[02] DI(Dependency Injection) 개념 및 활용

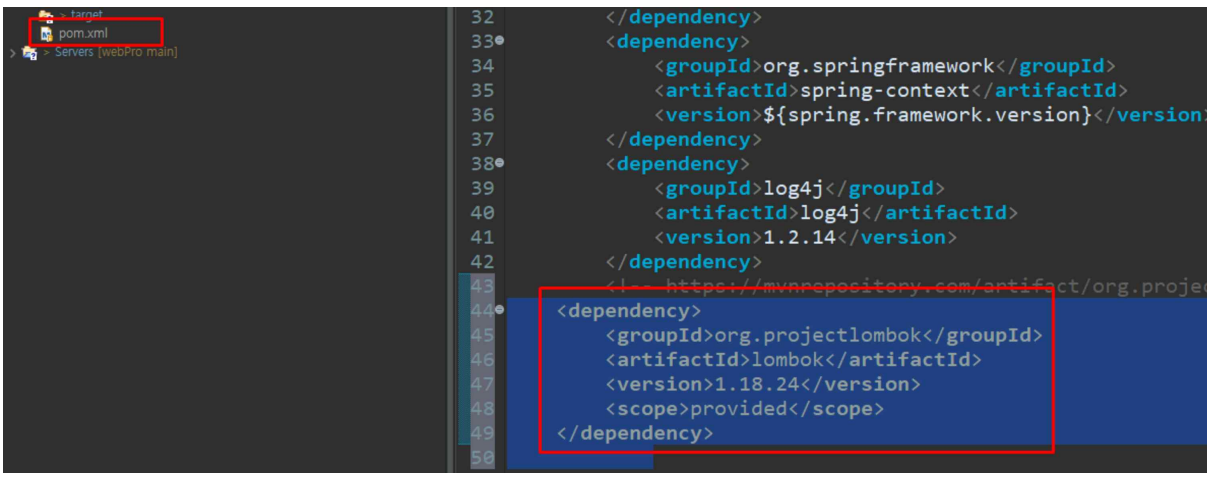
1. **스프링을 이용한 객체 생성과 조립(스프링은 객체 생성과 조립을 하는 컨테이너라 볼 수 있다)**

✓ 스프링을 사용하지 않은 프로젝트

```
Calculation cal = new Calculation();
cal.setNum1(10);
cal.setNum2(5);
cal.add();
cal.sub();
cal.mul();
cal.div();
```

1 ex01_cal Calculato.java 함수 생성 애를 의존하는 MyCalcuator.java 만들기

2.



Pom.xml에 복붙 항상 !

2. 함수 호출

MyCalculator.java

```java
package com.lec.ch02;

import lombok.Data;

@Data
public class MyCalculator {

        private Calculator calculator;
        private int num1;
        private int num2;
        public void add() {   // 더하기
            calculator.addition(num1, num2);
        }
        public void sub() {  // 빼기
            calculator.subtraction(num1, num2);
        }
        public void mul() {  // 곱하기
            calculator.multiplication(num1, num2);
        }
        public void div() {  // 나누기
            calculator.division(num1, num2);
        }
}

```

Calculator.java 의존 받는 MyCalcu
함수 호출

3. applicationCTX1 ...

ch02_DI/pom.xml applicationCTX1.xml

```xml
ml version="1.0" encoding="UTF-8"?>
ans 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:/
 <bean id="calculator" class="com.lec.ch02.ex1_cal.Calculator"/>

 <bean id="myColculator" class="com.lec.ch02.ex1_cal.MyCalculator"/>
eans>
```

urce Namespaces Overview beans Beans Graph

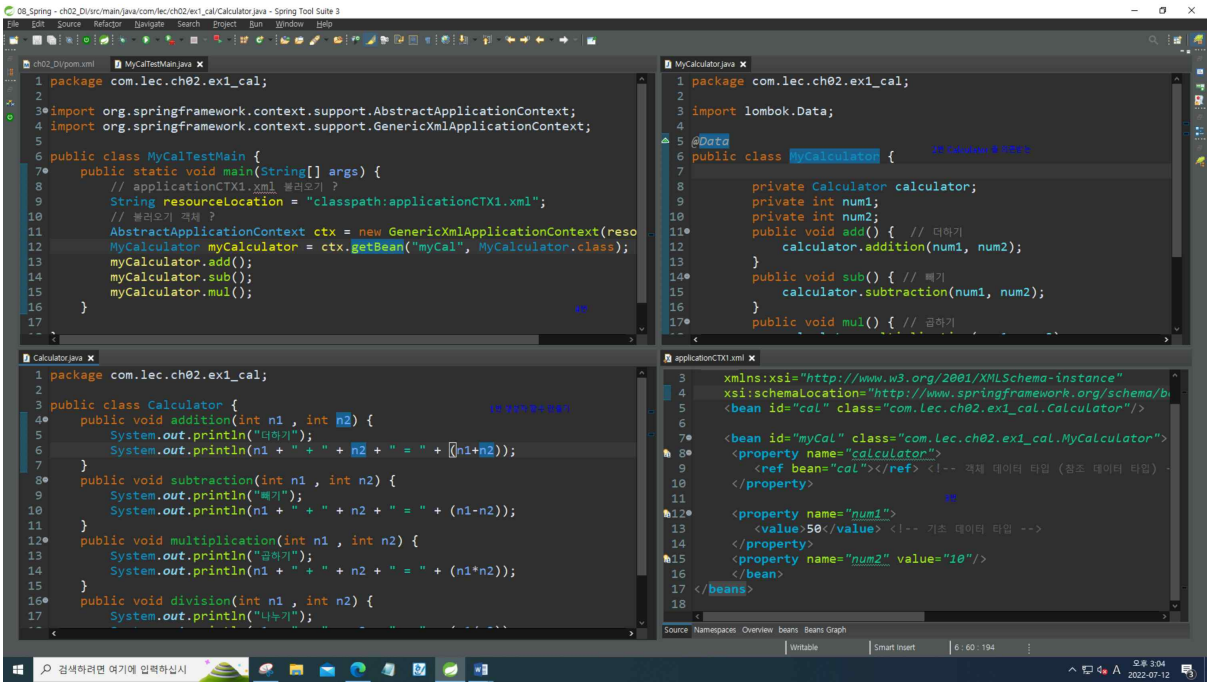
Calculator.java

```java
public class Calculator {
    public void addition(int n1 , int n2) {
        System.out.println("더하기");
```

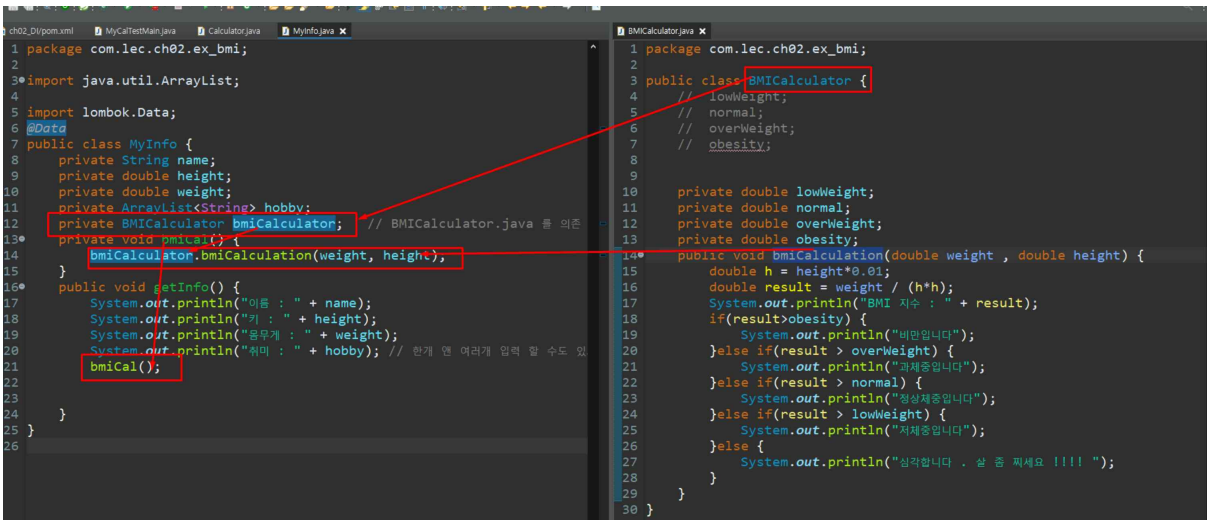
MyCalculator.java

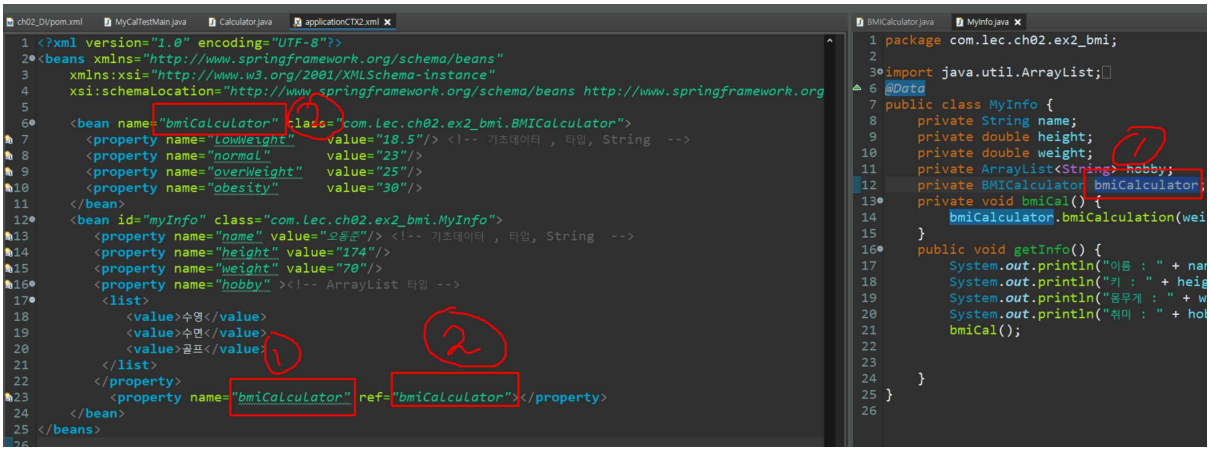
```java
package com.lec.ch02.ex1_cal;

import lombok.Data;

@Data
public class MyCalculator {

        private Calculator calculator;
        private int num1;
        private int num2;
        public void add() {  // 더하기
            calculator.addition(num1, num2);
        }
        public void sub() { // 빼기
            calculator.subtraction(num1, num2);
        }
        public void mul() { // 곱하기
            calculator.multiplication(num1, num2)
        }
        public void div() { // 나누기
            calculator.division(num1, num2);
        }
}
```

(순서)


08_Spring - ch02_DI/src/main/java/com/lec/ch02/ex1_cal/Calculator.java - Spring Tool Suite 3
MyCalTestMain.java
MyCalculator.java
Calculator.java
applicationCTX1.xml

Ex02_Bmi


MyInfo.java
BMICalculator.java



1번에서 값을 셋팅 하지 않으면 null 이 들어와 에러가 나 항상 셋팅

2번 ref는 위에 bean name 과 이름을 같게 해줘야함

-Student-

1. Student.java

-변수 선언

2. StudentInfo.java 생성

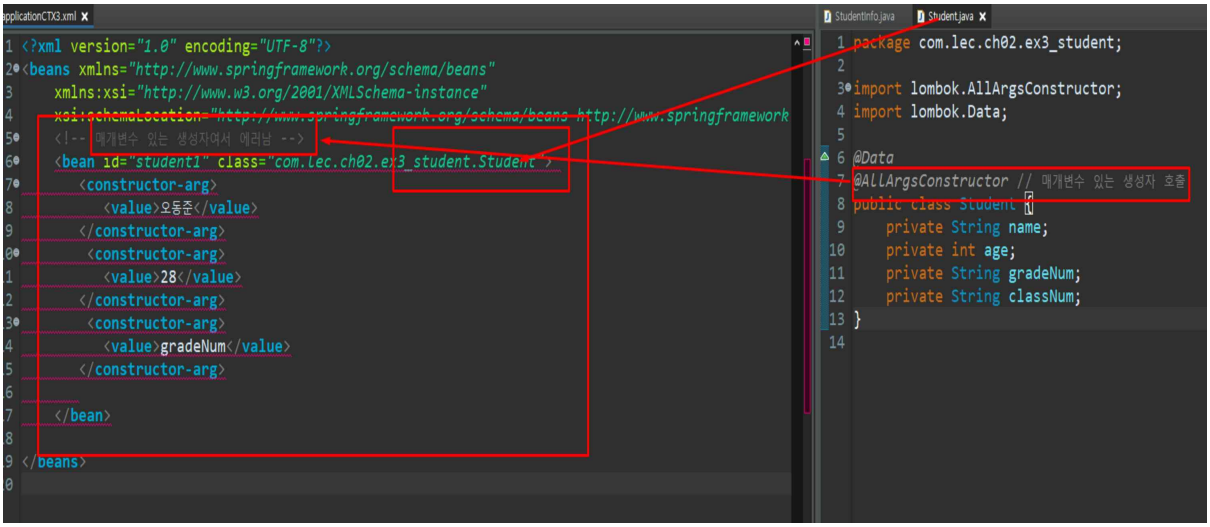
applicationCTX3.xml | StudentInfo.java | Student.java

```java
package com.lec.ch02.ex3_student;

import lombok.AllArgsConstructor;

@AllArgsConstructor
@Data
public class StudentInfo { //Student 를 의존 받는 StudentInfo
    private Student student; 1
    public void getInfo() { 2
        if(student != null) {
            System.out.println("이름 : " + student.getName());
            System.out.println("나이 : " + student.getAge());
            System.out.println("학년 : " + student.getGradeNum());
            System.out.println("반 : " + student.getClassNum());
            System.out.println("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~");
        }else {
            System.out.println("의존하는 Student형 객체가 없습니다");
        }
    }
}

```

1번 : Student.java에 의존 받음

2번 : getInfo() 생성자 함수 생성

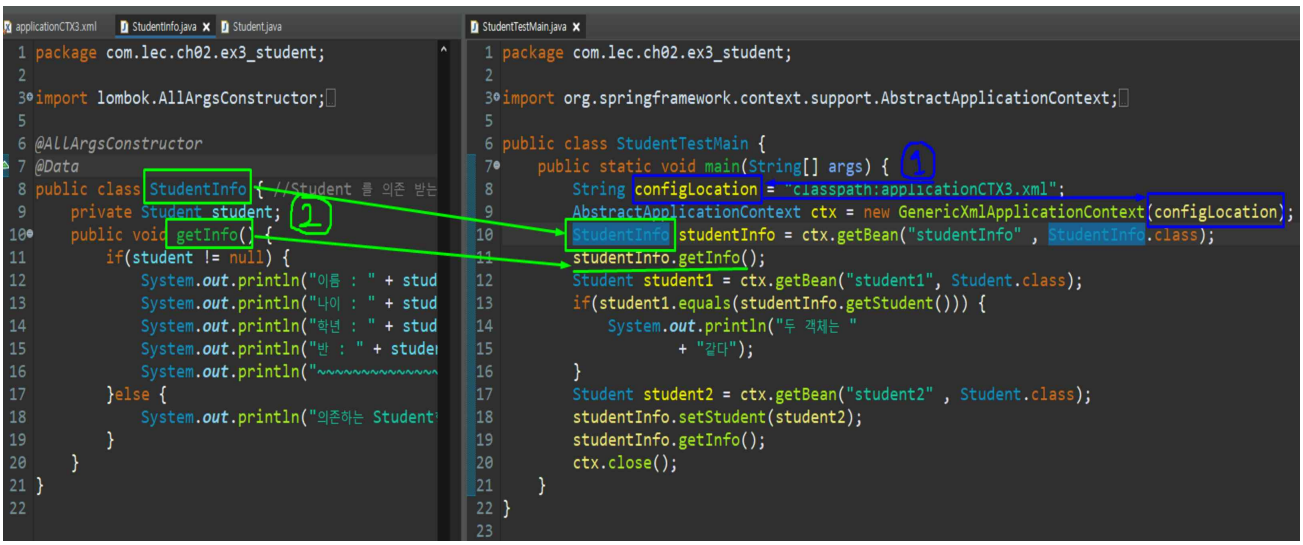
3 . CTX3.xml 생성



AllArgsConstructor : 매개변수 있는 생성자여서

Xml에 <value>값 안 넣으면 에러 발생

4. TestMain(실행화면) StudentInfo.java -> StudentTestMain

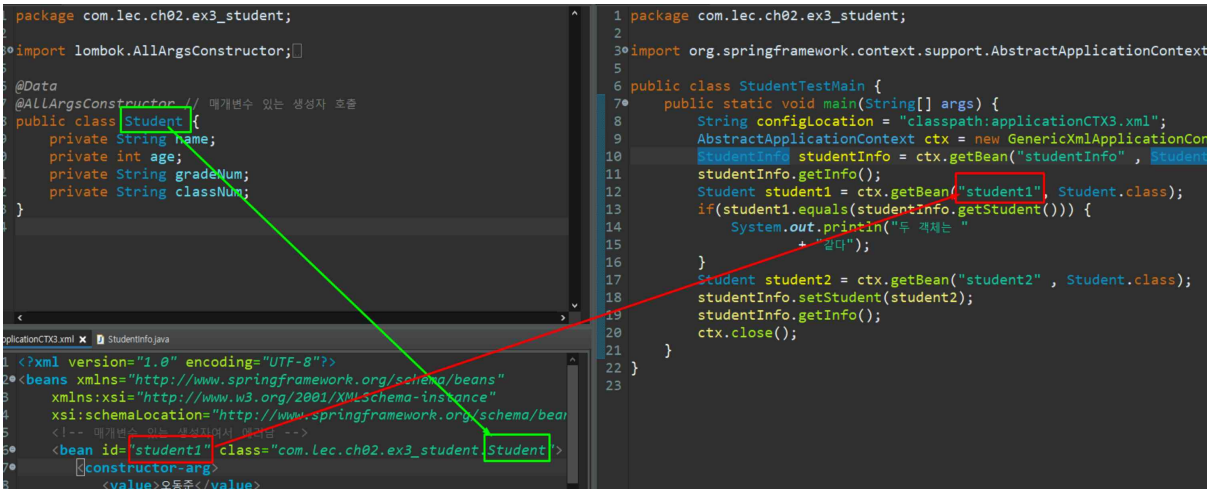


1.

CTX3.xml 받아오기

2.StudentInfo 호출 , getInfo 생성자 함수 호출

밑에 이어서 ..



1.Student 클래스 xml 에 id = student1 에 받아오고

2.testMain 에서 “student1” 받아와서 새로운 변수 sstudent1 에 넣기