Lập trình hướng đối tượng

Giao diện và xử lý sự kiện

GV: ThS. Ngô Tiến Đức

Nội dung chính

- Các đối tượng giao diện cơ bản
- Các dạng layout cơ bản
- Xử lý sự kiện
- Java Swing

Các đối tượng giao diện cơ bản

AWT (Abstract Window Toolkit):

- Cung cấp các lớp dùng để lập trình giao diện người dùng dạng đồ họa (Graphical User Interface - GUI)
  - Để sử dụng cần import các lớp trong gói `java.awt`
- Là framework GUI cũ của Java
- Phụ thuộc vào các thành phần GUI của hệ điều hành

Các đối tượng giao diện cơ bản

Container: Đối tượng chứa, bao gồm các đối tượng thành phần (**component**) bên trong

- Quản lý các đối tượng component dưới dạng mảng
  - Chỉ số của đối tượng component là số thứ tự khi được thêm vào container
- Một số phương thức chung: `add(Object)`, `remove(Object)`, `removeAll()`, `getComponent(int)`, `getComponents()`, `countComponents()`,…

Các đối tượng giao diện cơ bản

Frame: Đối tượng có thể sử dụng như một container độc lập hoặc đóng vai trò component của một đối tượng container khác

- Thường dùng để tạo ra cửa sổ chính của chương trình

```
import java.awt.*;
...
Frame myFrame = new Frame("My frame");
myFrame.setSize(300, 150);
myFrame.setResizable(false);
myFrame.setVisible(true); // hiển thị frame
```

Các đối tượng giao diện cơ bản

Panel: Sử dụng để nhóm các đối tượng giao diện với nhau, thường nằm trong Frame

```
Frame myFrame = new Frame("My frame");
myFrame.setSize(300, 150);
Panel myPanel = new Panel();
myFrame.add(myPanel); // thêm panel vào frame
myFrame.setVisible(true);
```

Các đối tượng giao diện cơ bản

Dialog: Cửa sổ con của chương trình, phải đi cùng với Frame

- Modal: Khi hiển thị sẽ khóa các cửa sổ khác của chương trình
- Non-modal: Khi hiển thị vẫn có thể thao tác với các cửa sổ khác

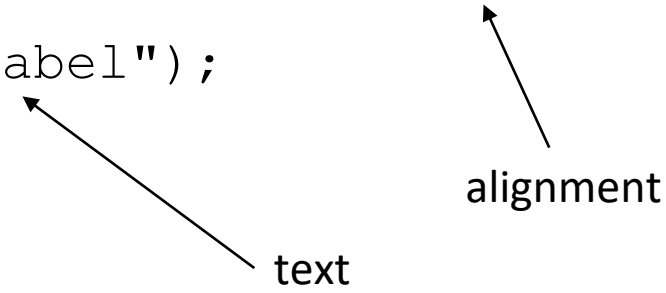
```
Frame myFrame = new Frame("My frame");
myFrame.setSize(300, 150);
Dialog myDialog = new Dialog(myFrame, "My dialog", true);
myDialog.setSize(200, 100);
myFrame.setVisible(true);
myDialog.setVisible(true);
```

is modal?

Các đối tượng giao diện cơ bản

Label: Component hiển thị văn bản tĩnh (không thể sửa trực tiếp)

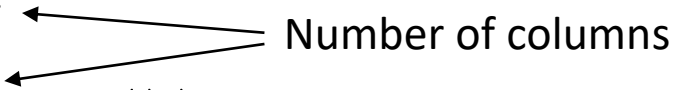
```
Frame myFrame = new Frame("My frame");
myFrame.setSize(300, 150);
// Label myLabel = new Label("My label", Label.CENTER);
// Label myLabel = new Label("My label");
Label myLabel = new Label();
myLabel.setText("My label");
myLabel.setAlignment(Label.CENTER);
myFrame.add(myLabel);
myFrame.setVisible(true);
```



Các đối tượng giao diện cơ bản

TextField: Component cho phép người dùng nhập một dòng văn bản

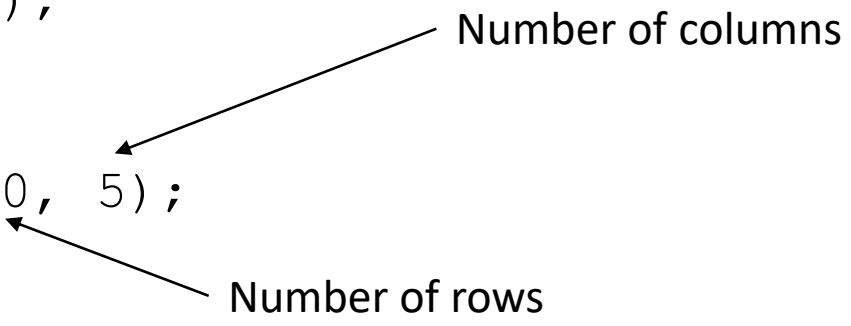
```
Frame myFrame = new Frame("My frame");
myFrame.setSize(300, 150);
TextField myTextField = new TextField("A text field!");
// TextField myTextField = new TextField(20);
// myTextField.setText(""+myTextField.getColumns());
// myTextField.setEditable(false);
myFrame.add(myTextField);
myFrame.setVisible(true);
```



Các đối tượng giao diện cơ bản

TextArea: Component cho phép người dùng nhập đoạn văn bản có chứa nhiều dòng

```
Frame myFrame = new Frame("My frame");
myFrame.setSize(300, 150);
TextArea myTextArea = new TextArea(20, 5);
myTextArea.setText("text\narea");
// TextArea myTextArea = new TextArea("text\narea", 20, 5);
myFrame.add(myTextArea);
myFrame.setVisible(true);
```



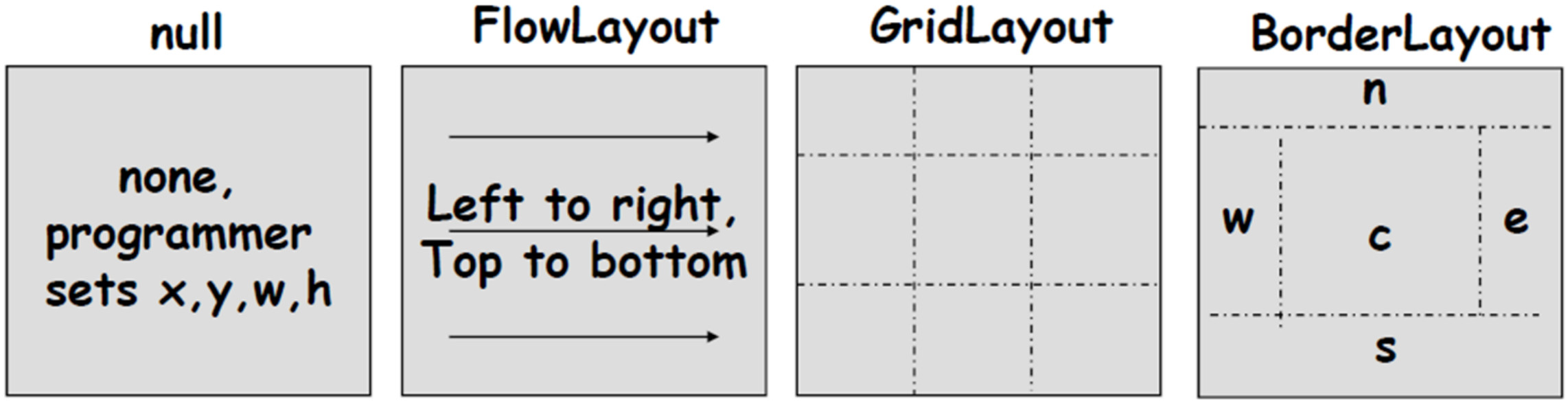
Các đối tượng giao diện cơ bản

Một số đối tượng khác:

- **Button:** Nút bấm
- **Checkbox:** Đánh dấu/chọn một hoặc nhiều ô lựa chọn, các ô lựa chọn được nhóm lại và trong **CheckboxGroup**
- **Radio:** Đánh dấu/chọn một ô lựa chọn, các ô lựa chọn được nhóm lại và đặt trong **RadioGroup**
- **Choice:** Menu sổ xuống cho phép chọn một trong số các lựa chọn (item)
- **List:** Menu sổ xuống cho phép chọn một hoặc nhiều item

Các dạng layout cơ bản

Layout Manager: Cách sắp xếp bố cục các component trên container

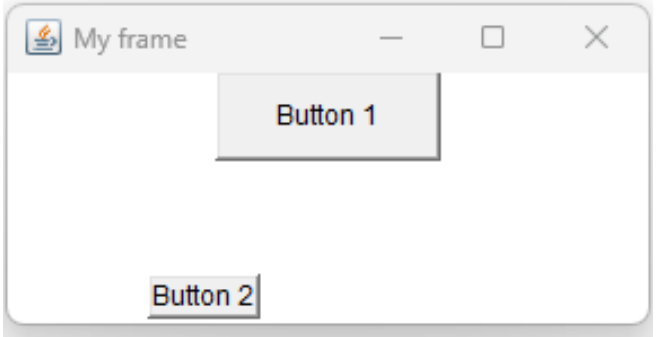


Các dạng layout cơ bản

Null Layout: Không có quy tắc, người dùng tự định vị và thiết lập kích thước cho từng component

```
myFrame.setLayout(null);
Button btn1 = new Button("Button 1");
btn1.setSize(100, 40);
btn1.setLocation(new Point(100, 30));
Button btn2 = new Button("Button 2");
btn2.setBounds(70, 120, 50, 20);
myFrame.add(btn1);
myFrame.add(btn2);
```

size position



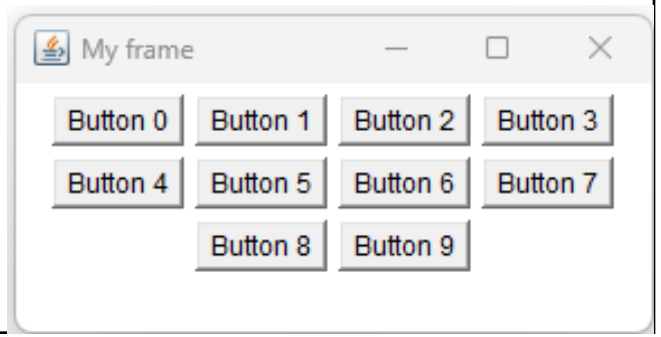
Các dạng layout cơ bản

Flow Layout: Sắp xếp các đối tượng trên một hướng theo dòng

- Nếu container không đủ rộng thì component nhảy xuống dòng mới
- Kích thước và khoảng cách của các component được set tự động

```
myFrame.setLayout(new FlowLayout());
// myFrame.setLayout(new FlowLayout(FlowLayout.RIGHT));
for (int i = 0; i < 10; i++) {
      myFrame.add(new Button("Button " + i));
}
```

alignment



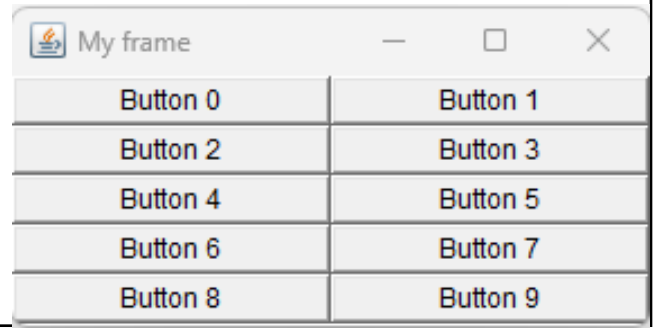
Các dạng layout cơ bản

Grid Layout: Sắp xếp các đối tượng theo dạng bảng gồm hàng + cột

- Các component được định vị theo vị trí các ô (cell)
- Số component thiếu/vượt quá số lượng ô: Số cột được tự điều chỉnh
- Để tính số hàng/cột tự động: Cho một trong hai nhận giá trị 0

```
myFrame.setLayout(new GridLayout(5, 2));
// myFrame.setLayout(new GridLayout(3, 0, 3, 5));
for (int i = 0; i < 10; i++) {
      myFrame.add(new Button("Button " + i));
}
```

Distance of row/columns

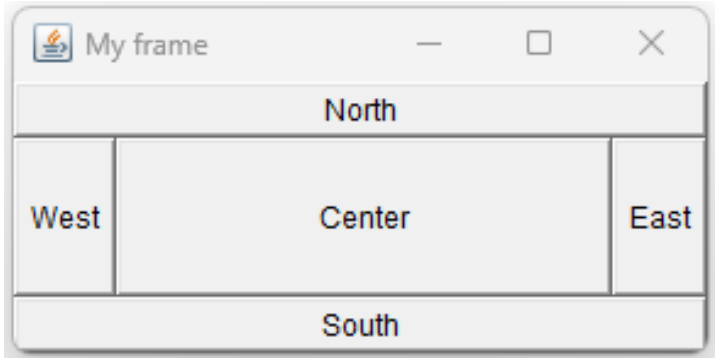


Các dạng layout cơ bản

Border Layout: Chia frame thành 5 vùng cố định

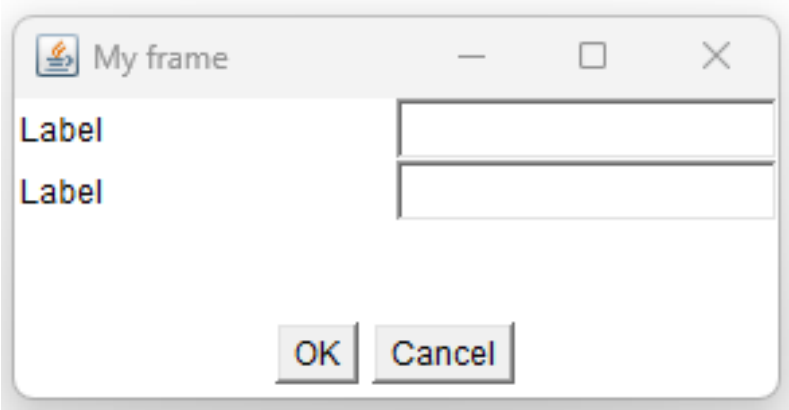
- Nếu chỉ có vùng Center: Tự động lấp đầy frame
- Nếu gắn nhiều đối tượng vào cùng một vùng: Chỉ thấy đối tượng sau

```
myFrame.setLayout(new BorderLayout());
myFrame.add("West", new Button("West"));
myFrame.add("East", new Button("East"));
myFrame.add("North", new Button("North"));
myFrame.add("South", new Button("South"));
myFrame.add("Center", new Button("Center"));
```



Các dạng layout cơ bản

- Một số layout khác: GridBag Layout, Card Layout, Box Layout,...
- Có thể kết hợp nhiều layout trên cùng frame bằng cách thêm nhiều các panel vào frame



Các dạng layout cơ bản

```
Panel panel1 = new Panel();
Panel panel2 = new Panel();
panel1.setLayout(new GridLayout(2, 2));
panel1.add(new TextField());
panel1.add(new Label("Label")); ...
panel2.setLayout(new FlowLayout());
panel2.add(new Button(“OK”)); ...
myFrame.setLayout(new BorderLayout());
myFrame.add(BorderLayout.NORTH, panel1);
myFrame.add(BorderLayout.SOUTH, panel2);
```

Xử lý sự kiện

- Sự kiện (Events): Một kiểu tín hiệu thông báo tới chương trình có một sự việc/hành động nào đó đã xảy ra
- Sự kiện được sinh ra bởi các hành động của người dùng, ví dụ như di chuyển chuột, click chuột, gõ phím,...
- Khi khai báo hàm xử lý sự kiện cho một đối tượng, hàm sẽ được gọi bất cứ khi nào sự kiện xảy ra
- Java cung cấp một số lớp sự kiện cơ bản nằm trong gói `java.awt.event`

Xử lý sự kiện

Các lớp sự kiện cơ bản:

- **ActionEvent**: Xuất hiện khi một nút bị click vào, một danh sách (list) được chọn, một menu được chọn
- **ComponentEvent**: Xuất hiện khi một component bị thay đổi kích cỡ, vị trí, trạng thái
- **FocusEvent**: Xuất hiện khi một component có hoặc mất focus
- **ItemEvent:** Xuất hiện khi một menu item được chọn hoặc bỏ, khi checkbox hoặc list item được click vào

Xử lý sự kiện

Các lớp sự kiện cơ bản:

- **WindowEvent**: Xuất hiện khi một của sổ được mở ra, kích hoạt, đóng lại hoặc thoát ra
- **TextEvent**: Xuất hiện khi giá trị văn bản của các đối tượng TextField và TextArea bị thay đổi
- **MouseEvent**: Xuất hiện khi chuột được click, di chuyển qua, nhấn xuống và thả ra
- **KeyEvent**: Xuất hiện khi có đầu vào từ bàn phím

Xử lý sự kiện

Listener: Lắng nghe và xử lý các sự kiện đã được khai báo

- Các interface **Listener** được cài đặt để xử lý các sự kiện tương ứng:
  - **ActionListener**: ComponentListener, FocusListener
  - **ItemListener**: WindowListener, TextListener, KeyListener, MouseListener, MouseMotionListener
- Nếu lớp xử lý sự kiện `implements` interface Listener thì phải viết cài đặt cho phương thức `actionPerformed` của interface
- Xác định component phát sinh sự kiện: Dùng phương thức `getSource`

Xử lý sự kiện

- Đơn giản: Gọi các phương thức listener từ component
- Ví dụ ấn nút để hiển thị dialog:

```
Dialog myDialog = new Dialog(myFrame, "Dialog");
myDialog.setSize(200, 100);
myDialog.add(new Label("This is dialog"));
Button button = new Button("Show dialog");
button.addActionListener((e) -> {
      myDialog.setVisible(true);
});
```

Xử lý sự kiện

Ví dụ: Đóng dialog hoặc frame khi ấn nút đóng mặc định

```
myDialog.addWindowListener(new WindowAdapter() {
      public void windowClosing(WindowEvent e) {
            myDialog.dispose();
      }
});
myFrame.addWindowListener(new WindowAdapter() {
      public void windowClosing(WindowEvent e) {
            myFrame.dispose(); // or System.exit(0);
      }
});
```

Xử lý sự kiện

```
import java.awt.*;
import java.awt.event.*;
public class Demo extends Frame implements ActionListener {
    Button button;
    public Demo() {
        super("My frame");
        button = new Button("Show Dialog");
        button.addActionListener(this);
        this.add(button);
        this.setSize(300, 150);
        this.addWindowListener(new WindowAdapter() {
            @Override
            public void windowClosing(WindowEvent e) {
                System.exit(0);
            }
        });
    }
```

Xử lý sự kiện

```
public void actionPerformed(ActionEvent e) {
        Dialog myDialog = new Dialog(this, "My Dialog");
        myDialog.setSize(200, 100);
        myDialog.add(new Label("This is dialog"));
        myDialog.addWindowListener(new WindowAdapter() {
            @Override
            public void windowClosing(WindowEvent e) {
                myDialog.dispose();
            }
        });
        myDialog.setVisible(true);
    }
    public static void main(String[] args) {
        Demo frame = new Demo();
        frame.setVisible(true);
    }
}
```

Java Swing

Java Swing:

- Để sử dụng cần import các lớp trong gói `javax.swing`
- Framework GUI mới của Java mở rộng từ AWT
- Không phụ thuộc vào các thành phần GUI của hệ điều hành
- Các đối tượng container và component cơ bản trong Swing có thêm chữ "J" ở đầu tên lớp. VD: Frame -> JFrame, Button -> JButton, Label -> JLabel,...

Java Swing

Các lớp mở rộng có đầy đủ phương thức của các lớp cơ bản của và được bổ sung một số phương thức tạo hiệu ứng giao diện. VD với **JButton**

- JButton(String, Icon): Khởi tạo một nút với ảnh nền kiểu icon
- setBorder(new MatteBorder(int, int, int, int, Icon)): Thiết lập khung nền cho nút với các tham số: Khoảng cách từ chữ đến biên theo các chiều trên -> dưới -> trái -> phải
- setBorder(new LineBorder(int)): Thiết lập viền cho nút dạng hình chữ nhật + màu viền
- setToolTipText(String): Thiết lập dòng chữ hiển thị khi di chuột lên nút

Java Swing

JFrame có nhiều tầng chồng lên nhau -> cảm giác các đối tượng được trình bày trên cùng một mặt phẳng như Frame

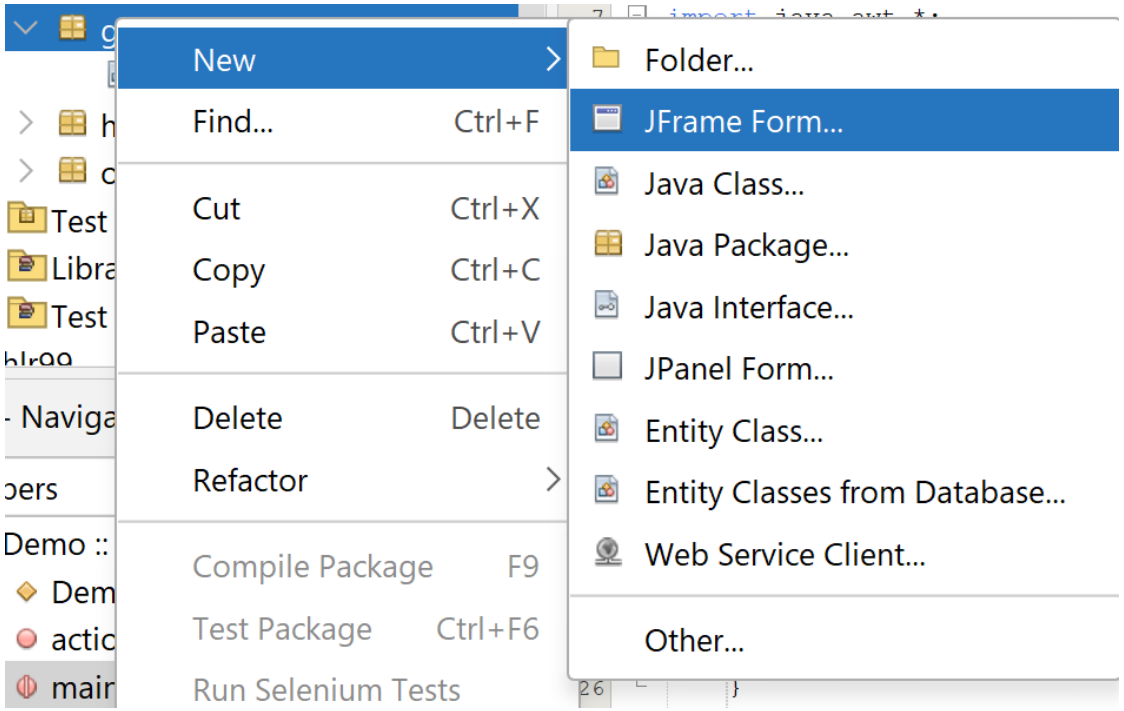
- **ContentPane**: Tầng thường dùng nhất, chứa các component cơ bản
- **TabbedPane**: Chứa các container như JPanel trong một cửa sổ duy nhất và tổ chức thành các tab khác nhau
- **MenubarPane**: Chứa các loại menu. VD: Menubar, PopupMenu,...
- **GlassPane**: Chứa các đối tượng nằm trên cùng. VD: Tooltip,...
- ...

Ví dụ:

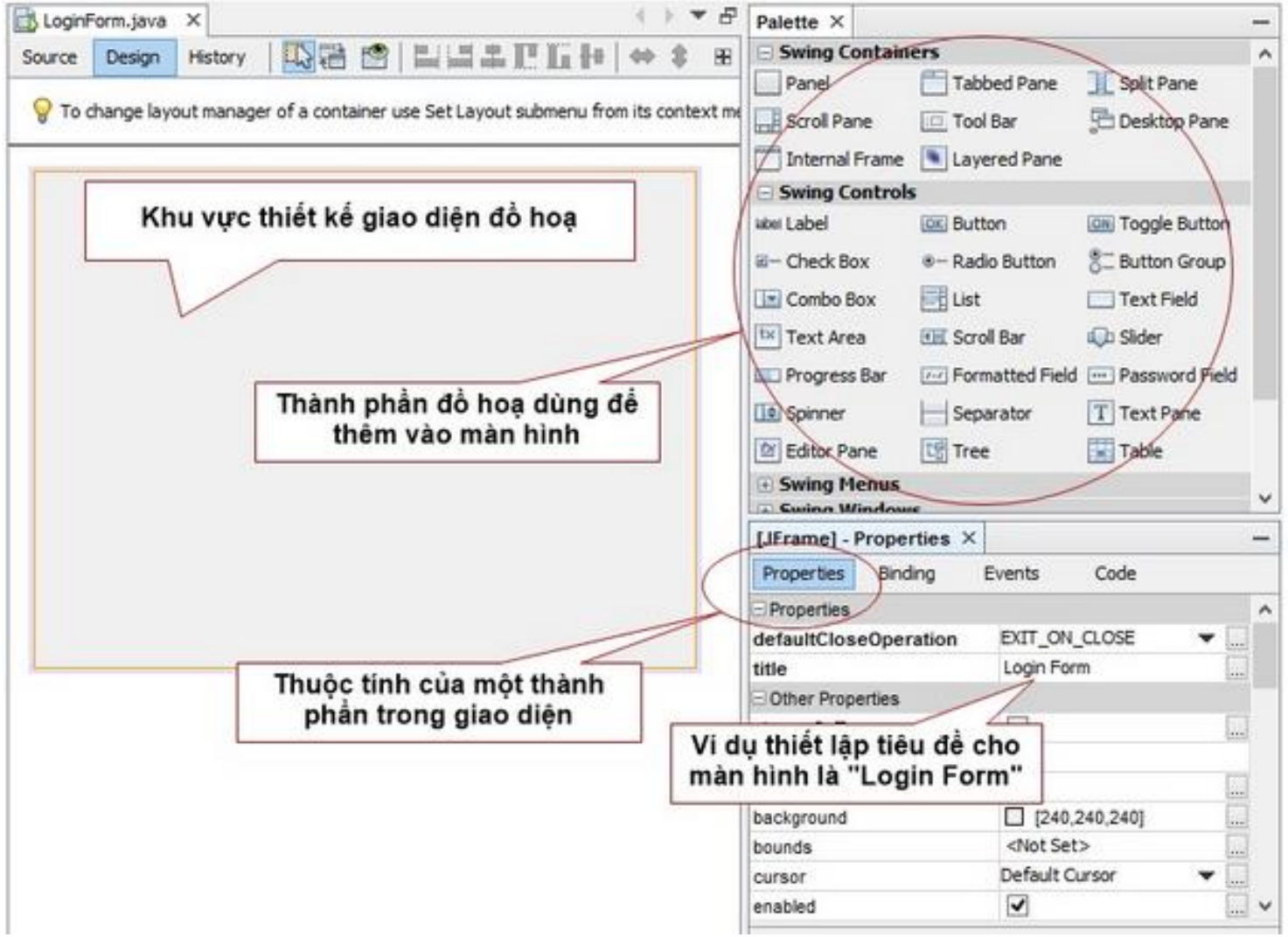
```
public JFrameDemo(){
        super("JFrame demo");
        JButton btn = new Jbutton();
        this.getContentPane().add("Center", btn);
        // Gắn button vào tầng ContentPane
        …
```

Java Swing

Kéo thả (Drag and Drop): Các thao tác thiết kế giao diện trực quan và nhanh hơn sử dụng Java Swing



Java Swing


LoginForm.java
Source
Design
History
To change layout manager of a container use Set Layout submenu from its context me
Khu vực thiết kế giao diện đồ hoạ
Thành phần đồ hoạ dùng để thêm vào màn hình
Thuộc tính của một thành phần trong giao diện
Palette
Swing Containers
Panel
Tabbed Pane
Split Pane
Scroll Pane
Tool Bar
Desktop Pane
Internal Frame
Layered Pane
Swing Controls
Label
Button
Toggle Button
Check Box
Radio Button
Button Group
Combo Box
List
Text Field
Text Area
Scroll Bar
Slider
Progress Bar
Formatted Field
Password Field
Spinner
Separator
Text Pane
Editor Pane
Tree
Table
Swing Menus
[JFrame] - Properties
Properties
Binding
Events
Code
Properties
defaultCloseOperation
EXIT_ON_CLOSE
title
Login Form
Other Properties
background
[240,240,240]
bounds
<Not Set>
cursor
Default Cursor
enabled
Ví dụ thiết lập tiêu đề cho màn hình là "Login Form"

Luyện tập

Viết chương trình thiết kế một giao diện đăng nhập gồm có tên đăng nhập và mật khẩu, hai nút "Đăng nhập" và "Đặt lại"

- Khi nhập đúng tên đăng nhập "java" và mật khẩu "ptit" sau đó ấn nút đăng nhập, hiện dialog "Đăng nhập thành công", nhập sai thì hiện dialog "Sai tên đăng nhập hoặc mật khẩu"
- Khi ấn nút "Đặt lại" thì xóa trắng 2 trường tên đăng nhập và mật khẩu

Tài liệu tham khảo

- Liang, Y. Daniel, Introduction to Java programming and Data Structures, 11th edition, Pearson Prentice Hall, 2019
- N. M. Sơn, Bài giảng Lập trình hướng đối tượng, HVCNBCVT, 2020
- N. M. Sơn, Slide giảng dạy môn Lập trình hướng đối tượng
- T. T. V. Anh, Slide giảng dạy môn Lập trình hướng đối tượng