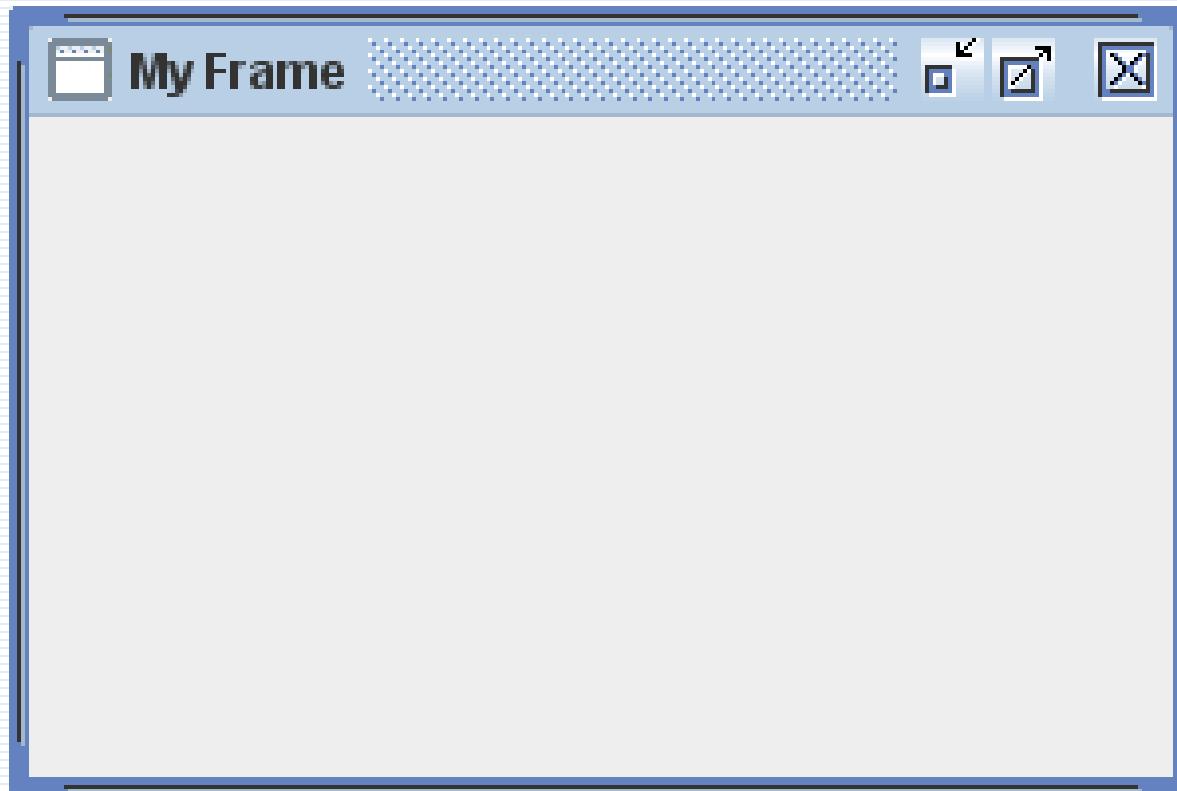
Võ Phương Bình - ITDLU

Đổi giao diện JFrame

```
1. import javax.swing.*;  
2. class FrameDemo extends JFrame{  
3.     public FrameDemo(String title){  
4.         super(title);  
5.     }  
6.     public static void main(String args[]){  
7.         JFrame.setDefaultLookAndFeelDecorated(true);  
8.         FrameDemo f=new FrameDemo("My Frame");  
9.         f.setSize(300,200);  
10.        f.setVisible(true);  
11.    }  
12. }
```



Đổi giao diện JFrame

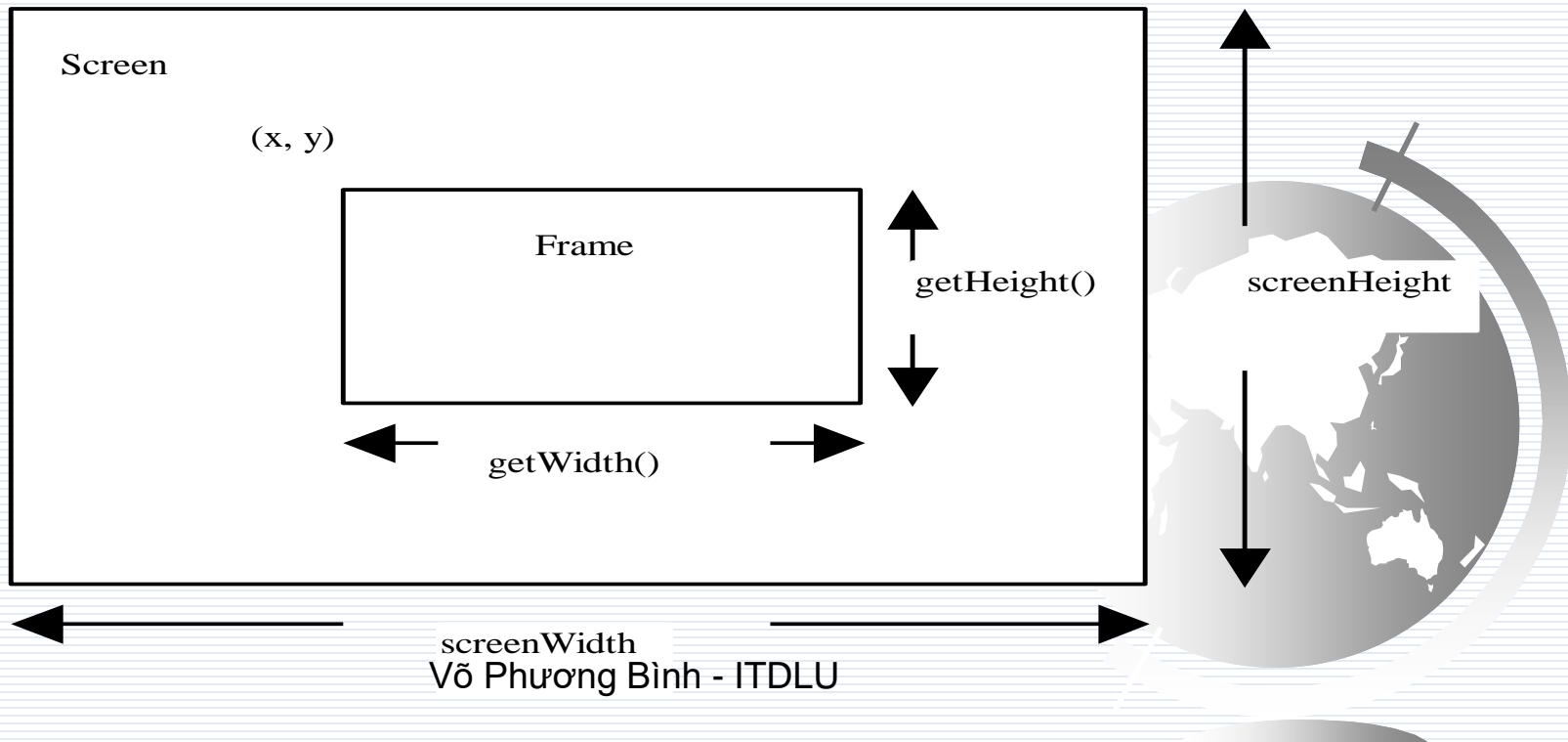


Võ Phương Bình - ITDLU

Giới thiệu về JFC

- Mặc định, Frame hiển thị tại góc left-top
- Để hiển thị tại vị trí có tọa độ (x, y) , ta dùng phương thức `setLocation(x, y)` của lớp `JFrame`

$(0, 0)$



Giới thiệu về JFC

➤ Lấy kích thước màn hình

➤ Dimension d =

```
Toolkit.getDefaultToolkit().getScreenSize();
```

➤ *d.height;*

➤ *d.width;*

➤ *Canh giữa màn hình*

➤ *setLocation(x,y)?*

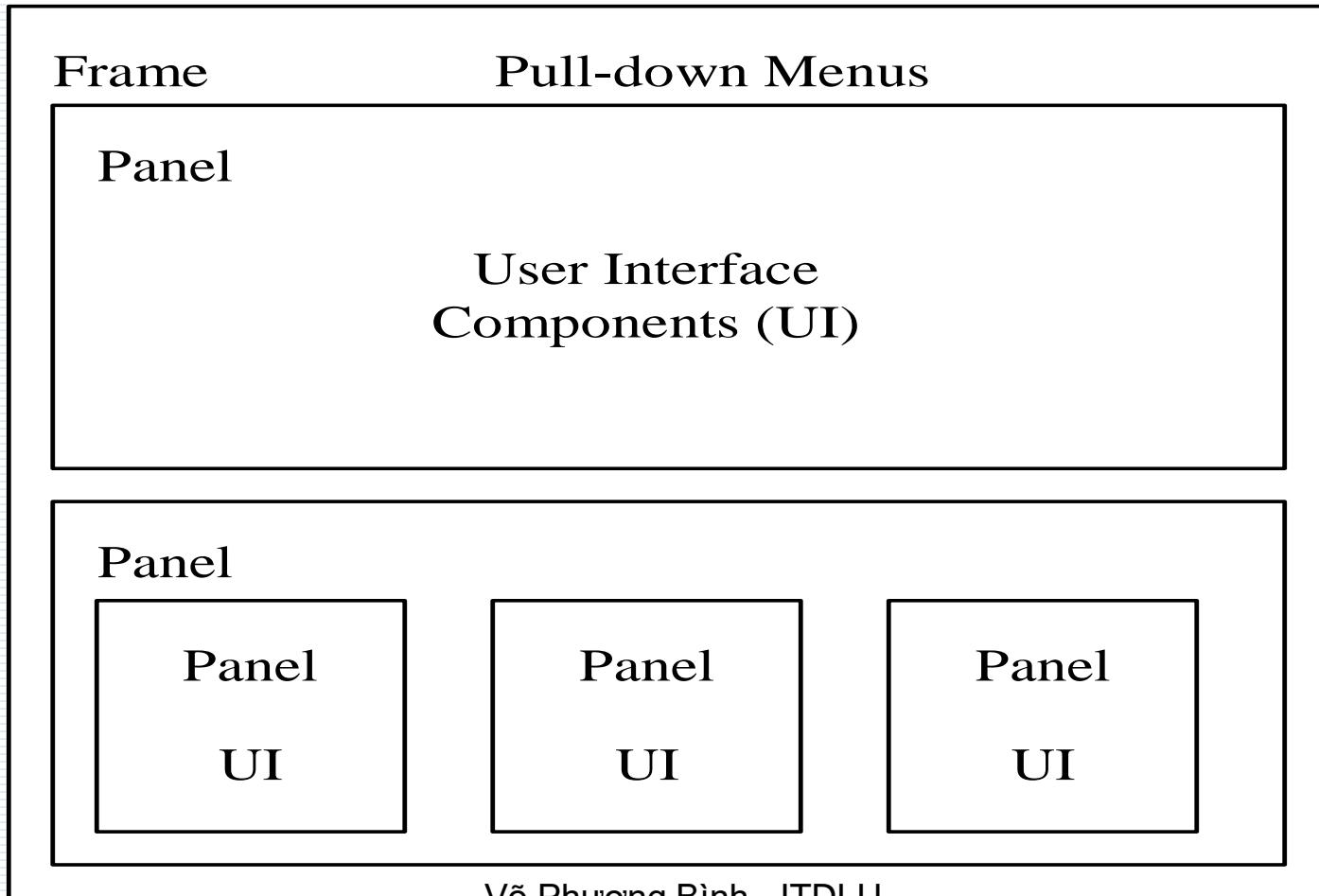


LẬP TRÌNH JAVA

Tiết 2: Sử dụng Container JPanel, Dialog

Sử dụng Panel

- ✓ Panel: nhóm các thành phần lại với nhau.



Ví dụ không sử dụng Panel

```
1. import javax.swing.*;  
2. class FrameDemo extends JFrame{  
3.     JButton button = new JButton("A");  
4.     public FrameDemo(String title){  
5.         super(title);  
6.         add(button);  
7.     }  
8.     public static void main(String args[]){  
9.         FrameDemo f=new FrameDemo("My Frame");  
10.        f.setSize(300,200);  
11.        f.setVisible(true);  
12.    }  
13. }
```



Ví dụ không sử dụng Panel



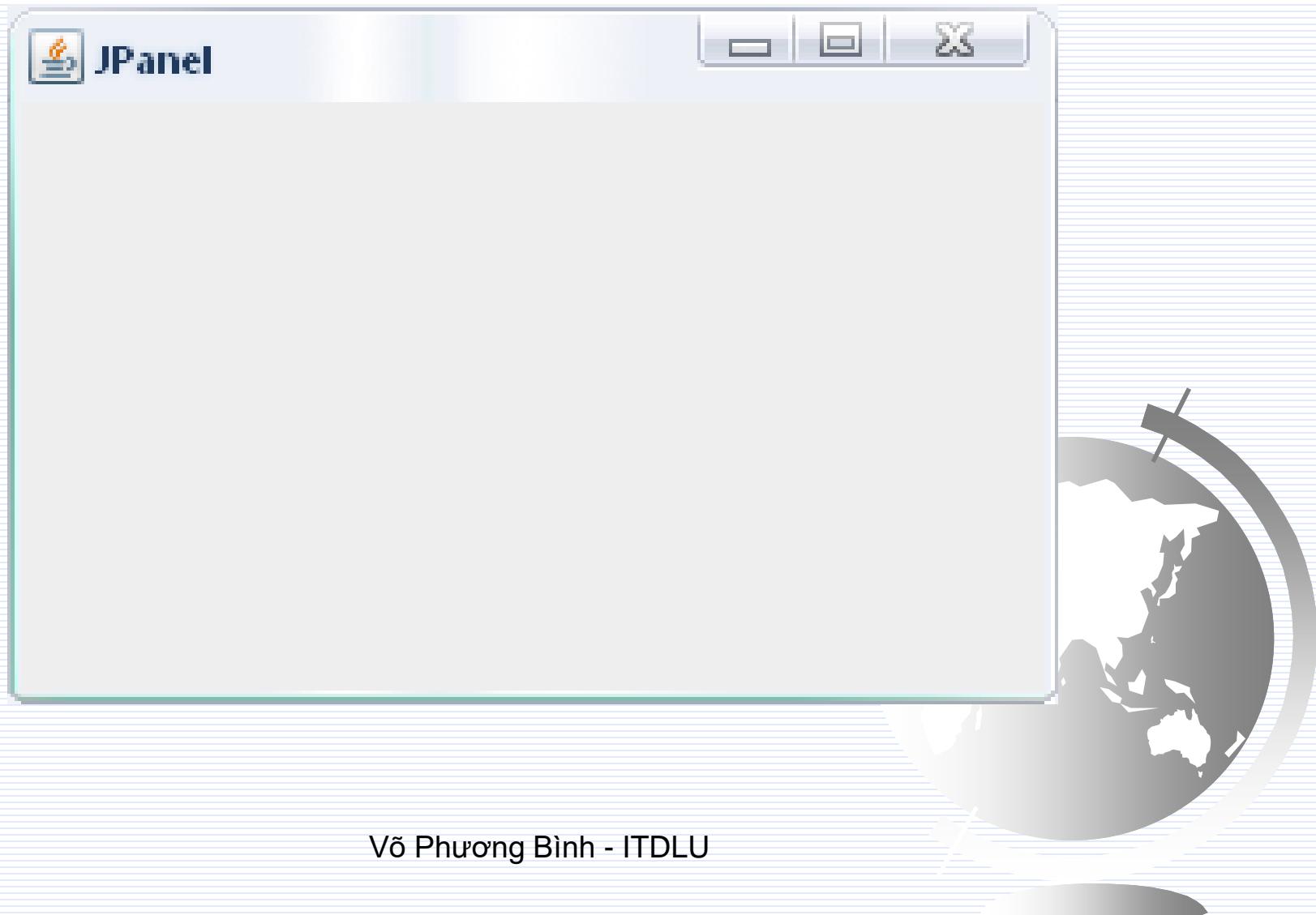
Võ Phương Bình - ITDLU

Cách sử dụng Panel

```
1. import javax.swing.*;  
2. class Paneltest extends JPanel{  
3.     public static void main(String args[]){  
4.         Paneltest p=new Paneltest();  
5.         JFrame f=new JFrame("JPanel");  
6.         f.add(p);  
7.         f.setSize(300,200);  
8.         f.setVisible(true);  
9.     }  
10. }
```



Cách sử dụng Panel



Võ Phương Bình - ITDLU

Ví dụ sử dụng Panel

```
1. import javax.swing.*;  
2. class Paneltest extends JPanel{  
3.     public static void main(String args[]){  
4.         Paneltest p=new Paneltest();  
5.         JButton button = new JButton("A");  
6.         p.add(button);  
7.         JFrame f=new JFrame("JPanel");  
8.         f.add(p);  
9.         f.setSize(300,200);  
10.        f.setVisible(true);  
11.    }  
12. }
```



Sử dụng Panel



Võ Phương Bình - ITDLU

Sử dụng Dialog

- ✓ Lớp JDialog tương tự như lớp JFrame.
- ✓ Đối tượng Dialog được tạo như sau:
 - JDialog dialog = new JDialog();
 - dialog.setSize(int width, int height);
 - dialog.show();
- ✓ Dialog có tham số Frame:
 - JDialog dialog = new JDialog(JFrame f, String title, boolean modal);



Ví dụ sử dụng Dialog

✓ import javax.swing.*;

✓ class DialogEx {

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

```
        JDialog d = new JDialog();
```

```
        d.setTitle("My Dialog");
```

```
        d.setSize(300, 200);
```

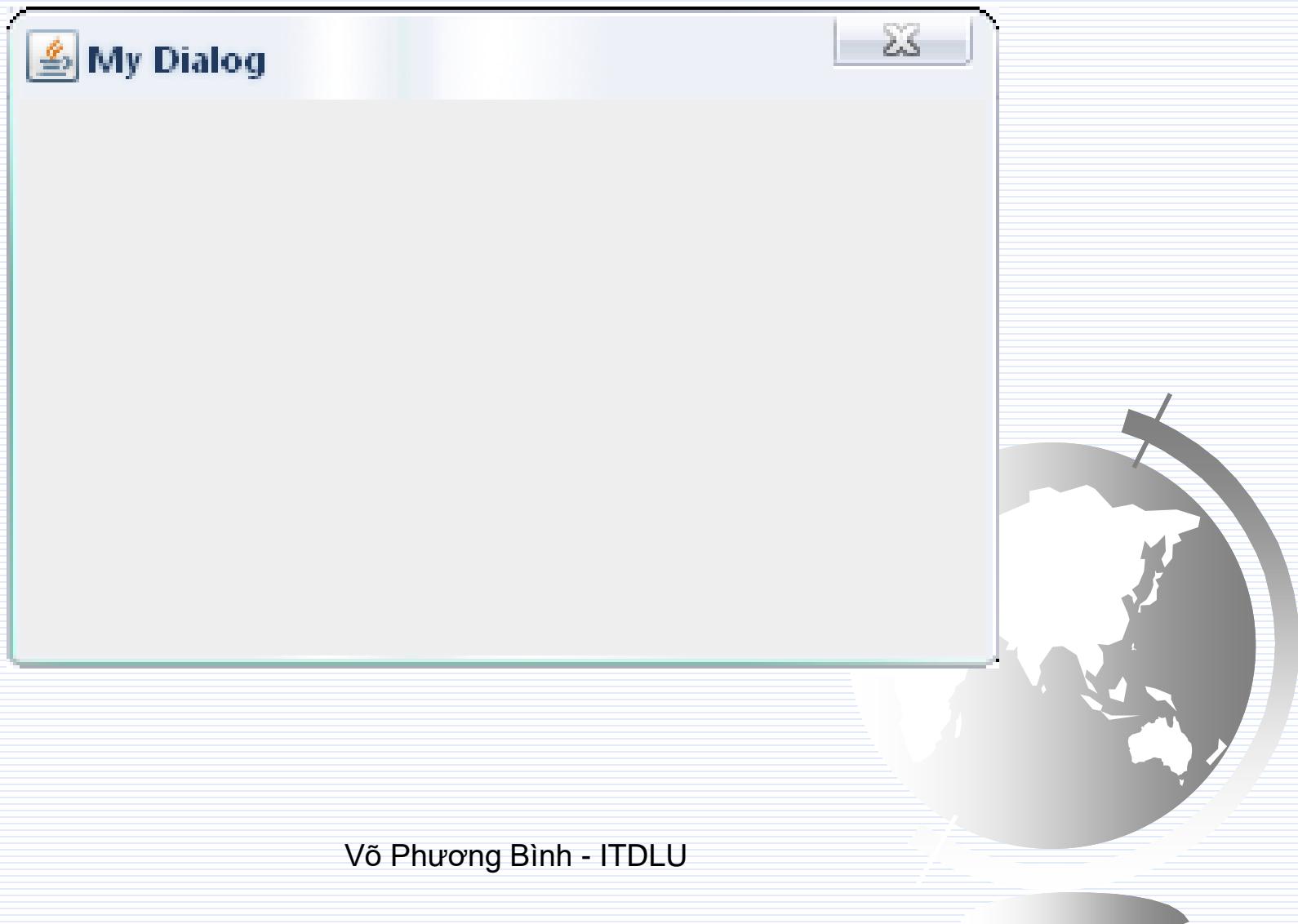
```
        d.show();
```

```
}
```

✓ }



Ví dụ sử dụng Dialog



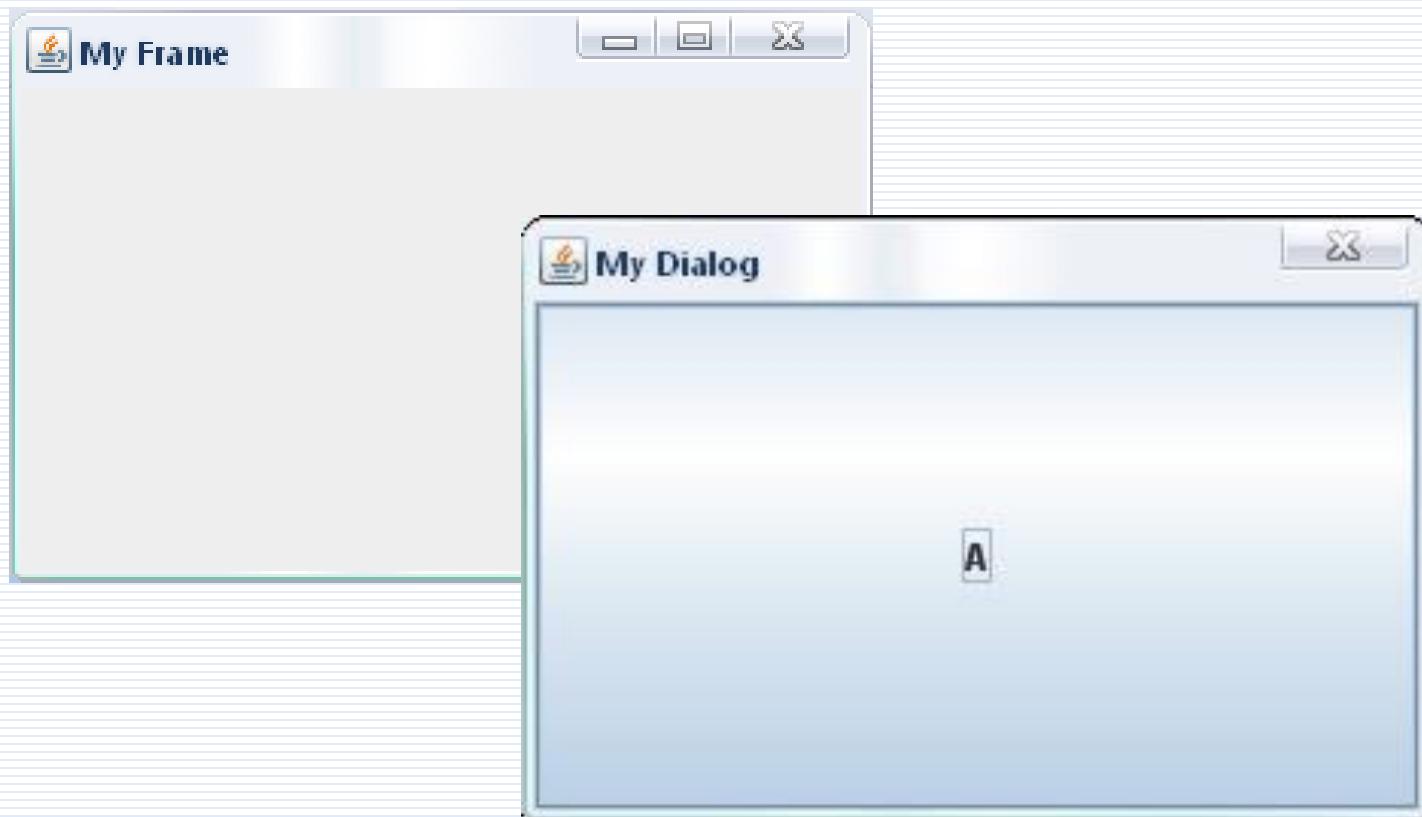
Võ Phương Bình - ITDLU

Ví dụ sử dụng Dialog

```
✓ class DialogEx {  
✓     public static void main(String args[]){  
✓         JFrame f=new JFrame ("My Frame");  
✓         f.setSize(300, 200);  
✓         f.show();  
✓         JDialog dialog = new JDialog(f,"My Dialog",true);  
✓         dialog.setSize(300, 200);  
✓         dialog.add(new JButton("A"));  
✓         dialog.show();  
✓     }  
✓ }
```



Ví dụ sử dụng Dialog

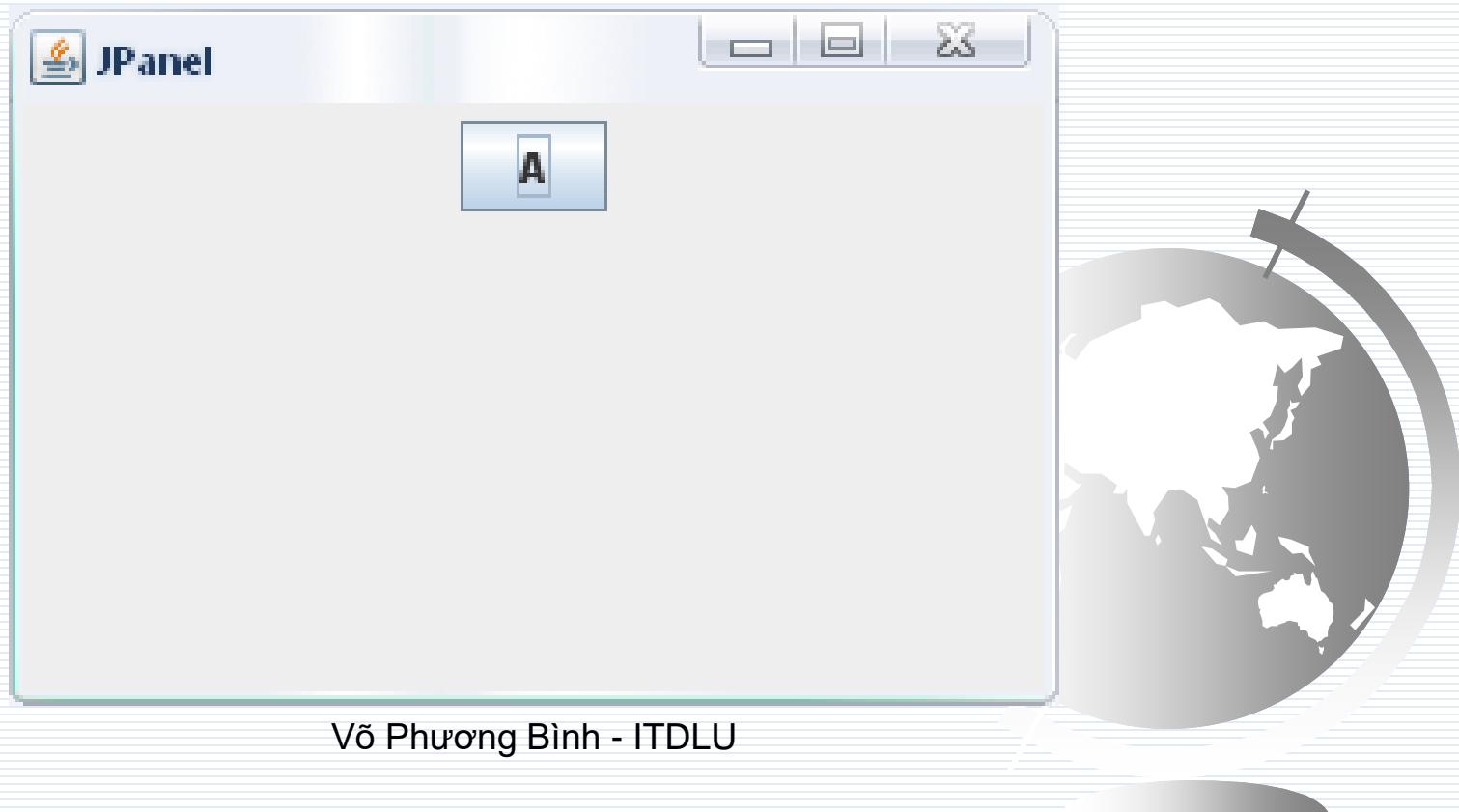


LẬP TRÌNH JAVA

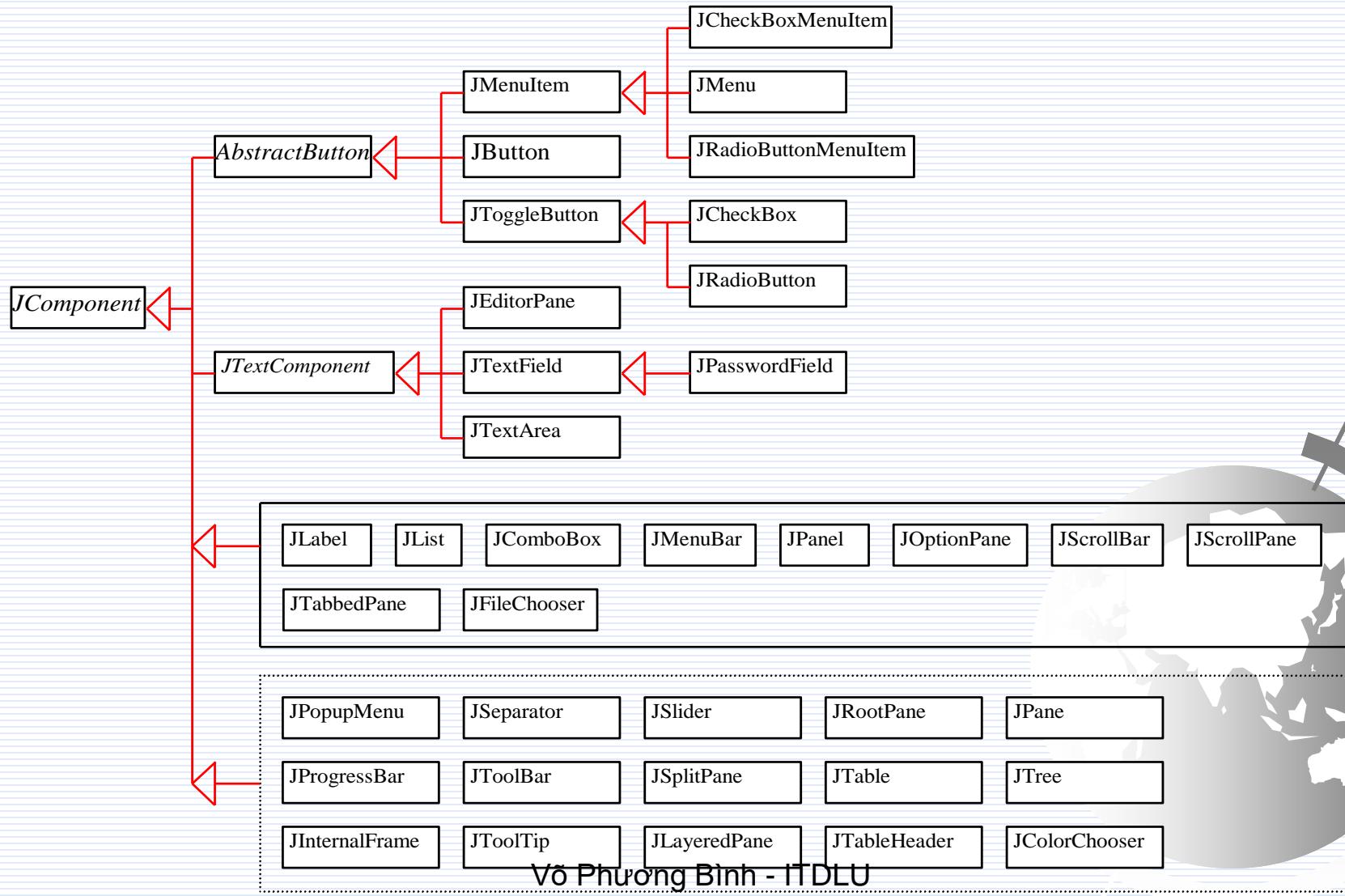
Tiết 3: Các thành phần giao diện, **JLabel**,
JTextField

Các thành phần giao diện (Component)

- ✓ Các thành phần (Component) được chứa trong Container (Frame, Panel).
- ✓ Ví dụ: các thành phần Button, Label, TextField, ...



Nhắc lại JComponent



JLabel

- ✓ Lớp JLabel được sử dụng để trình bày một chuỗi.
- ✓ Các constructor tạo một Label:
 - **JLabel()**
 - Tạo một Label trống.
 - **JLabel(String labelText)**
 - Tạo một Label với nội dung được cho.
 - **JLabel(String labelText, int alignment)**
 - Tạo một Label với một chế độ canh lè (alignment):
JLabel.LEFT, JLabel.RIGHT, JLabel.CENTER.

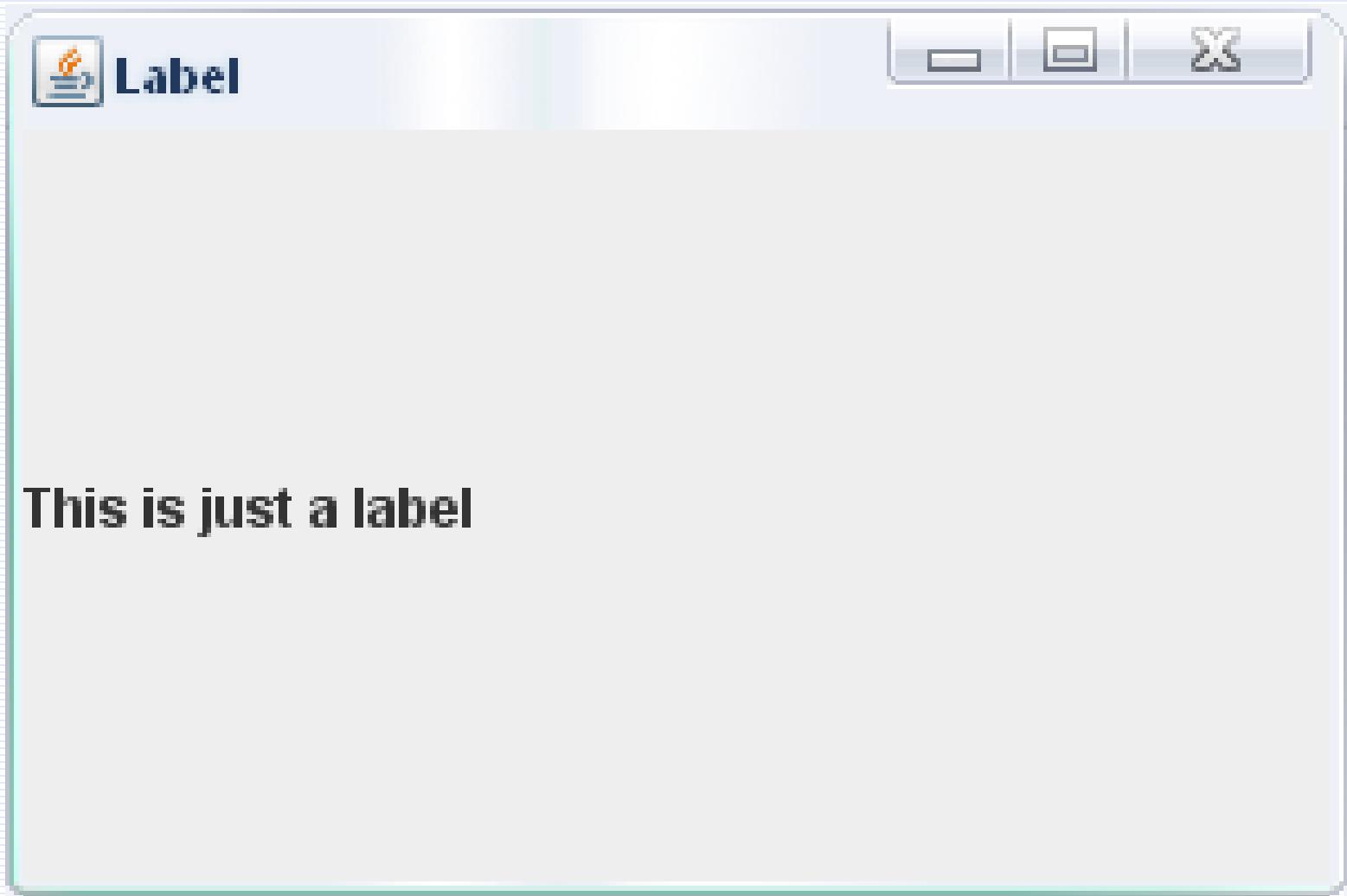


Ví dụ JLabel

```
1. import javax.swing.*;  
2. class Labeltest extends JFrame{  
3.     JLabel label1=new JLabel("This is just a label");  
4.     public Labeltest(String title){  
5.         super(title);  
6.         add(label1);  
7.     }  
8.     public static void main(String args[]){  
9.         Labeltest f=new Labeltest("Label");  
10.        f.setSize(300,200);  
11.        f.show();  
12.    }  
13. }
```



Ví dụ JLabel



- ✓ Các phương thức được sử dụng phổ biến của label được trình bày ở bảng bên dưới:

Phương thức	Chức năng
setFont(Font f)	Thay đổi phông chữ của Label
setText(String s)	Thiết lập nhãn cho Label
getText()	Lấy nội dung hiện tại của nhãn

JTextField

- ✓ JTextField là vùng chứa một dòng văn bản, trong đó văn bản có thể được hiển thị hay được nhập vào bởi người dùng.
- ✓ Các constructor tạo một JTextField:
 - JTextField(): Tạo một textfield mới.
 - JTextField(int columns): Tạo một textfield mới với số cột được cho trước.
 - JTextField(String s): Tạo một textfield mới với chuỗi văn bản được cho trước.
 - JTextField(String s, int columns): Tạo một textfield mới với nội dung và số cột được cho trước.

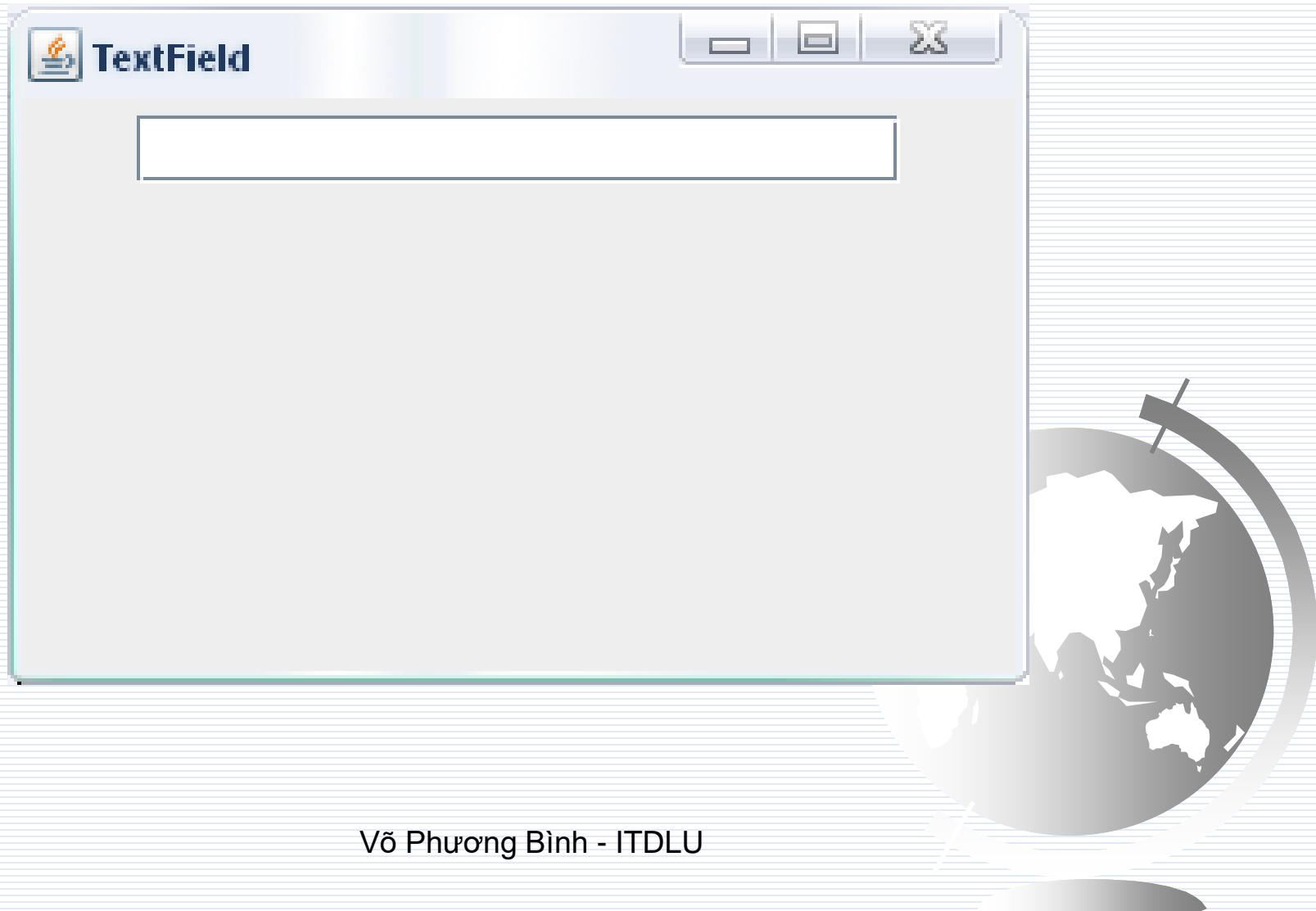


Ví dụ JTextField

```
1. import javax.swing.*;  
2. class TextFieldtest extends JFrame{  
3.     JTextField tf1=new JTextField(20);  
4.     public TextFieldtest(String title){  
5.         super(title);  
6.         JPanel p = new JPanel();  
7.         p.add(tf1);  
8.         add(p);  
9.     }  
10.    public static void main(String args[]){  
11.        TextFieldtest f=new TextFieldtest("TextField");  
12.        f.setSize(300,200);  
13.        f.show();  
14.    }  
15. }
```



Ví dụ JTextField



JTextField

- ✓ Các phương thức thường được sử dụng của đối tượng JTextField được tóm tắt trong bảng sau:

Phương thức	Chức năng
setText(String s)	Gán nội dung cho TextField.
getText()	Lấy nội dung của TextField.
setEditable(boolean)	Xác định TextField có soạn thảo được hay không. Nó chỉ được soạn thảo khi giá trị tham số truyền vào là True.
isEditable()	Xác định xem có đang trong chế độ soạn thảo hay không. Giá trị trả về kiểu Boolean.

LẬP TRÌNH JAVA

Tiết 4: Thành phần JTextArea, JButton

JTextArea

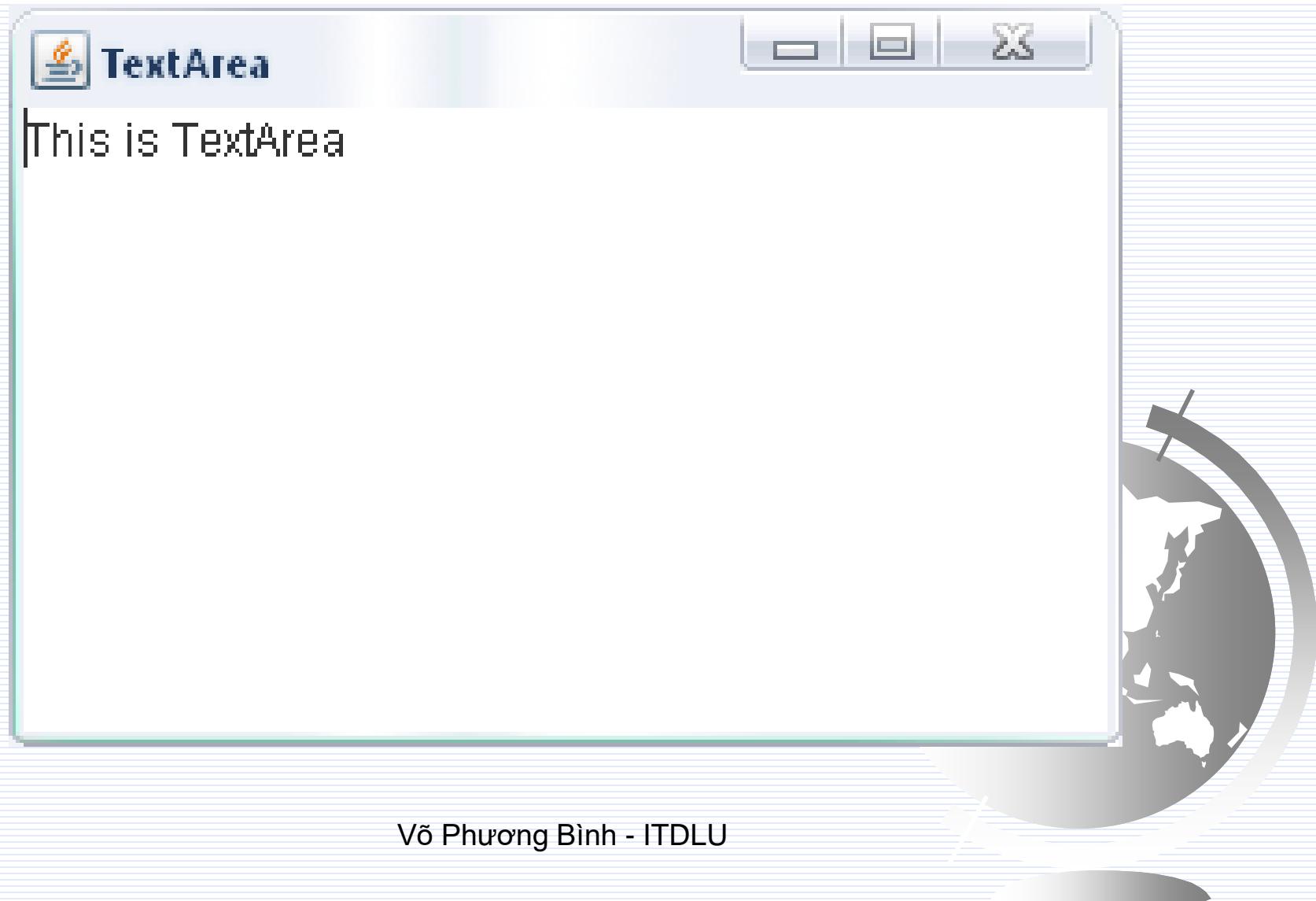
- ✓ TextArea được sử dụng khi văn bản nhập vào có từ hai hay nhiều dòng.
- ✓ TextArea có một scrollbar.
- ✓ Các constructor tạo TextArea:
 - **JTextArea()**: Tạo một TextArea mới.
 - **JTextArea(int rows, int cols)**: Tạo một TextArea mới với số lượng cột và dòng được cho trước.
 - **JTextArea(String text)**: Tạo một TextArea mới với dung được cho trước.
 - **JTextArea(String text, int rows, int cols)**: Tạo một TextArea mới với dung, số dòng và số cột được cho trước.

Ví dụ JTextArea

```
1. import javax.swing.*;  
2. class TextAreatest extends JFrame{  
3.     JTextArea ta1=new JTextArea("This is a TextArea");  
4.     public TextAreatest(String title){  
5.         super(title);  
6.         add(ta1);  
7.     }  
8.     public static void main(String args[]){  
9.         TextAreatest t=new TextAreatest("TextArea");  
10.        t.setSize(300,200);  
11.        t.show();  
12.    }  
13. }
```



Ví dụ JTextArea



JTextArea

- ✓ Các phương thức thường được sử dụng của TextArea:

Phương thức	Chức năng
<code>setText(String)</code>	Gán nội dung cho TextArea.
<code>getText()</code>	Trả về nội dung của TextArea.
<code>setEditable(boolean)</code>	Xác định xem TextArea có thể được soạn thảo hay không. TextArea có thể được soạn thảo khi giá trị này là True.
<code>isEditable()</code>	Xác định xem TextArea có đang trong chế độ soạn thảo được không. Trả về giá trị là kiểu Boolean.
<code>insert(String, int)</code>	Chèn chuỗi vào vị trí được cho trước.
<code>replaceRange(String s, int x, int y)</code>	Thay thế văn bản nằm giữa vị trí x, y cho trước bởi s.

JButton

- ✓ Button hay còn gọi là nút lệnh là một phần không thể thiếu của bất kỳ giao diện nào.
- ✓ Sử dụng button là cách dễ nhất để nhận các tác động của người dùng.
- ✓ Các constructor tạo button:
 - JButton()
 - JButton(String text)
 - JButton(Icon icon)
 - JButton(String text, Icon icon)
- ✓ Sử dụng setLabel() và getLabel() để thiết lập và lấy giá trị nhãn của button.



Ví dụ JButton

```
1. import javax.swing.*;  
2. class Buttontest extends JFrame{  
3.     JButton b1 = new JButton("OK");  
4.     public Buttontest(String title){  
5.         super(title);  
6.         add(b1);  
7.     }  
8.     public static void main(String args[]){  
9.         Buttontest t= new Buttontest("Button");  
10.        t.setSize(300,200);  
11.        t.show();  
12.    }  
13. }
```



Ví dụ JButton

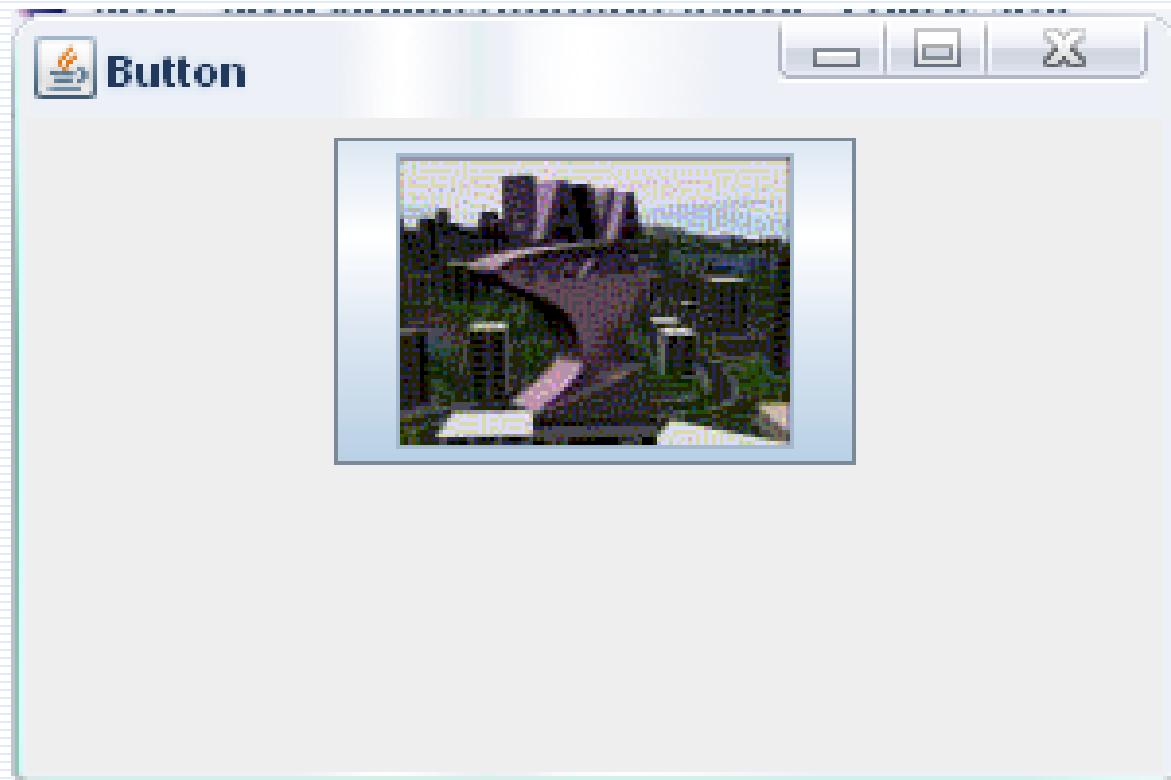


Ví dụ 2 JButton

```
✓ class Buttontest extends JFrame{  
✓   JButton b1 = new JButton();  
  public Buttontest(String title){  
    super(title);  
    ImageIcon ic = new ImageIcon("jc_frame_0.png");  
    b1.setIcon(ic);  
    JPanel p = new JPanel();  
    p.add(b1);  
    add(p);  
  }  
✓  public static void main(String args[]){  
✓    Buttontest t= new Buttontest("Button");  
✓    t.setSize(300,200);  
✓    t.show();  
  }  
✓ }
```



Ví dụ 2 JButton



Võ Phương Bình - ITDLU

LẬP TRÌNH JAVA

Tiết 5: Thành phần Jcheckbox và
JRadioButton, JComboBox

JCheckBox và JRadioButton

- ✓ Checkbox được sử dụng khi người dùng tiến hành chọn một hay nhiều tùy chọn.
- ✓ Người dùng click trên các checkbox để chọn hay bỏ chọn chúng.
- ✓ Các constructor tạo checkbox:
 - **Checkbox()**: Tạo một checkbox trống.
 - **Checkbox(String text, boolean selected)**: Tạo một checkbox với nhãn và giá trị được chọn hay không.
 - **Checkbox(Icon icon, boolean selected)**
 - **Checkbox(String text, Icon icon)**
 - **Checkbox(String text, Icon icon, boolean selected)**



Ví dụ JCheckbox và JRadioButton



Ví dụ JCheckbox và JRadioButton

```
1. import java.awt.GridLayout;  
2. import javax.swing.*;  
3. class Checkboxtest extends JFrame{  
4.     JLabel l1 = new JLabel("Check Boxes");  
5.     JCheckBox b1= new JCheckBox("red",true);  
6.     JCheckBox b2= new JCheckBox("green",false);  
7.     JCheckBox b3= new JCheckBox("blue",false);  
8.     JLabel l2 = new JLabel("Radiobuttons");  
9.     ButtonGroup bg = new ButtonGroup();  
10.    JRadioButton b4=new JRadioButton("small",true);  
11.    JRadioButton b5=new JRadioButton("medium",false);  
12.    JRadioButton b6=new JRadioButton("large",false);
```



Ví dụ JCheckbox và JRadioButton

```
13. public Checkboxtest(String title){  
14.         super(title); setLayout(new GridLayout(8,1));  
15.         add(l1);  
16.         add(b1);  
17.         add(b2);  
18.         add(b3);  
19.         add(l2);  
20.         bg.add(b4);bg.add(b5);bg.add(b6);  
21.         add(b4);  
22.         add(b5);  
23.         add(b6);  
24.     }
```



Ví dụ JCheckbox và JRadioButton

```
25. public static void main(String args[]){  
26.     Checkboxtest t=new Checkboxtest("Checkbox and  
        radiobutton");  
27.     t.setSize(300,200);  
28.     t.show();  
29. }  
30. }
```

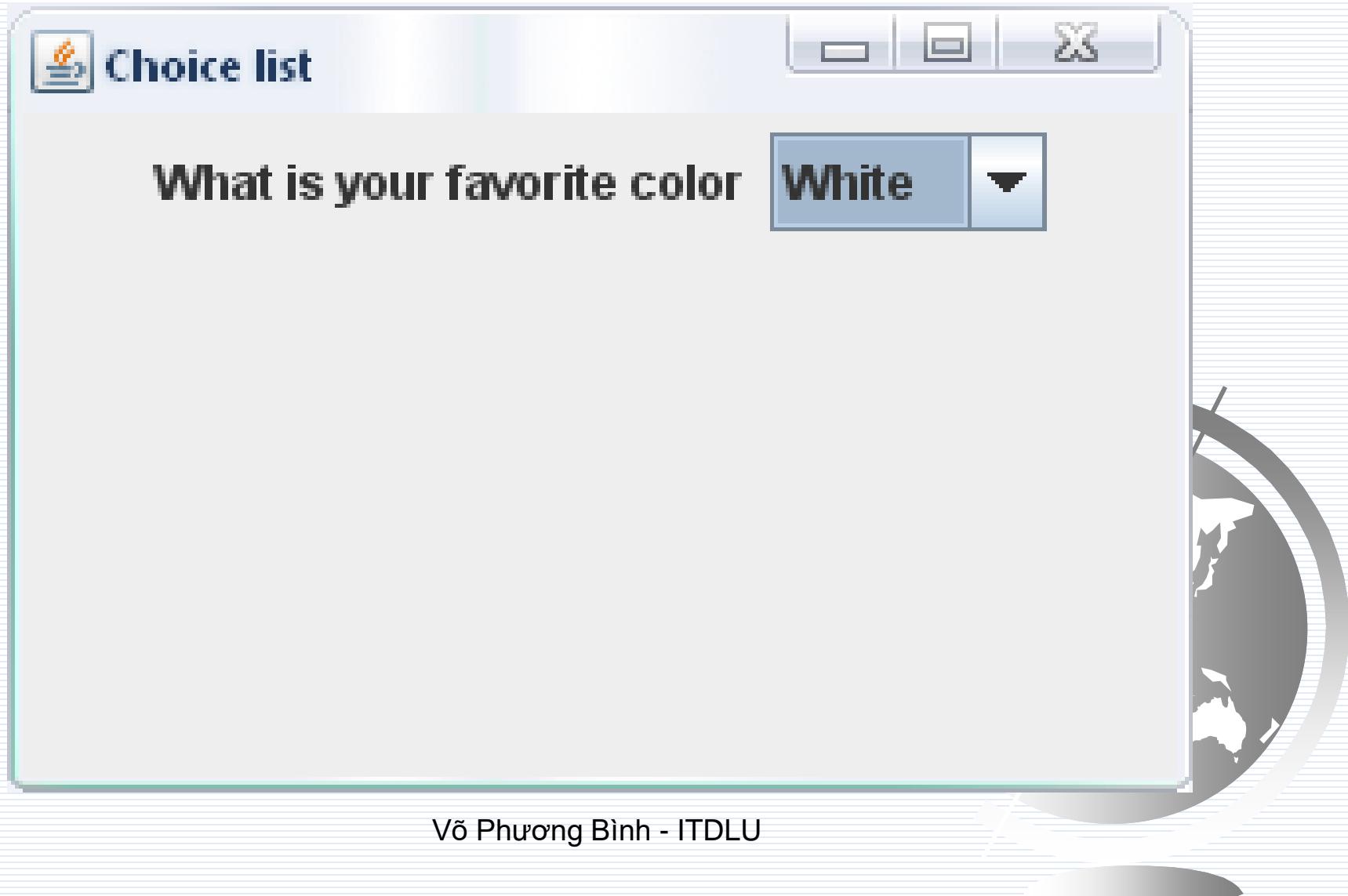


JComboBox

- ✓ JComboBox cho phép tạo danh sách chứa nhiều mục. Người dùng chỉ có thể click vào một mục từ danh sách.
- ✓ Các hàm Constructor tạo ComboBox:
 - JComboBox(): Tạo Combobox rỗng
 - JComboBox(String[] stringArray): Tạo Combobox với danh sách stringArray.
- ✓ Thêm một item bằng cách sử dụng phương thức addItem như sau:
 - addItem("Chuỗi");



Ví dụ JComboBox



Ví dụ JComboBox

```
1. import java.awt.FlowLayout;  
2. import javax.swing.*;  
3. class Choicetest extends JFrame{  
4.     JLabel l1=new JLabel("What is your favorite color");  
5.     JComboBox colors=new JComboBox();  
6.     public Choicetest(String title)  
7.     {  
8.         super(title); setLayout(new FlowLayout());  
9.         add(l1);  
10.        colors.addItem("White"); colors.addItem("Red");  
11.        colors.addItem("Orange"); colors.addItem("Green");  
12.        add(colors);  
13.    }
```



Ví dụ JComboBox

```
14. public static void main(String args[]){  
15.     Choicetest t=new  
    Choicetest("Choice list");  
16.     t.setSize(300,200);  
17.     t.show();  
18. }  
19. }
```



JComboBox

- ✓ Các phương thức thường được sử dụng của JComboBox :

Phương thức	Chức năng
setBackground(Color)	Xác định màu nền
setForeground(Color)	Xác định màu chữ
int getSelectedIndex()	Lấy giá trị chỉ số mục được chọn, trả về số nguyên.
Object getSelectedItem()	Lấy giá trị dữ liệu được chọn, trả về kiểu Object

LẬP TRÌNH JAVA

Tiết 6: Sử dụng Layout để sắp xếp các
thành phần trên JFrame

FlowLayout

BorderLayout

Layout manager

- ✓ Layout manager điều khiển cách trình bày vật lý của các phần tử GUI như là button, textbox, label, ...
- ✓ Các kiểu Layout khác nhau:
 - Flow layout
 - Border layout
 - Card layout
 - Grid layout
 - GridBagConstraints



Flow layout

- ✓ FlowLayout:
 - Các thành phần được sắp xếp theo hàng, từ trái sang phải.
- ✓ Các constructor của FlowLayout:
 - FlowLayout mylayout = new FlowLayout() // constructor
- ✓ Các điều khiển có thể được canh về bên trái, bên phải hay ở giữa:
 - **setLayout(new FlowLayout(FlowLayout.RIGHT));**



Ví dụ Flow layout

```
1. import java.awt.FlowLayout;  
2. import javax.swing.*;  
3. class Fltest extends JFrame{  
4.     JButton b1=new JButton("Center Aligned Button 1");  
5.     JButton b2=new JButton("Center Aligned Button 2");  
6.     JButton b3=new JButton("Center Aligned Button 3");  
7.     public Fltest(String title){  
8.         super(title);  
9.         setLayout(new  
        FlowLayout(FlowLayout.CENTER));  
10.        add(b1);add(b2);add(b3);  
11.    }
```

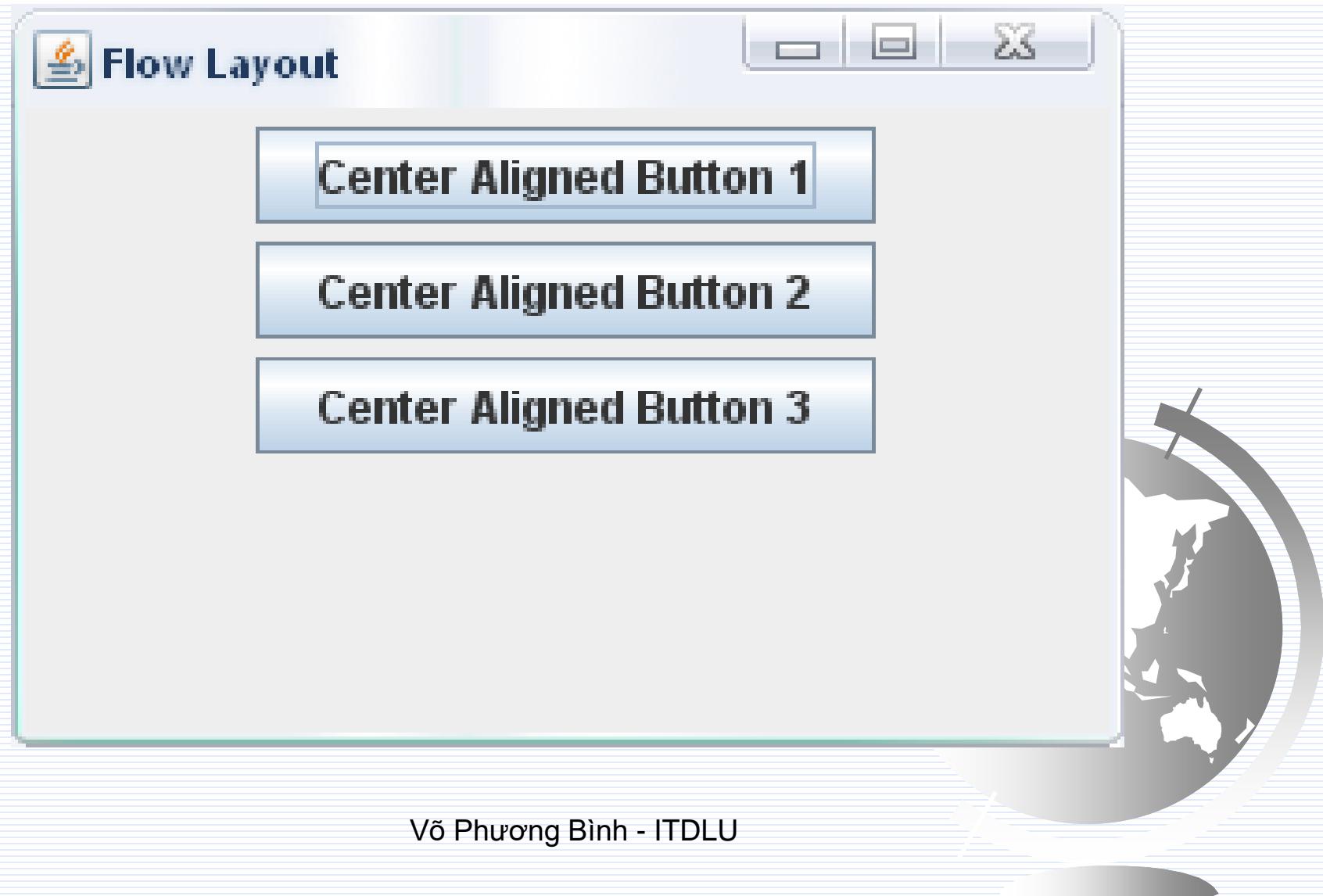


Ví dụ Flow layout

```
12. public static void main(String args[]){  
13.     Fltest t=new Fltest("Flow Layout");  
14.     t.setSize(300,200);  
15.     t.show();  
16. }  
17. }
```



Ví dụ Flow layout



BorderLayout

✓ BorderLayout

- Mặc định cho Window, Frame và Dialog.
- Sắp xếp tối đa 5 thành phần trong một container:
 - **NORTH** – Đặt ở đỉnh của container.
 - **EAST** – Đặt phía bên phải của container.
 - **SOUTH** – Đặt ở phía dưới của container.
 - **WEST** – Đặt phía bên trái của container.
 - **CENTER** – Đặt ở giữa của container.

✓ Constructor tạo BorderLayout

- **BorderLayout()**
- **BorderLayout(int, int)**



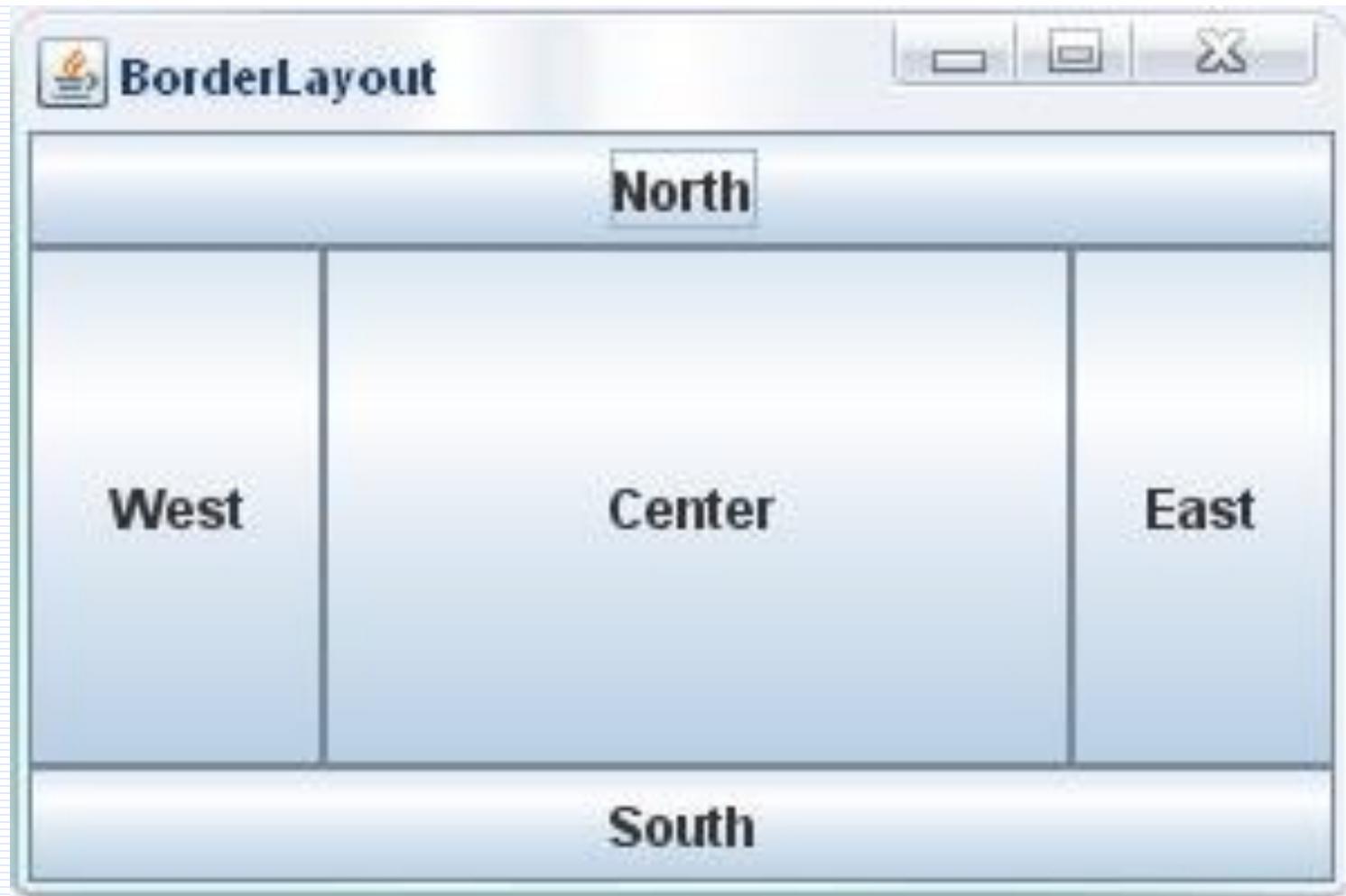
BorderLayout

✓ Ví dụ:

- Thêm một thành phần Button vào vùng ‘North’ như sau:
 - JButton b1=new JButton(“North Button”); // khai báo thành phần
 - setLayout(new BorderLayout()); // thiết lập layout
 - add(b1,BorderLayout.NORTH); // thêm thành phần vào layout



Ví dụ BorderLayout



Ví dụ BorderLayout

```
1. import java.awt.BorderLayout;  
2. import java.awt.FlowLayout;  
3. import javax.swing.*;  
4. class Fltest extends JFrame{  
5.     JButton b1=new JButton("North");  
6.     JButton b2=new JButton("South");  
7.     JButton b3=new JButton("West");  
8.     JButton b4=new JButton("East");  
9.     JButton b5=new JButton("Center");  
10.    public Fltest(String title){  
11.        super(title);  
12.        setLayout(new BorderLayout());
```



Ví dụ BorderLayout

```
13.         add(b1, BorderLayout.NORTH);
14.         add(b2, BorderLayout.SOUTH);
15.         add(b3, BorderLayout.WEST);
16.         add(b4, BorderLayout.EAST);
17.         add(b5, BorderLayout.CENTER);
18.     }
19.     public static void main(String args[]){
20.         Fltest t=new Fltest("BorderLayout");
21.         t.setSize(300,200);
22.         t.show();
23.     }
24. }
```



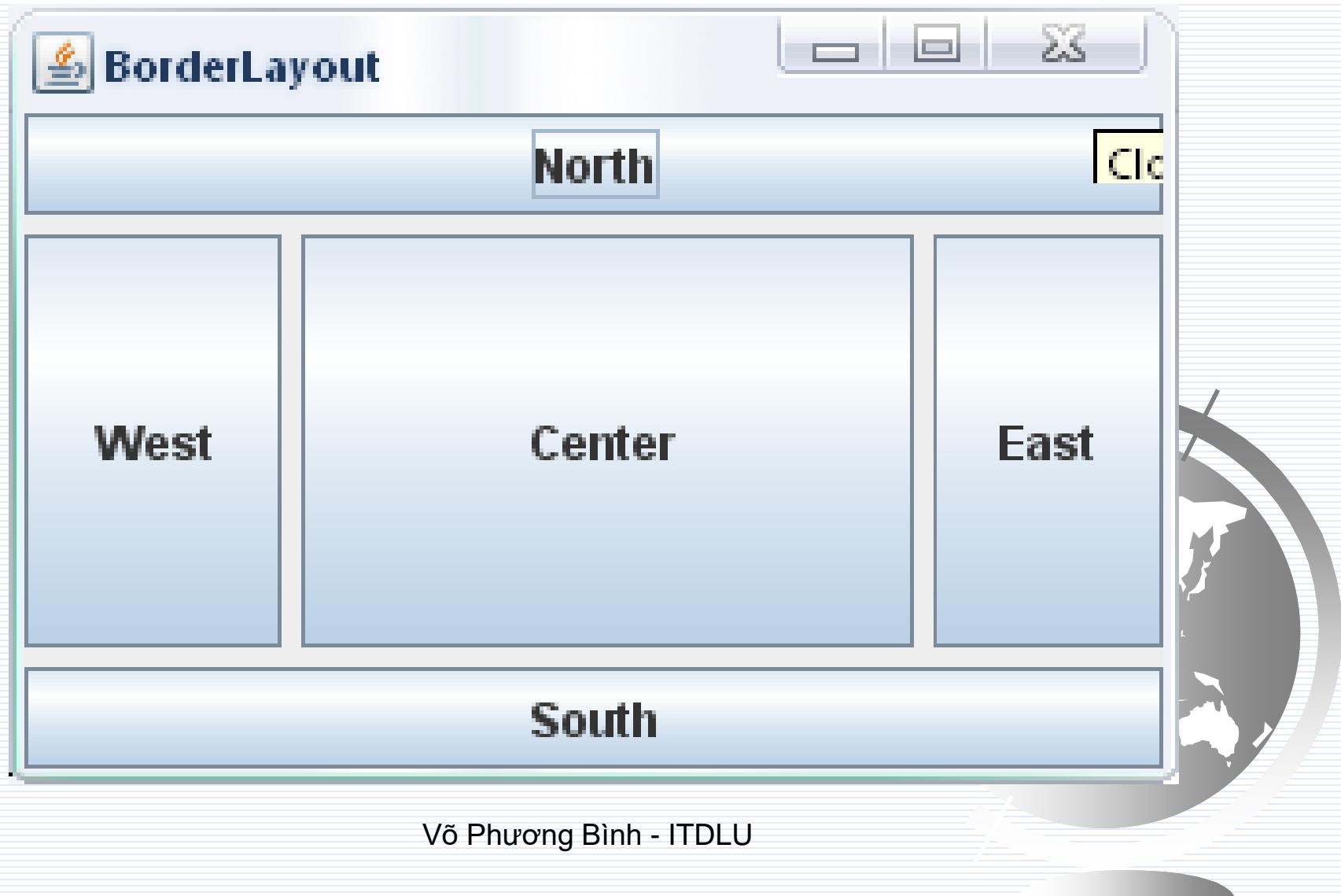
Ví dụ 2 BorderLayout

✓ Dùng contructor có tham số:

- **setLayout(new BorderLayout(5,5));**
- **add(b1, BorderLayout.NORTH);**
- **add(b2, BorderLayout.SOUTH);**
- **add(b3, BorderLayout.WEST);**
- **add(b4, BorderLayout.EAST);**
- **add(b5, BorderLayout.CENTER);**



Ví dụ 2 BorderLayout

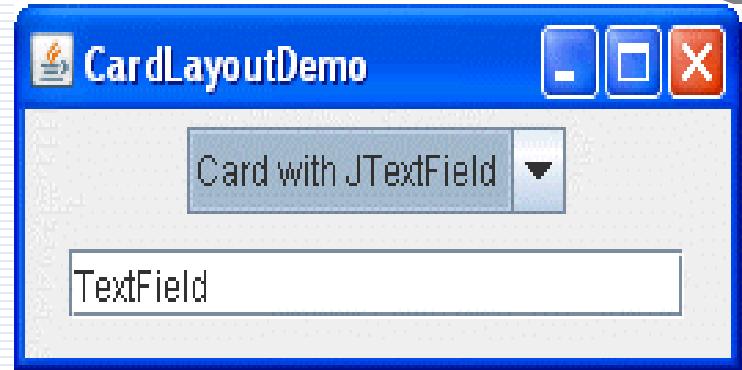
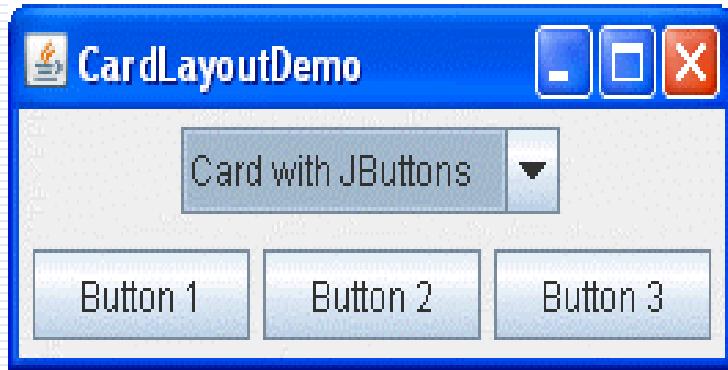


LẬP TRÌNH JAVA

Tiết 7: CardLayout, GridLayout

CardLayout

- ✓ CardLayout có thể lưu trữ một ngăn xếp các giao diện.
- ✓ Ta có thể chuyển qua lại các giao diện:



Ví dụ CardLayout

```
1. import java.awt.*;  
2. import javax.swing.*;  
3. public class CardLayoutDemo extends JFrame{  
4.     public CardLayoutDemo(String text) {  
5.         super(text);  
6.         JPanel cards = new JPanel(new CardLayout());  
7.         cards.add(new JButton("Button 1"), "B1");  
8.         cards.add(new JButton("Button 2"), "B2");  
9.         CardLayout cl = (CardLayout)(cards.getLayout());  
10.        cl.show(cards, "B2");  
11.        add(cards);  
12.    }
```



Ví dụ CardLayout

```
✓ public static void main(String args[]){  
✓     CardLayoutDemo t=new  
CardLayoutDemo("Card Layout");  
✓     t.setSize(300,200);  
✓     t.show();  
✓ }  
✓ }
```



Võ Phương Bình - ITDLU



Ví dụ CardLayout

- ✓ Ta có thể chuyển sang giao diện Button 1:

```
cl.show(cards, "B1");
```



GridLayout

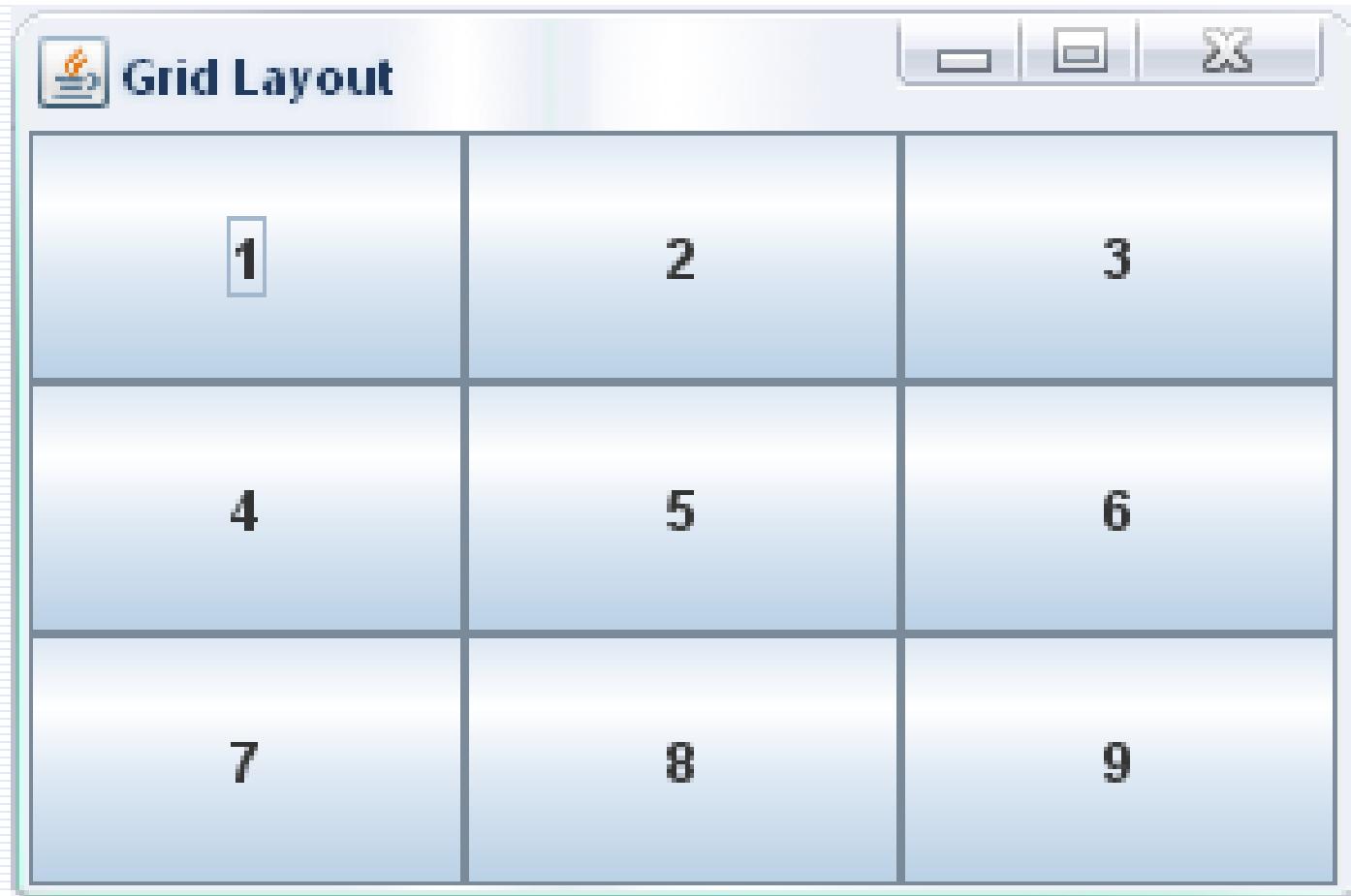
✓ GridLayout:

- Chia container thành các ô lưới đều nhau dạng ma trận.
- Các thành phần được đặt trong các ô giao theo dòng và cột theo thứ tự từ trái sang phải và từ trên xuống dưới..

✓ GridLayout được tạo như sau:

- **GridLayout(int rows, int columns);**
 - rows là số dòng và columns là số cột.

Ví dụ GridLayout



Ví dụ GridLayout

```
1. import java.awt.GridLayout;  
2. import javax.swing.*;  
3. class Gltest extends JFrame{  
4.     JButton btn[];  
5.     String str[]={ "1", "2", "3", "4", "5", "6", "7", "8", "9" };  
6.     public Gltest(String title){  
7.         super(title);setLayout(new GridLayout(3,3));  
8.         btn=new JButton[str.length];  
9.         for (int i=0; i<str.length;i++){  
10.             btn[i]=new JButton(str[i]); add(btn[i]);  
11.         }  
12.     }  
13. }
```



Ví dụ GridLayout

```
14. public static void main(String args[])
15. {
16.     Gltest t=new Gltest("Grid Layout");
17.     t.setSize(300,200);
18.     t.show();
19. }
20. }
```



LẬP TRÌNH JAVA

Tiết 8: Không dùng Layout để sắp xếp các
thành phần trên JFrame

Không dùng Layout

- ✓ Sắp xếp các thành phần trên Container mà không cần dùng Layout:

`setLayout(null);`

- ✓ Xác định phạm vi cần hiển thị thành phần:

`setBound(int x, int y, int width, int height);`

- Tọa độ góc top-left x, y
- Kích thước đối tượng width, height

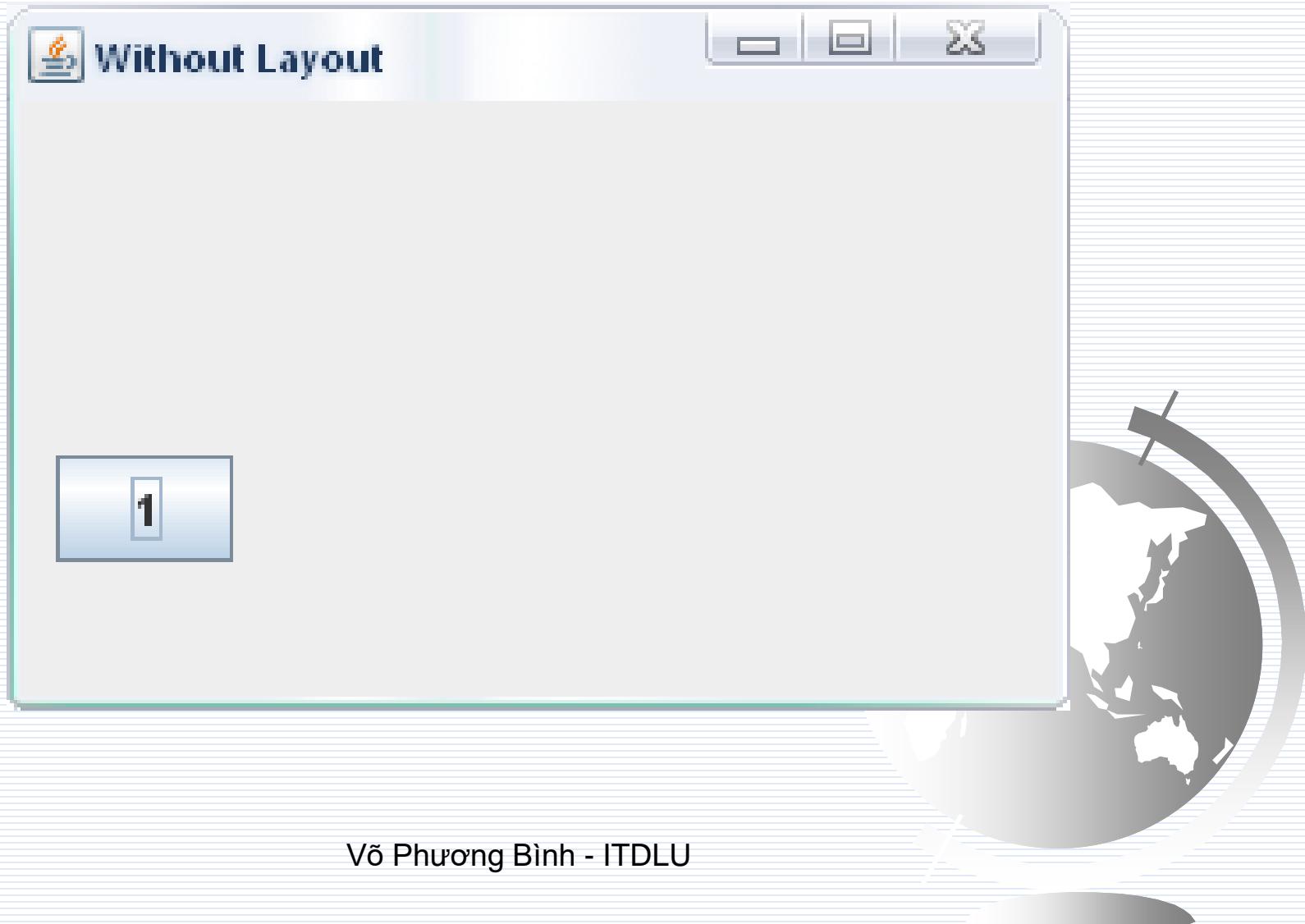


Ví dụ không dùng Layout

```
✓ import javax.swing.*;  
✓ class Gbltest extends JFrame{  
✓     public Gbltest(String title){  
✓         super(title);  
✓         setLayout(null);  
✓         JButton b1 = new JButton("1");  
✓         b1.setBounds(10, 100, 50, 30);  
✓         add(b1);  
✓     }  
✓     public static void main(String args[]){  
✓         Gbltest t=new Gbltest("Without Layout");  
✓         t.setSize(300,200); t.show();  
✓     }  
✓ }
```



Ví dụ không dùng Layout



LẬP TRÌNH JAVA

Tiết 9: Xử lý các loại sự kiện (Event)

Các giao diện lắng nghe sự kiện

Xử lý sự kiện

✓ Nhu cầu người dùng:

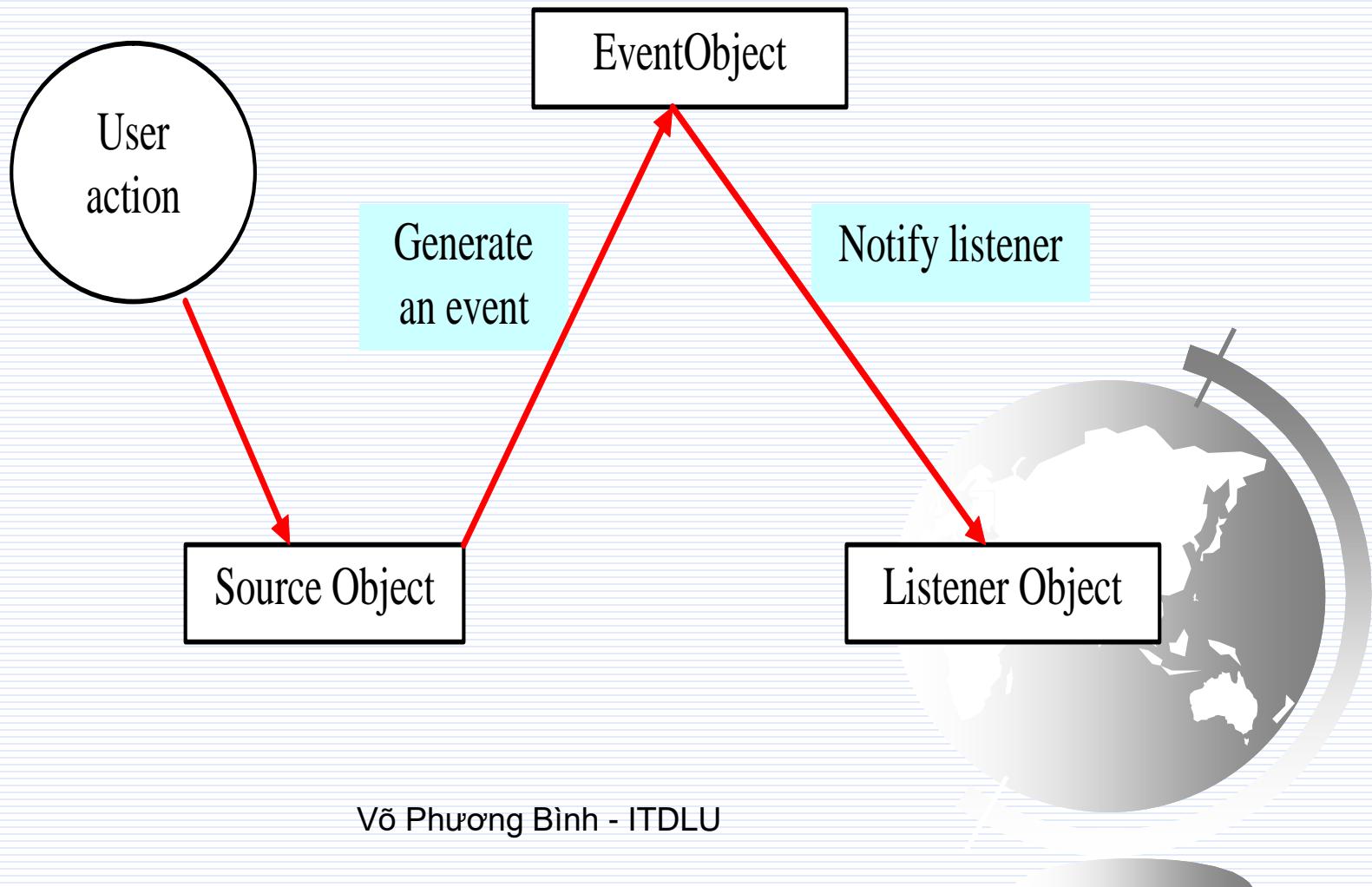
- Cần xử lý các tương tác: di chuyển chuột, nhấn phím, nhả phím, ... được gọi là **sự kiện**.

✓ Xử lý sự kiện thông qua các **giao diện lắng nghe sự kiện** (*Event Listener*):

- Mỗi giao diện lắng nghe một loại sự kiện.
- Mỗi giao diện cung cấp các phương thức xử lý những sự kiện này.
- Lớp thi hành giao diện cần phải định nghĩa những phương thức này.



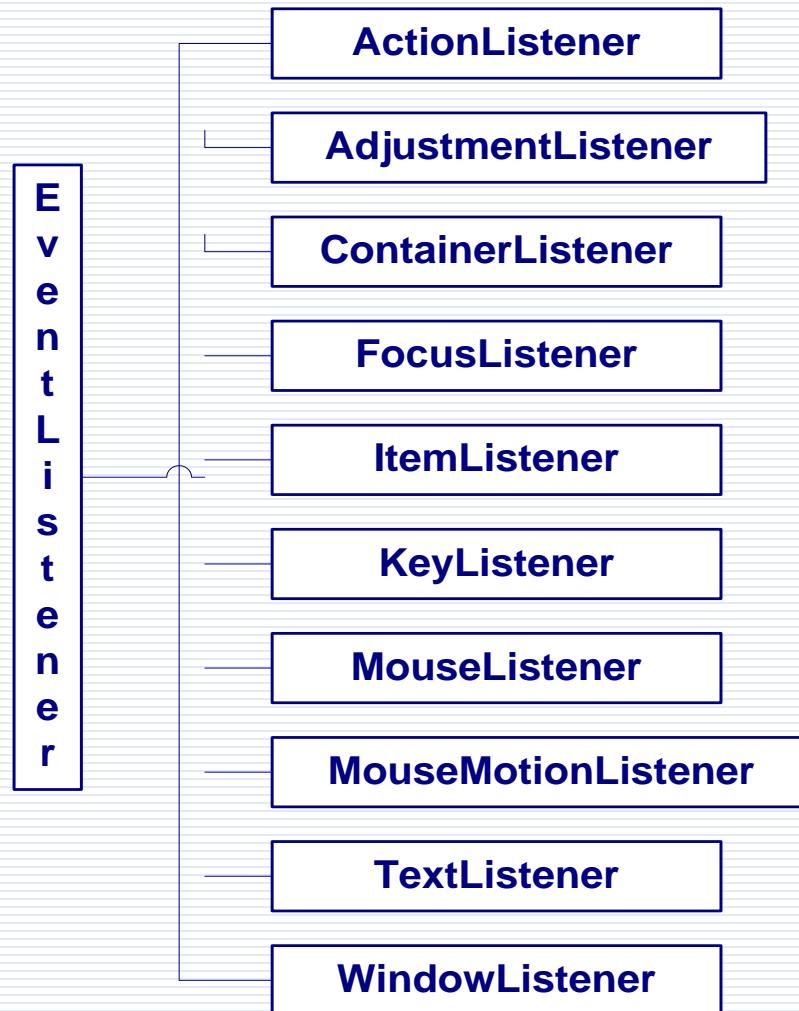
Xử lý sự kiện



Các loại sự kiện

Lớp sự kiện	Mô tả
ActionEvent	Phát sinh khi một button được nhấn, một item trong danh sách chọn lựa được nhấn đúp (double-click) hay một menu được chọn.
AdjustmentEvent	Phát sinh khi một thanh scrollbar được sử dụng.
ComponentEvent	Phát sinh khi một thành phần được thay đổi kích thước, được di chuyển, bị ẩn hay làm cho hoạt động được.
FocusEvent	Phát sinh khi một thành phần nhận focus từ bàn phím.
ItemEvent	Phát sinh khi một mục menu được chọn hay bỏ chọn; hay khi một checkbox hay một item trong danh sách được click.
WindowEvent	Phát sinh khi một cửa sổ được kích hoạt, được đóng, được mở hay thoát.
TextEvent	Phát sinh khi giá trị trong thành phần textfield hay textarea bị thay đổi.
MouseEvent	Phát sinh khi chuột di chuyển, được click, được kéo hay thả ra.
KeyEvent	Phát sinh khi bàn phím ấn, nhả.

Các giao diện Event Listener



Các giao diện Event Listener và Event tương ứng



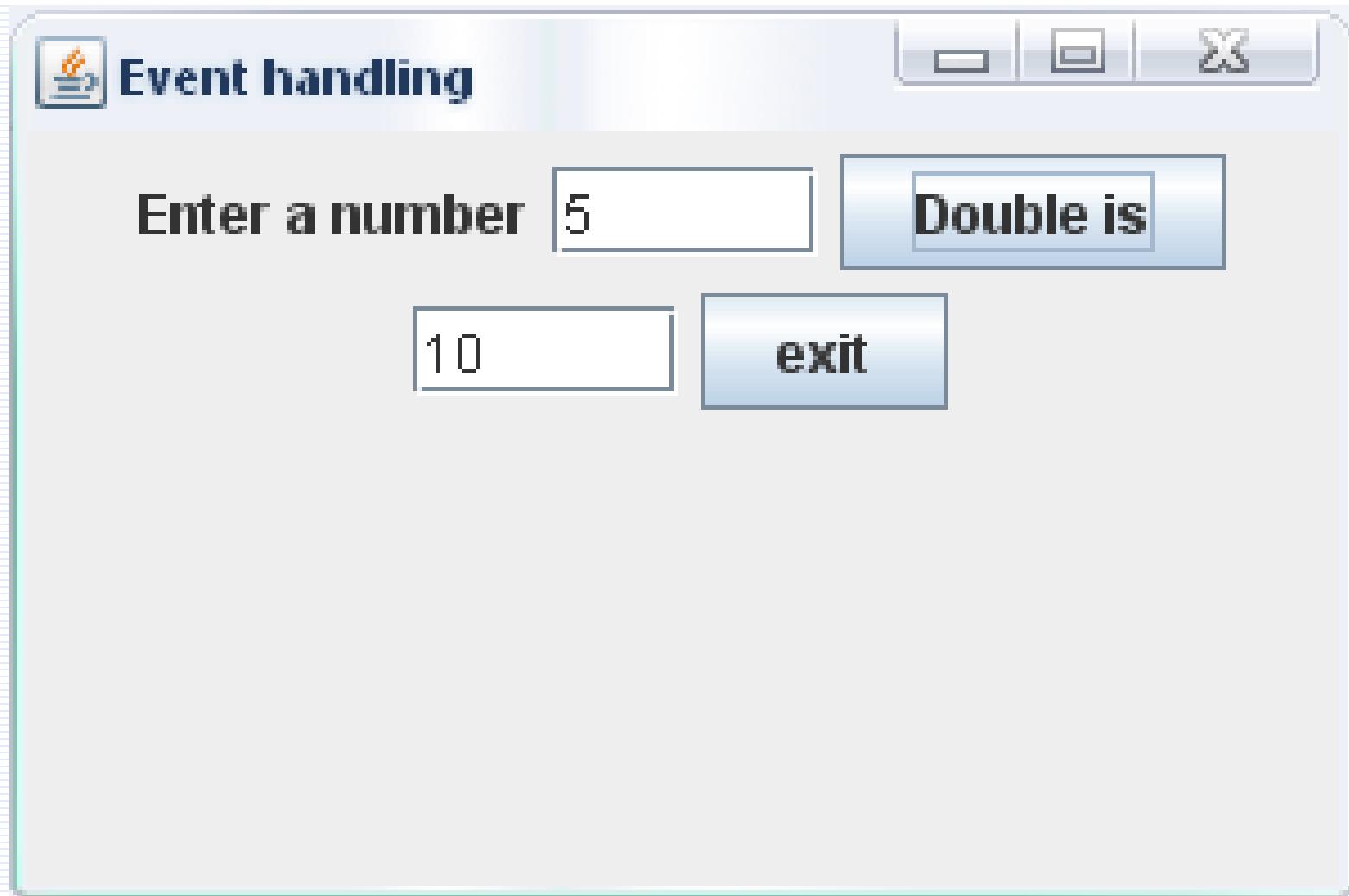
Xử lý sự kiện

✓ Ví dụ xử lý sự kiện trên Button:

- Cài đặt giao diện Event Listener thích hợp.
 - **public class MyApp extends JFrame implements ActionListener**
- Đăng ký một thành phần với listener, ta có thể sử dụng:
 - **addActionListener(this);**
- Xử lý sự kiện:
 - **public void actionPerformed(ActionEvent ae){**
 //code xử lý
• **}**



Ví dụ Xử lý sự kiện



Ví dụ Xử lý sự kiện

```
1. import java.awt.FlowLayout; import java.awt.event.ActionEvent;  
2. import java.awt.event.ActionListener; import javax.swing.*;  
3. class evttest extends JFrame implements ActionListener{  
4.     JLabel lab=new JLabel("Enter a number");  
5.     JTextField tf1=new JTextField(5);  
6.     JTextField tf2=new JTextField(5);  
7.     JButton btnResult=new JButton("Double is");  
8.     JButton ext=new JButton("exit");  
9.     public evttest(String title){  
10.         super(title); setLayout(new FlowLayout());  
11.         btnResult.addActionListener(this);  
12.         ext.addActionListener(this);  
13.         add(lab);add(tf1);add(btnResult);add(tf2);add(ext);  
14.     }
```



Ví dụ Xử lý sự kiện

```
15. public void actionPerformed(ActionEvent ae){  
16.     if (ae.getSource()==btnResult){  
17.         int num=Integer.parseInt(tf1.getText())*2;  
18.         tf2.setText(String.valueOf(num));  
19.     } else if (ae.getSource()==ext){  
20.         System.exit(0);  
21.     }  
22. }  
23. public static void main(String args[]){  
24.     evttest t=new evttest("Event handling");  
25.     t.setSize(300,200);  
26.     t.show();  
27. }  
28. }
```

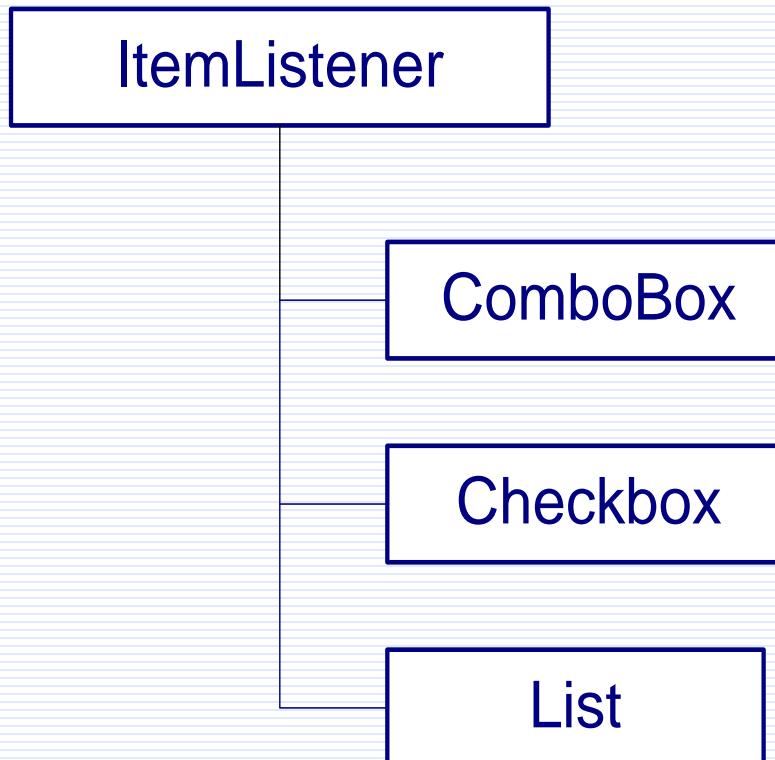


LẬP TRÌNH JAVA

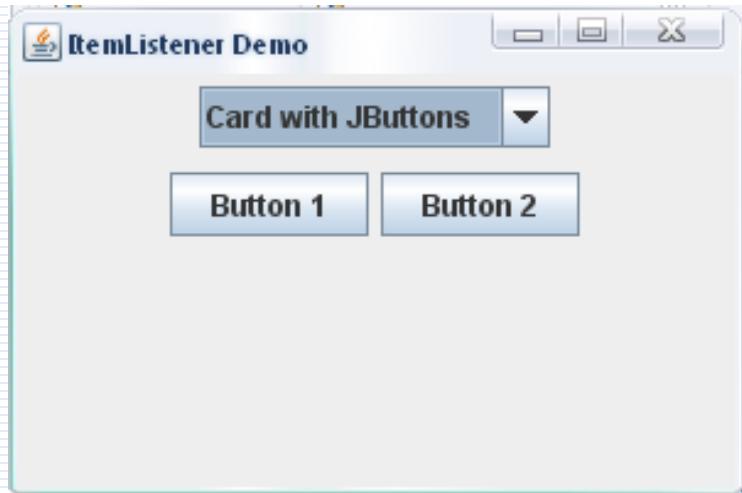
Tiết 10: Giao diện ItemListener

Giao diện ItemListener

- ✓ Giao diện ItemListener xử lý sự kiện ItemEvent trên các thành phần như ComboBox, CheckBox, List.



Ví dụ ItemListener và JComboBox



- ✓ Bước 1: Thiết kế giao diện
- ✓ Bước 2: Xử lý sự kiện trên JComboBox

Bước 1: Thiết kế giao diện

- ✓ public class ItemListenerDemo extends JFrame {
- ✓ JPanel cards;
- ✓ public ItemListenerDemo(String title) {
- ✓ super(title);
- ✓ JPanel comboBoxPane = new JPanel();
- ✓ String comboBoxItems[] = { "Card with JButtons", "Card with JTextField" };
- ✓ JComboBox cb = new JComboBox(comboBoxItems);
- ✓ comboBoxPane.add(cb);



Bước 1: Thiết kế giao diện

- ✓ JPanel card1 = new JPanel();
- ✓ card1.add(new JButton("Button 1"));
- ✓ card1.add(new JButton("Button 2"));
✓
- ✓ JPanel card2 = new JPanel();
- ✓ card2.add(new JTextField("TextField", 20));
✓
- ✓ cards = new JPanel(new CardLayout());
- ✓ cards.add(card1, "Card with JButtons");
- ✓ cards.add(card2, "Card with JTextField");
✓
- ✓ add(comboBoxPane, BorderLayout.NORTH);
- ✓ add(cards, BorderLayout.CENTER);
✓ }

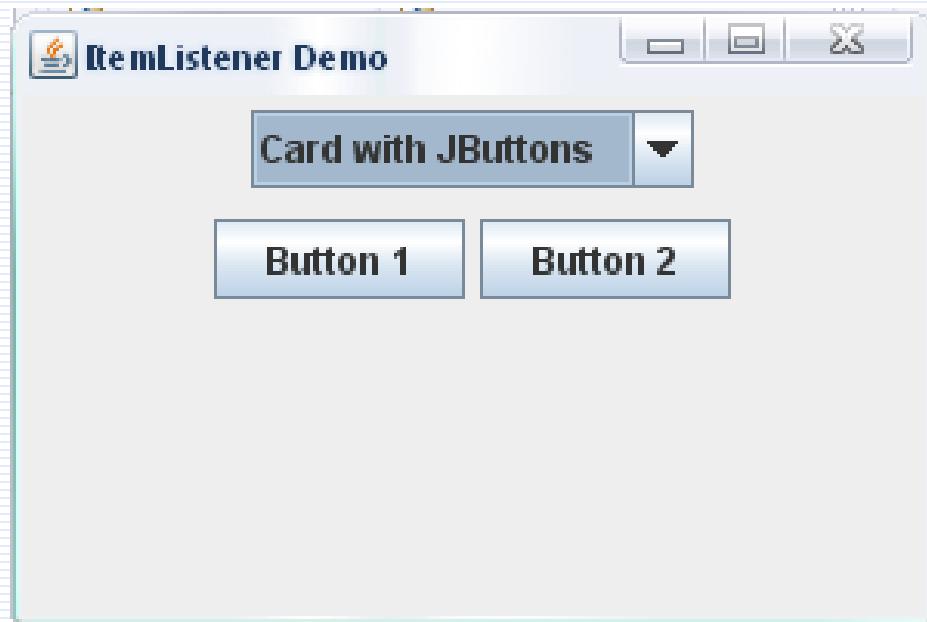


Bước 1: Thiết kế giao diện

- ✓ **public static void main(String[] args) {**
- ✓ **ItemListenerDemo demo = new**
 ItemListenerDemo("ItemListener Demo");
- ✓ **demo.setSize(300,200);**
- ✓ **demo.show();**
- ✓ **}**
- ✓ **}**



Bước 1: Thiết kế giao diện



Bước 2: Xử lý sự kiện

- ✓ Thực thi giao diện ItemListener:

```
public class ItemListenerDemo extends JFrame  
implements ItemListener {
```

- ✓ Lắng nghe sự kiện trên JComboBox

```
JPanel comboBoxPane = new JPanel();  
  
String comboBoxItems[] = { "Card with JButtons", "Card with  
JTextField" };  
  
JComboBox cb = new JComboBox(comboBoxItems);  
  
cb.addItemListener(this);  
  
comboBoxPane.add(cb);
```



Bước 2: Xử lý sự kiện

- ✓ Xử lý sự kiện click trên JComboBox
- ✓

```
public void itemStateChanged(ItemEvent evt) {
```
- ✓

```
    String strItem = (String) evt.getItem();
```
- ✓
- ✓

```
    CardLayout cl = (CardLayout)(cards.getLayout());
```
- ✓

```
    cl.show(cards, strItem );
```
- ✓

```
}
```



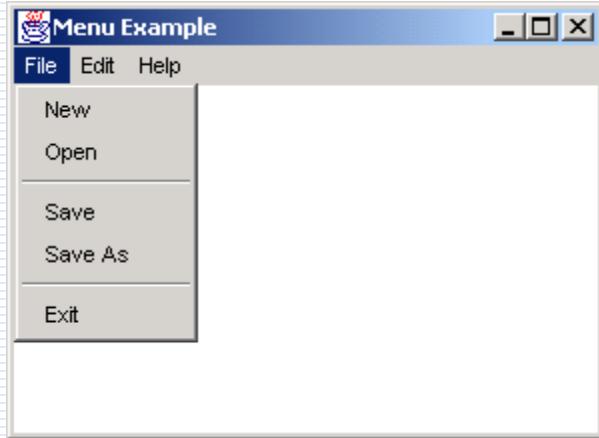
LẬP TRÌNH JAVA

Tiết 12: Tạo và xử lý 2 loại Menu
Pull-down và Pop-up

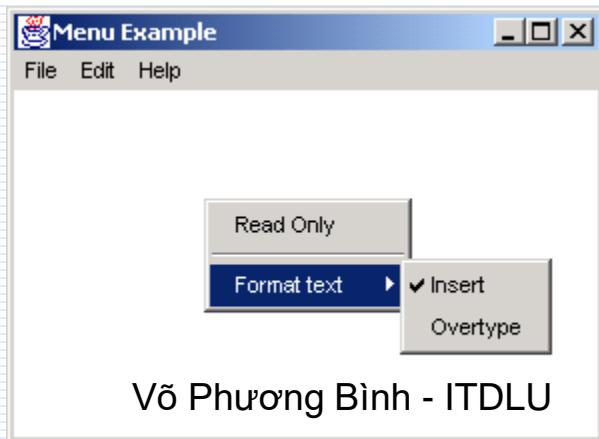
Menu

✓ Có 2 loại menu

- Pull-down



- Pop-up



Menu Pull-down

- ✓ Chỉ có duy nhất **1 Menubar** đặt trong **1 Frame**.
- ✓ Các thành phần của Menu:
 - **Menubar** là một thanh nằm ngang được đặt tại đỉnh của frame.
 - **Menu** liệt kê các mục chọn khác nhau (File, Edit, ...)
 - **MenuItem** các mục chọn con (Open, Close, ...)

Menu Pull-down

✓ TạoMenuBar

- **JMenuBar mbar=new JMenuBar();**

✓ Tạo Menu

- **JMenu fileMenu=new JMenu("File");**
- **mbar.add(fileMenu);**

✓ Tạo MenuItem

- **exitItem=new JMenuItem("Exit");**
- **exitItem.addActionListener(this);**
- **fileMenu.add(exitItem);**



Menu Pull-down

Xử lý MenuItem (exitItem)

Cách 1:

```
public void actionPerformed(ActionEvent ae) {  
    if (ae.getActionCommand().equals("Exit")) {  
        System.exit(0);  
    }  
}
```

Cách 2:

```
if (ae.getSource == exitItem){  
    System.exit(0);  
}
```



Menu Pop-up

Xử lý Click chuột phải:

1. Tạo MenuPopup

```
JPopupMenu optionsMenu=new JPopupMenu("Options");
```

2. Tạo MenuItem của MenuPopup

```
JMenuItem readItem=new JMenuItem("Read Only");
```

```
optionsMenu.add(readItem);
```

3. Xử Lý

```
public void mouseClicked(MouseEvent m) {  
    if(m.isMetaDown()){  
        optionsMenu.show(this,m.getX(),m.getY());  
    }  
}
```



LẬP TRÌNH JAVA

Tiết 13: Đồ họa trong Java

Sử dụng lớp JPanel để vẽ đồ họa

Lớp Graphics

Đồ họa trong Java

- ✓ Java cung cấp gói AWT cho phép ta vẽ các hình đồ họa trong một Panel.
- ✓ Để vẽ trong một panel
 - Tạo lớp kế thừa lớp JPanel
 - Định nghĩa lại phương thức **paintComponent** để chỉ cho panel vẽ các hình cần vẽ.



Đồ họa trong Java

```
✓ public class DrawMessage extends JPanel {  
✓   public static void main(String[] args) {  
✓     JFrame frame = new JFrame("DrawMessage");  
✓     frame.getContentPane().add(new DrawMessage());  
✓     frame.setSize(300, 200);  
✓     frame.setVisible(true);  
✓   }  
✓   public void paintComponent(Graphics g) {  
✓     super.paintComponent(g);  
✓     g.drawString("Welcome to Java!", 40, 40);  
✓   }  
✓ }
```



Đồ họa trong Java



Võ Phương Bình - ITDLU

Lớp Graphics

✓ Lớp Graphics bao gồm tập hợp rất nhiều phương thức, được sử dụng để vẽ trong các hình sau:

- Text
- Oval
- Rectangle
- Square
- Circle
- Lines



Vẽ các ký tự

- **drawChars (char array[], int offset, int length, int xCoor, int yCoor);**
 - Mảng các ký tự.
 - Vị trí bắt đầu, nơi các ký tự được vẽ.
 - Số các ký tự cần được vẽ.
 - Toạ độ xCoor, yCoor
- **drawBytes (byte array[], int offset, int length, int xCoor, int yCoor);**

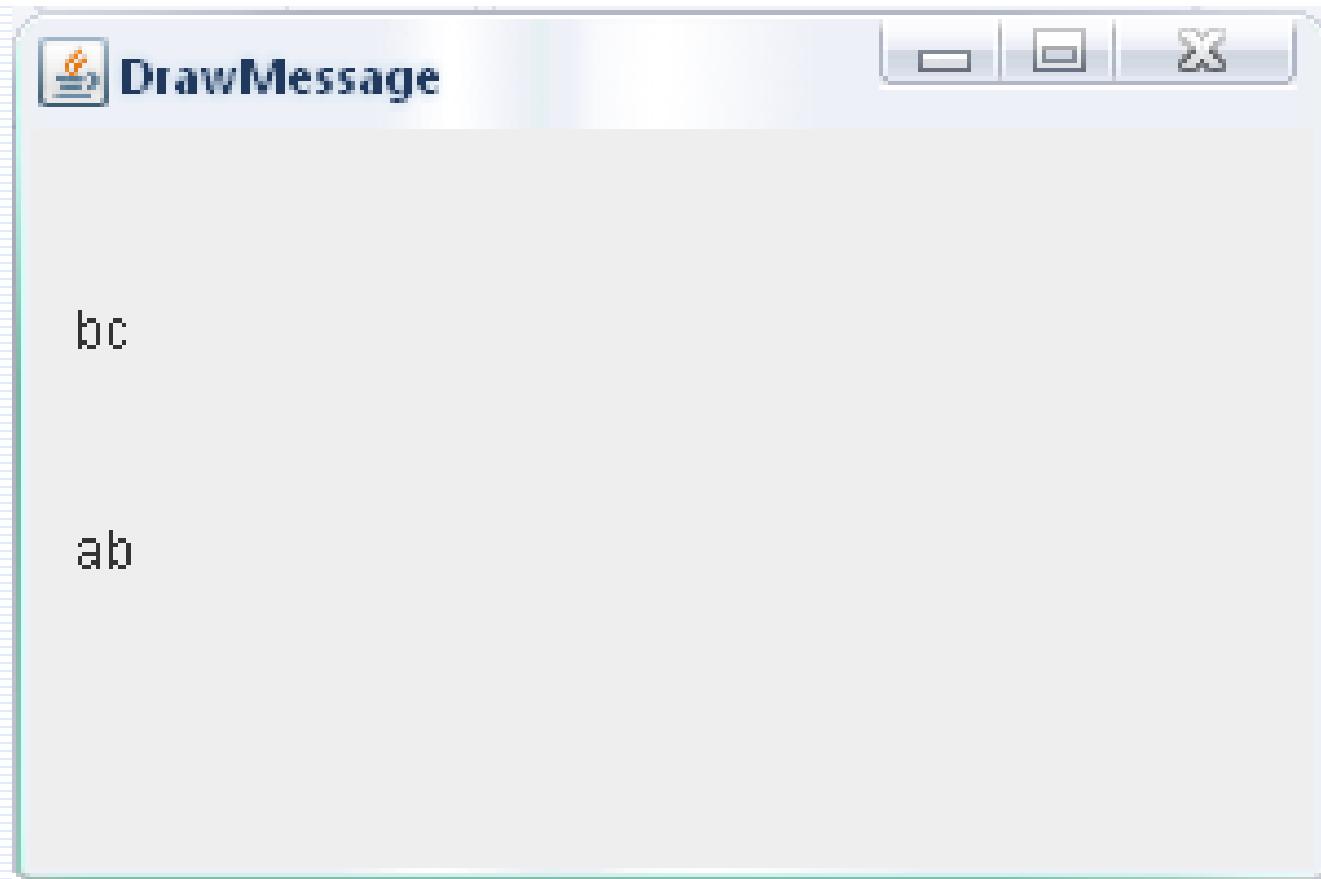


Ví dụ về các ký tự

```
✓ public class DrawMessage extends JPanel {  
✓     public static void main(String[] args) {  
✓         JFrame frame = new JFrame("DrawMessage");  
✓         frame.getContentPane().add(new DrawMessage());  
✓         frame.setSize(300, 200);  
✓         frame.setVisible(true);  
✓     }  
✓     public void paintComponent(Graphics g) {  
✓         super.paintComponent(g);  
✓         char[] c ={'a','b','c'};  
✓         byte[] b ={'a','b','c'};  
✓         g.drawChars(c, 1, 2, 10, 50);  
✓         g.drawBytes(b, 0, 2, 10, 100);  
✓     }  
✓ }
```



Ví dụ về các ký tự



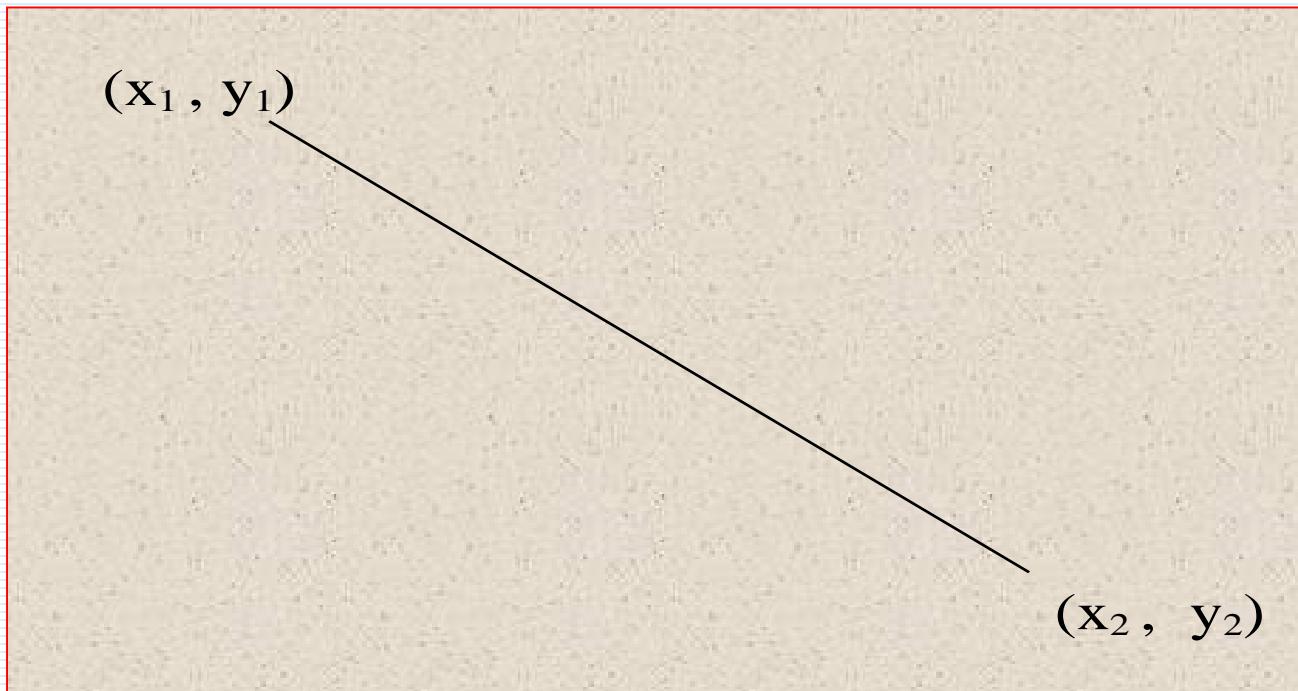
LẬP TRÌNH JAVA

Tiết 14: Vẽ đoạn thẳng với `drawLine`

Vẽ hình elipse, hình tròn với `drawOval`, `fillOval`

Vẽ đoạn thẳng (Line)

```
drawLine(x1, y1, x2, y2);
```



Ví dụ vẽ đoạn thẳng (Line)

```
✓ public class DrawMessage extends JPanel {  
✓     public static void main(String[] args) {  
✓         JFrame frame = new JFrame("DrawMessage");  
✓         frame.getContentPane().add(new DrawMessage());  
✓         frame.setSize(300, 200);  
✓         frame.setVisible(true);  
✓     }  
✓     public void paintComponent(Graphics g) {  
✓         super.paintComponent(g);  
✓         g.drawLine(50, 50, 150, 100);  
✓     }  
✓ }
```



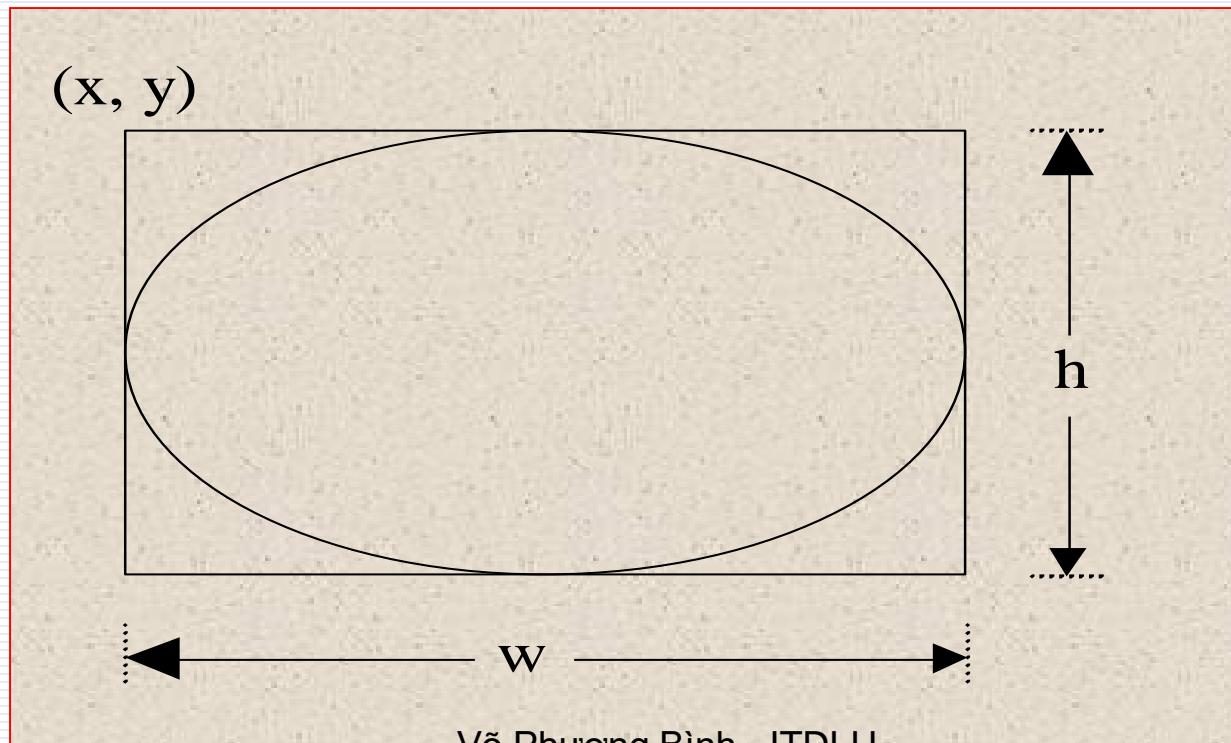
Ví dụ vẽ đoạn thẳng (Line)



Vẽ hình Elipse, hình Tròn (Oval)

`drawOval(x, y, w, h);`

`fillOval(x, y, w, h);`

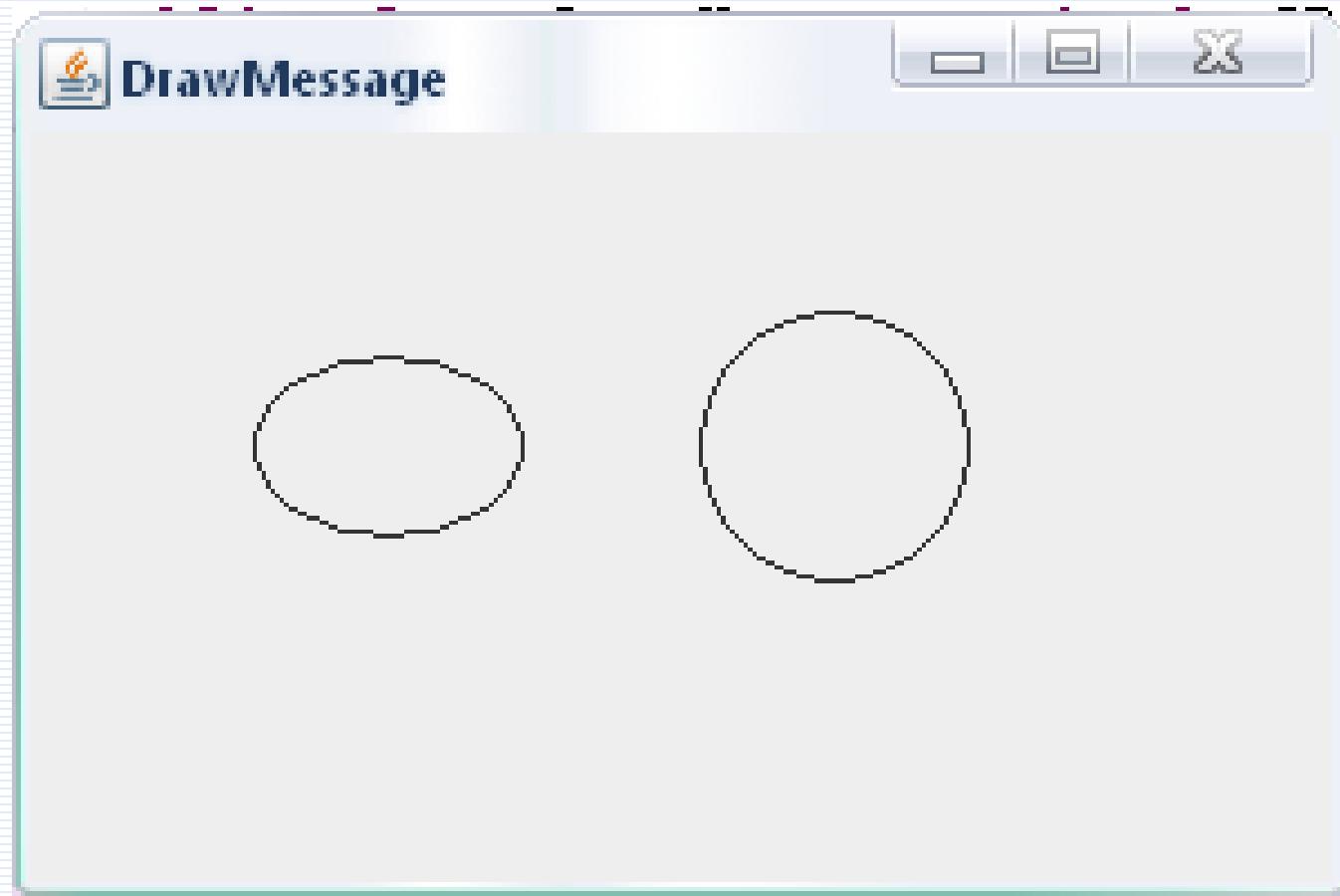


Ví dụ vẽ hình Elipse, hình Tròn (Oval)

```
✓ public class DrawMessage extends JPanel {  
✓     public static void main(String[] args) {  
✓         JFrame frame = new JFrame("DrawMessage");  
✓         frame.getContentPane().add(new DrawMessage());  
✓         frame.setSize(300, 200);  
✓         frame.setVisible(true);  
✓     }  
✓     public void paintComponent(Graphics g) {  
✓         super.paintComponent(g);  
✓         g.drawOval(50, 50, 60, 40);  
✓         g.drawOval(150, 40, 60, 60);  
✓     }  
✓ }
```



Ví dụ vẽ hình Elipse, hình Tròn (Oval)



Võ Phương Bình - ITDLU

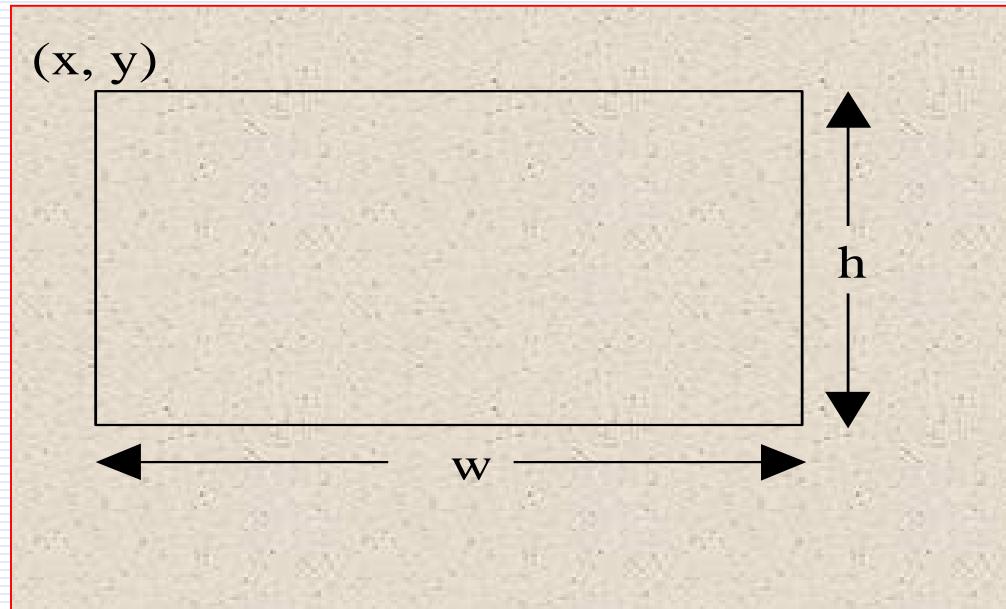
LẬP TRÌNH JAVA

Tiết 15: Vẽ hình chữ nhật với `drawRect`, `fillRect`
`drawRoundRect`, `fillRoundRect`

Vẽ hình chữ nhật

```
drawRect (x, y, w, h);
```

```
fillRect (x, y, w, h);
```

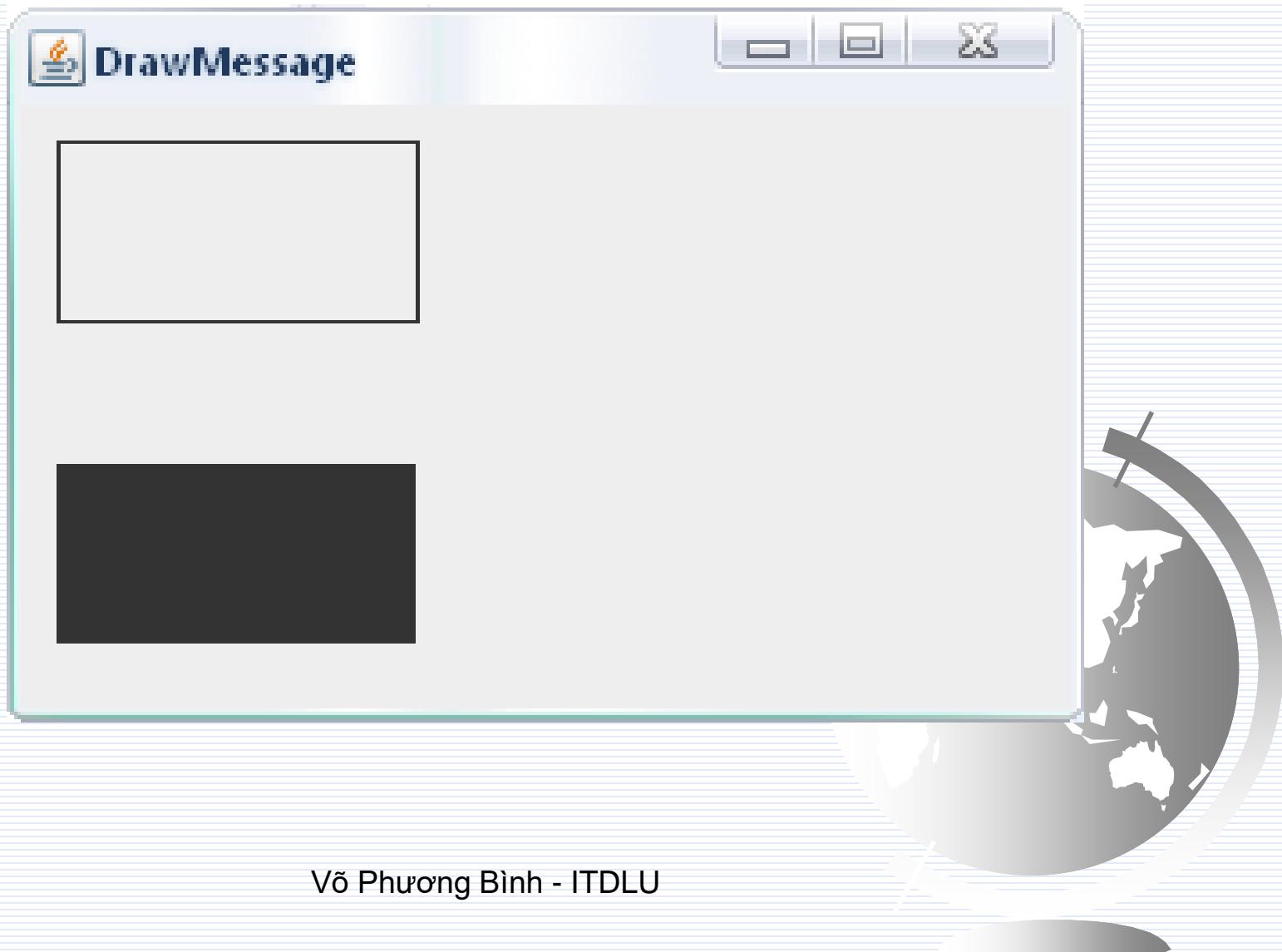


Ví dụ vẽ hình chữ nhật

```
✓ public class DrawMessage extends JPanel {  
✓     public static void main(String[] args) {  
✓         JFrame frame = new JFrame("DrawMessage");  
✓         frame.getContentPane().add(new DrawMessage());  
✓         frame.setSize(300, 200);  
✓         frame.setVisible(true);  
✓     }  
✓     public void paintComponent(Graphics g) {  
✓         super.paintComponent(g);  
✓         g.drawRect(10, 10, 100, 50);  
✓         g.fillRect(10, 100, 100, 50);  
✓     }  
✓ }
```

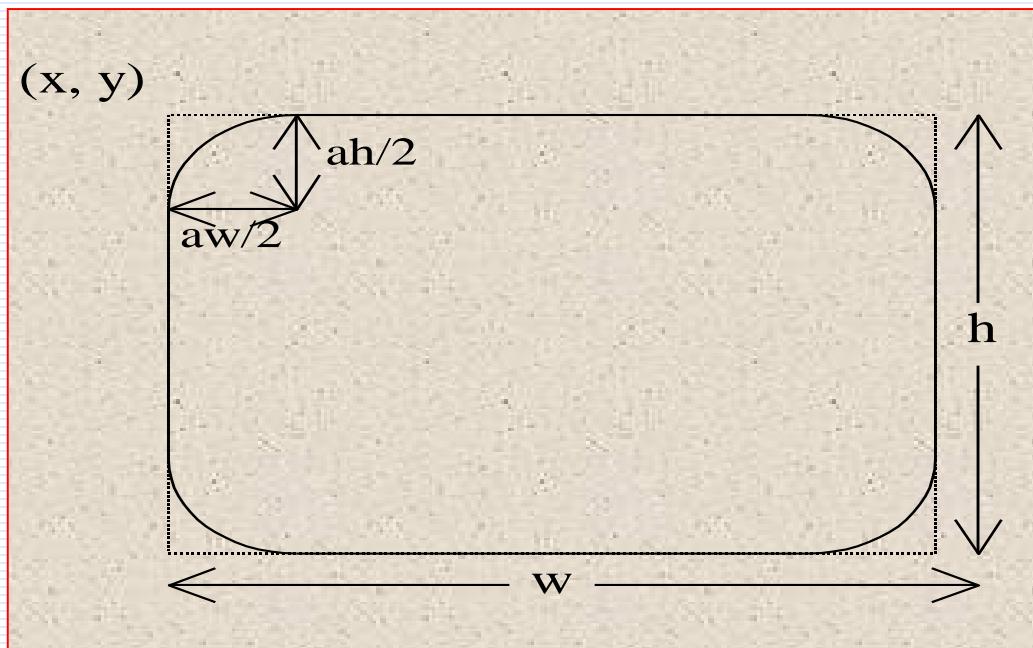


Ví dụ vẽ hình chữ nhật



Vẽ hình chữ nhật

```
drawRoundRect (x, y, w, h, aw, ah);  
fillRoundRect (x, y, w, h, aw, ah);
```

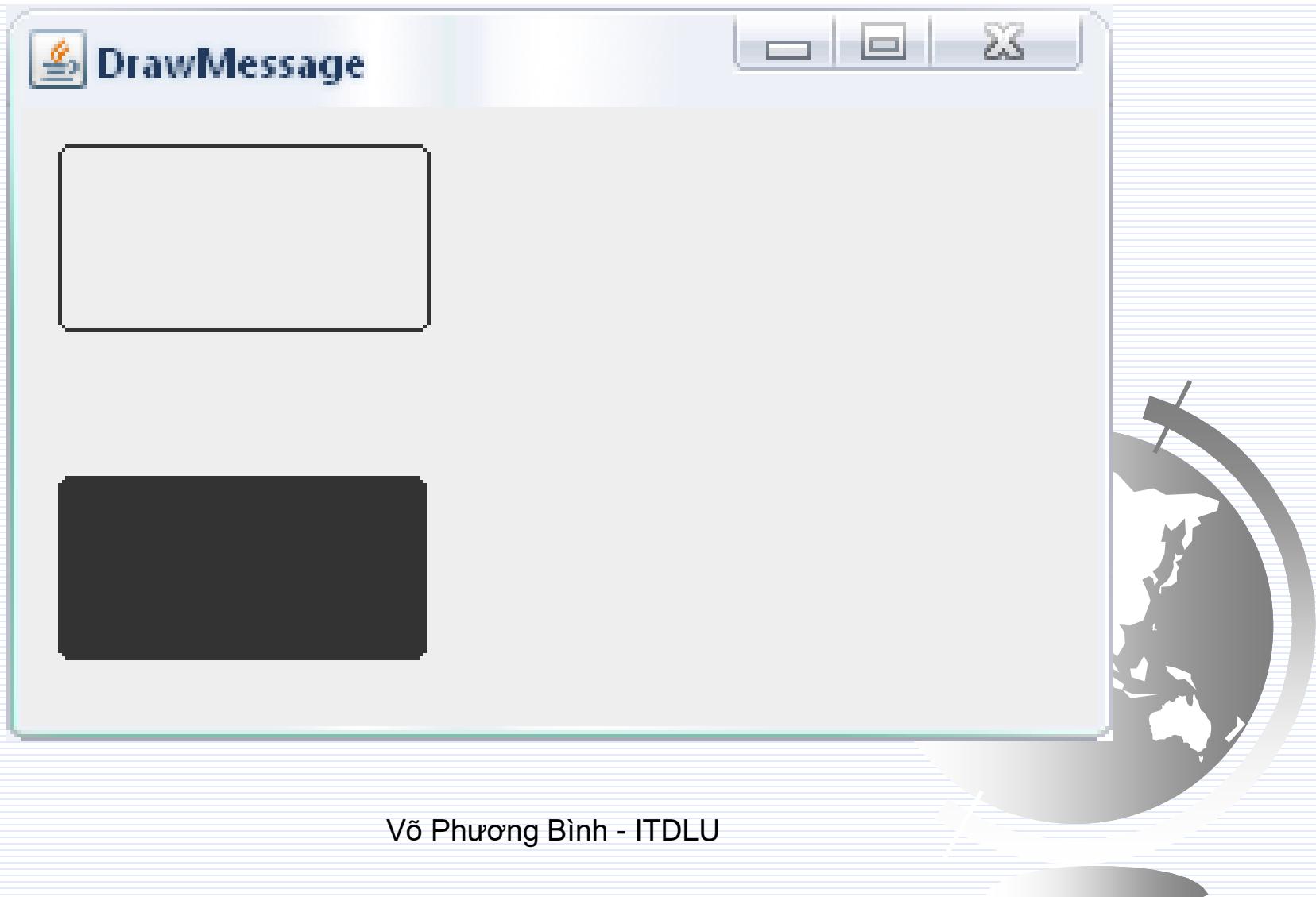


Ví dụ 2 vẽ hình chữ nhật

```
✓ public class DrawMessage extends JPanel {  
✓     public static void main(String[] args) {  
✓         JFrame frame = new JFrame("DrawMessage");  
✓         frame.getContentPane().add(new DrawMessage());  
✓         frame.setSize(300, 200);  
✓         frame.setVisible(true);  
✓     }  
✓     public void paintComponent(Graphics g) {  
✓         super.paintComponent(g);  
✓         g.drawRoundRect(10, 10, 100, 50, 5, 5);  
✓         g.fillRoundRect(10, 100, 100, 50, 5, 5);  
✓     }  
✓ }
```



Ví dụ 2 vẽ hình chữ nhật



Vẽ hình chữ nhật

✓ Vẽ hình chữ nhật 3D

- draw3DRect (int xCoord, int yCoord, int width, int height, boolean raised);

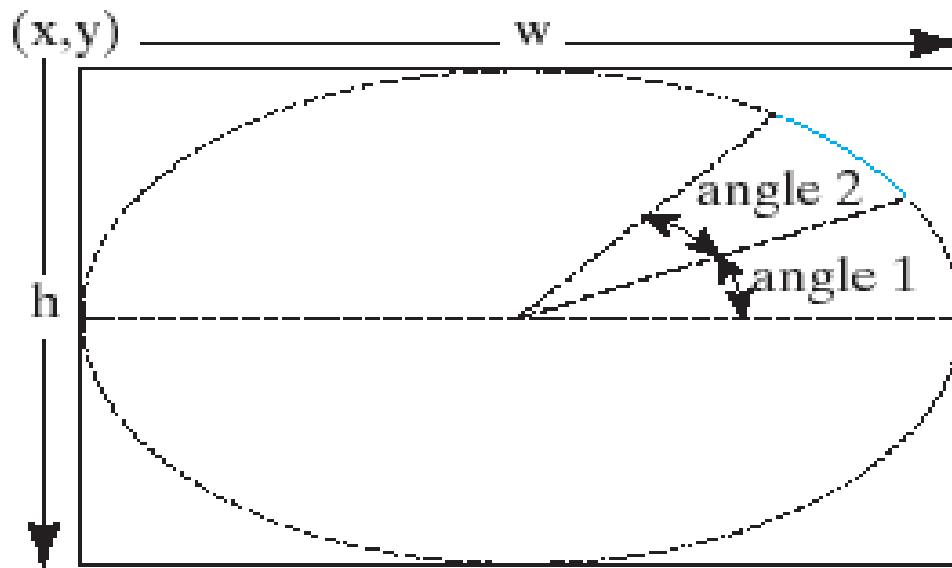


LẬP TRÌNH JAVA

Tiết 16: Vẽ hình chữ nhật 3D với `drawRect`,
`fillRect`
`drawRoundRect`, `fillRoundRect`

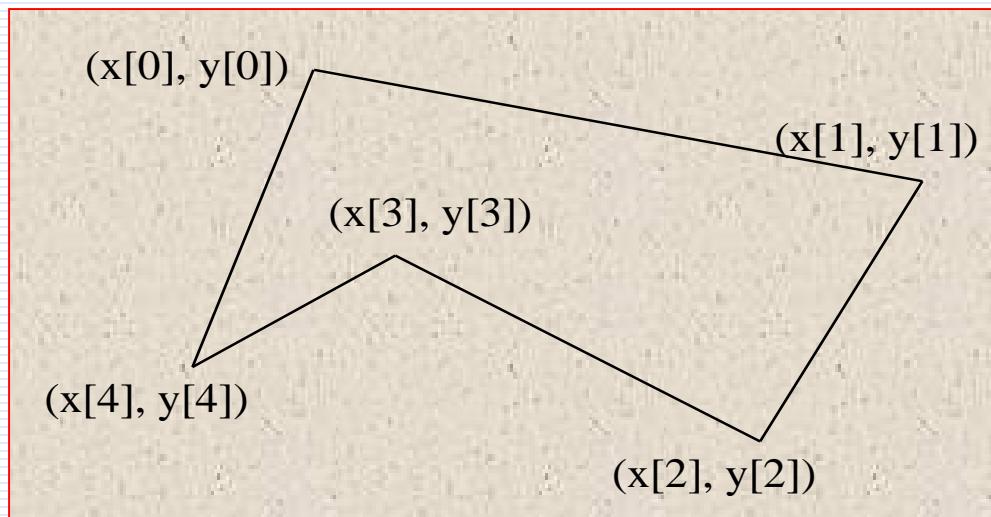
Vẽ hình cung (Arc)

- ✓ `drawArc(x, y, w, h, angle1, angle2);`
- ✓ `fillArc(x, y, w, h, angle1, angle2);`



Vẽ hình Polygon

```
int[] x = {40, 70, 60, 45, 20};  
int[] y = {20, 40, 80, 45, 60};  
g.drawPolygon(x, y, x.length);  
g.fillPolygon(x, y, x.length);
```



Điều khiển màu

- ✓ Java sử dụng mô hình màu RGB
 - color (int red, int green, int blue);
- ✓ Một số màu thông dụng

Màu	Red	Green	Blue
White	255	255	255
Light Gray	192	192	192
Gray	128	128	128
Dark Gray	64	64	64
Black	0	0	0
Pink	255	175	175
Orange	255	200	0
Yellow	255	255	0
Magenta	255	0	255

Điều khiển màu

- ✓ Lớp Color cung cấp sẵn một số màu:

color.white

color.black

color.orange

color.gray

color.lightgray

color.darkgray

color.red

color.green

color.blue

color.pink

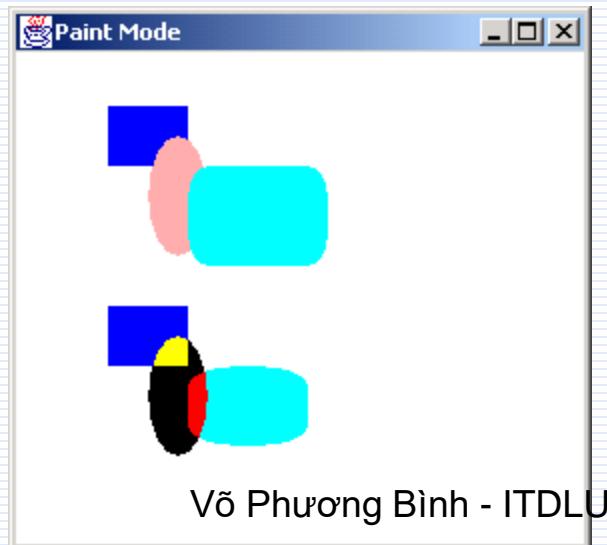
color.cyan

color.magenta

color.yellow

Chọn chế độ (mode) để vẽ

- ✓ Khi một đối tượng mới được vẽ, nó sẽ đè lên các hình đã vẽ trước đây.
- ✓ Để làm cho nội dung cũ và nội dung mới đều hiển thị trên cùng nền, lớp Graphics cung cấp phương thức `setXORMode (Color c)`.

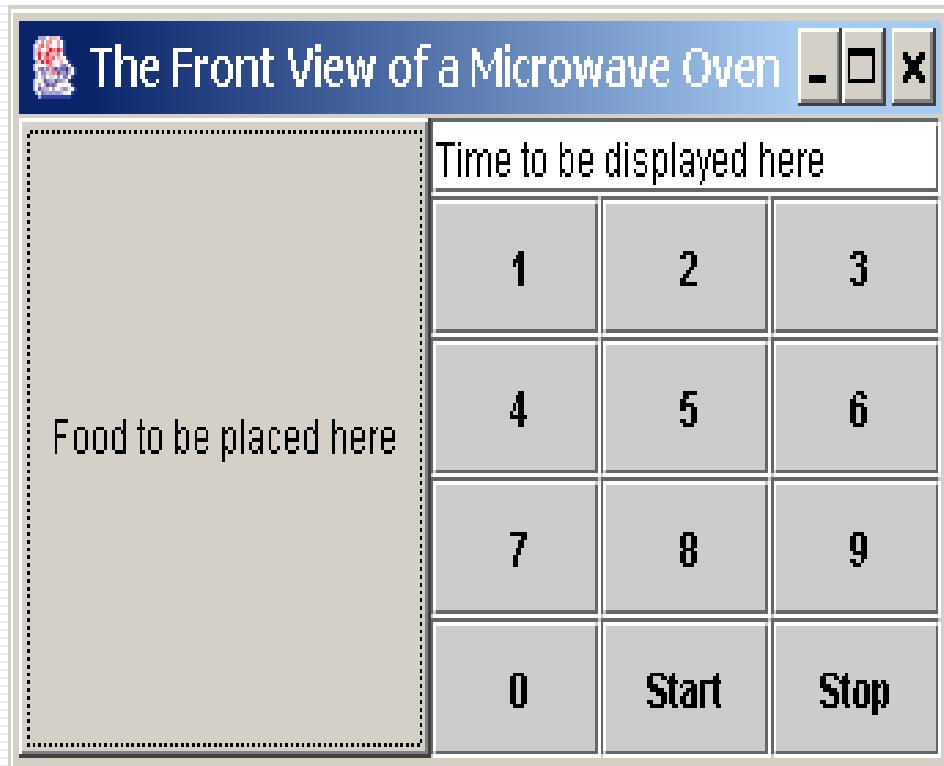


Chọn chế độ (mode) để vẽ

```
public class SetMode extends JPanel {  
    public static void main(String []args){  
        JFrame frm = new JFrame("SetMode");  
        frm.setSize(400,300);  
        frm.add(new SetMode());  
        frm.setVisible(true);  
    }  
    protected void paintComponent(Graphics g) {  
        super.paintComponent(g);  
        g.setPaintMode ();  
        g.setColor (Color.blue);  
        g.fillRect (50,50,40, 30);  
        g.setColor (Color.pink);  
        g.fillOval (70, 65, 30, 60);  
        g.setColor (Color.cyan);  
        g.fillRoundRect (90, 80, 70, 50, 20, 30);  
  
        g.setColor (Color.blue);  
        g.fillRect (50, 150, 40, 30);  
        g.setXORMode (Color.yellow);  
        g.fillOval (70, 165, 30, 60);  
        g.setXORMode (Color.magenta);  
        g.fillRoundRect (90, 180, 60, 40, 50, 20);  
    }  
}
```



1. Form mẫu



2. Form thông tin cá nhân

Exercise

Name

Address

Sex Male
 Female

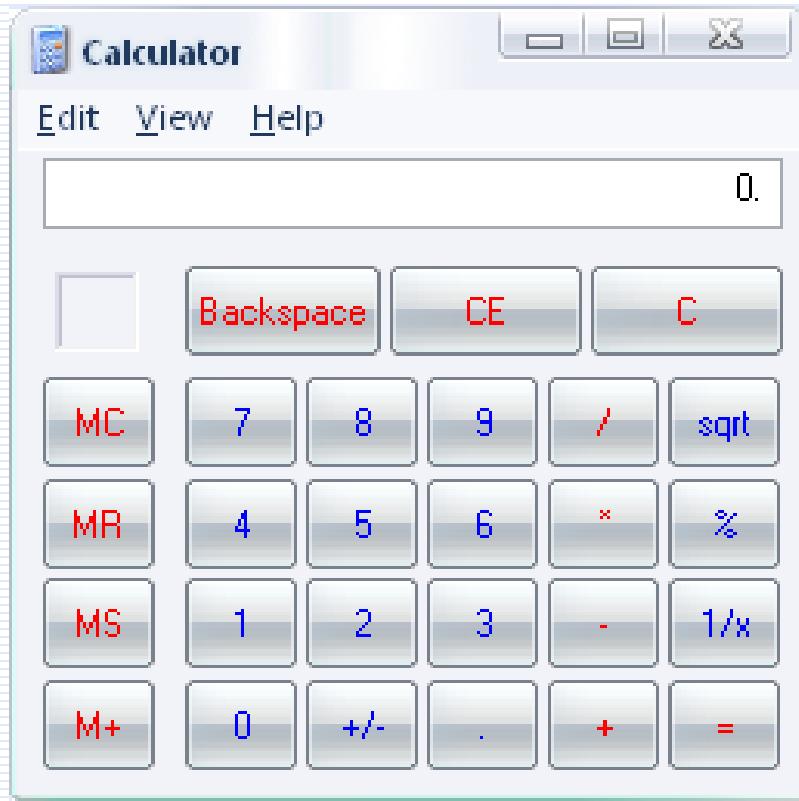
Hobbies Reading
 Sports
 Travelling

Age group

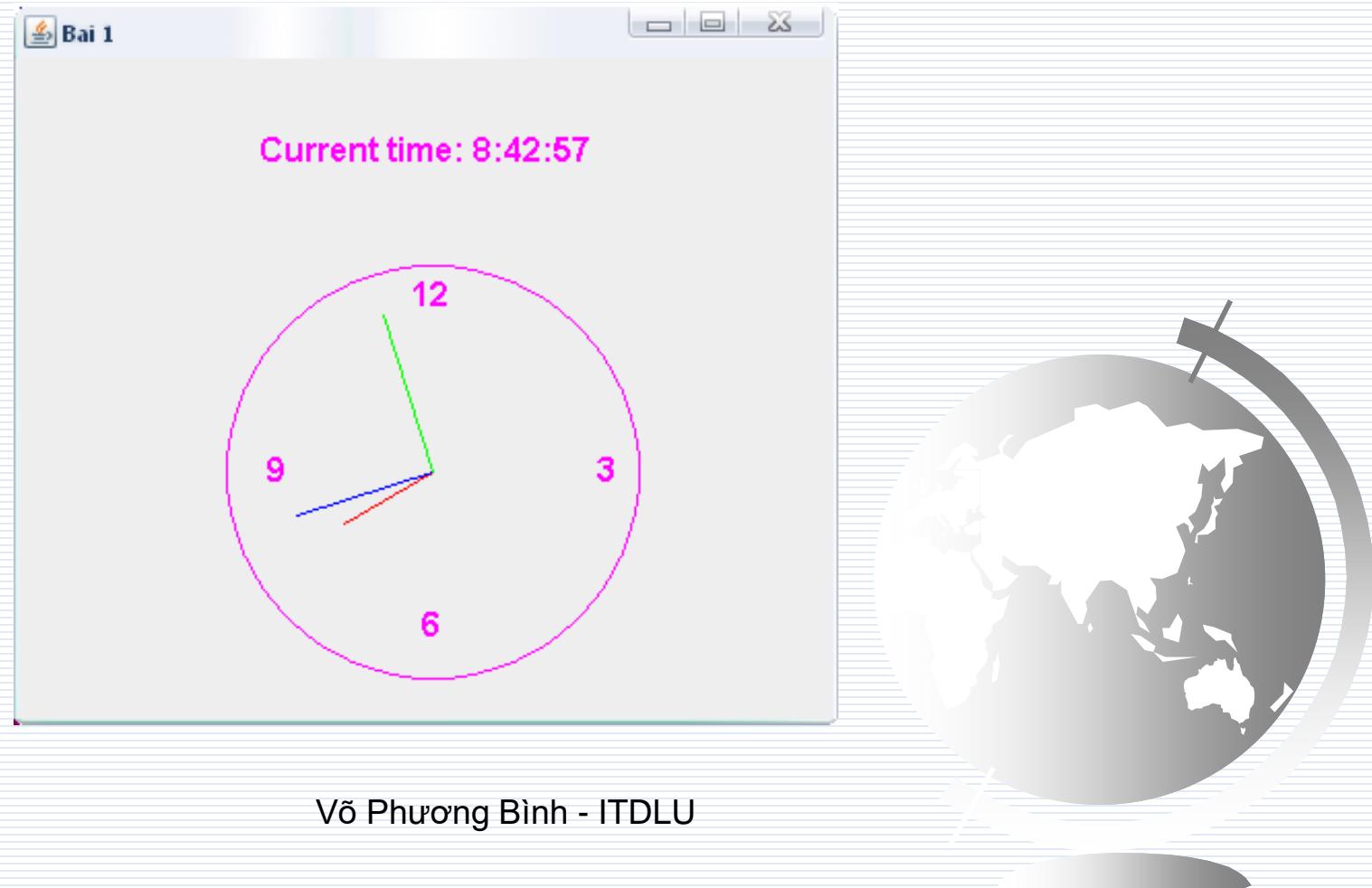
Ok Cancel



3. Form tính toán



4. Vẽ đồng hồ



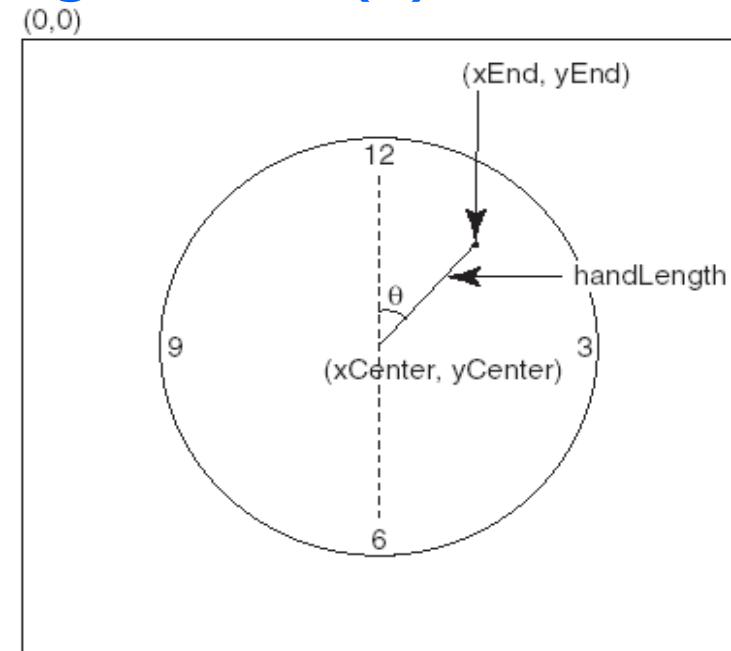
Drawing Clock

$$x_{\text{End}} = x_{\text{Center}} + \text{handLength} \times \sin(\theta)$$

$$y_{\text{End}} = y_{\text{Center}} - \text{handLength} \times \cos(\theta)$$

Since there are sixty seconds in one minute, the angle for the second hand is

$$\text{second} \times (2\pi/60)$$

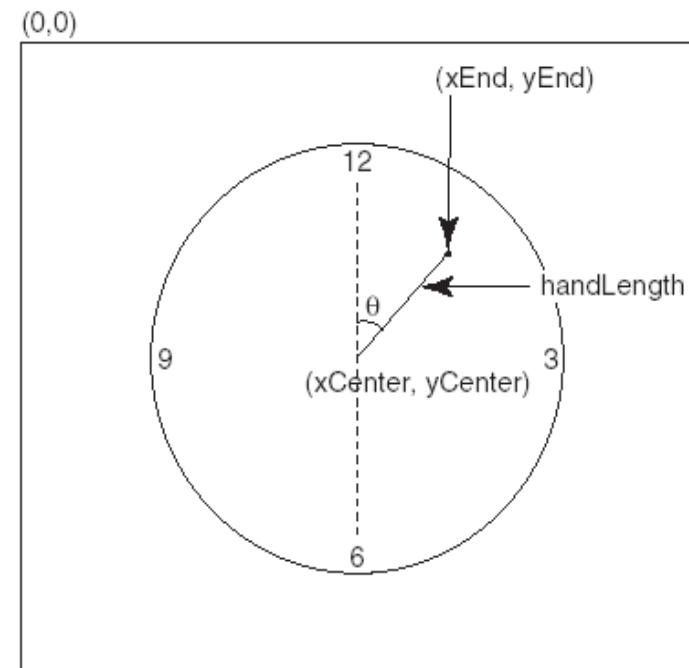


Drawing Clock, cont.

$$x_{\text{End}} = x_{\text{Center}} + \text{handLength} \times \sin(\theta)$$

$$y_{\text{End}} = y_{\text{Center}} - \text{handLength} \times \cos(\theta)$$

- The exact minute value combined with seconds is minute + second/60.
 - Since there are sixty minutes in one hour, the angle for the minute hand is $(\text{minute} + \text{second}/60) \times (2\pi/60)$



Drawing Clock, cont.

$$x_{\text{End}} = x_{\text{Center}} + \text{handLength} \times \sin(\theta)$$

$$y_{\text{End}} = y_{\text{Center}} - \text{handLength} \times \cos(\theta)$$

Since one circle is divided into twelve hours, the angle for the hour hand is $(\text{hour} + \text{minute}/60 + \text{second}/(60 \times 60))) \times (2\pi/12)$

