

1. 스프링의 정의

- Spring이란?
 - JAVA의 웹 프레임워크로 JAVA 언어를 기반으로 사용한다.
JAVA로 다양한 어플리케이션을 만들기 위한 프로그래밍 틀이라 할 수 있다.
 - JAVA의 활용도가 높아지면서, JAVA를 이용한 기술이 JSP, Mybatis, JPA 등의 기술이 생겨났다.
Spring은 다른 사람의 코드를 참조하기 쉽고 편리한 구조로
앞서 말한 기술들을 더 쉽게 사용해주는 오픈소스 프레임워크 이다.
-

2. 프레임워크(Frame Work)

- Frame Work란?
 - 프레임워크는 어떠한 목적을 달성하기 위해, 복잡하게 얹혀 있는 문제를 쉽게 해결하기 위한
약속이자 도구이며, 소프트웨어 개발에 하나의 뼈대 역할을 한다.
 - 프레임워크는 자주 쓰일 만한 기능들을 한데 모아 놓은 유틸(클래스)들의 모음이다.
 - 의자를 만든다고 가정 할때 의자를 만드는 망치 나 못 같은 개념

기본적인 설계나 필요한 라이브러리는 알아서(의존성 주입 해주면) 할테니 개발자는 개발 역량에 집중 !

3. 라이브러리와 프레임워크의 차이점

- 라이브러리
 - 개발자가 개발을 진행하다. 라이브러리의 필요한 순간을 인지하고 라이브러리를 추가한다.
 - 소프트웨어를 개발 할 때 사용되는 자원의 모임, 필요 할때 자유롭게 사용하는 도구
- 프레임워크
 - 개발을 수월하게 하기위해 제공된 소프트웨어 개발 환경이다.
사용자가 세세하게 신경쓰지 않아도 기능을 확장하거나 유지보수 할 수 있게 클래스의 구조나 데이터를 처리하는 절차 등에 대한 설계 등에 대한 구조를 제공받는 가이드라인이라 할 수 있다.
 - 라이브러리 + 설계도 의 개념으로
의자를 만든다고 가정 할때 의자를 만들 수 있는 공방의 개념

의자를 만드는 설계도, 망치나 못 같은 도구, 어느 정도 자동화된 도구 선택 방식 등이라 할 수 있다.

- API(Application Programming Interface)
 - 응용 프로그램에서 운영 체제나 프로그래밍 언어가 제공하는 기능을 제어 할 수 있는 인터페이스
 - URI를 통해 데이터를 전달 받는 형태가 많고, 구현 혹은 다른 코드와 독립적으로 사용방법이 정의 되어 있다.
 - 방을 꾸민다고 할 때 의자 그 자체, 개발과 별개로 다른 코드에 비교적 독립적으로 사용하고 개발자의 의도와 별개로 사용방법을 정의하고 있다.
-

4. 스프링 프레임워크의 특징

1. IoC(Inversion of Control, 제어 반전)

- 개발자는 JAVA 코딩시 new 연산자, 인터페이스 호출, 데이터 클래스 호출 방식으로 객체를 생성, 소멸시킨다.
여기서 IoC란 객체의 생성부터 소멸까지 개발자가 아닌 스프링컨테이너가 대신 해주는 것이다
- 제어권이 개발자가 아닌 IoC에 있으며,
IoC가 개발자의 코드를 호출해 필요한 객체를 생성, 소멸 하며 생명주기를 관리하는 것이다.

2. DI(Dependency Injection, 의존성 주입)

- 프로그램에서 구성 요소의 의존 관계가 소스코드 내부가 아닌 외부의 설정 파일을 통해 정의 되는 방식이다.
- 코드 간의 재사용을 높이고, 소스코드를 다양한 곳에 사용하며 모듈 간의 결합도를 낮출 수 있다.
- 대표적으로 라이브러리나 API, 프레임워크를 연동 할 때 연결하는 소스코드를 직접 작성하는게 아닌 외부 파일을 연결해 불러오는 방식이다.

3. AOP(Aspect Object Programming, 관점 지향 프로그래밍)

- 로깅, 트랜잭션, 보안 등 여러 모듈에서 공통적으로 사용하는 기능을 분리하여 관리 할 수 있다.
- 각각의 클래스가 있다고 가정하자. 각 클래스들은 서로 코드와 기능들이 중복되는 부분이 많다. 코드가 중복될 경우 실용성과 가독성 및 개발 속도에 좋지 않다. 중복된 코드를 최대한 배제하는 방법은 중복되는 기능들을 전부 빼놓은 뒤 그 기능이 필요할때만 호출하여 쓰면 훨씬 효율성이 좋다.

- 즉, AOP는 여러 객체에 공통으로 적용할 수 있는 기능을 구분함으로써 재사용성을 높여주는 프로그래밍 기법이다.

4. POJO(Plain Old Java Object) 방식

- POJO는 Java EE를 사용하면서 해당 플랫폼에 종속되어 있는 무거운 객체들을 만드는 것에 반발하여 나타난 용어이다.
- 별도의 프레임 워크 없이 Java EE를 사용할 때에 비해 인터페이스를 직접 구현하거나 상속 받을 필요가 없어 기존 라이브러리를 지원하기 용이하고, 객체가 가볍다.
- 즉, getter/setter를 가진 단순한 자바 오브젝트를 말한다.