**[ 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();

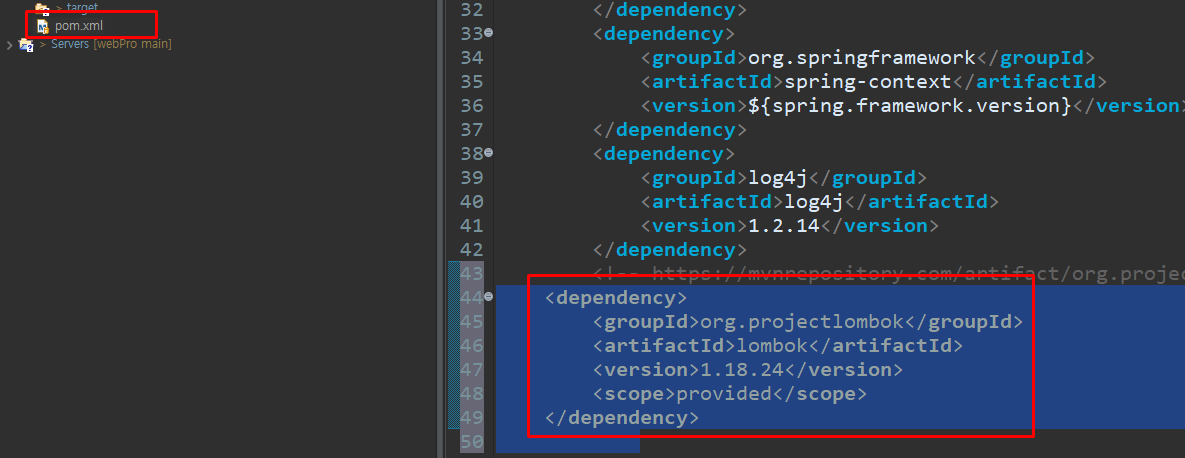
cal.sub();

cal.mul();

cal.div();

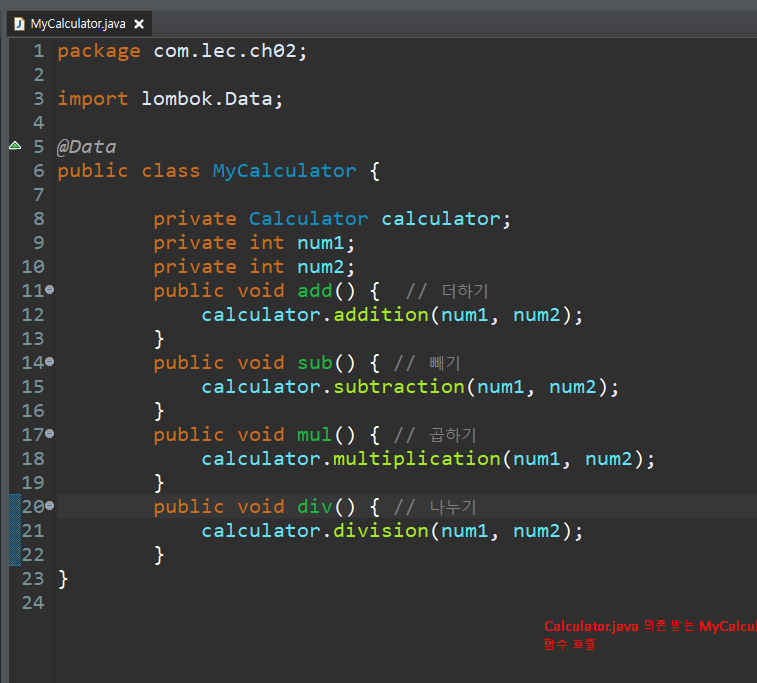
**1 ex01\_cal Calculato.java 함수 생성 애를 의존하는 MyCalcuator.java 만들기**

**2.**

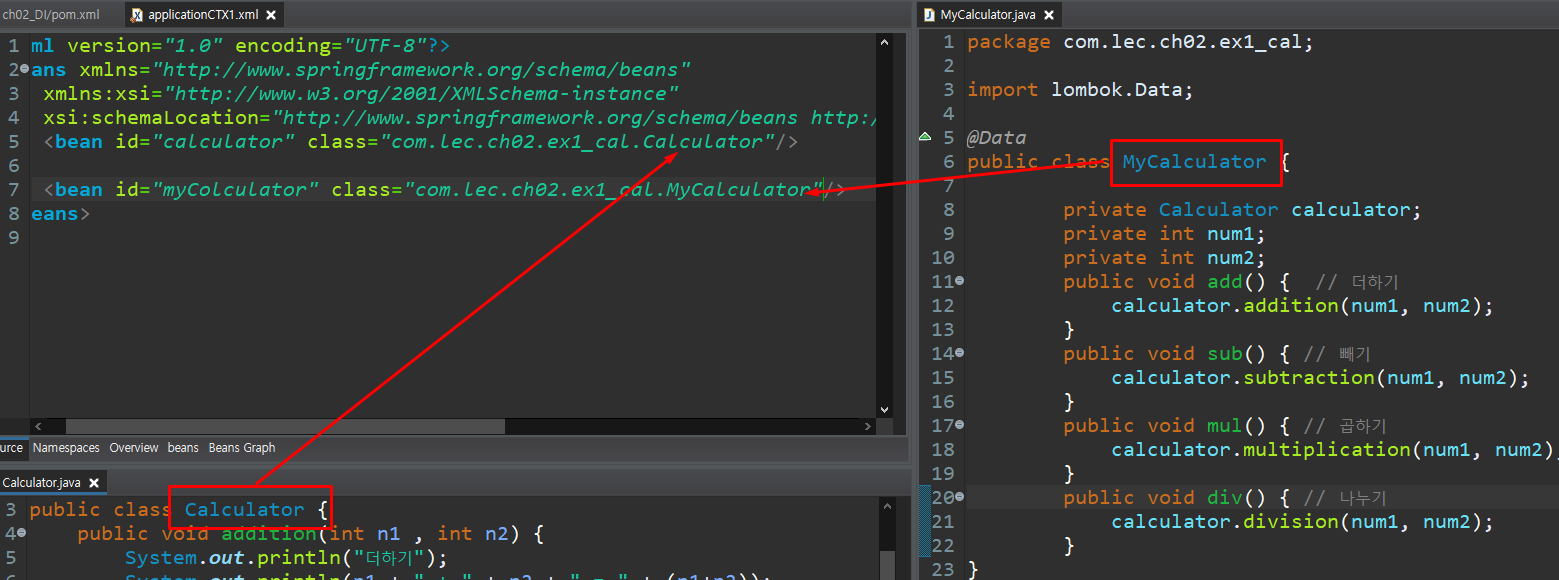


**Pom.xml에 복붙 항상 !**

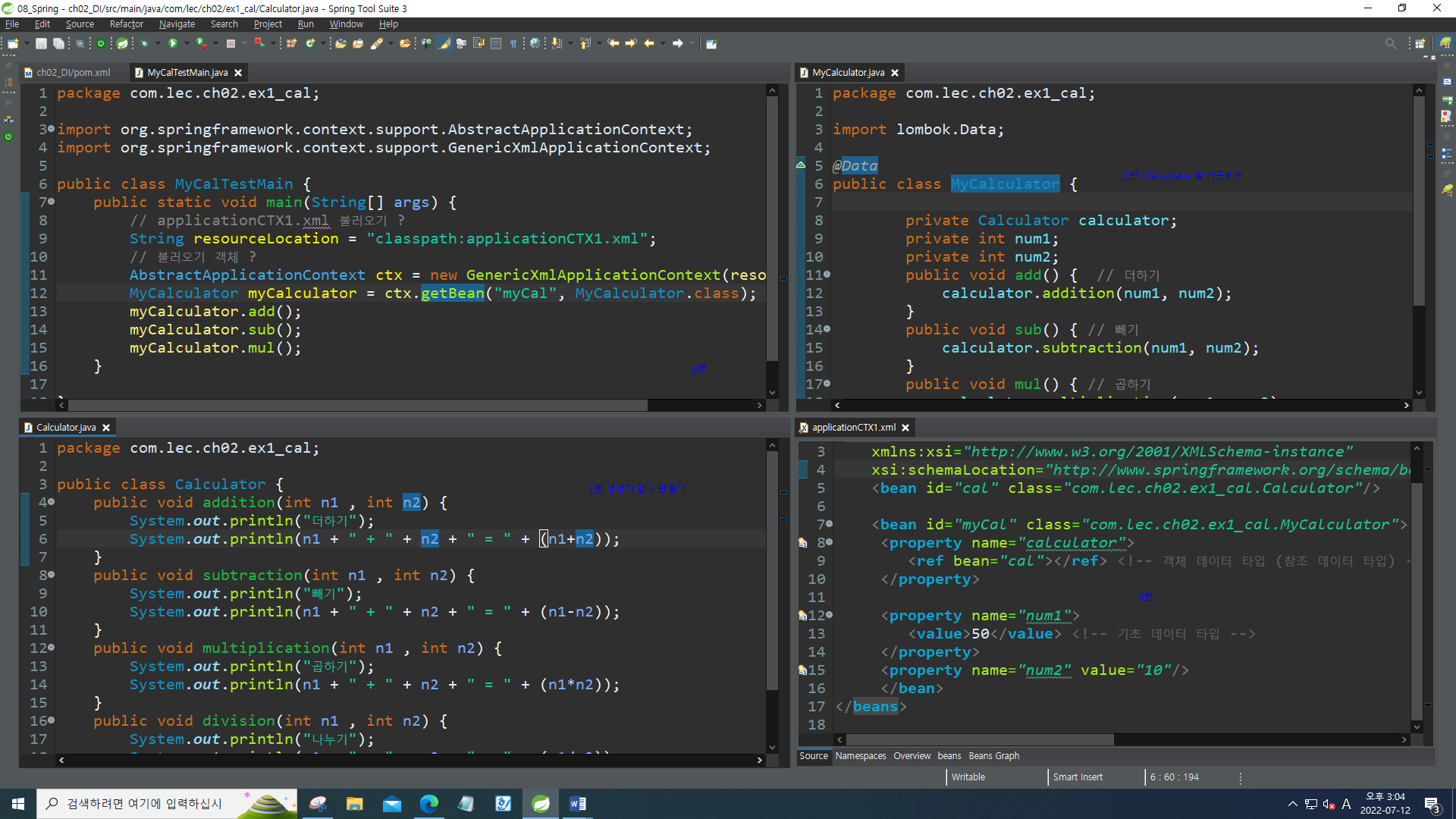
1. **함수 호출**



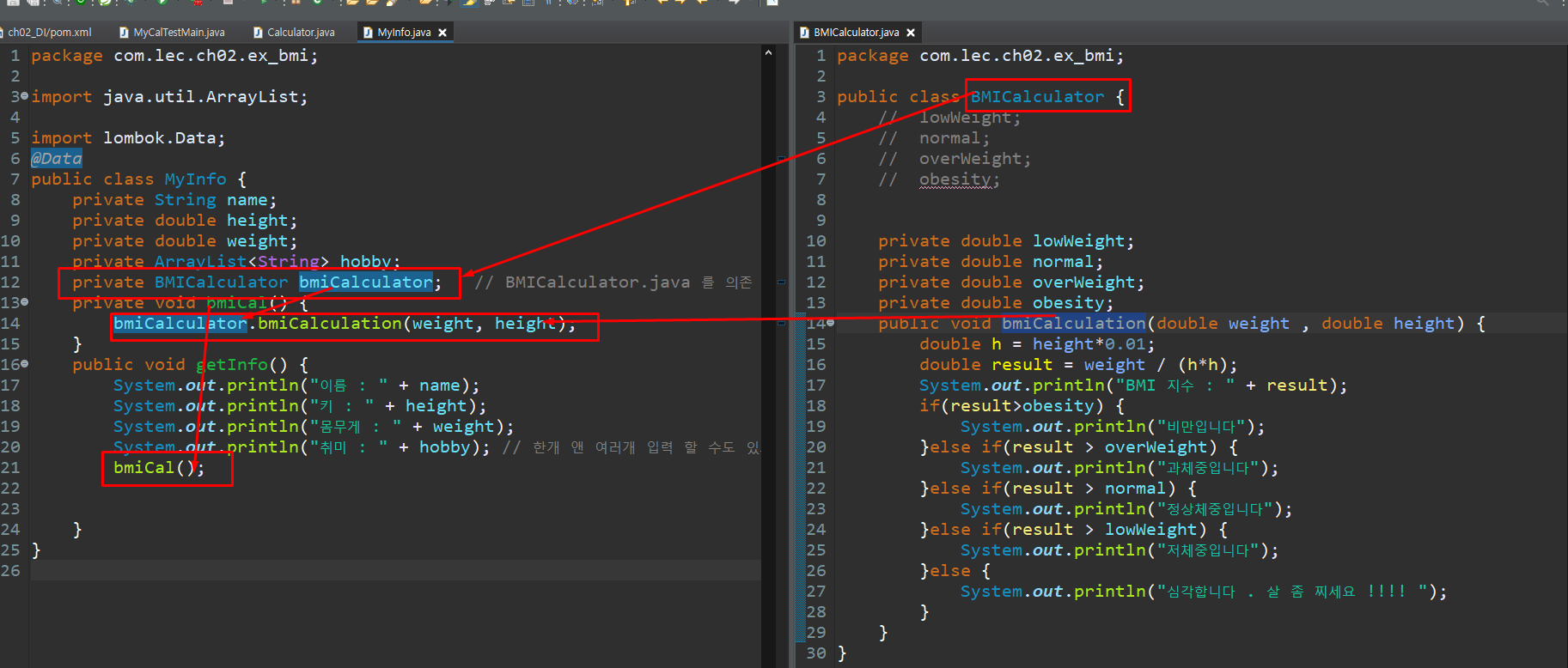
1. **applicationCTX1 …**

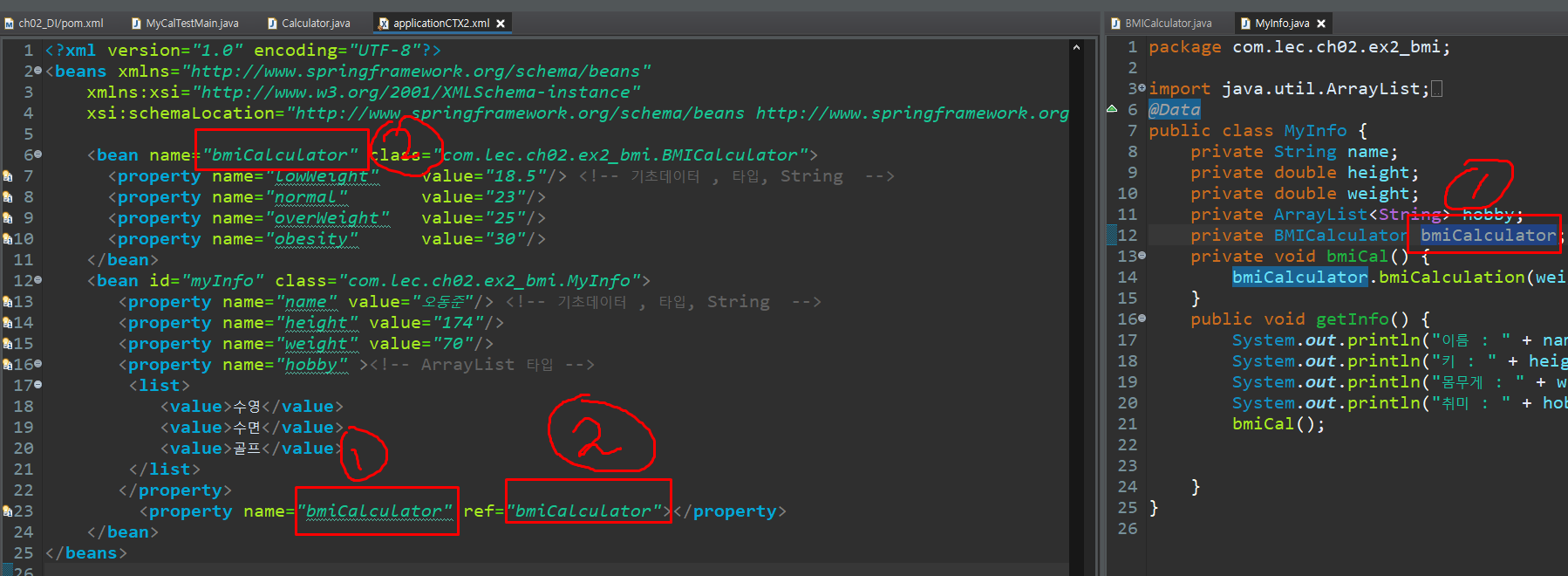


**(순서)**



**Ex02\_Bmi**





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

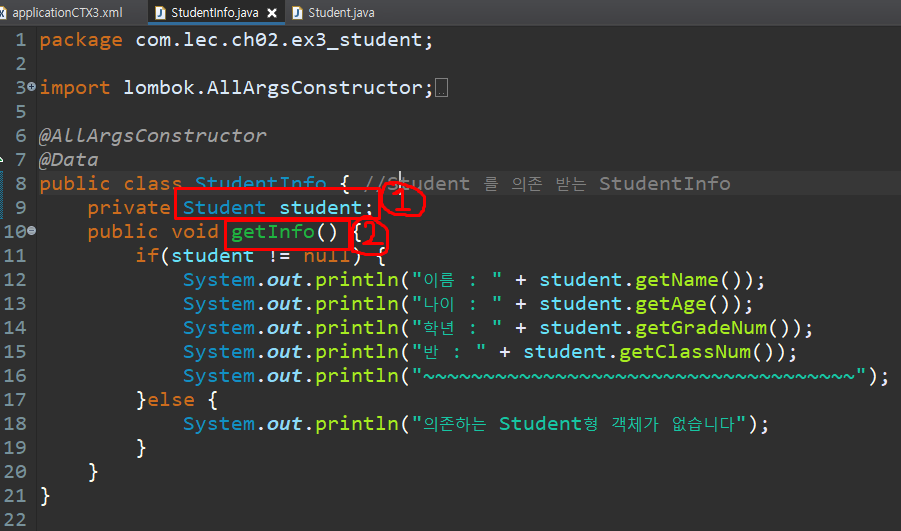
**2번 ref는 위에 bean name 과 이름을 같게 해줘야함**

**-Student-**

**1. Student.java**

**-변수 선언**

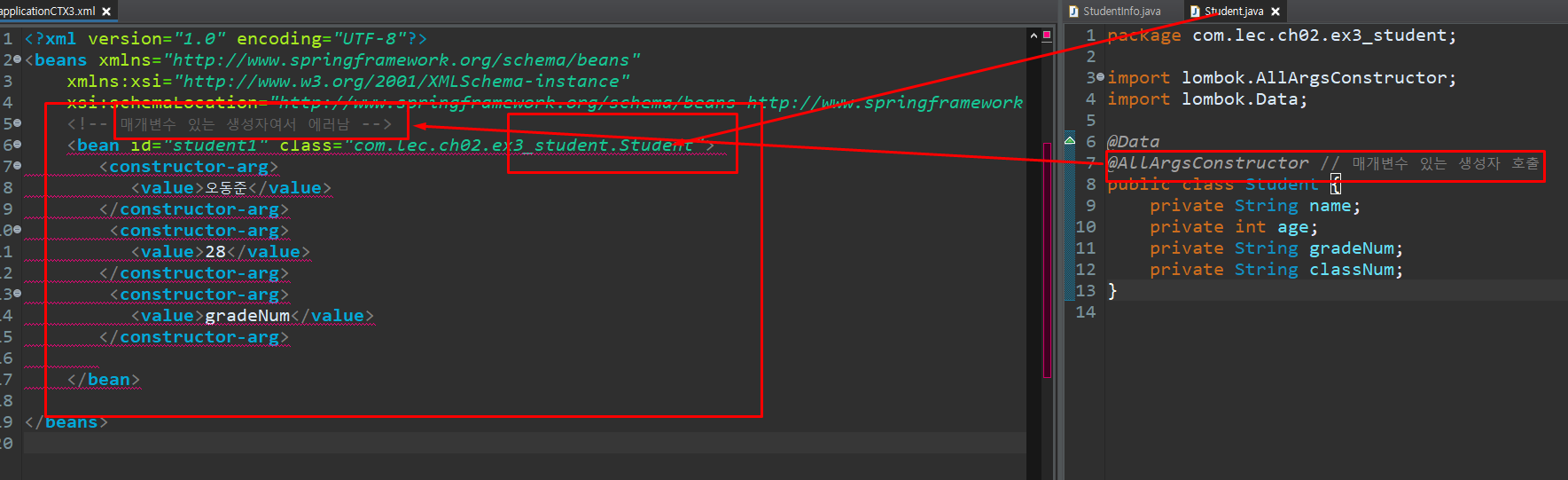
**2. StudentInfo.java 생성**



**1번 : Student.java에 의존 받음**

**2번 : getInfo() 생성자 함수 생성**

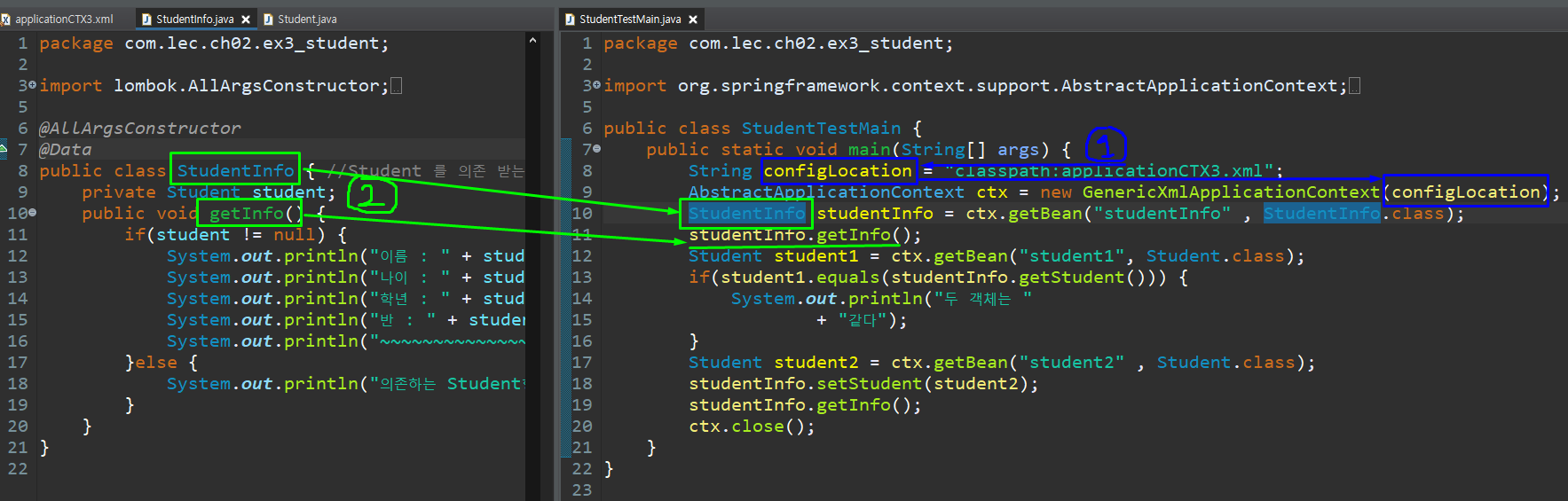
**3 . CTX3.xml 생성**



**AllArgsConstructor : 매개변수 있는 생성자여서**

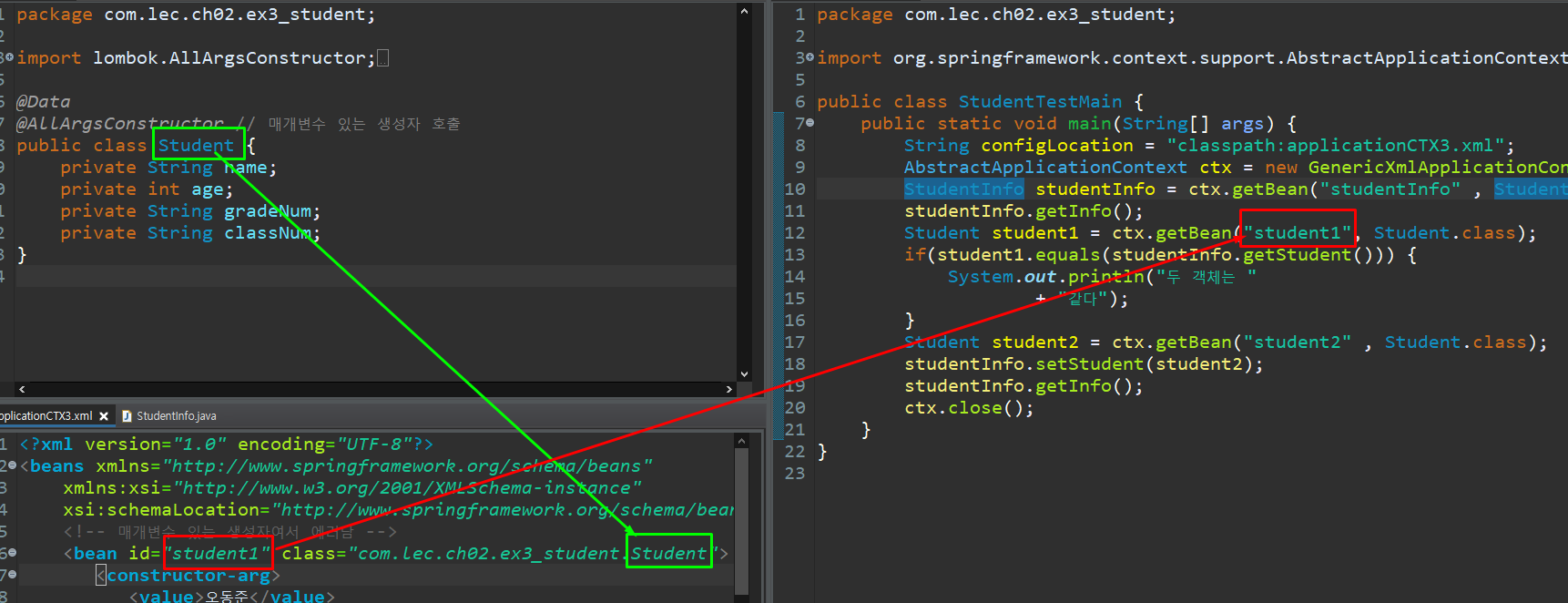
**Xml에 <value>값 안 넣으면 에러 발생**

1. **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**