

Bài 1: Tổng quan về Lập trình giao diện Java



Mục tiêu bài học

- Giới thiệu gói thư viện AWT
- So sánh AWT và Swing
- Cài đặt Swing trên NetBean
- Khởi tạo một Java App cơ bản
- Container Component (JFrame, JPanel)

Giới thiệu tổng quan về



JFC là gì ?

Java Foundation Classes

Tập hợp các tính năng để xây dựng giao diện người dùng đồ họa – GUI, để tạo ra các chức năng đồ họa phong phú và tương tác với các ứng dụng Java

JFC là một dự án phối hợp giữa Netscape's Internet Foundation Classes(IFC) và IBM's Taligent division and Lighthouse Design

JFC bao gồm 5 thư viện:

- Swing GUI Components
- AWT
- Pluggable Look and Feel Support
- Accessibility API
- Java 2D API

JFC là gì

- Look and feel Manager: Bộ phận quản lý diện mạo nhằm làm thay đổi bộ mặt của chương trình Java. Chẳng hạn, có thể làm cho ứng dụng chạy trên PC với hệ điều hành Windows trông giống như chương trình chạy trên máy với hệ điều hành Macintosh (Mac OS), hoặc của Motif trong UNIX, v.v...
- Thành phần hỗ trợ người khuyết tật (Accessibility Application Programming Interface), bằng cách làm cho ứng dụng tương tác với các phương tiện khác như bộ đọc chữ Braille, bộ đọc chữ từ màn hình (Screen Reader)

JFC là gì

- Bộ phận hỗ trợ đồ hoạ 2 và 3 chiều
- Drag and Drop Support, nhằm hỗ trợ tương tác giữa người và máy trên mọi hệ điều hành, qua đó có thể trao đổi dữ liệu giữa chương trình Java và không phải Jav


AWT

Giới thiệu chung

(**A**bstract **W**indows **T**oolkit)

- Thư viện API cung cấp các đối tượng GUI
- Tạo liên kết giao diện giữa ứng dụng Java và OS
- Chiếm nhiều tài nguyên hệ thống (Heavy-weight component)
- Package java.awt
- Gồm nhiều phần tử (class) để tạo GUI.
- Có các lớp quản lý việc bố trí các phần tử.
- Có (**event-oriented application**) mô hình ứng dụng hướng sự kiện.
- Có các công cụ xử lý đồ họa và hình ảnh.
- Các lớp sử dụng các tác vụ với clipboard (vùng nhớ đệm) như cut, paste.



DEMO 
AWT 

Demo AWT

```
import java.awt.*;
public class DemoAWT extends Frame
{
    public DemoAWT(String title)
    {
        super(title);
        this.setBounds(100,150,200,200);
        this.setVisible(true);
    }
    public static void main (String[] args) {
        new DemoAWT("First App");
    }
}
```

SWING

Giới thiệu chung

SWING không phải là một từ viết tắt, Swing là một sản phẩm của gia đình lớn Java, một phần của dự án JFC

Swing bắt đầu được phát triển từ bản beta của JDK 1.1, khoảng mùa xuân năm 1997, đến tháng 3 năm 1998 thì bắt đầu được công bố rộng rãi

Khi phát hành thư viện Swing 1.0 chứa 250 class và 80 interface, đến nay phiên bản Swing 1.4 chứa 451 class và 85 interface

Mặc dù Swing được phát triển riêng rẽ với phần lõi của Java Development Kit, nó đòi hỏi ít nhất JDK 1.1.5 để chạy

SWING

Giới thiệu chung

SWING là sự mở rộng AWT

- Tạo giao diện nhất quán độc lập với môi trường(cross-platform GUI)
- Đa luồng và nhẹ(Light-weight component) với nhiều đặc điểm nâng cấp
- Có khả năng tùy biến tại thời điểm runtime
- Không sử dụng trộn lẫn AWT và SWING GUI component trên cùng một giao diện
- Package java.swing

SWING

Giới thiệu chung

- Sự xuất hiện thêm Swing từ Java 1.2 nhằm giúp khắc phục sự khó khăn của AWT khi cần bổ sung thêm các widget mới
- Đặc biệt, khắc phục bộ mặt nghèo nàn của chương trình viết bằng AWT so với các giao diện đẹp đẽ khác (chẳng hạn với MFC (Microsoft Foundation Classes) của Microsoft

So sánh AWT và SWING

AWT

- Xây dựng bằng native code
- Khó phát triển thêm các linh kiện(widget) mới

SWING

- Xây dựng hoàn toàn bằng JAVA API
- Dễ phát triển các linh kiện
- Có thể thay đổi diện mạo của linh kiện lúc runtime
- Mô hình MVC
(Model – View – Controller)

So sánh AWT và SWING

AWT

SWING

Có thể bổ sung thêm biểu tượng bên cạnh dòng chữ vào Label, Button, Menu item

- Tạo đường viền và ghi tựa cho các thành
- phần Swing

So sánh AWT và SWING

AWT

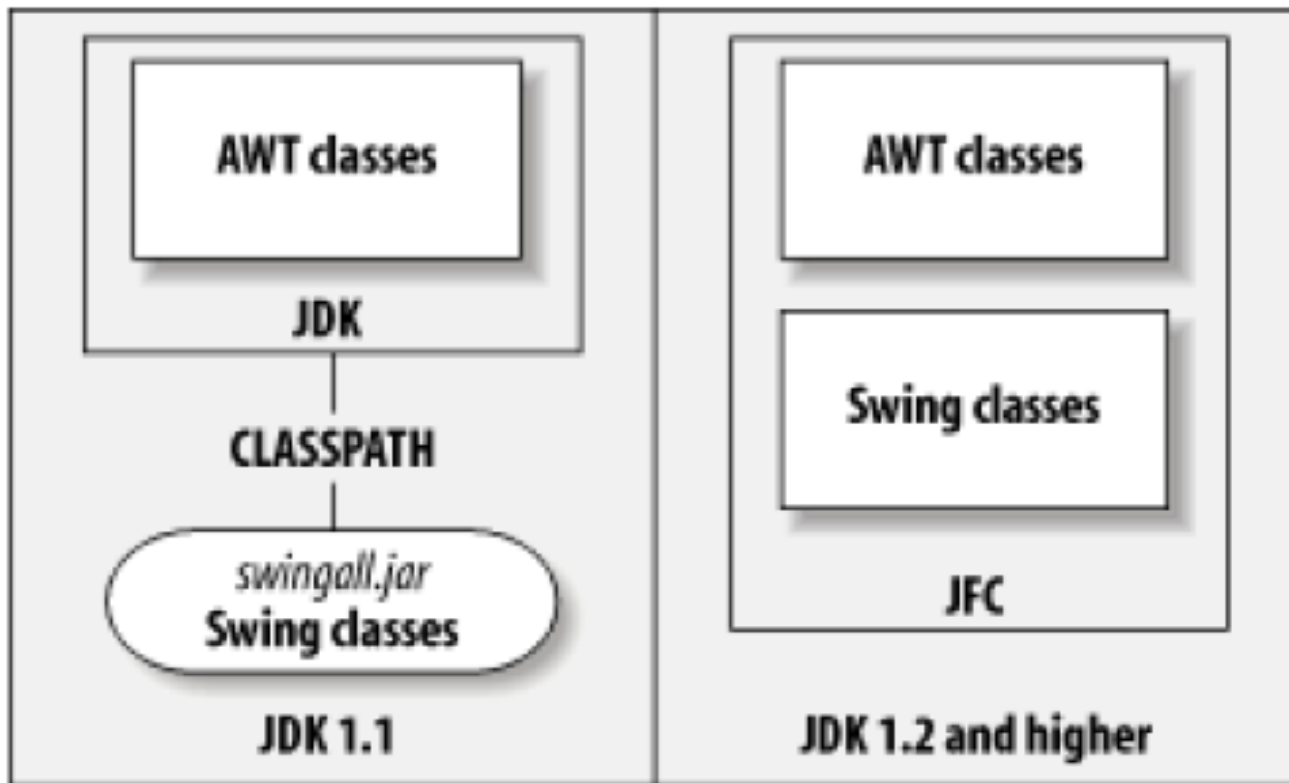
SWING

- Có thể chọn một thành phần Swing bằng
- Cách dùng phím thay cho chuột
- Các thành phần Swing sử dụng các nhãn Unicode, vì vậy có thể đưa tiếng Việt vào các thành phần này

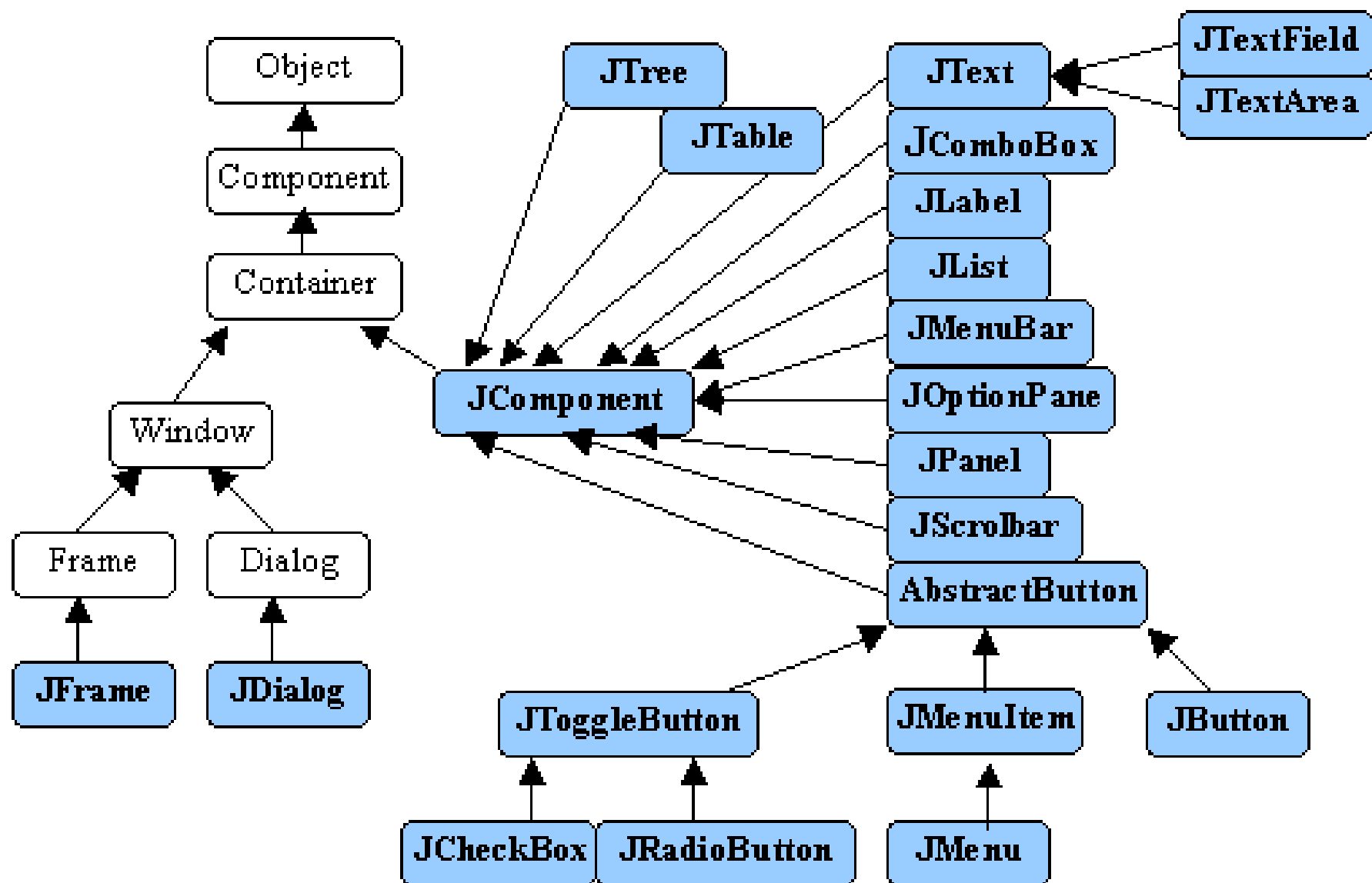
Vậy SWING là sự thay thế của AWT?

Không. Swing thực tế được xây dựng trên phần lõi của AWT, bởi vì Swing không chứa bất kỳ mã dành cho nền tảng nào (native code)

Mô tả mối quan hệ giữa AWT, SWING, và JDK:



Kiến trúc SWING



Dịch và chạy một chương trình Swing

- Bước **1**: tải và cài đặt phiên bản JDK mới nhất tại <http://www.oracle.com/technetwork/java/javase/downloads/index.html>
- Bước **2**: cấu hình biến môi trường, JavaHome và PATH
- Bước **3**: download chương trình Swing demo "HelloWorldSwing.java"
<https://docs.oracle.com/javase/tutorial/uiswing/examples/start/HelloWorldSwingProject/src/start/HelloWorldSwing.java>

Dịch và chạy một chương trình Swing

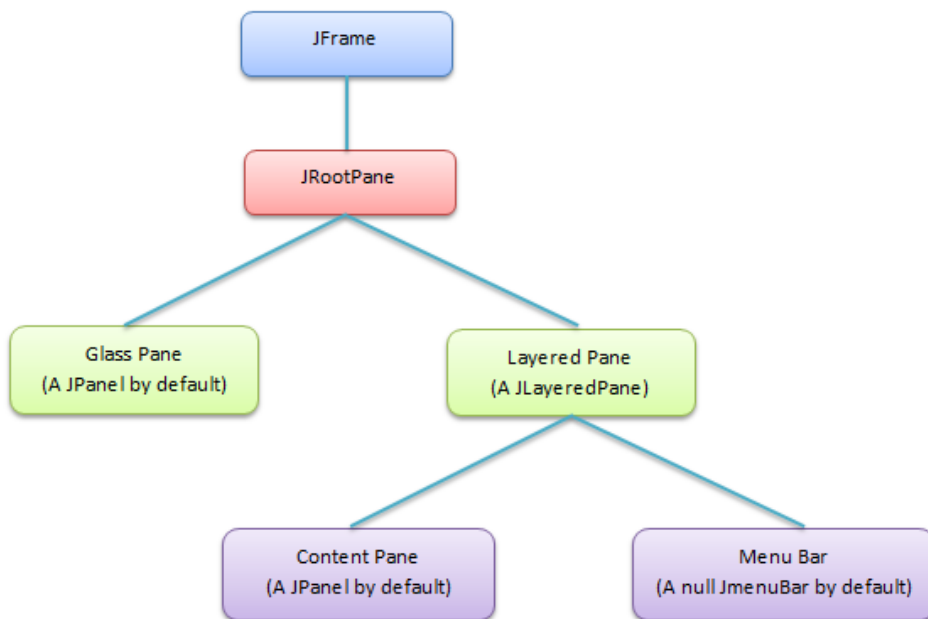
- Bước **4**: Giống như tất cả các chương trình swing, chương trình này được tạo ra bên trong một Package. Hãy nhìn vào dòng đầu tiên của mã nguồn
 - `package start;`
 - Nghĩa là bạn phải đặt file HelloWorldSwing.java vào trong thư mục Start
- Bước **5**: dịch chương trình, sử dụng lệnh
 - `javac start/HelloWorldSwing.java`
 - Nhận được file HelloWorldSwing.class
- Bước **6**: chạy chương trình, sử dụng lệnh
 - `java start.HelloWorldSwing`

Container trong Java Swing

- Thành phần chứa trong Swing, hay còn gọi là Container
- Có 2 kiểu Container trong Swing, đó là Top-level Container và Multi-purpose Container
- Swing cung cấp cho chúng ta 3 loại Top-level Container đó là:
 - JFrame
 - JDialog
 - JApplet: được sử dụng cho ứng dụng web
 - JWindow: loại này không có đặc điểm gì cả, chỉ là một màn hình chờ được bật lên trong lúc khởi động (Splash – Screen)
- General-purpose thì gồm có: JPanel, JLayered, JInternalFrame, và JDesktopPane

JFrame là gì? Chương trình Swing đầu tiên.

- **JFrame** là một Top-level Container thường được sử dụng để tạo các giao diện ứng dụng người dùng. Tất cả các đối tượng liên quan tới **JFrame** được quản lý bởi đứa con duy nhất của nó, một thể hiện (instance) của JRootPane. JRootPane có 4 phần chính là GlassPane, LayeredPane, ContentPane và MenuBar.



JFrame là gì? Chương trình Swing đầu tiên.

- Khi thêm các thành phần (component) vào JFrame chúng ta không thêm trực tiếp, thay vào đó, phần lớn các thành phần sẽ thêm vào ContentPane bằng cách gọi phương thức:
 - `getContentPane().add(component);`
- Có hai cách tạo JFrame như sau:
 - **Thứ nhất: Thừa kế `javax.swing.JFrame`**

```
1  Import javax.swing.JFrame;
2  public class DemoJframe extends JFrame {
3      // TO DO code here
4      public DemoJframe () {
5          .....
6          .....
7      }
8      public static void main (String [] args) {
9          DemoJframe objFrame = new DemoJframe ();
10         .....
11         .....
12     }
13 }
14
```

JFrame là gì? Chương trình Swing đầu tiên.

- Điều kiện sử dụng cách tạo **JFrame** kiểu này là khi lớp chúng ta muốn tạo một **JFrame** không được kế thừa từ bất cứ lớp nào khác. Sự thuận lợi của cách này là chúng ta có thể gọi các phương thức của lớp Cha một cách trực tiếp
- **Thứ hai: Khai báo javax.swing.JFrame**

```
1  import javax.swing.JFrame;
2  public class DemoJframe extends ClassABC {
3      // TO DO code here
4      .....
5      .....
6  public static void main (String [] args) {
7      DemoJframe objFrame = new DemoJframe();
8      JFrame objFrame = new JFrame();    //tao Jfram khong co tiêu đề
9      .....
10     .....
11 }
12 }
```

JFrame là gì? Chương trình Swing đầu tiên.

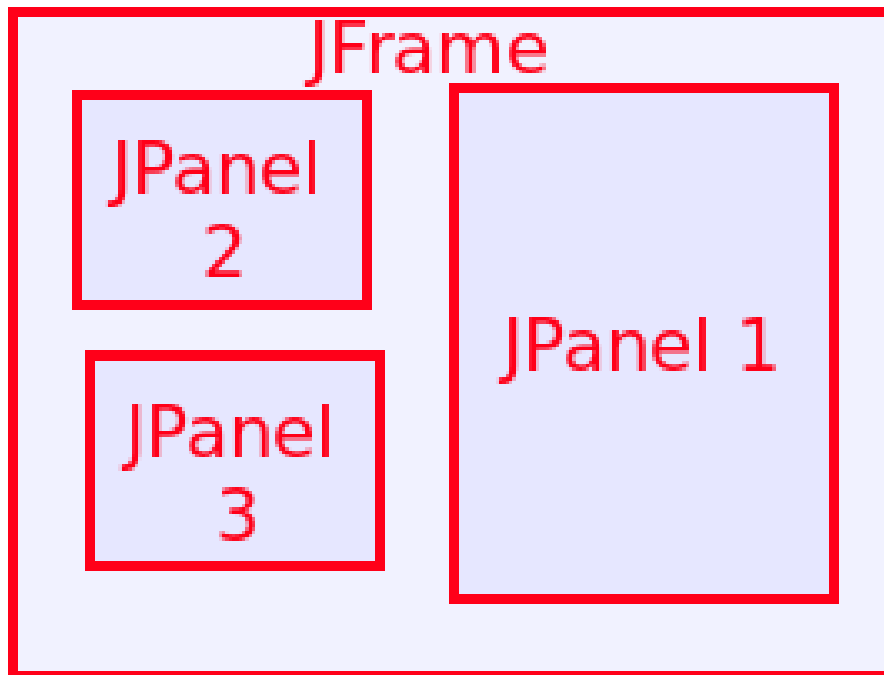
- JFrame sau khi được tạo thì chúng ta sẽ không nhìn thấy được nó. Để nhìn được nó chúng ta phải nhờ đến phương thức
 - **setVisible(boolean value).**
- Thiết lập độ rộng và cao của JFrame.
 - **setSize(width, height)**
- Khi **JFrame** đã được nhìn thấy, muốn đóng lại dùng phương thức
 - **setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);**
 - Nếu không dùng hàm này để đặt, thì mặc định là:
HIDE_ON_CLOSE: Khi đóng frame sẽ bị ẩn đi chứ hoàn toàn không đóng lại.
 - Các lựa chọn khác gồm:
 - DO_NOTHING_ON_CLOSE (0) – không làm gì cả
 - DISPOSE_ON_CLOSE (2) – Chỉ đóng frame đó, các frame khác liên quan sẽ không bị đóng.
 - EXIT_ON_CLOSE (3) – Đóng toàn bộ các frame liên quan tới nó.

JFrame là gì? Chương trình Swing đầu tiên.

- Đặt vị trí xuất hiện của JFrame trên màn hình.
 - **setLocation(x,y)**
- Đặt JFrame xuất hiện chính giữa màn hình
 - **setLocationRelativeTo(null)**
- Đặt kích thước JFrame vừa đủ với nội dung
 - **pack()**
- Đặt JFrame có thể được thay đổi kích thước hay không, mặc định là True.
 - **setResizable(boolean)**
- Đặt màu nền cho JFrame
 - **getContentPane().setBackground(Color."Color")**

JPanel là gì?

- JPanel là một container dùng để chứa các thành phần đồ họa khác (tương tự như JFrame tuy nhiên nó không phải là 1 JFrame).
- Trong một JFrame chứa các JPanel, trong mỗi JPanel lại có thể chứa các đối tượng hoặc thậm chí là các JPanel khác.



JPanel là gì?

- Chúng ta có 2 Phương thức khởi tạo JPanel đó là:
 - JPanel(): Tạo 1 JPanel với Layout **mặc định** là FlowLayout(cách bố trí mà các đối tượng nối tiếp nhau).
 - JPanel(LayoutManager layout): Tạo 1 JPanel với Layout được **chỉ định**.
- Ví dụ:

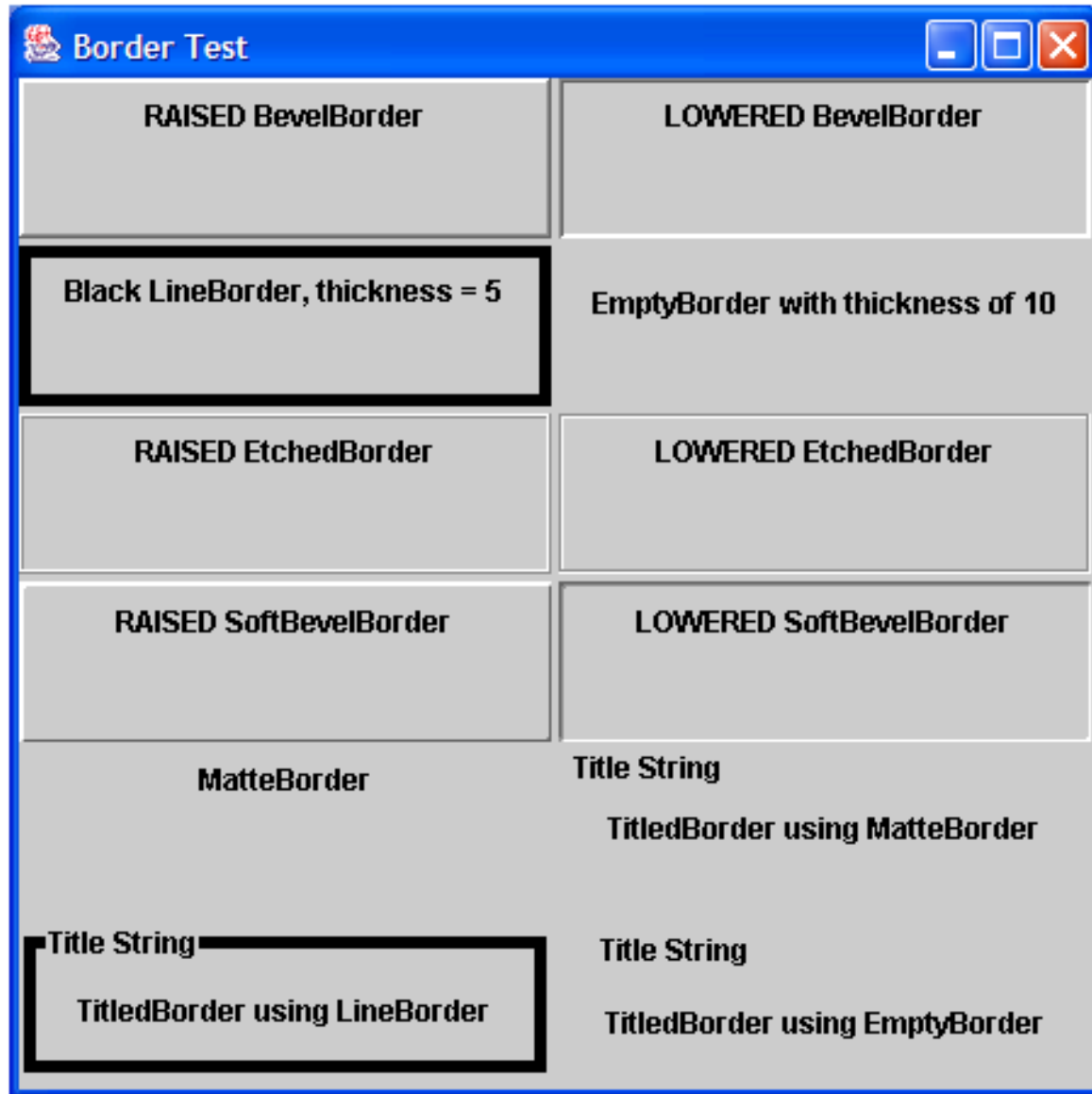
```
// create panel1 with layout default is FlowLayout  
JPanel panel1 = new JPanel();
```

```
// create panel2 with GridLayout  
JPanel panel2 = new JPanel(new GridLayout());
```

JPanel là gì?

- Đặt Layout cho JPanel
 - **setLayout(“layout”)**
- Thiết lập vị trí và kích thước cho JPanel
 - **setBound(x,y,width,heigh)**
- Đặt màu nền cho JPanel
 - **setBackground(Color.”color”)**
- Đặt đường viền cho JPanel
 - Có nhiều kiểu đường viền
 - Sử dụng phương thức **setBorder(new “kiểu-border”)**
 - Cùng xem ví dụ

JPanel là gì?



JPanel là gì?

```
p.setBorder(new BevelBorder(BevelBorder.RAISED));
```

```
p.setBorder(new BevelBorder(BevelBorder.LOWERED));
```

```
p.setBorder(new LineBorder(Color.black, 5));
```

```
p.setBorder(new EmptyBorder(10, 10, 10, 10));
```

```
p.setBorder(new EtchedBorder(EtchedBorder.RAISED));
```

```
p.setBorder(new EtchedBorder(EtchedBorder.LOWERED));
```

```
p.setBorder(new SoftBevelBorder(SoftBevelBorder.RAISED));
```

```
p.setBorder(new SoftBevelBorder(SoftBevelBorder.LOWERED));
```

```
p.setBorder(new MatteBorder(new ImageIcon("BALL.GIF"));
```

```
p.setBorder(new TitledBorder(new MatteBorder(new ImageIcon("java2sLogo.gif"), "Title String"));
```

```
p.setBorder(new TitledBorder(new LineBorder(Color.black, 5), "Title String"));
```

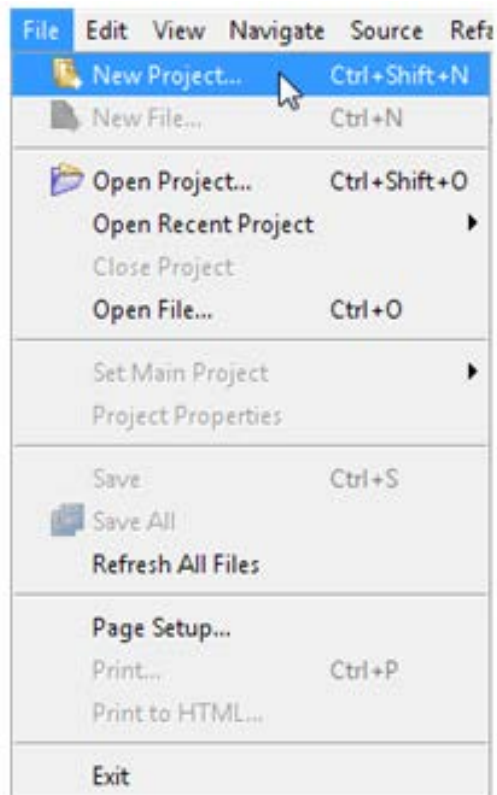
```
p.setBorder(new TitledBorder(new EmptyBorder(10, 10, 10, 10), "Title String"));
```

Lập trình Swing với công cụ NetBeans IDE

- Cài đặt công cụ NetBeans IDE 8.0.2
- Yêu cầu phần mềm:
 - Java SE Development Kit (JDK) 7 Update 10 hoặc cao hơn
- Download phần mềm
 - <http://download.netbeans.org/netbeans/8.0.2/final/bundles/netbeans-8.0.2-javase-windows.exe>

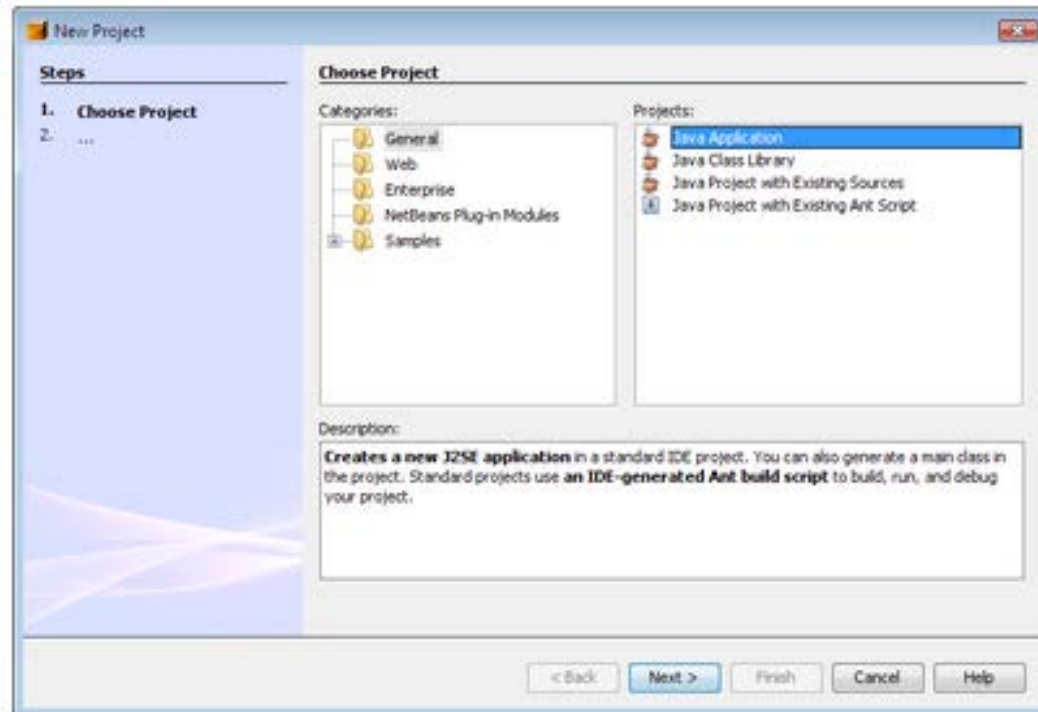
Lập trình Swing với công cụ NetBeans IDE

- Bước 1: Tạo mới một Project, mở chương trình, chọn thực đơn File, chọn New Project



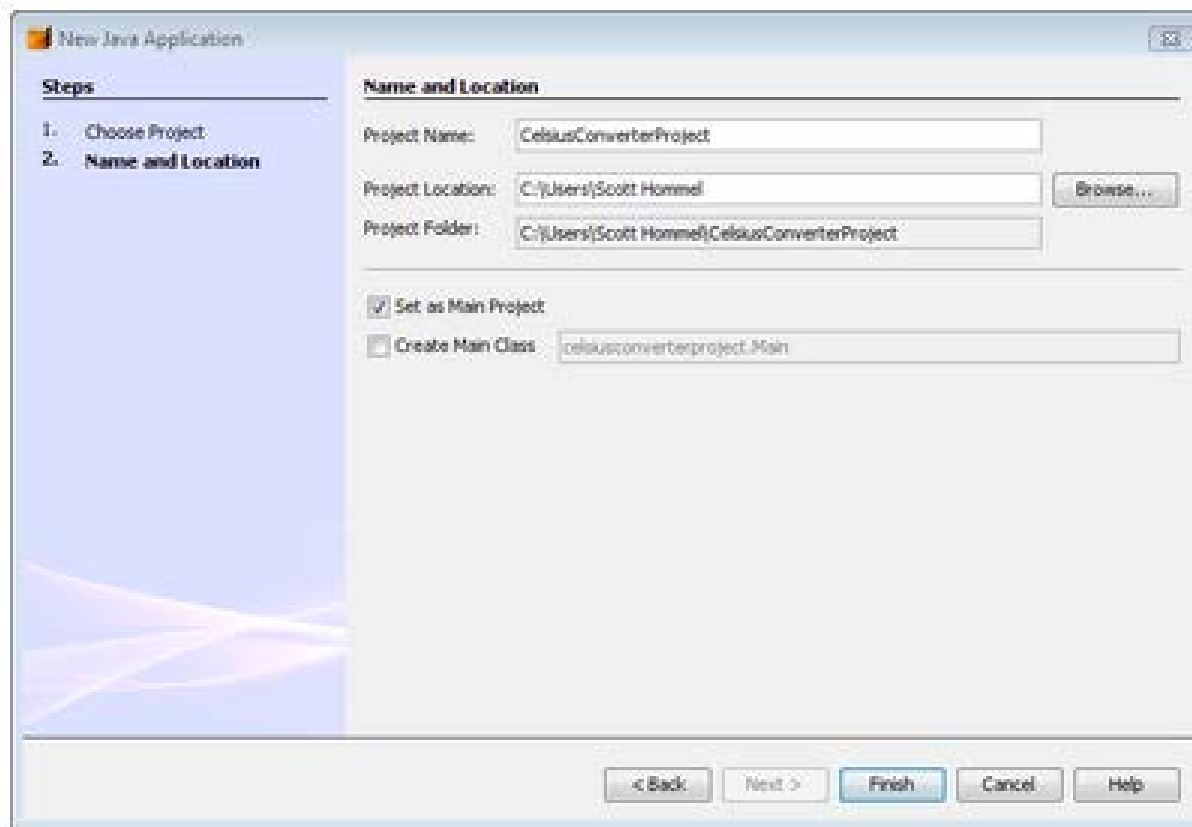
Lập trình Swing với công cụ NetBeans IDE

- Bước 2: chọn General – chọn Java Application



Lập trình Swing với công cụ NetBeans IDE

- Bước 3: Đặt tên cho Project, Project Location và Project Folder có thể để mặc định. **Nhớ** chọn dấu tích “Create Main Class”



- Cuối cùng, chọn Finish và bạn đã sẵn sàng sử dụng NetBeans

XIN CẢM ƠN!

