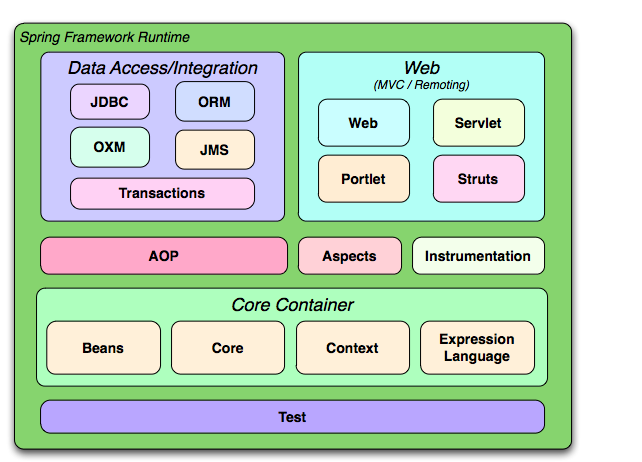
**#2.3 Cấu trúc mô hình Spring framework**

1. **Giới thiệu Spring Frame work**

* Đơn giản hóa phát triển ứng dụng : Spring Framework tập trung vào việc giảm bớt sự phức tạp của phát triển ứng dụng java bằng cách cung cấp cá khái niệm và công cụ giúp làm giảm boilerplate code, tạo điều kiện thuận lợi cho việc xây dựng các ứng dụng mạnh mẽ mà không cần đầu tư quá nhiều công sức
* Spring Framework cung cấp cá công cụ và tính năng để tương tác và tích hợp với các công nghệ và framework khác. Nó hỗ trợ việc tích hợp với các công nghệ như Java EE, Hibernate, JPA, RESTful Web Service, Thymeleaf, và nhiều hơn nữa
* Tích hợp dễ dàng với các container: Spring Framework hỗ trợ việc triển khai ứng dung trên các container như Apache Tomcat, Jetty, IBM WebSphere, và môi trường jva standalone.
  1. **Chèn phụ thuộc và đảo ngược quyền kiểm soát**
* Hỗ trợ nguyên tắc của lập trình hướng đối tượng(OOP) sprig framework của lập trình hương đối tượng như **dependency injection(DI), inversion of control(IoC)** và hợp thành(composition) để tạo ra các mã nguồn dễ đọc, linh hoạt và dễ bảo trì.

**1.2 Module**

- Spring Framework bao gồm các tính năng được tổ chức thành khoảng 20 mô-đun. Các mô-đun này được nhóm thành Bộ chứa lõi, Truy cập/Tích hợp dữ liệu, Web, AOP (Lập trình hướng theo khía cạnh), Thiết bị đo đạc và Kiểm tra, như minh họa trong sơ đồ sau.

1.2.1 Core container

- Beans : Bean là thành phần cốt lỗi trong core container của spring và là trách nhiệm chính của spring container. Bean đại diện cho các đối tượng trong ứng dụng được quản lý bởi Spring. Spring container sẽ tạo, cấu hình và quản lý các bean. Các bean được khai báo và cấu hình thông qua các tệp cấu hình XML, Anotation hoặc Java Base configuration

- Core: core là thành phần quan trọng khác trong core container. Nó chứa các lớp và giao diện cung cấp các tính năng cốt lỗi của Spring như Dependency injection (DI) và inversion control(IoC). Core cung cấp các lớp và giao diện như ApplicationContext, BeanFactory, ResourceLoader và các lơp tiện ích khác để hỗ trợ việc quản lý và tạo ra các Bean.

- Context: Context là thành phần thứ ba trong core container của spring. Nó cung cấp các tính năng phục vụ cho việc tạo và quản lý các Bean, bao gồm cả xử lý sự kiện, quản lý giao dịch và quản lý chuỗi thông báo. Context được xây dựng dự trên Core và cung cấp các chức năng mở rộng để giúp quản lý các bean trong quá trình thực thi ứng dụng

- SpEL( Spring Expression Language) là một ngôn ngữ biểu thức mạnh trong core container của spring. SpEL được sử dụng trong các tệp cấu hình XML, Annotations và Java-based configuration để được hiện các tác vụ

1.2.2  Data Access/Integration

- các Truy cập/Tích hợp dữ liệu lớp bao gồm các mô-đun JDBC, ORM, OXM, JMS và Giao dịch.

- các JDBC mô-đun này cung cấp một lớp trừu tượng JDBC giúp loại bỏ nhu cầu thực hiện mã hóa JDBC tẻ nhạt và phân tích cú pháp các mã lỗi cụ thể của nhà cung cấp cơ sở dữ liệu.

- các ORM mô-đun cung cấp các lớp tích hợp cho các API ánh xạ quan hệ đối tượng phổ biến, bao gồm JPA , JDO , Ngủ đông , Và iBatis . Khi sử dụng gói ORM, bạn có thể sử dụng tất cả các khung ánh xạ O/R này kết hợp với tất cả các tính năng khác mà Spring cung cấp, chẳng hạn như tính năng quản lý giao dịch khai báo đơn giản đã đề cập trước đó.

- các OXM mô-đun cung cấp lớp trừu tượng hỗ trợ triển khai ánh xạ Đối tượng/XML cho JAXB, Castor, XMLBeans, JiBX và XStream.

Dịch vụ nhắn tin Java ( JMS ) mô-đun chứa các tính năng để tạo và tiêu thụ tin nhắn.

* các Giao dịch mô-đun hỗ trợ quản lý giao dịch theo chương trình và khai báo cho các lớp thực hiện các giao diện đặc biệt và cho tất cả các POJO của bạn (các đối tượng Java cũ đơn giản) .

**1.2.3 Web**

- các Web lớp bao gồm các mô-đun Web, Web-Servlet, Web-Struts và Web-Portlet.

Spring Web mô-đun này cung cấp các tính năng tích hợp hướng web cơ bản như chức năng tải lên tệp nhiều phần và khởi tạo bộ chứa IoC bằng cách sử dụng trình nghe servlet và bối cảnh ứng dụng hướng web. Nó cũng chứa các phần liên quan đến web hỗ trợ từ xa của Spring.

các Web-Servlet mô-đun chứa bộ điều khiển mô hình-khung nhìn-mô hình của Spring ( MVC ) triển khai cho các ứng dụng web. Khung MVC của Spring cung cấp sự tách biệt rõ ràng giữa mã mô hình miền và biểu mẫu web, đồng thời tích hợp với tất cả các tính năng khác của Spring Framework.

các Web-Struts mô-đun chứa các lớp hỗ trợ để tích hợp tầng web Struts cổ điển trong ứng dụng Spring. Lưu ý rằng hỗ trợ này hiện không được dùng nữa kể từ Spring 3.0. Hãy cân nhắc việc di chuyển ứng dụng của bạn sang Struts 2.0 và tích hợp Spring của nó hoặc sang giải pháp Spring MVC.

các Web-Portlet mô-đun cung cấp triển khai MVC để sử dụng trong môi trường portlet và phản ánh chức năng của mô-đun Web-Servlet.

**1.2.4 AOP and Instrumentation**

- Spring AOP mô-đun cung cấp một Liên minh AOP -Việc triển khai lập trình hướng theo khía cạnh tuân thủ cho phép bạn xác định, chẳng hạn như các bộ chặn phương thức và các điểm cắt để tách mã rõ ràng thực hiện chức năng cần được tách riêng. Bằng cách sử dụng chức năng siêu dữ liệu cấp nguồn, bạn cũng có thể kết hợp thông tin hành vi vào mã của mình theo cách tương tự như thuộc tính .NET.

- Sự riêng biệt Các khía cạnh mô-đun cung cấp khả năng tích hợp với AspectJ.

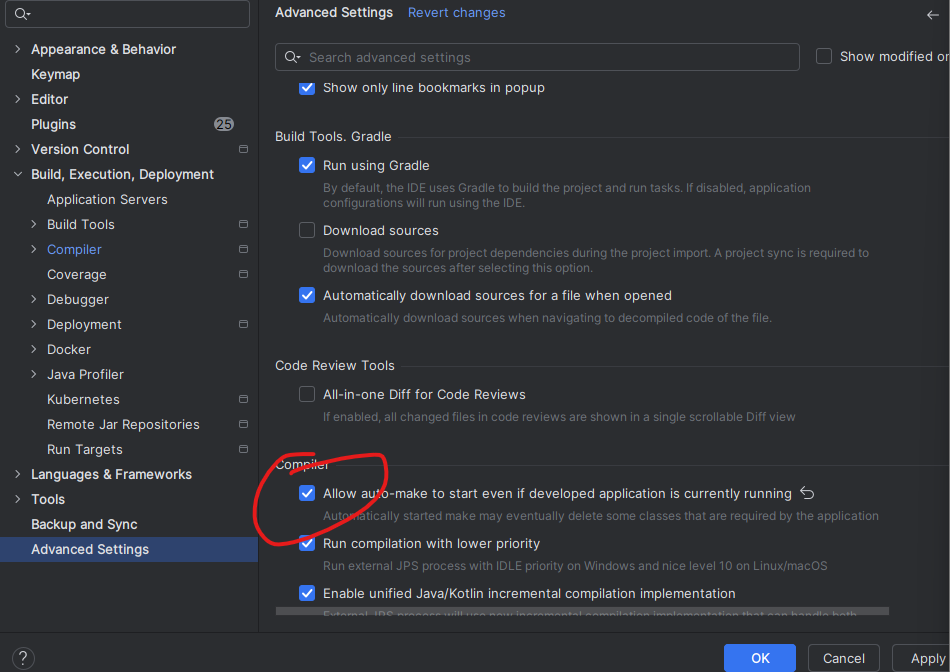
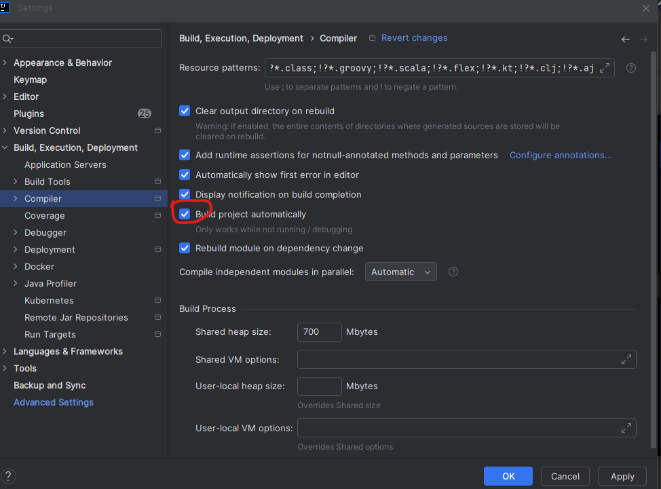
- các thiết bị đo đạc mô-đun cung cấp hỗ trợ thiết bị đo lớp và triển khai trình nạp lớp để sử dụng trong một số máy chủ ứng dụng nhất định.

**1.2.3 Test**

- các Bài kiểm tra mô-đun hỗ trợ thử nghiệm các thành phần Spring với JUnit hoặc TestNG. Nó cung cấp khả năng tải nhất quán của Spring ApplicationContexts và bộ nhớ đệm của các bối cảnh đó. Nó cũng cung cấp các đối tượng mô phỏng mà bạn có thể sử dụng để kiểm tra mã của mình một cách riêng biệt.

**#3.5 Spring boot devtools**

* Spring boot DevTools là một module hỗ trợ trong Sping Boot:
* Khởi động tự động: khi các tệp tin trong dự án của bạn thay đổi , ứng dụng sẽ tự động khởi động lại để áp dụng những thay đổi này mà không cần khởi động thủ công
* Tải tự động(Live Reload) : khi bạn thay đổi tệp tin trong tài nguyên tĩnh(css, JavaScript, HTML), trình duyệt sẽ tự động làm mới trang web để hiển thị những thây đổi mới nhất
* Khởi động nhanh: Sping boot Devtools sử dụng một số kỹ thuật như chia tách lớp, việc khởi tạo động và ẩn đi các thành không cần thiết để giúp giảm thời gian khởi động ứng dụng
* Cấu hình:



#3.7 Actuator

* Spring boot Actuator là một sub-project của Spring boot. Actuator cho phép ta theo dõi, giám sát ứng dụng, thu thập số liệu , lưu lượng truy cập hay trạng thái cơ sở dữ liệu , v.v mà không cần thêm bất cứ dòng code nào
* Khi project của ta được cấu hình Spring Actuator thì mặc định ta sẽ có sẵn cá endpoint để quản lý và theo dõi ứng dụng của ta.
* Trong trường hợp ta muốn có thêm những kiểm soát khác cho ứng dụng thì ta có thể thêm endpoint mới của riêng mình