**[ 01 ] Spirng 개요**

1. 프레임워크 : 특정한 목적에 맞게 프로그래밍을 쉽게 하기 위한 프로그래밍 틀을 약속
2. Spring이란?
   * 자바(JAVA)언어를 기반으로, 다양한 어플리케이션을 제작할 때 효율적으로 빠르게 개발할 수 있도록 하는 어플리케이션 프레임워크(프로그래밍 툴) ; Rod Johnson이 만든 오픈 소스 프레임워크
   * 스프링의 경우 톰캣을 이용할 수 있으며, EJB에 비해서 코드의 **경량화** 그리고 개발 중에 **테스트가 쉽다는** 점이 특징 입니다.
   * DI(Dependency Injection)지원
   * AOP(Aspect Oriented Programming)지원 : 공통기능을 쉽게 개발하고 관리하는 기능
   * 연속성과 관련된 다양한 API 지원 : JDBC, myBatis등 데이터 베이스 처리 라이브러리 연동 지원. 메일, 스케줄링 등을 지원
   * 스프링의 국내에서 자바개발자들에게 웹어플리케이션 표준프레임워크
   * 스프링의 기본을 정확히 학습한다면, 규모가 큰 다양한 어플리케이션을 효율적으로 개발할 수 있을 것 입니다.
   * 물론 스프링학습 이전에 JAVA언어 및 JSP&Servlet, MVC(model 2) 패턴에 대한 선행학습이 반드시 필요하며, 그 외에 html, javascript, jquery, css등의 스크립트 언어도 기본적인 학습이 필요합니다.
3. 개발환경 : Eclipse 또는 STS
4. 설치
   * JDK 설치 : <http://java.sun.com>
   * JAVA 환경변수 설정 : bin/javac.exe(.java파일을 JVM이 받아들일 수 있는 .class파일로 변환시켜주는 프로그램)를 어디서나 실행 가능하게 하도록 하기 위함.
   * 서버 (엄밀히 말하면 웹컨테이너 톰캣)설치 <http://tomcat.apache.org> 에서 다운후 이클립스에서 서버 연동
   * IDE(Integrated Development Environment 통합개발툴) 이클립스 다운로드 : <http://www.eclipse.org>에서 Eclips IDE for Java EE Developers 다운
   * 이클립스에 스프링(STS) 플로그인 설치
     1. 이클립스 Help – Eclipse Maketplace – sts 입력 후 맞는 STS 선택하여 설치
   * 이클립스 대신 STS 다운로드 받아도 됨

다운 로드 : http://spring.io

Spring Framework 5.0.2.RELEASE API : <http://docs.spring.io/spring/docs/current/javadoc-api/>

1. 처음 만들어보는 스프링 프로젝트
   * 스프링 프레임워크에 적합한 개념(DI, AOP 등)을 익히기 위한 기초를 튼튼히 하기 위한 예제를 3일정도 진행할 예정입니다. 기초를 튼튼하게 되면, 추후 스프링 레퍼런스 문서를 참고해서 스스로 학습할 수 있기 때문입니다

**public** **class** Calculation {

**private** **int** firstNum;

**private** **int** secondNum;

**public** **int** getFirstNum() {

**return** firstNum;

}

**public** **void** setFirstNum(**int** firstNum) {

**this**.firstNum = firstNum;

}

**public** **int** getSecondNum() {

**return** secondNum;

}

**public** **void** setSecondNum(**int** secondNum) {

**this**.secondNum = secondNum;

}

**public** **void** add(){

System.***out***.println("add()");

System.***out***.println(firstNum + "+" + secondNum + "=" + (firstNum+secondNum));

}

**public** **void** sub(){

System.***out***.println("sub()");

System.***out***.println(firstNum + "-" + secondNum + "=" + (firstNum-secondNum));

}

**public** **void** mult(){

System.***out***.println("mult()");

System.***out***.println(firstNum + "\*" + secondNum + "=" + (firstNum\*secondNum));

}

**public** **void** div(){

System.***out***.println("div()");

System.***out***.println(firstNum + "/" + secondNum + "=" + (firstNum/secondNum));

}

}

**public** **class** MainClass {

**public** **static** **void** main(String[] args) {

Calculation cal = **new** Calculation();

cal.setFirstNum(10);

cal.setSecondNum(2);

cal.add();

cal.sub();

cal.mult();

cal.div();

}

}

위에서 아주 간단한 스프링 프로젝트(Simple spring maven)를 만들어보았습니다. 위의 경우 MainClass는 Calculation 객체에 의존하는 프로젝트였습니다.

1. DI(Dependency Injection)와 IOC 컨테이너(Inversion of Control ; 컴포넌트의 재사용을 용이하게 해주고 단위 테스트를 쉽게 할 수 있도록 지원하는 스프링, Spring)

A객체는 B/C 객체에 의존한다. Ex. MainClass는 Calculation객체에 의존한다

A객체 : A객체가 B/C객체를 직접 생성(방법1)

A객체

방법1

New B()

New C()

A객체 : B/C객체를 외부에서 생성하여 주입(방법2)

**IOC 컨테이너**

= spring

<인터페이스를 통한 부품화>

Setters

방법2

New B()

New C()

방법1. A객체가 B/C객체를 직접 생성한다. new B() new C()로

방법2. B/C객체 외부에 생성하여 A객체에 넣어준다. setter()나 construct()로

결국 스프링이란 부품을 생성하고 조립하는 라이브러리 집합체라고 할 수 있다