**TRƯỜNG ĐẠI HỌC QUY NHƠN**

**KHOA CÔNG NGHỆ THÔNG TIN**

**CÔNG NGHỆ JAVA**

**TÌM HIỂU SPRING**



**GVHD : Nguyễn Thanh Bình**

**Tên thành viên :**

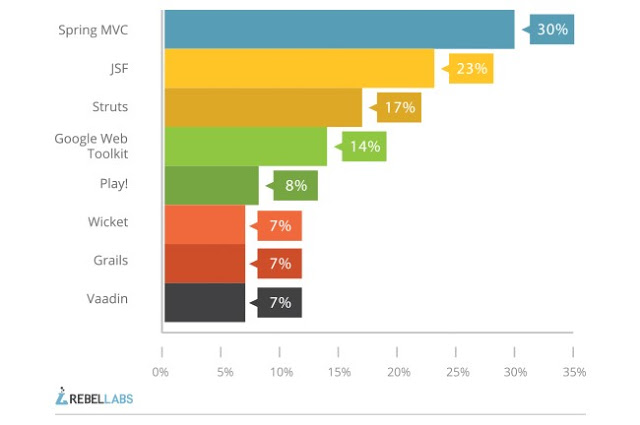
1. **Lê Phi Khánh**
2. **Nguyễn Văn Bình**
3. **Trần Văn Hoài Nhiên**

Quy Nhơn, Ngày 15 tháng 09 năm 2018

**I Giới thiệu về Spring**

1. **Spring là gì**

* Spring Framework(gọi tắt Spring), là một bộ khung ứng dụng và [bộ chứa](http://www.wikiwand.com/en/Servlet_container) [đảo ngược điều khiển](http://www.wikiwand.com/en/inversion_of_control) cho [nền tảng Java](http://www.wikiwand.com/vi/Java_(c%C3%B4ng_ngh%E1%BB%87)). Chức năng tính của bộ khung này có thể áp dụng cho bất kỳ ứng dụng Java nào, tuy vậy, nhiều bản mở rộng dành cho việc xây dựng ứng dụng nền web cũng được phát triển trên nền tảng Java EE
* Nó được sử dụng phổ biến nhất trong các Framework của Java, với cộng đồng sử dụng đông đảo, tài liệu trainning và support rất nhiều.
* Tại Việt Nam, Spring là Framework Java được ưa thích sử dụng và rất phổ biến.

[](http://2.bp.blogspot.com/-2GlYwZn3A5E/V-M_RwC9k1I/AAAAAAAAadg/DYOUrfKWWac3ki1QW4ngehIJ3LrIm9LjwCK4B/s1600/Picture1.jpg)

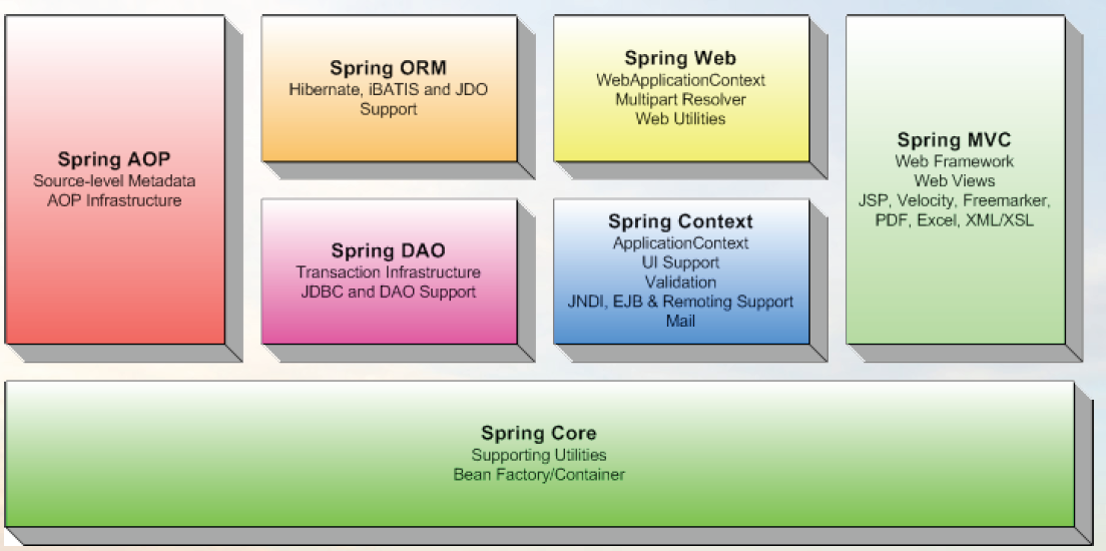
Biểu đồ độ phổ biến của các Framework trong Java.

1. **Lịch sử phiên bản**

* Phiên bản đầu tiên của Spring được viết bởi [Rod Johnson](http://www.wikiwand.com/en/Rod_Johnson_(programmer)), cùng lúc đó ông cũng cho ra cuốn sách *Expert One-on-One J2EE Design and Development* vào tháng 10 năm 2002.
* Spring Framework 4.2 được phát hành ngày 31 tháng 07 năm 2015 và ngay sau đó được nâng cấp lên phiên bản 4.2.1, được phát hành ngày 01 tháng 09 cùng năm. Phiên bản này được mô tả là *"tương thích với Java 6, 7 và 8, chú trọng vào những cải tiến cốt lõi và tính năng web hiện đại"*.
* Spring Framework 4.3 được phát hành ngày 10 tháng 06 năm 2016.
* Spring 5 được thông báo rằng sẽ dựng trên [Reactive Streams](http://www.wikiwand.com/en/Reactive_Streams) tương thích với Reactor Core.

1. **Thành phần của Spring Framework**

* Một vài module quan trọng của Spring framework sau:
* Spring Context – hỗ trợ dependency injection.
* Spring AOP – hỗ trợ aspect oriented programming.
* Spring DAO – hỗ trợ database operations sử dụng DAO pattern
* Spring JDBC – hỗ trợ JDBC and DataSource.
* Spring ORM – hỗ trợ ORM
* Spring Web Module – hỗ trợ phát triển ứng dụng web.
* Spring MVC – Model-View-Controller hỗ trợ phát triển web application, web services vv.



1. **Lý do sử dụng**

* Tất cả các framework đã được tích hợp rất tốt vào Spring.
* Hoạt động rất tốt khi áp dụng theo kiến trúc MVC.
* Sử dụng cơ chế plug-in.
* Kết hợp rất tốt với các O/R (object-relational) Mapping frameworks như là Hibernate.
* Dễ Testing ứng dụng.
* Ít phức tạp hơn so với các framework khác.
* Cộng đồng người sử dụng rất nhiều, nhiều sách mới được xuất bản.

1. **Ưu điểm của Spring**

**• Ưu điểm**

* Lightweight và very little overhead trong việc phát triển ứng dụng của bạn.
* Dependency Injection hoặc Inversion of Control được sử dụng để giúp các component tách rời, độc lập với nhau. Spring container sẽ giúp gắn kết những components này lại với nhau theo đặc tả business của bạn.
* Spring IoC container quản lý vòng đời của Spring Bean và các cấu hình của project chẳng hạn như JNDI lookup.
* Spring MVC framework được sử dụng cho phát triển ứng dụng web rất dễ dàng với việc hỗ trợ rất tốt các tính năng web services, json,…
* Hỗ trợ quản lý transaction, JDBC operations, File uploading, Exception Handling,… rất dễ dàng bằng cách cấu hình được rút gọn, thay vào đó là sử dụng annotation hoặc spring bean configuration file.
* Và một vài ưu điểm khi sử dụng Spring framework:
* Làm giảm đi sự phụ thuộc giữa các components khác nhau của ứng dụng, Spring IoC container làm nhiệm vụ khởi tạo resources hoặc beans và “tiêm – inject” chúng theo sự phụ thuộc khác nhau.
* Thực hiện unit test case rất dễ bởi vì business logic của bạn không có sự phụ thuộc trực tiếp. Việc thực hiện chỉ là viết test configuration và inject mock bean cho các mục đích test khác nhau.
* Làm giảm đi khối lượng code rất nhiều, chẳng hạn như việc khởi tạo đối tượng, open/close các resources,…
* Spring framework chia thành nhiều module riêng biệt, do đó việc sử dụng các features trong Spring framework rất tự do… Ví dụ như ứng dụng không sử dụng tính năng transaction, thì không cần thiết phải thêm dependency này vào.
* Spring framework hỗ trợ hầu hết các tính năng của Java EE, thậm chí còn nhiều hơn nữa.

**IV. Sử Dụng Spring :**

# 1. Một số dự án của Spring.

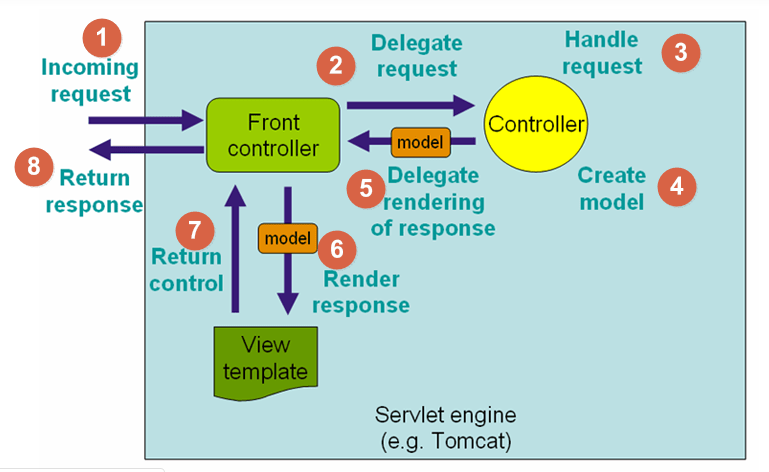
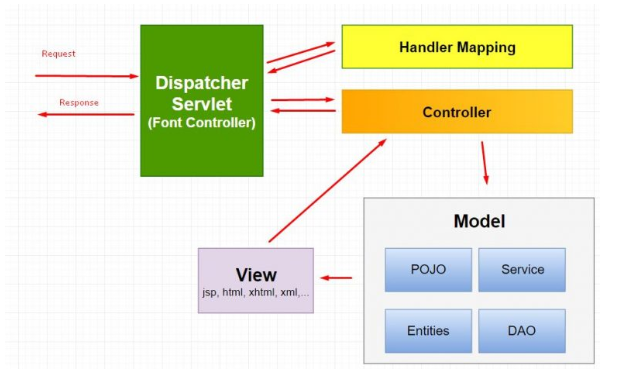
Cũng dựa trên các nguyên tắc thiết kế cơ bản của spring core. Spring còn phát triển nhiều project con như:

* **Spring MVC** được thiết kế dành cho việc xây dựng các ứng dụng nền tảng web.

**A. Giới thiệu Spring MVC, luồng đi trong Spring MVC**

1. Giới thiệu Spring MVC (Spring Web MVC)

Spring MVC là một Framework / 1 dự án mã nguồn mở của Spring, nó cung cấp kiến trúc MVC (Model – View - Controller) và các thành phần được sử dụng để phát triển các ứng dụng web một cách linh hoạt.

* 1. Luồng đi trong Spring MVC
* Bất kỳ request nào tới ứng dụng web đều sẽ được gởi tới Front Controller (Dispatcher Servlet).
* Front Controller sẽ sử dụng Handler Mapping để biết được controller nào sẽ xử lý request đó.
* Controller nhận request, gọi tới các class service thích hợp để xử lý yêu cầu.
* Sau khi xử lý xong, Controller sẽ nhận được model từ tầng Service hoặc tầng DAO.
* Controller gửi model vừa nhận được tới Front Controller (Dispatcher Servlet).
* Dispatcher Servlet sẽ tìm các mẫu view, sử dụng view resolver và truyền model vào nó.
* View template, model, view page được build và gửi trả lại Front Controller.
* Front Controller gửi một page view tới trình duyệt để hiển thị nó cho người dùng.

Trong mô hình MVC thì:

* Model: là các file POJO, Service, DAO thực hiện truy cập database, xử lý business.
* View: là các file JSP, html…
* Controller: là Dispatcher Controller, Handler Mapping, Controller – thực hiện điều hướn các request.

2. Các lợi ích của Spring MVC

* Các tầng trong Spring MVC độc lập nên việc viết unit test dễ dàng hơn.
* Phần giao diện có thể tích hợp với nhiều framework về UI (User Interface) như JSF, Freemarker, Themeleaf, …
* Spring MVC độc lập trên các POJO (Plain Old Java Object) class nên các hành động của nó khá đơn giản.
* Hỗ trợ cả Annotation và XML Config giúp việc phát triển nhanh hơn và sạch hơn.
* Cung cấp việc phân chia một cách rõ ràng, linh hoạt giữa Controller, service, data access layer.
* **Spring Security** Cung cấp các cơ chế xác thực (authentication) và phân quyền (authorization) cho ứng dụng của bạn.

*Đặc trưng:*

*Spring Security* là một khung tập trung vào việc cung cấp cả xác thực và ủy quyền cho các ứng dụng Java. Giống như tất cả các dự án Spring, sức mạnh thực sự của Spring Security được tìm thấy ở mức độ dễ dàng có thể được mở rộng để đáp ứng các yêu cầu tùy chỉnh

Hỗ trợ toàn diện và mở rộng cho cả Xác thực và Ủy quyền

Bảo vệ chống lại các cuộc tấn công như sửa lỗi phiên, nhấp chuột, giả mạo yêu cầu trang web chéo, v.v.

Tích hợp API của Servlet

Tích hợp tùy chọn với Spring Web MVC

* **Spring Boot** là một framework giúp chúng ta phát triển cũng như chạy ứng dụng một cách nhanh chóng.

*Đặc trưng:*

* + Tạo các ứng dụng Spring độc lập
  + Nhúng Tomcat, Jetty hoặc Undertow trực tiếp (không cần triển khai các tệp WAR)
  + Cung cấp các phụ thuộc 'khởi động' có ý kiến để đơn giản hóa cấu hình bản dựng của bạn
  + Tự động cấu hình thư viện Spring và bên thứ 3 bất cứ khi nào có thể
  + Cung cấp các tính năng sẵn sàng sản xuất như số liệu, kiểm tra sức khỏe và cấu hình bên ngoài
  + Hoàn toàn không tạo mã và không yêu cầu cấu hình XML
* **Spring Batch** dự án này giúp chúng ta dễ dàng tạo các lịch trình (scheduling) và tiến trình (processing) cho các công việc xử lý theo mẻ (batch job).

Spring Batch cung cấp các chức năng có thể sử dụng lại rất cần thiết trong việc xử lý khối lượng lớn hồ sơ, bao gồm ghi nhật ký / theo dõi, quản lý giao dịch, thống kê xử lý công việc, khởi động lại công việc, bỏ qua và quản lý tài nguyên

*Đặc trưng:*

* + *Quản lý giao dịch*
  + *Bắt đầu / Dừng / Khởi động lại* /*Thử lại / Bỏ qua*

* **Spring Data** là cung cấp một mô hình lập trình dựa trên Spring quen thuộc và nhất quán để truy cập dữ liệu trong khi vẫn giữ được các đặc điểm đặc biệt của kho lưu trữ dữ liệu cơ bản.

*Đặc trưng:*

* + - * + *Kho lưu trữ mạnh mẽ và trừu tượng ánh xạ đối tượng tùy chỉnh*
        + Dẫn xuất truy vấn động từ tên phương thức kho lưu trữ
        + Các lớp cơ sở miền thực hiện cung cấp các thuộc tính cơ bản
        + Hỗ trợ kiểm toán minh bạch (được tạo, thay đổi lần cuối)
        + Khả năng tích hợp mã kho lưu trữ tùy chỉnh
        + Tích hợp Easy Spring thông qua JavaConfig và các không gian tên XML tùy chỉnh
* **Spring Cloud** cung cấp các công cụ cho các nhà phát triển để nhanh chóng xây dựng một số mô hình phổ biến trong các hệ thống phân tán

*Đặc trưng:*

Spring Cloud tập trung vào việc cung cấp trải nghiệm tốt cho các trường hợp sử dụng điển hình và cơ chế mở rộng để bao quát những người khác.

* **Spring Mobile** là một khung cung cấp khả năng phát hiện loại thiết bị đưa ra yêu cầu đến trang web Spring của bạn và phục vụ các chế độ xem thay thế dựa trên thiết bị đó

*Đặc trưng:*

* + - * + Trình chuyển đổi trang web có khả năng chuyển người dùng sang trang web phù hợp nhất, trên thiết bị di động, máy tính bảng hoặc bình thường, dựa trên thiết bị của họ và tùy chọn trang web được chỉ định tùy chọn
        + Quản lý chế độ xem nhận biết thiết bị để tổ chức và quản lý các chế độ xem khác nhau cho các thiết bị cụ thể
* **Spring Android** là một khung được thiết kế để cung cấp các thành phần của dòng dự án Spring để sử dụng trong các ứng dụng Android

*Đặc trưng:*

Hỗ trợ xác thực để truy cập các API an toàn

* **Spring Flo** là một thư viện JavaScript cung cấp trình xây dựng hình ảnh HTML5 có thể nhúng cơ bản cho các đường ống và các biểu đồ đơn giản

*Đặc trưng:*

* + - * + Tạo, quản lý và giám sát các đường ống truyền phát bằng DSL, khung vẽ đồ họa hoặc cả hai
        + Viết đường ống qua DSL với hỗ trợ nội dung và tự động hoàn thành
        + Xem biểu diễn trực quan của các mô-đun qua triển khai phân tán
        + Sử dụng khả năng tự động điều chỉnh và bố trí lưới trong GUI để tổ chức các đường ống dễ dàng và hiệu quả hơn

**2. Các lỗi nên tránh**

* **Quá tập trung vào tiểu tiết** thường xuyên viết lại các đoạn mã thường được sử dụng và dường như có rất nhiều nhà phát triển bị ảnh hưởng.
* **Không thống nhất và xử lý lỗi kém**
* **Xử lý không hợp lý với Multithreading**
* **(Vẫn còn) Sử dụng Configuration dựa trên XML**
* **Quên các Profile**
* **Không test hoặc test không đúng**

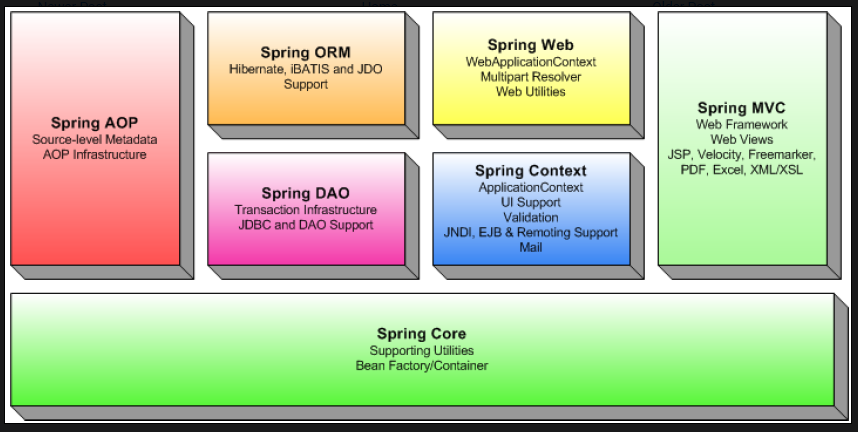
**V. Vài trò của Spring :**

A. Hoạt động:

### **1. Tác dụng:**

* **Trọng lượng nhẹ:** có lợi cho việc phát triển và triển khai các ứng dụng trên các máy tính có bộ nhớ và tài nguyên CPU hạn chế.
* **Inversion of Control (IoC):** Spring container chăm sóc các phụ thuộc dây của các đối tượng khác nhau, thay vì tạo hoặc tìm kiếm các đối tượng phụ thuộc.
  + **Lập trình hướng Aspect (AOP):**Spring hỗ trợ AOP để tách logic nghiệp vụ khỏi các dịch vụ hệ thống.
* **IoC container:**nó quản lý vòng đời Spring Bean và các cấu hình cụ thể của dự án.
* **Khung MVC:** được sử dụng để tạo các ứng dụng web hoặc các dịch vụ web RESTful, có khả năng trả về các phản hồi XML / JSON.
* **Quản lý Transaction:** giảm số lượng mã tấm nồi hơi trong các hoạt động JDBC, tải lên tệp, v.v. bằng cách sử dụng chú thích Java hoặc tệp cấu hình Spring Bean XML.
* **Xử lý ngoại lệ:** Spring cung cấp một API thuận tiện để dịch ngoại lệ dành riêng cho công nghệ thành các ngoại lệ không được kiểm soát.
* **Sử dụng:** 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.
* **Module Web**: 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.
* **Code:** Làm giảm đi khối lượng code rất nhiều, chẳng hạn như việc khởi tạo đối tượng, open/close các resources,…

### **2. Các chức năng:**



* **Spring AOP:**

Aspect oriented programming dịch là lập trình hướng khía cạnh. AOP giúp module hóa ứng dụng, biến chương trình thành các module hoạt động độc lập, mỗi module làm một chức năng riêng, từ đó dễ bảo trì và nâng cấp. Các module có thể áp dụng chung tính năng logging, transaction, data validation, authentication, …

* **Spring ORM:**

Là từ viết tắt của Object / Relational mapping. là phương pháp lập trình chuyển đổi ánh xạ từ mô hình database sang mô hình đối tượng.

* **Spring Web:**

Spring Web là 1 phần của Spring Web application development, bao gồm MVC.

* **Spring DAO:**

Data Access Object để truy cập dữ liệu như JDBC, Hibernate và JPA một cách nhất quán và dễ dàng.

* **Spring Context:**

Cung cấp cách để truy cập đối tượng. Kết thừa các đặc tính từ bean package và thêm vào chức năng đa ngôn ngữ (I18N), truyền sự kiện, resource-loading

* **Spring MVC:**

Triển khai MVC cho các ứng dụng web.

* **Spring Core:**

Chứa những thành phần **IoC, DI**và **BeanFactory**

* **DI:** Dependency Injection là khả năng liên kết giữa các thành phần lại với nhau, thuộc tính trong một đối tượng được “tiêm” tham chiếu lần lượt đến các đối tượng khác quản lý bởi Spring container.
* **IoC:**Inversion of Control dịch là đảo ngược điều khiển (hơi khó hiểu)  
  Ý của nó là làm thay đổi luồng điều khiển của ứng dụng. ví dụ như ở trên việc thay đổi thông tin trong file config.properties đã làm thay đổi luồng chạy của ứng dụng..
* **BeanFactory:**Cho phép móc nối giữa các đối tượng trong file cấu hình.

B. Phát triển:

* Phiên bản đầu tiên của Spring được viết bởi Rod Johnson, cùng lúc đó ông cũng cho ra cuốn sách Expert One-on-One J2EE Design and Development vào tháng 10 năm 2002. Ban đầu, bộ khung được phát hành dưới Giấy phéo Apache 2.0 vào tháng 06 2003. Cột mốc đầu tiên, bản 1.0, được phát hành tháng 03 năm 2004, và lần lượt các mốc phát hành kế tiếp vào tháng 09 năm 2004 và tháng 03 năm 2005. Phiên bản Spring 1.2.6 nhận Giải năng suất Jolt (Jolt Productivity award) và một Giải Cải tiến JAX (JAX Innovation Award) vào năm 2006. Spring 2.0 was released in October 2006, Spring 2.5 in November 2007, Spring 3.0 in December 2009, Spring 3.1 in December 2011, and Spring 3.2.5 in November 2013.
* Spring Framework 4.0 được phát hành vào tháng 12 năm 2013. Cải tiến đáng chú ý của phiên bản này hỗ trợ Java SE 8, Groovy 2, một số phần của Java EE7, và Websocket. Thu hẹp khoảng cách giữa mã nguồn của bạn và chạy các ứng dụng Spring Boot.
* Spring Framework 4.2.0 được phát hành ngày 31 tháng 07 năm 2015 và ngay sau đó được nâng cấp lên phiên bản 4.2.1, được phát hành ngày 01 tháng 09 cùng năm. Phiên bản này được mô tả là "tương thích với Java 6, 7 và 8, chú trọng vào những cải tiến cốt lõi và tính năng web hiện đại".
* Spring Framework 4.3 được phát hành ngày 10 tháng 06 năm 2016. Phiên bản 4.3.0.RC1 có hiệu lực. Nó "sẽ là thế hệ cuối cùng cho những yêu cầu của Spring 4 (Java 6+, Servlet 2.5+), chuẩn bị cho phiên bản mở rộng 4.3.x với thời gian hỗ trợ đến năm 2019".
* Spring 5 đã được ra đời và công bố sớm từ những ngày tháng 7 năm 2017 với phiên bản milestone đầu tiên ( hay viết tắt là M1) .Spring 5 được thông báo rằng sẽ dựng trên Reactive Stream tương thích với Reactor Core. Hỗ trợ ngôn ngữ lập trình mới Kotlin với mục đích cú pháp ngắn gọn, dễ nhìn hơn,..tương thích với các thư viện JVM hiện hành.

**III. Kiến trúc Spring :**

**Kiến trúc Spring** được tổ chức theo kiểu module. Mặc dù số lượng các gói và các lớp là khá nhiều, nhưng bạn chỉ cần quan tâm đến những gì bạn cần để đáp ứng đúng requirement và không cần quan tâm đến những phần còn lại.

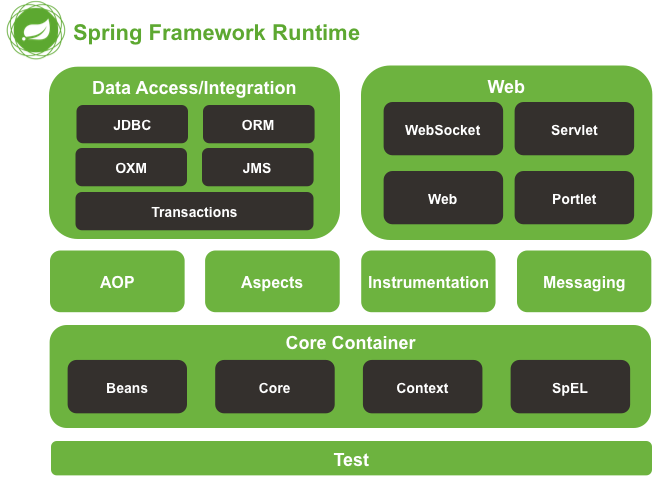
## 1.Dependency Injection và Inversion of Control:

**Dependency Injection** (DI) là một design pattern để loại bỏ sự phụ thuộc vào mã lập trình, vì vậy nó giúp dễ dàng quản lý và test ứng dụng. Dependency Injection giúp mã lập trình liên kết lỏng với nhau.

**Inversion of Control (IoC) Container** có trách nhiệm khởi tạo, configure, gắn kết các đối tượng. Có hai kiểu là **BeanFactory**và **ApplicationContext**

## 2. Các module của Spring Framework:

**Spring Framework** bao gồm nhiều tính năng được tổ chức thành khoảng 20 module. Các module này được phân nhóm thành Core Container, Data Access/Integration, Web, AOP (Aspect Oriented Programming), Instrumentation, Messaging, và Test.



### Core Container :

**Core Container** bao gồm các module: *spring-core*, *spring-beans*, *spring-context*, *springcontext-support*, và *spring-expression*(Spring Expression Language).

* Module **spring-core** và **spring-beans** cung cấp những phần cơ bản của framework, bao gồm IoC và Dependency Injection.
* Module **spring-context**xây dựng trên một nền tảng vững chắc được cung cấp bởi các module Core và Beans, được sử dụng để truy cập vào các đối tượng trong framework-style tương tự như việc đăng ký một JNDI. **spring-context-support**hỗ trợ tích hợp libraries của bên thứ 3 (third-party) vào ứng dụng Spring như caching (EhCache, Guava, JCache), mailing (JavaMail), schedule (CommonJ, Quatz) và các template engine (FreeMarker, JasperReports, Velocity)).
* Module **spring-expression** cung cấp một **Expression Language** mạnh mẽ cho việc truy vấn và tính toán một đồ thị đối tượng lúc runtime. Nó được kế thừa và mở rộng từ unified expression language (unified EL). Ngôn ngữ này hỗ trợ setting và getting giá trị các property, gọi phương thức, truy cập vào nội dung của mảng, tập hợp và chỉ mục, operator logic và tính toán, đặt tên biến, và truy xuất các đối tượng theo tên từ IoC container của Spring.

**Data Access/Integration :**

**Data Access/Integration** bao gồm các module JDBC, ORM, OXM, JMS và Transaction.

* Module spring-jdbc cung cấp một lớp JDBC-abstraction để loại bỏ những code tẻ nhạt cả JDBC và phân tích những mã lỗi cụ thể của database-vendor.
* Module spring-orm cung cấp lớp tích hợp với các ORM API phổ biến như JPA, JDO và Hibernate.
* Module spring-oxm cung cấp lớp abstraction hỗ trợ triển khai Object/XML mapping như AXB, Castor, XMLBeans, JiBX and XStream.
* Module spring-jms chứa các tính năng tạo và sử dụng các message. Từ Spring Framework 4.1, nó được tích hợp với spring-messaging.
* Module spring-transaction hỗ trợ quản lý giao dịch theo chương trình và khai báo cho các lớp mà thực hiện các giao diện đặc biệt và cho tất cả các POJO của bạn.

### Web :

**Tầng Web** bao gồm các module spring-web , spring-webmvc, spring-websocket và springwebmvc-portlet.

* Module spring-web cung cấp tính năng tích hợp web theo định hướng cơ bản như chức năng tập tin tải lên nhiều phần dữ liệu và khởi tạo các container IoC sử dụng nghe servlet và một bối cảnh ứng dụng web theo định hướng.
* Module spring-webmvc bao gồm việc triển khai Model-View-Controller (MVC) của Spring cho các ứng dụng web.
* Module spring-websocket cung cấp hỗ trợ cho WebSocket-based, giao tiếp hai chiều giữa client và server trong các ứng dụng web.
* Module springwebmvc-portlet cung cấp việc triển khai MVC được sử dụng trong môi trường portlet và ánh xạ chức năng của module Web-Servlet.

### Các module khác :

Có vài module quan trọng khác như module AOP, Các khía cạnh, Instrumentation, Web và Test mô tả chi tiết như sau:

* Module **AOP** cung cấp một thực hiện lập trình hướng khía cạnh cho phép bạn xác định phương pháp-chặn và pointcuts để sạch tách mã thực hiện chức năng đó nên được tách ra.
* Module **Aspects** cung cấp tích hợp với AspectJ, mà lại là một khuôn khổ AOP mạnh mẽ và trưởng thành.
* Module **Instrumentation** cung cấp thiết bị đo đạc lớp hỗ trợ và triển khai lớp bộ nạp được sử dụng trong các máy chủ ứng dụng nhất định.
* Module **Messaging** cung cấp hỗ trợ cho STOMP như WebSocket sub-protocol để sử dụng trong các ứng dụng. Nó cũng hỗ trợ một mô hình lập trình chú thích cho việc định tuyến và xử lý tin nhắn STOMP từ các máy khách WebSocket.
* Module **Test** hỗ trợ việc kiểm tra các thành phần mùa xuân với JUnit hoặc TestNG khuôn khổ.