

## 1. So sánh Top-level Containers:

	AWT	Swing
Containers	Frame (java.awt.Frame) Dialog (java.awt.Dialog) Applet (java.awt.Applet)	JFrame (javax.swing.JFrame) JDialog (javax.swing.JDialog) JApplet (javax.swing.JApplet)
Loại thành phần	Heavyweight – phụ thuộc hệ điều hành	Lightweight – vẽ bằng Java thuần
Phụ thuộc nền tảng	Có	Không – Swing hỗ trợ đa nền tảng, giao diện có thể tùy chỉnh

Ta thấy Swing thường dùng các containers giống AWT nhưng khác cú pháp và hơi có sự khác biệt về tên.

## 2. So sánh tên lớp các thành phần (Components):

AWT (java.awt)	Swing (javax.swing)
Label	JLabel
Button	JButton
TextField	JTextField
TextArea	JTextArea
CheckBox	JCheckBox
List	JList
Menu, MenuBar, MenuItem	JMenu, JMenuBar, JMenuItem
Panel	JPanel

Các lớp Swing thường có tiền tố “J”, hỗ trợ nhiều tính năng nâng cao hơn AWT.

## 3. So sánh xử lý sự kiện (Event Handling):

	AWT	Swing
Mô hình sự kiện	Delegation Event Model	Giống AWT (kế thừa và mở rộng)
Cách gán	Listener Interface	Giống AWT
Tính tùy biến	Cơ bản	Linh hoạt hơn với class ẩn danh hoặc lambda
Khả năng mở rộng	Hạn chế	Mạnh mẽ, dễ tùy chỉnh hành vi

Swing sử dụng cùng một mô hình xử lý sự kiện với AWT nhưng hỗ trợ tốt hơn cho các ứng dụng phức tạp.

#### 4. So sánh giao diện và hiển thị (Appearance & Rendering):

	AWT	Swing
Giao diện	Phụ thuộc hệ điều hành	Tùy chỉnh
Cách hiển thị	Do OS xử lý	Java tự xử lý
Tùy biến	Giới hạn	Dễ dàng tùy chỉnh với UIManager
Hiệu suất	Nhanh hơn vì dùng native GUI	Có thể chậm hơn chút nhưng đồng nhất
Đồng nhất giữa các nền tảng	Không	Có – Swing hiển thị giống nhau trên mọi hệ điều hành

Swing thường có giao diện đẹp hơn, hiện đại hơn và dễ tùy biến hơn AWT.

- ➔ **AWT** phù hợp cho những ứng dụng đơn giản, cần tích hợp chặt với hệ điều hành.
- ➔ **Swing** phù hợp với các ứng dụng GUI hiện đại, dễ tùy biến, hoạt động đồng nhất trên nhiều nền tảng.
- ➔ Nếu đang bắt đầu với Java GUI, **nên dùng Swing** vì tính linh hoạt và cộng đồng hỗ trợ rộng rãi hơn.