JavaFX là nền tảng để tạo và ophaan phối các ứng dụng dành cho máy tính để bàn cũng như các ứng dụng RIAs(Rich Internet Apllications) có thể chạy trên nhiều thiết bị khác nhau.

Nó được phát triển bởi Chris Oliver, lúc đầu có tên là F3 đến năm 2005 được Sun Micro- systems mua lại đổi tên thành javaFX.

JavaFX dự định thay thế hoàn toàn Swing làm thư viện chuẩn cho JavaSE, JavaFX hỗ trợ máy tính để bàn và các reang web có hệ điều hành window, linux, macOS.

Ưu điểm nổi bật của javaFX là:

* Thư viện bao gồm các lớp và giao diện được viết bằng java nguyên gốc.
* FXML là ngôn ngữ khai báo dựa trên XML để xây dựng giao diện người dùng cho ứng dụng.
* Có thể tùy biến giao diện để sinh động hơn bằng CSS.
* Hỗ trợ đồ họa 2D và 3D cũng như âm thanh và video.
* Có webView dựa trên trình duyệt webKit, có thể nhúng các trang web hoặc cá công cụ web bên trong javaFX.

JavaFX chỉ hỗ trợ lập trình trên netBEANS, Eclipe, IntelliJ Idiea

**AWT**

**Java AWT** (Bộ công cụ cửa sổ trừu tượng) là một API để phát triển Giao diện người dùng đồ họa (GUI) hoặc các ứng dụng dựa trên cửa sổ trong Java.

Các thành phần Java AWT phụ thuộc vào nền tảng tức là các thành phần được hiển thị theo chế độ xem của hệ điều hành. AWT có trọng lượng nặng tức là các thành phần của nó đang sử dụng tài nguyên của hệ điều hành cơ bản (OS).

AWT độc lập với nền tảng vì:

Java AWT gọi nền tảng gốc gọi chương trình con nền tảng gốc (hệ điều hành) để tạo các thành phần API như TextField, ChechBox, nút, v.v.

Hệ thống phân cấp AWT của Java

Phân cấp của các lớp Java AWT được đưa ra dưới đây.

Rich client flatform

Xây dựng một ứng dụng phần mềm đòi hỏi sự tích hợp phức tạp của nhiều thành phần. Nền tảng máy khách phong phú (RCP) là một công cụ [**lập trình viên**](https://www.practicaladultinsights.com/what-does-a-programmer-do.htm) giúp tích hợp các thành phần phần mềm độc lập dễ dàng hơn.

Một nền tảng khách hàng phong phú cho phép tích hợp liền mạch các mô-đun phần mềm độc lập vào một ứng dụng phần mềm. Điều này thường bao gồm các công cụ đồ họa nâng cao, bảng tính và công nghệ lập bản đồ không gian địa lý. Bằng cách sử dụng RCP, nhà phát triển có thể tích hợp các thành phần độc lập vào một ứng dụng chỉ bằng một cú nhấp chuột đơn giản.

Nền tảng ứng dụng khách phong phú tương tự như [**giao diện người dùng**](https://www.easytechjunkie.com/what-is-a-user-interface.htm) iTunes® . Nó cho phép các nhà phát triển một bảng điều khiển trung tâm để quản lý tất cả các thành phần của một ứng dụng phần mềm. Cách tiếp cận [**phát triển phần mềm**](https://www.easytechjunkie.com/what-is-software-development.htm) này cung cấp một điểm tích hợp duy nhất cho tất cả các mô-đun phần mềm.

Spring

Spring là một Framework phát triển các ứng dụng Java được sử dụng bởi hàng triệu lập trình viên. Nó giúp tạo các ứng dụng có hiệu năng cao, dễ kiểm thử, sử dụng lại code…

Spring nhẹ và trong suốt (nhẹ: kích thước nhỏ, version cơ bản chỉ khoảng 2MB; trong suốt: hoạt động một cách trong suốt với lập trình viên)

Spring là một mã nguồn mở, được phát triển, chia sẻ và có cộng đồng người dùng rất lơn.

Spring Framework được xây dựng dựa trên 2 nguyên tắc design chính là: [Dependency Injection](https://stackjava.com/design-pattern/dependency-injection-di-la-gi.html) và Aspect Oriented Programming.

Những tính năng core (cốt lõi) của Spring có thể được sử dụng để phát triển Java Desktop, ứng dụng mobile, Java Web. Mục tiêu chính của Spring là giúp phát triển các ứng dụng J2EE một cách dễ dàng hơn dựa trên mô hình sử dụng POJO (Plain Old Java Object)

## Spring Core Container

Bao gồm các module spring core, beans, context và expression languate (EL)

* Spring core, bean cung cấp tính năng [IOC và Dependency Injection](https://stackjava.com/design-pattern/dependency-injection-di-la-gi.html).
* Spring Context hỗ trợ đa ngôn ngữ (internationalization), các tính năng Java EE như EJB, JMX.
* Expression Language được mở rộng từ Expresion Language trong JSP. Nó cung cấp hỗ trợ việc setting/getting giá trị, các method cải tiến cho phép truy cập collections, index, các toán tử logic…

## AOP, Aspects and Instrumentation

Những module này hỗ trợ cài đặt lập trình hướng khía cạnh (Aspect Oriented Programming), hỗ trợ tích hợp với AspectJ.

## Data Access / Integration

Nhóm này bao gồm JDBC, ORM, OXM, JMS và module Transaction. Những module này cung cấp khả năng giao tiếp với database

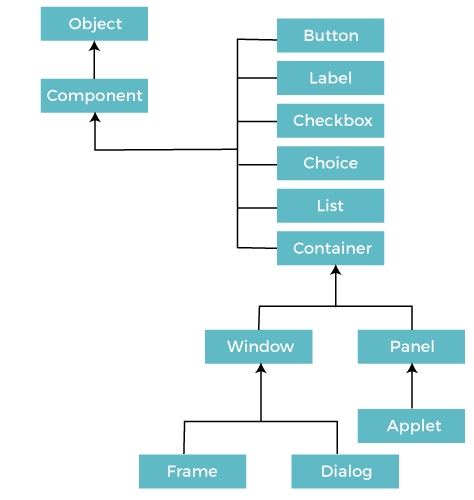
## Web

Hay còn gọi là Spring MVC Nhóm này gồm Web, Web-Servlet… hỗ trợ việc tạo ứng dụng web.

Spring là gì? Giới thiệu Spring Framework trong Java

# 3. Các lợi ích của Spring Framework.

* Spring cho phép lập trình viên sử dụng POJOs. Việc sử dụng POJOs giúp bạn không phải làm việc với EJB, ứng dụng, các luồng chạy, cấu hình… đơn giản hơn rất nhiều.
* Spring được tổ chức theo kiểu mô đun. Số lượng các gói và các lớp khá nhiều, nhưng bạn chỉ cần quan tâm đến những gì bạn cần và không cần quan tâm đến phần còn lại.
* Spring hỗ trợ sử dụng khá nhiều công nghệ như [ORM Framework](https://stackjava.com/uncategorized/orm-la-gi-tong-quan-ve-orm-framework.html), các logging framework, JEE, các thư viện tạo lịch trình (Quartz và JDK timer)…
* Module Web của Spring được thiết kế theo mô hình MVC nên nó cung cấp đầy đủ các tính năng giúp thay thế các web framework khác như Struts.



### **Thùng đựng hàng**

Vùng chứa là một thành phần trong AWT có thể chứa các thành phần khác như [nút](https://www.javatpoint.com/java-awt-button) , trường văn bản, nhãn, v.v. Các lớp mở rộng lớp Vùng chứa được gọi là vùng chứa như **Khung, Hộp thoại** và **Bảng điều khiển** .

Các phương pháp hữu ích của lớp thành phần

|  |  |
| --- | --- |
| **Phương pháp** | **Sự mô tả** |
| public void add (Thành phần c) | Chèn một thành phần trên thành phần này. |
| public void setSize (int width, int height) | Đặt kích thước (chiều rộng và chiều cao) của thành phần. |
| public void setLayout (LayoutManager m) | Xác định trình quản lý bố cục cho thành phần. |
| public void setVible (trạng thái boolean) | Thay đổi khả năng hiển thị của thành phần, theo mặc định là false. |