**[ 01 ] Spring 개요**

1. 프레임워크 : 특정한 목적에 맞게 프로그래밍을 쉽게 하기 위한 프로그래밍 틀을 약속
2. Spring이란?
   * 자바(JAVA)언어(JSP)를 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.
   * Spring Framework는 종속성 주입, 트랜잭션 관리, 웹 앱, 데이터 액세스, 메시징 등을 위한 핵심 지원을 제공한다(Spring Framework privides core support for dependency injection, transaction management, web apps, data access, messaging, and more).
     1. DI(Dependency Injection)지원
     2. AOP(Aspect Oriented Programming)지원 : 공통기능을 쉽게 개발하고 관리하는 기능
     3. 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>
     1. 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 서버 연동
   * 이클립스에 스프링(Spring Tools 3 Add-On for Spring Tools 4 3.9.14.SNAPSHOT) 플로그인 설치
   * 이클립스 대신 STS 다운로드 받아도 됨
     1. <http://spring.io/tools> 에서 sts 다운로드 (자바와 톰캣은 이미 설치되어 있어야 한다)

다운 받은 spring-tool-suite-3.9.13.RELEASE-e4.16.0-win32-x86\_64.zip(spring-tool-suite-3.9.14.RELEASE-e4.17.0-win32-x86\_64.zip 다운로드시 ini수정)을 d:/webPro/IDE 압축을 푼다

* + 1. Lombok을 다운받아 Lombok 추가실행(cmd창에서 java –jar Lombok.jar) 및 설정
    2. Lombok 및 한글 인코딩 설정(sts.ini)

-vmargs

-javaagent:lombok.jar

-Dosgi.requiredJavaVersion=1.8

-vm

C:\Program Files\Java\jdk1.8.0\_321\bin\javaw.exe

-javaagent:D:\Nam\IDE\sts-3.9.14.RELEASE\lombok.jar

**-Dfile.encoding=utf-8**

* + 1. sts (-clean)실행 (Gradle IDE pack 플러그인)
    2. Spring Framework 5.0.2.RELEASE API : <http://docs.spring.io/spring/docs/current/javadoc-api/>
    3. 필요한 메이븐 라이브러리 : C:\Users\컴퓨터이름\.m2\repository

1. 처음 만들어보는 스프링 프로젝트(Simple Spring Utility Project vs. Spring MVC Project)
   * 스프링 프레임워크에 적합한 개념(DI, AOP 등)을 익히기 위한 기초를 튼튼히 하기 위한 예제를 웹프로젝트 전에 진행할 예정입니다. 기초를 튼튼하게 되면, 추후 스프링 레퍼런스 문서를 참고해서 스스로 학습할 수 있기 때문입니다
   * peom.xml에 추가

<!-- https://mvnrepository.com/artifact/org.projectlombok/lombok -->

<dependency>

<groupId>org.projectlombok</groupId>

<artifactId>lombok</artifactId>

<version>1.18.26</version>

<scope>provided</scope>

</dependency>

**import** lombok.AllArgsConstructor;

**import** lombok.Data;

**import** lombok.NoArgsConstructor;

@Data

@NoArgsConstructor

@AllArgsConstructor

**public** **class** Calculation { // 럼복을 이용하여 생성자, setter&getter,toString 추가

**private** **int** num1;

**private** **int** num2;

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

System.***out***.println("더하기");

System.***out***.println(num1 + " + " + num2 + "=" +(num1+num2));

}

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

System.***out***.println("빼기");

System.***out***.println(num1 + " - " + num2 + "=" +(num1-num2));

}

**public** **void** mul() {

System.***out***.println("곱하기");

System.***out***.println(num1 + " \* " + num2 + "=" +(num1\*num2));

}

**public** **void** divS() {

System.***out***.println("나누기");

System.***out***.println(num1 + " / " + num2 + "=" +(num1/num2));

}

}

**public** **class** CalTestMain {

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

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

cal.setNum1(10);

cal.setNum2(5);

cal.add();

cal.sub();

cal.mul();

cal.div();

}

}

위의 경우 CalTestMain는 Calculation 객체에 의존하는 간단한 프로젝트였습니다.

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

CalTestMain 클래스객체는 X와 C 객체에 의존한다. Ex. CalTestMain은 Calculation객체에 의존한다

CalTestMain객체 : X/C객체를 직접 생성(방법1)

CalTestMain 클래스 객체

방법1

New X()

New C()

CalTestMain객체 : X/C객체를 외부에서 생성하여 주입(방법2)

방법2

**IOC 컨테이너**

= spring

<인터페이스를 통한 부품화>

New X()

New C()

방법1. CalTestMain객체가 X/C객체를 직접 생성한다. new X() new C()로

방법2. X/C객체 외부에 생성하여 CalTestMain객체에 넣어준다.

결국 스프링이란 필요한 객체를 생성하고 조립하는 라이브러리 집합체라고 할 수 있다