# Java Overview

Java-core là java cơ bản

J2SE: hỗ trợ xây dựng các Desktop Application **<desktop>**

J2ME là lập trình cho thiết bị di động. **<Mobile>**  
  
J2EE: hỗ trợ xây dựng Web Application và các dịch vụ Web **<Web>**  
  
<j2me < j2se < j2ee. j2me là tập hợp con của j2se, j2se là con của j2ee. >

Trong mỗi thằng lại có thêm những công nghệ đặc thù nữa

ví dụ: j2ee có jsp/servlet, ejb, webservice;

trong khi j2se có thêm awt, swing, swt (eclipse)...  
java core là nền tảng cơ bản về ngôn ngữ, ko phải là một công nghệ, ví dụ như: lớp như thế nào, exception ra sao, các kiểu dữ liệu (primitive, wrapper)...  
  
Còn các framwork tức là bộ khung làm việc cho mình lập trình dễ hơn, như : **Spring**, hiberate,....

Java đã trải qua 3 bước phát triển quan trọng: **Java 1.0 gắn liền với bản JDK đầu tiên**, **Java 2** gắn với **JDK** **1**.2 và **Java 5 gắn với J2SDK 1.5**  
Ngày nay, khi nhắc đến Java người ta không còn chỉ nhắc đến Java như là một ngôn ngữ mà nhắc đến Java như là một công nghệ hay một nền tảng phát triển. Nó bao gồm các bộ phận:  
  
\* Máy ảo Java: JVM  
\* Bộ công cụ phát triển: J2SDK  
\* Các đặc tả chi tiết kĩ thuật (specifications)  
\* Ngôn ngữ lập trình (programming language)

**Lịch sử**

Java là một ngôn ngữ lập trình được Sun Microsystems giới thiệu vào tháng 6 năm 1995.

Java được xây dựng trên nền tảng của C và C++

**Đặc trưng của java:**

* Đơn giản
* Hướng đối tượng
* Độc lập với hệ nền
* Hỗ trợ lập trình đa tuyến
* Phân tán
* Thông dịch
* Hỗ trợ internet

**Các kiểu ứng dụng Java**

Ứng dụng **Applets**:

**Applet là chương trình Java được tạo ra để sử dụng trên Internet** thông qua các trình duyệt hỗ trợ Java như IE hay Netscape. **Applet được nhúng bên trong trang Web**. **Khi trang Web hiển thị trong trình duyệt, Applet sẽ được tải về và thực thi tại trình duyệt.**

Ứng dụng dòng lệnh

Các chương trình này chạy từ dấu nhắc lệnh và không sử dụng giao diện đồ họa. Các thông tin nhập xuất được thể hiện tại dấu nhắc lệnh.

Ứng dụng đồ họa -- giao diện đồ họa.

**JSP**/**Servlet**

Java thích hợp để phát triển ứng dụng nhiều lớp. **Applet là chương trình đồ họa chạy trên trình duyệt tại máy trạm (clients)**. Ở các ứng dụng Web, máy trạm gửi yêu cầu tới máy chủ. Máy chủ xử lý và gửi kết quả trở lại máy trạm. Các Java API chạy trên máy chủ chịu trách nhiệm xử lý tại máy chủ và trả lời các yêu cầu của máy trạm. Các Java API chạy trên máy chủ này mở rộng khả năng của các ứng dụng Java API chuẩn. **Các ứng dụng trên máy chủ này được gọi là các JSP/Servlet**. **hoặc Applet tại máy chủ**. Xử lý Form của HTML là cách sử dụng đơn giản nhất của JSP/Servlet. Chúng còn có thể được dùng để xử lý dữ liệu, thực thi các giao dịch và thường được thực thi thông qua máy chủ Web.

Ứng dụng cơ sở dữ liệu

Các ứng dụng này sử dụng **JDBC API** để kết nối tới cơ sở dữ liệu. Chúng có thể là Applet hay ứng dụng, nhưng Applet bị giới hạn bởi tính bảo mật.

Ứng dụng mạng

Java là một ngôn ngữ rất thích hợp cho việc xây dựng các ứng dụng mạng. Với thư viện Socket bạn có thể lập trình với hai giao thức: **UDP và TCP.**

Ứng dụng nhiều tầng

Với Java bạn có thể xây dựng phân tán nhiều tầng với nhiều hỗ trợ khác nhau như: RMI, CORBA, EJB, Web Service

Ứng dụng cho các thiết bị di động

Hiện nay phần lớn các thiết bị di động như: Điện thoại di động, máy trợ giúp cá nhân… đều hỗ trợ Java. Thế nên bạn có thể xây dựng các ứng dụng chạy trên các thiết bị di động này. Đây là một kiểu ứng dụng khá hấp dãn, bởi vì các thiết bị di động này ngày càng phổ biến và nhu cầu có các ứng dụng chạy trên đó, đặc biệt là các ứng dụng mang tính chất giải trí như game…

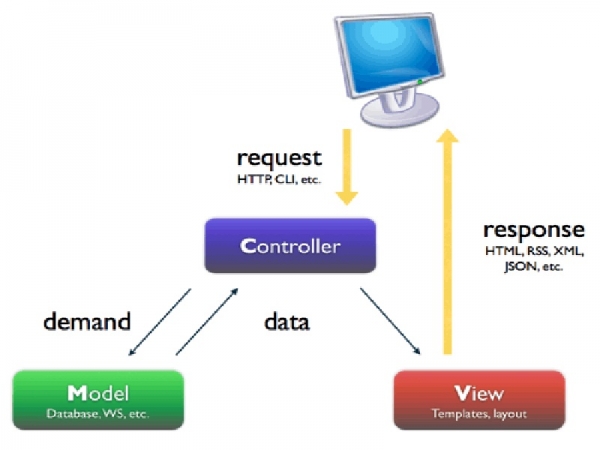
<http://3i.com.vn/index.php/vi/cong-ngh/java>

# MVC là gì ?

Mô hình MVC (Model - View - Controller) là một kiến trúc phần mềm hay mô hình thiết kế được sử dụng trong kỹ thuật phần mềm. Nó giúp cho các developer tách ứng dụng của họ ra 3 thành phần khác nhau Model, View và Controller. Mỗi thành phần có một nhiệm vụ riêng biệt và độc lập với các thành phần khác.

**Các thành phần trong MVC**

Chúng ta khoan hãy tìm hiểu đến cách thức nó hoạt động mà hãy xem nó gồm những gì! Đây là mô hình MVC

[](http://lmt.com.vn/home/media/k2/items/cache/6ddb2450462828abf9aabc88d6bfb7fe_XL.jpg)

**Model**

Đây là thành phần chứa tất cả các nghiệp vụ logic, phương thức xử lý, truy xuất database, đối tượng mô tả dữ liệu như các Class, hàm xử lý...

**View**

Đảm nhận việc hiển thị thông tin, tương tác với người dùng, nơi chứa tất cả các đối tượng GUI như textbox, images...Hiểu một cách đơn giản, nó là tập hợp các form hoặc các file HTML.

**Controller**

Giữ nhiệm vụ nhận điều hướng các yêu cầu từ người dùng và gọi đúng những phương thức xử lý chúng... **Chẳng hạn thành phần này sẽ nhận request từ url và form để thao tác trực tiếp với Model.**

**MVC làm việc như thế nào?**

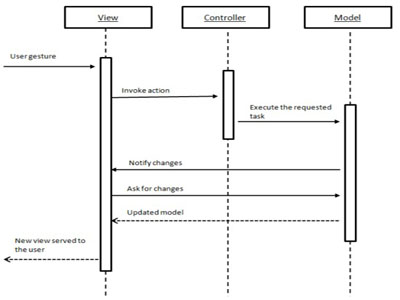
 Đây là một cách đơn giản để mô tả lại luồng sự kiện được xử lý trong MVC:

- User tương tác với View, bằng cách click vào button, user gửi yêu cầu đi.

- Controller nhận và điều hướng chúng đến đúng phương thức xử lý ở Model.

- Model nhận thông tin và thực thi các yêu cầu.

- Khi Model hoàn tất việc xử lý, View sẽ nhận kết quả từ Model và hiển thị lại cho người dùng.



**Ưu điểm và nhược điểm của MVC**

**1. Ưu điểm:**

Thể hiện tính chuyên nghiệp trong lập trình, phân tích thiết kế. Do được chia thành các thành phần độc lập nên giúp phát triển ứng dụng nhanh, đơn giản, dễ nâng cấp, bảo trì..

**2. Nhược điểm:**

Đối với dự án nhỏ việc áp dụng mô hình MC gây cồng kềnh, tốn thời gian trong quá trình phát triển. Tốn thời gian trung chuyển dữ liệu của các thành phần.

# Definitions, Acronyms, and Abbreviations

**Struts** là framework mã nguồn mở được tạo ra để hổ trợ cho các nhà phát triển xây dựng ứng dụng web trên nền tảng kỹ thuật Java Servlet và JSP. Đó là toà nhà chỉ có nền móng vững chắc mà từ đó chúng ta phải xây dựng phần còn lại của cấu trúc, ứng dụng web phải được xây dựng với cùng nguyên tắc đã quy định trong framework struts. Struts cung cấp cho các nhà phát triển một framework đồng bộ và từ đó ứng dụng web được xây dựng, và các nhà phát triển chỉ phải tập trung xây bộ phận nghiệp vụ cho ứng dụng.

Hầu hết chúng ta đều biết **Struts** là **ứng dụng web mã nguồn mở** được phát triển theo khuôn khổ. Struts là một dự án độc lập, là một bộ khung(framework) mã nguồn mở cho việc phát triển ứng dụng web bằng J2EE(Java 2 Enterprise Edition). Nó sử dụng và mở rộng Java Servlet API để giúp cho các nhà phát triển áp dụng kiến trúc MVC(Model-view-controller).

Struts là một tập thư viện các thẻ JSP tùy chọn (Custom JSP Tag): Struts cung cấp các thư viện thẻ tùy chọn cho việc thể hiện các thuộc tính của bean, quản lý các HTML forms, lặp lại các kiểu cấu trúc dữ liệu, và đưa ra các HTML có điều kiện.

## Các thành phần liên quan tới J2EE:

**L**à một chương trình Java chạy trên web browser.

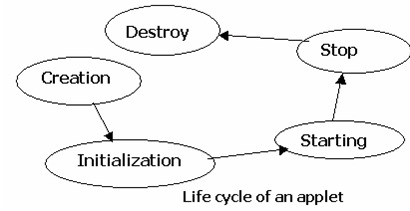
Để tạo ra một applet, ta phải import:

* + java.applet
  + java.awt

Một applet có 4 phương thức. Mỗi phương thức được tự động gọi khi có event tương ứng.

Methods:

* init( )
* start( )
* stop( )
* destroy( )



Sự khác nhau giữa **Applet và Application**

* **Applications** chạy bằng Java interpreter, còn **applets** chạy trên browser có hổ trợ Java, hoặc dùng “**AppletViewer**” có trong JDK.
* Việc thực hiện một applications bắt đầu trong hàm ‘main()’. Còn applet thì không cần hàm main.
* Applications dùng ‘**System.out.println()’** để xuất, trong khi **applets** dùng hàm “**drawString()”** để hiển thị.

**b. Java Bean:**

* JavaBeans là software component model.
* Nó có thể **tương tác với các đối tượng của các software khác** và **có thể được dùng lại**.
* **Các software components mà có thể dùng lại này được gọi là ‘beans’**
* Components được dùng trong **distributed environments**. JavaBeans **được thiết kế để dùng khi cần.**
* **Java thường xây dựng các software thông qua các JavaBeans.**
* Khi mà Java Beans được tạo ra, thì chúng có thể được dùng như một thư viện không phụ thuộc nền.

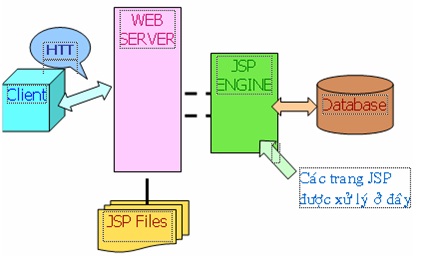
**Các qui tắc để tạo ra một Beans**

* Là một class.
* phải có một default constructor.
* Có ít nhất một thuộc tính và là non-public
* Các thuộc tính được truy xuất từ bên ngoài thông qua các hàm getXXX và setXXX.

**c. JSP (Java Server Pages):**

* JSP technology đươc **xây dựng trên ngôn ngư lap trình Java.**
* Nó **thừa kế tất cả các ưu điểm cũng như các đặc tính của ngôn ngữ Java.**
* **JSP** có thể dùng trên **tất cả các platforms.**

**Cách làm việc của JSP:**



**<tìm hiểu sau…>**

**d. Servlets**  
  
  
Servlets đươc dùng để mở rộng các chức năng của Java-enabled server  
  
Dùng để thay thế cho CGI Scripts  
  
Trước servlets, chúng ta có các công nghệ để tạo ra web applications:

* Common Gateway Interface (CGI)
* FastCGI
* ASP
* Server-side Java Script

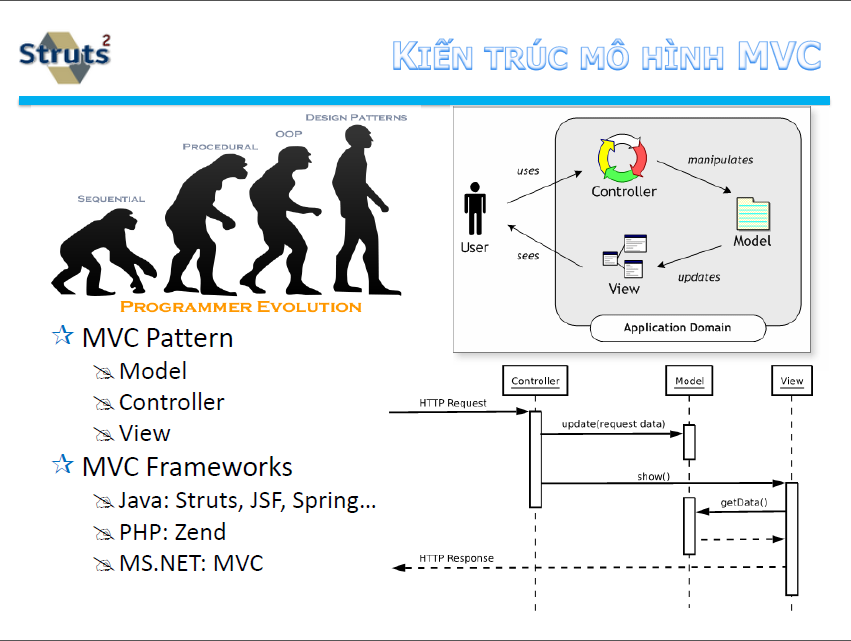
**Đặc điểm Servels:**

* Không bị phụ thuộc
* Làm việc trên nhiều web servers
* Là công nghệ đầu tiên của java để tạo ra web Application
* Không nằm trong core Java API mà **nằm trong hai package javax.servlet và javax.servlet.http của Java Servlet Development Kit (JSDK)**

# Công cụ và môi trường



# Kiến trúc mô hình MVC



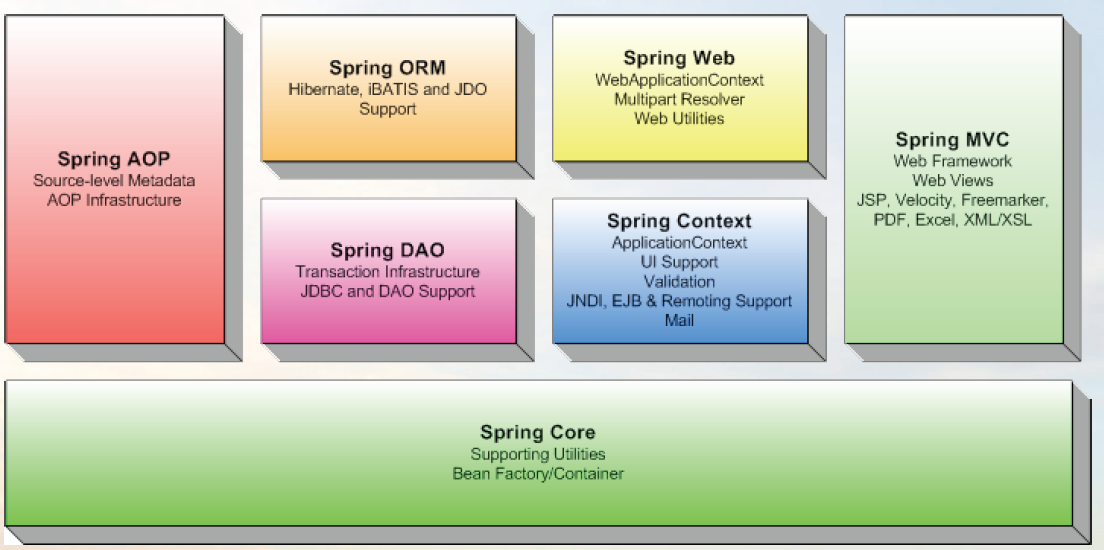
# J2EE Dev : Spring Framework

Dù hiện tại đã có quá nhiều framework J2EE thậm chí nhiều người còn tự xây dựng framework cho chính mình. Nhưng hệ nền Spring mới xuất hiện đã dần khẳng định được vị trí của mình nhờ những ưu điểm rõ rệt và có rất nhiều các nguyên nhân kỹ thuật khác.

## Spring Framework là gì?

**Bản chất framework là các gói thư viện đã implement một hoặc nhiều design pattern nào đó để giúp cho việc xây dựng các ứng dụng dễ dàng hơn và ít tốn effort hơn**. Nói đơn giản thì framework giống như là phần móng và khung sườn của một ngôi nhà, nếu bạn sử dụng lại phần đó thì việc xây nhà thì sẽ hiệu quả hơn nhiều.

Spring là open-source framework được phát triển từ Java. Spring gồm những module chính sau đây:



* **IoC Container**: Đây là phần quan trọng nhất và cũng là phần cơ bản, nền tảng của Spring. Nó giữ vai trò về **cấu hình** và **quản lý lifecycle của các java object**.
* **DAO, ORM, AOP, WEB:**Các module này là **tool** hoặc là **framework** có sẵn được tích hợp vào Spring.

**1. Core Container:**Core container **cung cấp chức năng cơ bản của Spring**. Thành phần chính của nó là Bean Factory, một cài đặt của Factory pattern. BeanFactory áp dụng IoC pattern để đặc tả sự phụ thuộc từ code của ứng dụng.  
  
**2. Spring Context/Application Context:**  
Spring context là một **file cấu hình để cung cấp thông tin ngữ cảnh của Spring**. Spring context **cung cấp các service như JNDI access**, **EJB integration, e-mail, internalization, validation, và scheduling functionality.**  
3. **Spring AOP** (Aspect-Oriented):  
**Spring AOP module tích hợp chức năng lập trình hướng khía cạnh** vào Spring framework thông qua cấu hình của nó. Spring AOP module cung cấp các dịch vụ quản lý giao dịch cho các đối tượng trong bất kỳ ứng dụng nào sử dụng Spring. Với Spring AOP chúng ta có thể tích hợp declarative transaction management vào trong ứng dụng mà không cần dựa vào EJB component.  
  
Spring AOP module cũng đưa lập trình metadata vào trong Spring. Sử dụng cái này chúng ta có thể thêm annotation vào source code để hướng dẫn Spring nơi và làm thế nào để liên hệ với aspect.  
  
**4. Spring DAO:**  
**Tầng JDBC và DAO đưa ra một cây phân cấp exception để quản lý kết nối đến database**, điều khiển exception và thông báo lỗi được ném bởi vendor của database. Tầng exception đơn giản điều khiển lỗi và giảm khối lượng code mà chúng ta cần viết như mở và đóng kết nối. Module này cũng cung cấp các dịch vụ quản lý giao dịch cho các đối tượng trong ứng dụng Spring.  
  
**5. Spring ORM:**  
**Spring có thể tích hợp với một vài ORM framework để cung cấp Object Relation tool bao gồm: JDO, Hibernate, OJB và iBatis SQL Maps.**

**< ORM** là một phương pháp lập trình để chuyển đổi từ mô hình database sang mô hình đối tượng.  
ORM có nhiều thuận lợi hơn so với những phương pháp truy cập dữ liệu (data access) khác:  
ORM tự động hóa việc chuyển đổi từ object sang table và từ table sang object, giúp giảm thời gian và chi phí phát triển  
ORM cần ít code hơn *store procedures*, thay thế số lượng lớn *store procedudres* cần phát triển  
Tăng tốc độ thực thi của hệ thống  
Một giải pháp ORM tốt sẽ giúp ứng dụng nhanh hơn và dễ hỗ trợ hơn.**>**  
**6. Spring Web module:**  
Nằm trên application context module, **cung cấp context cho các ứng dụng web**. **Spring cũng hỗ trợ tích hợp với Struts, JSF và Webwork (frameworks others)** . Web module cũng làm giảm bớt các công việc điều khiển nhiều request và gắn các tham số của request vào các đối tượng domain.  
  
**7. Spring MVC Framework:**  
MVC Framework thì cài đặt đầy đủ đặc tính của MVC pattern để xây dựng các ứng dụng Web. MVC framework thì cấu hình thông qua giao diện và chứa được một số kỹ thuật view bao gồm: JSP, Velocity, Tiles và generation of PDF và Excel file.

**Một định nghĩa khác**

Ví dụ:  
**Có một số kỹ thuật tuyệt vời cho tầng web như: Spring MVC framework, Struts, JSF, WebWork, JSP, Tapestry, FreeMarker,...Developer sẽ bị lúng túng đối chiếu những điểm mạnh và xấu của tất cả chúng. Mỗi khi họ chọn một kỹ thuật và bắt đầu cài đặt, thì sau đó nếu họ muốn thay đổi một kỹ thuật khác thì rất khó. Nhưng Spring đưa ra các module cho tất cả các kỹ thuật trên, và rất đơn giản để thay đổi file cấu hình.**  
Với phương pháp này, nó có khả năng cho cả **team thử và test các tất cả các hình thức trên và xem ảnh hưởng cùng tốc độ trước khi quyết định chọn lựa.**  
  
  
**Spring framework** là một platform hổ trợ lập trình ứng dụng bằng java. Nó được phát minh bởi Rod Johnson. Nó được giới thiệu lần đầu tiên trong cuốn Expert One-on-One J2EE Design and Development, vào năm 2002. Spring đảm nhiệm(xử lý) phần cơ sở hạ tầng của phần mềm để bạn có thể tập trung vào xây dựng ứng dụng của mình.  
  
Spring cho phép chúng ta xây dựng ứng dụng từ POJOs.  
POJOs là viết tắt của "plain old Java objects" chỉ những object java bình thường, chúng không có gì đặc biệt, nó không theo một mô hình hay quy ước nào cả chỉ bao gồm các thuộc tính và phương thức.

**1. Core package** là phần cơ bản nhất của framework, cung cấp những đặc tính IoC và Dependency Injection. Khái niệm cơ bản là BeanFactory - cài đặt factory pattern cho phép bạn móc nối sự phụ thuộc giữa các đối tượng trong file cấu hình. (**IoC và Dependency Injection** mình sẽ giải thik sau )

2. Phía trên của Core package là Context package - cung cấp cách để truy cập đối tượng. Context package kết thừa các đặc tính từ bean package và thêm vào chức năng đa ngôn ngữ (I18N), truyền sự kiện, resource-loading,...

3. DAO package cung cấp cho tầng JDBC, bỏ bớt những coding dài dòng của JDBC và chuyển đổi mã lỗi được xác định bởi database vendor. JDBC package cung cấp cách lập trình tốt như declarative transaction management, không chỉ cho các lớp cài đặt các giao tiếp đặc biệt mà còn cho tất cả POJO (plain old Java objects).

4. ORM package cung cấp tầng tích hợp với object-relational mapping API bao gồm: JDO, Hibernate, iBatis. Sử dụng ORM package bạn có thể sử dụng tất cả các O/R mapper đó kết hợp với tất cả các đặc tính của Spring như declarative transaction management.

5. Spring AOP package cung cấp aspect-oriented programming cho phép bạn định nghĩa method-interceptor và pointcut để móc nối các chức năng được cài đặt trong các module. Sử dụng chức năng metadata bạn có thể kết hợp tất cả thông tin vào code.

6. Spring Web package cung cấp đặc tính của web như: chức năng file-upload, khởi tạo IoC container sử dụng trình lắng nghe serlvet và web-oriented application context. Package này để tích hợp với WebWork và Struts.  
7. Spring MVC package cung cấp mô hình MVC cho ứng dụng web. Spring MVC framework cung cấp sự phân biệt rõ ràng giữa domain model và web form - cho phép bạn sử dụng tất cả các đặc tính khác của Spring framework.  
  
src : <http://khoinguonit.com/f40/hoc-spring-framework-1110/>

<http://springsource-vn.blogspot.com/2013/08/gioi-thieu-ve-spring.html> <-- Ref source

Vd Hello world demo: <https://www.google.com.vn/url?sa=t&rct=j&q=&esrc=s&source=web&cd=6&cad=rja&ved=0CFoQFjAF&url=http%3A%2F%2Fkieutrongkhanh.net%2Findex.php%2Fspring%2F67-spring-30-hello-world&ei=5vKCUuWpIK77iQeVxIHoCQ&usg=AFQjCNGHXlJqiN5_XQzarVpv58nPTSbu2A&sig2=Ig3vhdi5EMG5755R0-nalg&bvm=bv.56343320,d.aGc>

**Một định nghĩa khác**

The **Spring Framework** is an [open source](http://en.wikipedia.org/wiki/Open_source) [application framework](http://en.wikipedia.org/wiki/Application_framework) and [inversion of control](http://en.wikipedia.org/wiki/Inversion_of_control) container for the [Java platform](http://en.wikipedia.org/wiki/Java_platform).[[2]](http://en.wikipedia.org/wiki/Spring_Framework#cite_note-2)

The first version was written by [Rod Johnson](http://en.wikipedia.org/wiki/Rod_Johnson_(programmer)), who released the framework with the publication of his book *Expert One-on-One J2EE Design and Development* in October 2002. The framework was first released under the [Apache 2.0 license](http://en.wikipedia.org/wiki/Apache_License) in June 2003. The first milestone release, 1.0, was released in March 2004, with further milestone releases in September 2004 and March 2005. The Spring 1.2.6 framework won a [Jolt productivity award](http://en.wikipedia.org/wiki/Jolt_Awards)and a [JAX Innovation Award](http://en.wikipedia.org/w/index.php?title=JAX_Innovation_Award&action=edit&redlink=1) in 2006.[[3]](http://en.wikipedia.org/wiki/Spring_Framework#cite_note-3)[[4]](http://en.wikipedia.org/wiki/Spring_Framework#cite_note-4) Spring 2.0 was released in October 2006, Spring 2.5 in November 2007, Spring 3.0 in December 2009, and Spring 3.1 in December 2011. The current version is 3.2.4, which was released in Aug 2013.[[5]](http://en.wikipedia.org/wiki/Spring_Framework#cite_note-5) Spring Framework 4.0 is expected to be released by the end of 2013[[6]](http://en.wikipedia.org/wiki/Spring_Framework#cite_note-6) with support for Java SE 8, [Groovy](http://en.wikipedia.org/wiki/Groovy_(programming_language)) 2, some aspects of Java EE7, and [WebSockets](http://en.wikipedia.org/wiki/WebSocket).

The core features of the Spring Framework can be used by any Java application, but there are extensions for building web applications on top of the[Java EE](http://en.wikipedia.org/wiki/Java_EE) platform. Although the Spring Framework does not impose any specific [programming model](http://en.wikipedia.org/wiki/Programming_model), it has become popular in the Java community as an alternative to, replacement for, or even addition to the [Enterprise JavaBean](http://en.wikipedia.org/wiki/Enterprise_JavaBean) (EJB) model.

* [**Inversion of control**](http://en.wikipedia.org/wiki/Inversion_of_control)**container**: configuration of application components and lifecycle management of Java objects, done mainly via [dependency injection](http://en.wikipedia.org/wiki/Dependency_injection)
* [**Aspect-oriented programming**](http://en.wikipedia.org/wiki/Aspect-oriented_programming): enables implementing [cross-cutting concerns](http://en.wikipedia.org/wiki/Cross-cutting_concern).
* [**Data access**](http://en.wikipedia.org/wiki/Data_access): working with [relational database management systems](http://en.wikipedia.org/wiki/RDBMS) on the Java platform using [JDBC](http://en.wikipedia.org/wiki/JDBC) and [object-relational mapping](http://en.wikipedia.org/wiki/Object-relational_mapping) tools and with [NoSQL](http://en.wikipedia.org/wiki/NoSQL) databases
* [**Transaction management**](http://en.wikipedia.org/wiki/Transaction_processing): unifies several transaction management APIs and coordinates transactions for Java objects
* [**Model–view–controller**](http://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller): an [HTTP](http://en.wikipedia.org/wiki/HTTP)- and [servlet](http://en.wikipedia.org/wiki/Java_Servlet_API)-based framework providing hooks for extension and customization for web applications and [RESTful](http://en.wikipedia.org/wiki/REST) web services.
* **Remote access framework**: configurative [RPC](http://en.wikipedia.org/wiki/Remote_procedure_call)-style [marshalling](http://en.wikipedia.org/wiki/Marshalling_(computer_science)) of Java objects over networks supporting [RMI](http://en.wikipedia.org/wiki/Java_remote_method_invocation), [CORBA](http://en.wikipedia.org/wiki/CORBA) and [HTTP](http://en.wikipedia.org/wiki/HTTP)-based protocols including [web services](http://en.wikipedia.org/wiki/Web_services) ([SOAP](http://en.wikipedia.org/wiki/SOAP_(protocol)))
* [**Convention over configuration**](http://en.wikipedia.org/wiki/Convention_over_configuration): a rapid application development solution for Spring-based enterprise applications is offered in the [Spring Roo](http://en.wikipedia.org/wiki/Spring_Roo) module
* [**Authentication**](http://en.wikipedia.org/wiki/Authentication)**and**[**authorization**](http://en.wikipedia.org/wiki/Authorization): configurable security processes that support a range of standards, protocols, tools and practices via the [Spring Security](http://en.wikipedia.org/wiki/Spring_Security) sub-project (formerly *Acegi Security System for Spring*).
* **Remote management**: configurative exposure and management of Java objects for local or remote configuration via [JMX](http://en.wikipedia.org/wiki/JMX)
* **Messaging**: configurative registration of message listener objects for transparent message-consumption from [message queues](http://en.wikipedia.org/wiki/Message_queue) via [JMS](http://en.wikipedia.org/wiki/Java_Message_Service), improvement of message sending over standard JMS APIs
* [**Testing**](http://en.wikipedia.org/wiki/Software_testing): support classes for writing unit tests and integration tests