**Spring Framework**

**설치**

**목표 :**  이클립스에서 별도의 세팅없이 모두 한번에 스프링 개발을 메이븐으로 관리하는 것을 목표로 한다.

**1. JAVA, STS install, tomcat, maven**

- java, tomcat, maven 환경변수 설정

**2. 이클립스에서 maven 플러그인 설치**

- SpringMVC프로젝트가 자동으로 작성해준 pom.xml 정보를 메이븐 플러그인이 읽어들여 프로젝트가 생성됨과 동시에 필요한 라이브러리 목록을 가져와 준다.

이제 당신은 힘겹게 인터넷을 돌아다니며 필요한 라이브러리를 받을 필요가 전혀 없어졌다. 필요한 라이브러리가 있다면 아래의 그림과 같이 pom.xml을 열어 Dependencies에서 검색한 뒤 pom.xml을 저장만 하면 통합 메이븐 플러그인이 알아서 해당 플러그인을 찾아준다. 그야말로 최고의 제작환경이 갖추어 졌다고 할 수 있다.

**ORM : Object Relation Mapping.**

 오브젝트와 RDB 사이에 존재하는 개념과 접근방법, 성격의 차이 때문에 요구되는 불편한 작업을 제거해줘서 자바 개발자가 오브젝트를 가지고 정보를 다루면 ORM 프레임워크가 이를 RDB에 적절한 형태로 변환해주거나 그 반대로 RDB에 저장되어 있는 정보를 자바 오브젝트가 다루기 쉬운 형태로 변환해주는기술이다.

MyBatis

- 스프링 -

**컨테이너란?**

컨테이너는 적절한 설정만 되있다면 누구의 도움없이도 프로그래머가 작성한 코드를 스스로 참조한 뒤 알아서 객체의 생성과 소멸을 컨트롤 해준다.

프로그래머가 작성한 코드는 컨테이너를 사용하게 됨으로서 프로그래머의 손을 떠나 컨테이너의 영역으로 떠나게 된다. 스프링과 더블어 대표적 컨테이너는 톰캣과 같은 WAS(web application service)이다.

**Loc/Di란? = 획기적인 프로그래밍 작성 방식**

Inversion of Control과 Dependency Injection의 줄임말

제어의 역전(대신해줌), 의존적 주입(미리 찜해놓음)

제어의 역전 : 컨테이너다. 당신이 작성한 코드 컨트롤(객체 생성과 소멸)을 대신 해주는데, loc덕분에 우리는 언제 호출될지 모르는 코드를 마음 껏 작성할 수 있다.

의존성 주입 : 이것은 마치 당신이 받아먹기도 전에 미리 받아먹겠노라고 선언하는..주는놈은 생각도 않았는데 먼저 말부터 꺼내놓은 파렴치한 코드방식이다.

간단하게 오버라이딩을 생각해 보면, 인터페이스에 메서드를 호출하면 이를 구현한 객체의 메서드가 호출된다.(오버라이딩) 구현부는 나중에 주입을 통해 해결하므로획기덕 분업과 동시에 구현 클래스를 쉽게 교체할 수 있다는 장점을 얻게 된다.