**스프링(Spring)이란**

@ 스프링(Spring)

- EJB를 주 프레임워크로 사용할 때 불편했던 점들을 해소

- Design and Development에 선보인 코드가 Spring의 근간이 됨

@ 스프링 프레임워크란?

- Java 엔터프라이즈 개발을 편하게 해주는 오픈소스 경량급 애플리케이션 프레임워크

@ 애플리케이션 프레임워크

특정 계층이나 기술, 업무 분야에 국한되지 않고 애플리케이션의 전 영역을 포괄하는

범용적인 프레임워크를 말함

@ 경량급 프레임워크

단순한 웹 컨테이너에서도 엔터프라이즈 개발의 고급기술을 대부분 사용할 수 있음

@ 엔터프라이즈 개발 용이

개발자가 복잡하고 실수하기 쉬운 low level에 많이 신경 쓰지 않으면서 Business Logic

개발에 전념할 수 있도록 해줌

@ 오픈소스

Spring은 OpenSource의 장점을 충분히 취하면서 동시에 오픈소스 제품의 단점과 한계를

잘 극복함

@ Spring Framework 특징

1. 컨테이너 역할

- Spring 컨테이너는 Java객체의 Life Cycle을 관리하며, Spring 컨테이너로부터

필요한 객체를 가져와 사용할 수 있다

2. DI (Dependency Injection) 지원

- Spring은 설정 파일이나 어노테이션을 통해서 객체 간의 의존관계를 설정할 수

있도록 하고 있다

3. AOP (Aspect Oriented Programming) 지원

- Spring은 트랜잭션이나 로깅, 보안과 같이 공통적으로 필요로 하는 모듈들을

실제 핵심 모듈에서 분리해서 적용할 수 있다

4. POJO(Plain Old Java Object) 지원

- Spring 컨테이너에 저장되는 Java객체는 특정한 인터페이스를 구현하거나,

특정 클래스를 상속받지 않아도 된다

5. 트랜잭션 처리를 위한 일관된 방법을 지원

- JDBC, JTA 등 어떤 트랜잭션을 사용하던 설정을 통해 정보를 관리하므로

트랜잭션 구현에 상관없이 동일한 코드 사용 가능

6. 영속성(Persistence)과 관련된 다양한 API지원

- Spring은 MyBatis, Hibernate 등 데이터베이스 처리를 위한

ORM(Object Relational Mapping) 프레임워크들과의 연동 지원

@ 메이븐(Maven)

Spring을 사용하기 위해서는 Maven Build가 필요함

- 자바 개발의 사실상 표준 빌드 툴. 이전에는 ANT를 많이 사용

- XML 설정파일을 사용. 이후에 gradle이 등장

@ Java 기반 Web Application의 Best Practice를 따름

- 개발 디렉토리 구조

- 빌드 단계

@ 의존성 관리를 자동으로 수행

- Maven 중앙 저장소(Central Repository)를 제공하여 자바 라이브러리에 대한 생태계조성

@ POM.xml (메이븐 메인 설정파일)

- 프로젝트 루트에 위치

- 메이븐 프로젝트를 의미. IDE에서 불러오기 쉬움

@ 프로젝트명

- artifact ID로 사용

@ 그룹 아이디

- 주로 프로젝트 생성 조직이나 기관의 도메인 명 역순으로 표기

- ex) kr.co.company

- 일반적으로 package명으로 사용됨

@ 버전

- Release처럼 배포 버전 등을 사용

@ IoC (Inversion of Control)

개념

- Ioc(제어의 역전)이란, 객체의 생성, 생명주기의 관리까지 모든 객체에 대한 제어권이

바뀌었다는 것을 의미한다

- 컴포넌트 의존관계 결정 설정 및 생명주기(Life Cycle)를 해결하기 위한 디자인 패턴

@ IoC컨테이너

- 스프링 프레임워크도 객체에 대한 생성 및 생명주기를 관리할 수 있는 기능을

제공하고 있음. 즉, IoC컨테이너 기능을 제공한다

- IoC컨테이너는 객체의 생성을 책임지고, 의존성을 관리한다

- POJO의 생성, 초기화, 서비스, 소멸에 대한 권한을 가진다

- 개발자들이 직접 생성할 수 있지만 컨테이너에게 맡긴다

@ DI (의존성 주입)

각 클래스 간의 의존관계를 빈 설정(Bean Definition) 정보를 바탕으로 컨테이너가

자동으로 연결해 주는 것

@ Managed Bean

- 스프링 컨테이너에 의해서 관리되는 객체

- 스프링 빈이라고도 함

- 스프링 설정파일에 등록되어 사용

@ Spring Container

- 관리되어지는 빈이 모여 있는 곳

- IoC 컨테이너로써 Application Context 클래스로 구현됨

@ Application Context

- Bean을 등록, 생성, 조회, 반환 관리하는 기능

- Spring의 각종 부가 서비스를 추가로 제공함

@ 스프링 MVC 패턴의 개념

- MVC(Model-View-Controller) 패턴의 개념

모델 뷰 컨트롤러(MVC)는 소프트웨어 공학에서 사용되는 아키텍처 패턴으로

MVC 패턴의 주 목적은 Business Logic과 Presentation Logic을 분리하기 위함

- MVC 패턴을 사용하면 사용자 인터페이스로부터 비즈니스 로직을 분리하여

애플리케이션의 시각적 요소나 그 이면에서 실행되는 비즈니스 로직을 서로

영향 없이 쉽게 고칠 수 있는 애플리케이션을 만들 수 있다

@ 각각의 MVC 컴포넌트의 역할

- Model : 애플리케이션의 정보 (데이터, Business Logic 포함)

1. 데이터 저장소와 연동하여 사용자가 입력한 데이터나 사용자에게 출력할

데이터를 다루는 일을 함

2. 여러 개의 데이터 변경(추가, 변경, 삭제)을 하나의 작업으로 묶는 트랜잭션을

다루는 일도 함

3. DAO, Service 클레스에 해당

- View : 사용자에게 제공할 화면 (Presentation Logic)

1. Controller의 처리 결과 화면에 대한 정보를 보유한 객체

2. 클라이언트에게 보여지는 영역

- Controller : Model과 View 사이의 상호 작용을 관리

1. 클라이언트의 요청을 받았을 때 그 요청에 대해 실제 업무를 수행하는 모델

컴포넌트를 호출하는 일을 함

2. 클라이언트가 보낸 데이터가 있다면, 모델을 호출할 때 전달하기 쉽게

데이터를 적절히 가공하는 일을 함

3. 모델이 업무 수행을 완료하면, 그 결과를 가지고 화면을 생성하도록 뷰에게

전달 (모델과 뷰를 결정하여 클라이언트에 전달)

4. Servlet과 JSP를 사용하여 작성(이젠 거의 JSP만 사용)

@ Spring MVC의 특징

- 스프링은 DI나 AOP와 같은 기능 뿐만 아니라, 서블릿 기반의 웹 개발을 위한 MVC

프레임워크를 제공

- 스프링 MVC 프레임워크는 Spring을 기반으로 하고 있기 때문에 Spring이 제공하는

트랜잭션 처리나 DI 및 AOP 등을 손쉽게 사용가능

@ Log4j

- 로그 출력 등급

FATAL : 애플리케이션을 중지해야 할 심각한 오류

ERROR : 오류가 발생했지만, 애플리케이션은 계속 실행할 수 있는 상태

WARN : 잠재적인 위험을 안고 있는 상태

INFO : 애플리케이션의 주요 실행 정보

DEBUG : 애플리케이션의 내부 실행 상황을 추적해 볼 수 있는 상세 정보

TRACE : 디버그보다 더 상세한 정보

@ 마이바티스(MyBatis) 프레임워크

MyBatis는 자바 오브젝트와 SQL문 사이의 자동 Mapping 기능을 지원하는 Object

Mapper이다

- MyBatis는 Sql을 별도의 파일로 분리해서 관리하게 해주며, 객체-SQL 사이의 파라미터

Mapping 작업을 자동으로 해주기 때문에 많은 인기를 얻고 있는 기술이다

- 마이바이티스나 Hibernate와 같은 JPA(Java Persistence Api)처럼 새로운 DB프로그래밍

패러다임을 익혀야 하는 부담이 없이, 개발자가 익숙한 sql을 그대로 이용하면서

JDBC 코드 작성의 불편함도 제거해주고, DTO 객체나 VO 객체를 중심으로 개발이

가능하다는 장점이 있다

@ MyBatis의 특징

1. 쉬운 접근성과 코드의 간결함

- 간단한 퍼시스턴스 프레임워크

- XML 형태로 서술된 JDBC 코드라고 생각해도 될 만큼 JDBC의 모든 기능을

MyBatis가 대부분 제공함

- 복잡한 JDBC 코드를 걷어내며 깔끔한 소스코드를 유지할 수 있음

- 수동적인 파라미터 설정과 쿼리 결과에 대한 맵핑 구문을 제거할 수 있음

2. SQL문과 프로그래밍 코드의 분리

- SQL에 변경이 있을 때마다 자바 코드를 수정하거나 컴파일 하지 않아도 됨

- SQL작성과 관리 또는 검토를 DBA와 같은 개발자가 아닌 다른 사람에게

맡겨도 된다

3. 다양한 프로그래밍 언어로 구현 가능(지원)

- Java, C#, NET, Ruby 등등

@ ORM이란?

Object Relational Mapping, 객체 관계 매핑

- 객체와 관계형 데이터 베이스의 데이터를 자동으로 매핑(연결) 해주는 기술을

말한다

- 객체지향 프로그래밍은 클래스를 사용하고, 관계형 데이터베이스는 테이블을

사용한다

- ORM을 통해 객체 간의 관계를 바탕으로 SQL을 자동으로 생성하여 불일치를

해결해준다