

[01] Spring 개요

1. 프레임워크 : 특정한 목적에 맞게 프로그래밍을 쉽게 하기 위한 프로그래밍
툴을 약속

2. Spring이란?

자바 언어를 기반으로 , 다양한 어플리케이션을 제작할 때 효율적으로
빠르게 개발할 수 있도록 하는 어플리케이션 프레임 워크(프로그래밍 툴);
Rod Johnson 이 만든 오픈 소스 프레임워크

3. JDK 설치 : <http://java.sun.com>

- i. JAVA 환경변수 설정 : bin/javac.exe(java파일을 JVM이 받아들일 수 있는 .
class파일로 변환시켜주는 프로그램)를
어디서나 실행 가능하게 하도록 하기 위함.

서버 <http://tomcat.apache.org> 에서 다운후 이클립스에서 서버 연동

IDE(Integrated Development Environment 통합개발툴) 이클립스 다운로드

: <http://www.eclipse.org>에서 Eclipse IDE for Java EE Developers 다운

-IDE 톱캣 하나 더만들기

-spring.io 사이트가기

-projects - > Spring Tool 4 -> 3.9.14 다운

- :D 에 파일 이름 줄여서 sts 바꾸고 알집으로 압축풀기

- sts-bundle IDE에 옮기기

4 .<https://projectlombok.org/download> 다운 dto 만들때 게터 세터 생성자 toString 자동 생성

-mvnrepository 가서 lombok 1.18.24 복붙

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

```
<dependency>
```

```
    <groupId>org.projectlombok</groupId>
```

```
    <artifactId>lombok</artifactId>
```

```
    <version>1.18.24</version>
```

```
    <scope>provided</scope>
```

```
</dependency>
```

-----Calculation.java-----

* 어노테이션

@Data : toString

@AllArgsConstructor : 매개변수 있는 생성자 함수

@NoArgsConstructor : 매개변수 없는 생성자함수

-----CalTestMain.java -----

CalTestMain -> Calculation 의존 !

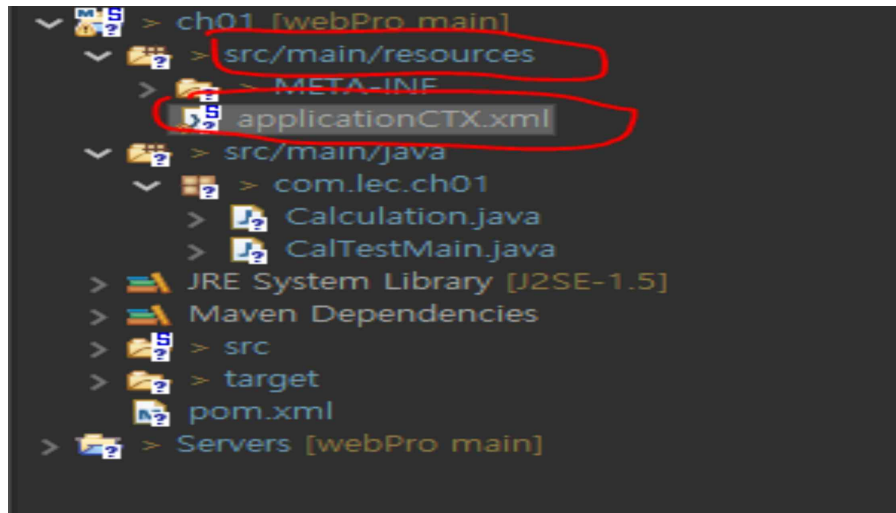
★ .DI(Dependency Injection; 의존주입 = 부품조립)

방법 1 . A객체 : A객체가 B/C객체를 직접 생성(방법1)

- New B() / New C()

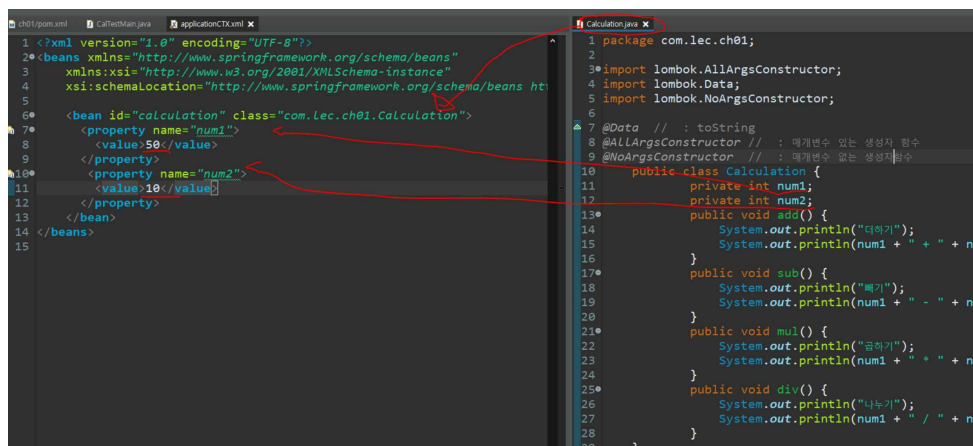
```
public class CalTestMain {
    // CalTestMain -> Calculation 의존
    public static void main(String[] args) {
        Calculation cal = new Calculation();
        cal.setNum1(50);
        cal.setNum2(10);
        cal.add();
        cal.sub();
        cal.mul();
        cal.div();
        System.out.println(cal);
    }
}
```

방법2 : B/C객체를 외부에서 생성하여 주입(방법2)



-Spring Bean 으로 생성

-----applicationCtx.xml-----



1. applicationCtx에 num1 , num2 받아오고 벨류값 설정

-----다시 CalTestMain.java-----

```
package com.lec.ch01;

import org.springframework.context.support.AbstractApplicationContext;
import org.springframework.context.support.GenericXmlApplicationContext;

public class CalTestMain {
    // CalTestMain -> Calculation 의존
    public static void main(String[] args) {
        Calculation cal = new Calculation();
        // cal.setNum1(50);
        // cal.setNum2(10);
        AbstractApplicationContext ctx
            = new GenericXmlApplicationContext("classpath:applicationCTX.xml");
        Calculation cal = ctx.getBean("calculation", Calculation.class);
        cal.add();
        cal.sub();
        cal.mul();
        cal.div();
        System.out.println(cal);
    }
}
```

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans.xsd">
    <!-- 객체이름 -->
    <bean id="calculation" class="com.lec.ch01.Calculation">
        <property name="num1">
            <value>50</value>
        </property>
        <property name="num2">
            <value>10</value>
        </property>
    </bean>
</beans>
```

Calculation : 객체이름

applicationCTX.xml : 파일명

순서 1. Calculation.java 에서 num1, num2 생성

더하기 빼기 곱하기 나누기 함수 생성

2. applicationCTX.xml 에서

- id = "calculation" (객체이름) class = "파일경로?"
- 벨류값 설정

3. CalTestMain.java

- 함수 호출 결과값 출력