

[02] DI(Dependency Injection) 개념 및 활용

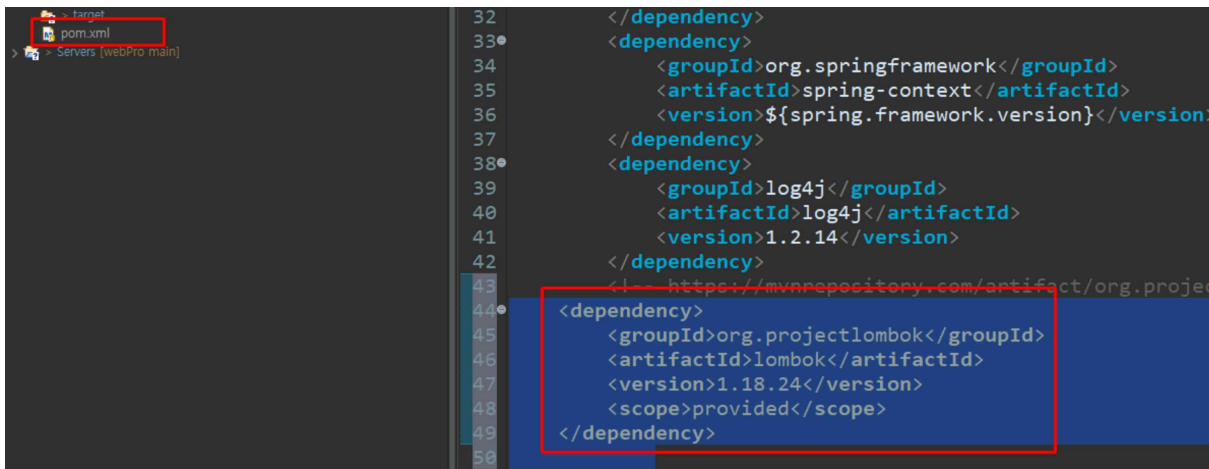
1. 스프링을 이용한 객체 생성과 조립(스프링은 객체 생성과 조립을 하는 컨테이너라 볼 수 있다)

✓ 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 x
1 package com.lec.ch02;
2
3 import lombok.Data;
4
5 @Data
6 public class MyCalculator {
7
8     private Calculator calculator;
9     private int num1;
10    private int num2;
11    public void add() { // 더하기
12        calculator.addition(num1, num2);
13    }
14    public void sub() { // 빼기
15        calculator.subtraction(num1, num2);
16    }
17    public void mul() { // 곱하기
18        calculator.multiplication(num1, num2);
19    }
20    public void div() { // 나누기
21        calculator.division(num1, num2);
22    }
23 }
24
```

Calculator.java 의 본 명는 MyCalcu
알수 있음

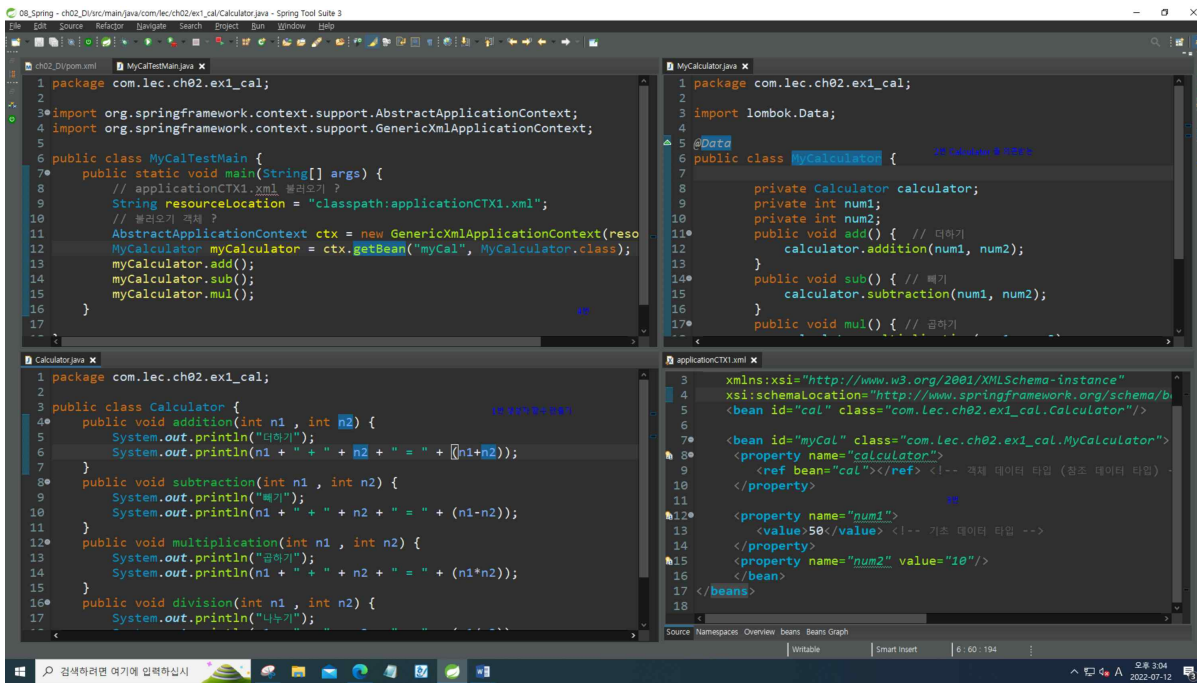
3. applicationCTX1 ...

```
ch02_Div pom.xml applicationCTX1.xml x
1 <?xml version="1.0" encoding="UTF-8"?>
2 <beans xmlns="http://www.springframework.org/schema/beans"
3       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4       xsi:schemaLocation="http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans.xsd">
5     <bean id="calculator" class="com.lec.ch02.ex1_cal.Calculator"/>
6
7     <bean id="myCalculator" class="com.lec.ch02.ex1_cal.MyCalculator"/>
8 </beans>
9
```

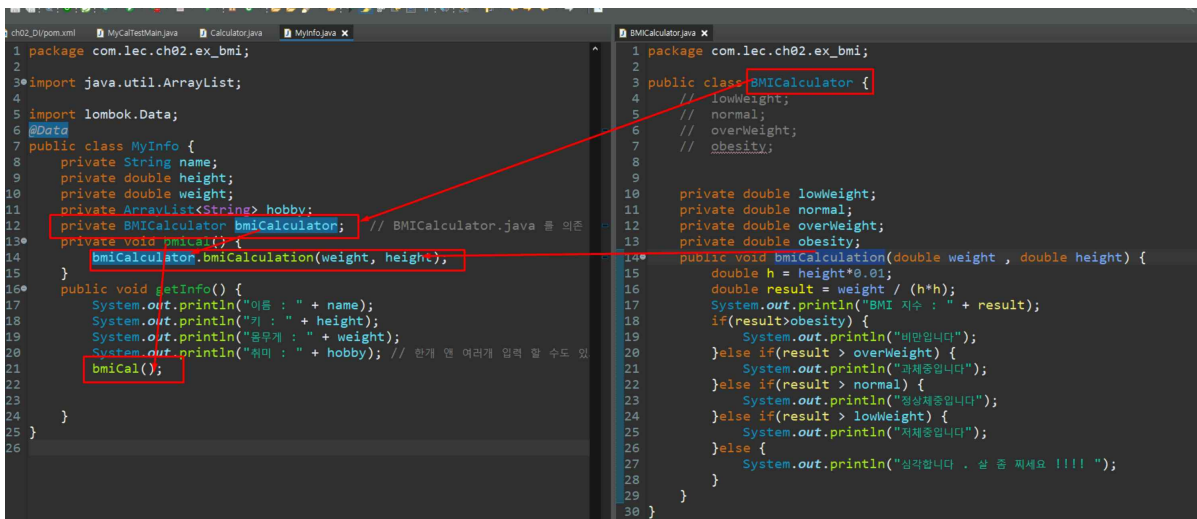
```
MyCalculator.java x
1 package com.lec.ch02.ex1_cal;
2
3 import lombok.Data;
4
5 @Data
6 public class MyCalculator {
7
8     private Calculator calculator;
9     private int num1;
10    private int num2;
11    public void add() { // 더하기
12        calculator.addition(num1, num2);
13    }
14    public void sub() { // 빼기
15        calculator.subtraction(num1, num2);
16    }
17    public void mul() { // 곱하기
18        calculator.multiplication(num1, num2);
19    }
20    public void div() { // 나누기
21        calculator.division(num1, num2);
22    }
23 }
```

```
Calculator.java x
1 public class Calculator {
2
3     public void addition(int n1, int n2) {
4         System.out.println("더하기");
5     }
6 }
```

(순서)



Ex02_Bmi



```
<?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
6
<bean name="bmiCalculator" class="com.lec.ch02.ex2_bmi.BMICalculator">
7
<property name="lowWeight" value="18.5"/> <!-- 기조데이터 , 타입, String -->
8
<property name="normal" value="23"/>
9
<property name="overWeight" value="25"/>
10
<property name="obesity" value="30"/>
11
</bean>
12
<bean id="myInfo" class="com.lec.ch02.ex2_bmi.MyInfo">
13
<property name="name" value="오승준"/> <!-- 기조데이터 , 타입, String -->
14
<property name="height" value="174"/>
15
<property name="weight" value="70"/>
16
<property name="hobby" ><!-- ArrayList 타입 -->
17
<list>
18
<value>수영</value>
19
<value>수면</value>
20
<value>골프</value>
21
</list>
22
</property>
23
<property name="bmiCalculator" ref="bmiCalculator"></property>
24
</bean>
25
</beans>
26

package com.lec.ch02.ex2_bmi;
import java.util.ArrayList;
6
@Data
7
public class MyInfo {
8
private String name;
9
private double height;
10
private double weight;
11
private ArrayList<String> hobby;
12
private BMICalculator bmiCalculator;
13
private void bmiCal() {
14
bmiCalculator.bmiCalculation(wei
15
}
16
public void getInfo() {
17
System.out.println("이름 : " + name);
18
System.out.println("키 : " + height);
19
System.out.println("몸무게 : " + weight);
20
System.out.println("취미 : " + hobby);
21
bmiCal();
22
}
23
}
24
}
25
}
26
}
```

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

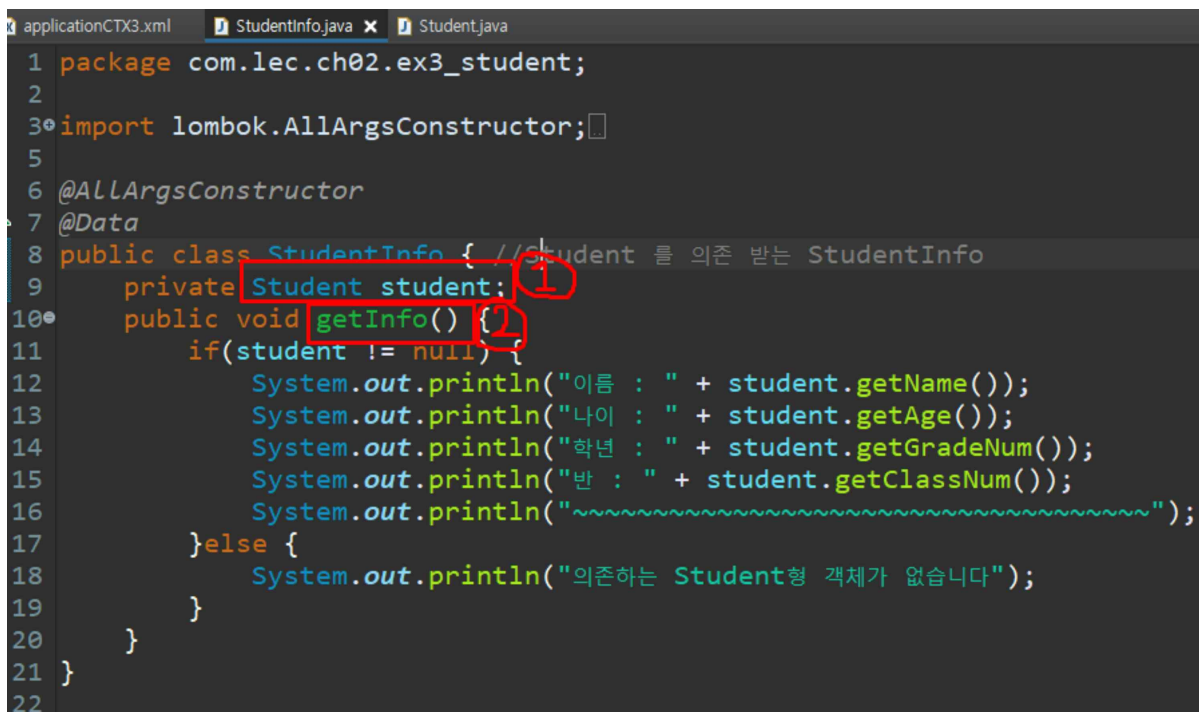
2번 ref는 위에 bean name 과 이름을 같게 해줘야함

-Student-

1. Student.java

-변수 선언

2. StudentInfo.java 생성



```
1 package com.lec.ch02.ex3_student;
2
3 import lombok.AllArgsConstructor;
4
5
6 @AllArgsConstructor
7 @Data
8 public class StudentInfo { //Student 를 의존 받는 StudentInfo
9     private Student student; 1
10    public void getInfo() {} 2
11    if(student != null) {
12        System.out.println("이름 : " + student.getName());
13        System.out.println("나이 : " + student.getAge());
14        System.out.println("학년 : " + student.getGradeNum());
15        System.out.println("반 : " + student.getClassNum());
16        System.out.println("~~~~~");
17    } else {
18        System.out.println("의존하는 Student형 객체가 없습니다");
19    }
20 }
21 }
22
```

1번 : Student.java에 의존 받음

2번 : getInfo() 생성자 함수 생성

3 . CTX3.xml 생성

```
applicationCTX3.xml
1 <?xml version="1.0" encoding="UTF-8"?>
2 <beans xmlns="http://www.springframework.org/schema/beans"
3       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4       xsi:schemaLocation="http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans.xsd">
5     <!-- 매개변수 있는 생성자여서 에러남 -->
6     <bean id="student1" class="com.lec.ch02.ex3.Student">
7       <constructor-arg>
8         <value>오동준</value>
9       </constructor-arg>
10      <constructor-arg>
11        <value>28</value>
12      </constructor-arg>
13      <constructor-arg>
14        <value>gradeNum</value>
15      </constructor-arg>
16    </bean>
17  </beans>
```

```
StudentInfo.java
1 package com.lec.ch02.ex3_student;
2
3 import lombok.AllArgsConstructor;
4 import lombok.Data;
5
6 @Data
7 @AllArgsConstructor // 매개변수 있는 생성자 호출
8 public class Student {
9     private String name;
10    private int age;
11    private String gradeNum;
12    private String classNum;
13 }
14
```

AllArgsConstructor : 매개변수 있는 생성자여서

Xml에 <value>값 안 넣으면 에러 발생

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

```
StudentInfo.java
1 package com.lec.ch02.ex3_student;
2
3 import lombok.AllArgsConstructor;
4
5 @AllArgsConstructor
6 @Data
7 public class StudentInfo { // Student 를 의존 받는
8     private Student student;
9     public void getInfo() {
10         if(student != null) {
11             System.out.println("이름 : " + student.getName());
12             System.out.println("나이 : " + student.getAge());
13             System.out.println("학년 : " + student.getGradeNum());
14             System.out.println("반 : " + student.getClassNum());
15             System.out.println("~~~~~");
16         } else {
17             System.out.println("의존하는 Student가 없습니다.");
18         }
19     }
20 }
21
```

```
StudentTestMain.java
1 package com.lec.ch02.ex3_student;
2
3 import org.springframework.context.support.AbstractApplicationContext;
4
5 public class StudentTestMain {
6     public static void main(String[] args) {
7         String configLocation = "classpath:applicationCTX3.xml";
8         AbstractApplicationContext ctx = new GenericXmlApplicationContext(configLocation);
9         StudentInfo studentInfo = ctx.getBean("studentInfo", StudentInfo.class);
10        studentInfo.getInfo();
11        Student student1 = ctx.getBean("student1", Student.class);
12        if(student1.equals(studentInfo.getStudent())) {
13            System.out.println("두 객체는 " + student1.getName() + "입니다.");
14        }
15        Student student2 = ctx.getBean("student2", Student.class);
16        studentInfo.setStudent(student2);
17        studentInfo.getInfo();
18        ctx.close();
19    }
20 }
21
```

1.

CTX3.xml 받아오기

2.StudentInfo 호출 , getInfo 생성자 함수 호출

밑에 이어서 ..


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

import lombok.AllArgsConstructor;

@Data
@AllArgsConstructor // 매개변수 있는 생성자 호출
public class Student {
    private String name;
    private int age;
    private String gradeNum;
    private String classNum;
}

<?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">
    <!-- 매개변수 있는 생성자에서 매개변수 -->
    <bean id="student1" class="com.lec.ch02.ex3_student.Student">
        <constructor-arg>
            <value>오동준</value>
        </constructor-arg>
    </bean>

```

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

import org.springframework.context.support.AbstractApplicationContext;

public class StudentTestMain {
    public static void main(String[] args) {
        String configLocation = "classpath:applicationCTX3.xml";
        AbstractApplicationContext ctx = new GenericXmlApplicationContext(configLocation);
        StudentInfo studentInfo = ctx.getBean("studentInfo", StudentInfo.class);
        studentInfo.getInfo();
        Student student1 = ctx.getBean("student1", Student.class);
        if(student1.equals(studentInfo.getStudent())) {
            System.out.println("두 객체는 " + "같다");
        }
        Student student2 = ctx.getBean("student2", Student.class);
        studentInfo.setStudent(student2);
        studentInfo.getInfo();
        ctx.close();
    }
}
```

1.Student 클래스 xml 에 id = student1 에 받아오고

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