

```
1 HOL : Spring DI
2 -----
3 Task 1. Non-DI Java Project
4 1. Project 유형 : Java Project
5 2. Project Name : BeforeSpring
6 3. Package Name : com.example
7
8
9 4. Calculator Class
10 com.example.Calculator.java
11 package com.example;
12
13 public class Calculator {
14     public void addAction(int a, int b){
15         System.out.println("Called addAction()");
16         System.out.printf("%d + %d = %d\n", a, b, (a + b));
17     }
18     public void subAction(int a, int b){
19         System.out.println("Called subAction()");
20         System.out.printf("%d - %d = %d\n", a, b, (a - b));
21     }
22     public void multiAction(int a, int b){
23         System.out.println("Called multiAction()");
24         System.out.printf("%d x %d = %d\n", a, b, (a * b));
25     }
26     public void divAction(int a, int b){
27         System.out.println("Called divAction()");
28         System.out.printf("%d / %d = %d\n", a, b, (a / b));
29     }
30 }
31
32
33 5. MyCalculator Class
34 com.example.MyCalculator.java
35 package com.example;
36
37 public class MyCalculator {
38     private Calculator calculator;
39     private int firstNum;
40     private int secondNum;
41
42     public void setFirstNum(int firstNum) {
43         this.firstNum = firstNum;
44     }
45     public void setSecondNum(int secondNum) {
46         this.secondNum = secondNum;
47     }
48     public void setCalculator(Calculator calculator){
49         this.calculator = calculator;
50     }
51
52     public void add(){
53         this.calculator.addAction(firstNum, secondNum);
54     }
55     public void sub(){
56         this.calculator.subAction(firstNum, secondNum);
57     }
58     public void multi(){
```

```
59     this.calculator.multiAction(firstNum, secondNum);
60 }
61 public void div(){
62     this.calculator.divAction(firstNum, secondNum);
63 }
64 }
65
66
67 6. MainClass Class
68 com.example.MainClass
69 package com.example;
70
71 public class MainClass {
72     public static void main(String[] args) {
73         MyCalculator myCalculator = new MyCalculator();
74         myCalculator.setCalculator(new Calculator());
75
76         myCalculator.setFirstNum(10);
77         myCalculator.setSecondNum(2);
78
79         myCalculator.add();
80         myCalculator.sub();
81         myCalculator.multi();
82         myCalculator.div();
83     }
84 }
85
86
87 7. Result
88 Called addAction()
89 10 + 2 = 12
90 Called subAction()
91 10 - 2 = 8
92 Called multiAction()
93 10 x 2 = 20
94 Called divAction()
95 10 / 2 = 5
96
97
98
99 -----
100 Task 2. DI Demo in Spring
101 1. New > Java Project
102     1)Project Name : StartSpring
103     2)JRE : Use default JRE 'jdk-13.0.2' and workspace compiler preferences
104     3)Next
105     4)Uncheck [Create module-info.java file]
106     5)Finish
107
108
109 2. Create package to src : com.example
110
111 3. Copy MyCalculator.java, Calculator.java from BeforeSpring project to StartSpring's package
112
113 4. Create class : com.example.MainClass.java
114     package com.example;
115
116     public class MainClass {
```

```
117     public static void main(String[] args) {
118
119     }
120 }
121
122
123 5. Java Project를 Spring Project로 변환
124 1)StartSpring Project > right-click > Configure > Convert to Maven Project
125     -Project : /StartSpring
126     -Group Id : StartSpring
127     -Artifact Id : StartSpring
128     -version : 0.0.1-SNAPSHOT
129     -Packaging : jar
130     -Finish
131     -Package Explorer에서 보이는 Project icon에 Maven의 'M'자가 보임.
132
133 2)StartSpring Project > right-click > Spring > Add Spring Project Nature
134     -Package Explorer에서 보이는 Project icon에 'M'자와 Spring의 'S'가 보임.
135
136 3)pom.xml file에 Spring Context Dependency 추가하기
137     -https://mvnrepository.com에서 'spring context'로 검색
138     -[Spring Context] click
139     -현재 Spring 5.x의 현재 version인 5.2.5.RELEASE click
140     -Copy하여 pom.xml에 paste
141
142     <version>0.0.1-SNAPSHOT</version>
143     <dependencies> <--- dependencies element 추가
144         <dependency> <---여기에 paste
145             <groupId>org.springframework</groupId>
146             <artifactId>spring-context</artifactId>
147             <version>5.2.5.RELEASE</version>
148         </dependency>
149     </dependencies>
150
151 4)pom.xml Save
152
153 5)pom.xml > right-click > Run As > Maven install
154     [INFO] BUILD SUCCESS 확인
155
156
157 6. config folder 생성
158 1)StartSpring project > right-click > New > Source Folder
159     -Folder name : config
160     -Finish
161
162
163 7. Bean Configuration XML 작성
164 1)config > right-click > New > Other > Spring > Spring Bean Configuration File > Next
165 2)Name : applicationContext.xml > Finish
166     <?xml version="1.0" encoding="UTF-8"?>
167     <beans xmlns="http://www.springframework.org/schema/beans"
168         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
169         xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans.xsd">
170
171         <bean id="calculator" class="com.example.Calculator" />
172
173         <bean id="myCalculator" class="com.example.MyCalculator">
```

```
174     <property name="calculator">
175         <ref bean="calculator" />
176     </property>
177     <property name="firstNum" value="10" />
178     <property name="secondNum" value="2" />
179 </bean>
180 </beans>
```

181

182

183 8. MainClass.java

184 package com.javasoft;

185

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

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

188

189 public class MainClass {

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

191 String configFile = "classpath:applicationContext.xml";

192 AbstractApplicationContext ctx = new GenericXmlApplicationContext(configFile);

193 MyCalculator myCalculator = ctx.getBean("myCalculator", MyCalculator.class);

194

195 myCalculator.add();

196 myCalculator.sub();

197 myCalculator.multi();

198 myCalculator.div();

199

200 ctx.close();

201 }

202 }

203

204

205 9. Result

206 BeforeSpring과 같음.

207 Called addAction()

208 10 + 2 = 12

209 Called subAction()

210 10 - 2 = 8

211 Called multiAction()

212 10 x 2 = 20

213 Called divAction()

214 10 / 2 = 5

215

216

217

218 -----

219 Task 3. 간단한 DI Project

220 1. In Package Explorer &gt; right-click &gt; New &gt; Java Project

221 1)Project name : DIDemo

222 2)JRE : Use default JRE 'jdk-13.0.2' and workspace compiler preferences

223 3)Next

224 4)Uncheck [Create module-info.java file]

225 5)Finish

226

227

228 2. src &gt; right-click &gt; New &gt; Package

229 1)Package name : com.example

230 2)Finish

231

```
232
233 3. Interface 작성
234 1)com.example > right-click > New > Interface
235 2)Name : Printer
236
237 3)Printer.java
238 package com.example;
239
240 public interface Printer{
241     void print(String message);
242 }
243
244
245 4. POJO class 작성
246 1)com.example > right-click > New > Class
247 2)Name : Hello
248 3)Finish
249 4)Hello.java
250 package com.example;
251
252 public class Hello{
253     private String name;
254     private Printer printer;
255
256     public Hello(){ }
257
258     public void setName(String name){
259         this.name = name;
260     }
261
262     public void setPrinter(Printer printer){
263         this.printer = printer;
264     }
265
266     public String sayHello(){
267         return "Hello " + name;
268     }
269
270     public void print(){
271         this.printer.print(sayHello());
272     }
273 }
274
275
276 5. Printer interface의 child class 작성하기
277 1)com.example > right-click > New > Class
278 -Name : StringPrinter
279 -Interfaces : com.example.Printer
280 -Finish
281
282 2)StringPrinter.java
283 package com.example;
284
285 public class StringPrinter implements Printer{
286     private StringBuffer buffer = new StringBuffer();
287
288     @Override
289     public void print(String message){
```

```
290     this.buffer.append(message);
291 }
292
293     public String toString(){
294         return this.buffer.toString();
295     }
296 }
```

297

298 3)com.example > right-click > New > Class

299 -Name : ConsolePrinter  
300 -Interface : com.example.Printer  
301 -Finish

302

303 4)ConsolePrinter.java

```
304     package com.example;
305
306     public class ConsolePrinter implements Printer{
307
308         @Override
309         public void print(String message){
310             System.out.println(message);
311         }
312     }
```

313

314

315 6. Java Project를 Spring Project로 변환

316 1)DIDemo Project > right-click > Configure > Convert to Maven Project

317 -Project : /DIDemo  
318 -Group Id : DIDemo  
319 -Artifact Id : DIDemo  
320 -version : 0.0.1-SNAPSHOT  
321 -Packaging : jar  
322 -Finish

323

324 2)DIDemo Project > right-click > Spring > Add Spring Project Nature

325

326 3)pom.xml file에 Spring Context Dependency 추가하기

```
327     <version>0.0.1-SNAPSHOT</version>
328     <dependencies>
329         <dependency>
330             <groupId>org.springframework</groupId>
331             <artifactId>spring-context</artifactId>
332             <version>5.2.5.RELEASE</version>
333         </dependency>
334     </dependencies>
```

335

336 4)pom.xml > right-click > Run As > Maven install

337 [INFO] BUILD SUCCESS 확인

338

339

340 7. config folder 생성

341 1)StartSpring project > right-click > New > Source Folder

342 -Folder name : config  
343 -Finish

344

345

346 8. Bean Configuration XML 작성

347 1)config > right-click > New > Other > Spring > Spring Bean Configuration File > Next

```
348 2)File name : beans.xml
349 3)Finish
350
351 <?xml version="1.0" encoding="UTF-8"?>
352 <beans xmlns="http://www.springframework.org/schema/beans"
353       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
354       xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans.xsd">
355
356     <bean id="hello" class="com.example.Hello">
357         <property name="name" value="Spring" />
358         <property name="printer" ref="printer" />
359     </bean>
360     <bean id="printer" class="com.example.StringPrinter" />
361     <bean id="consolePrinter" class="com.example.ConsolePrinter" />
362
363 </beans>
364
365
366 9. Beans Graph 사용하기
367 1)Window menu > Show View > Other > Spring > Spring Explorer > Open
368 2)Spring Explorer
369 -DIDemo > Beans > beans.xml > right-click > Open Beans Graph
370
371
372 10. DI Test class 작성
373 1)src/com.example > right-click > New > Package
374 -Name : com.example.test
375 -Finish
376
377 2)/src/com.example.test > New > Class
378 -Name : HelloBeanTest.java
379
380 package com.example.test;
381
382 import org.springframework.context.ApplicationContext;
383 import org.springframework.context.support.GenericXmlApplicationContext;
384
385 import com.example.Hello;
386 import com.example.Printer;
387
388 public class HelloBeanTest {
389     public static void main(String [] args){
390         //1. IoC Container 생성
391         ApplicationContext context =
392             new GenericXmlApplicationContext("classpath:beans.xml");
393
394         //2. Hello Beans 가져오기
395         Hello hello = (Hello)context.getBean("hello");
396         System.out.println(hello.sayHello());
397         hello.print();
398
399         //3. SpringPrinter 가져오기
400         Printer printer = (Printer)context.getBean("printer");
401         System.out.println(printer.toString());
402
403         Hello hello2 = context.getBean("hello", Hello.class);
404         hello2.print();
```

```

405
406     System.out.println(hello == hello2); //Singleton Pattern
407 }
408 }
409
410
411 11. Result
412     Hello Spring
413     Hello Spring
414     true
415
416
417
418 -----
419 Task 4. JUnit을 사용한 DI test class 작성하기
420 1. JUnit을 사용한 DI test class(HelloBeanJUnitTest.java) 작성
421     1)pom.xml에 아래 코드 붙여넣기
422         <dependency>
423             <groupId>junit</groupId>
424             <artifactId>junit</artifactId>
425             <version>4.12</version>
426             <scope>test</scope>
427         </dependency>
428
429     2)pom.xml > right-click > Run As > Maven install
430         [INFO] BUILD SUCCESS 확인
431
432     3)src/com.example.test > New > Class
433         -Name : HelloBeanJUnitTest.java
434
435         package com.example.test;
436
437         import org.junit.Before;
438         import org.junit.Test;
439         import org.springframework.context.ApplicationContext;
440         import org.springframework.context.support.GenericXmlApplicationContext;
441
442         import com.example.Hello;
443         import com.example.Printer;
444
445         import static org.junit.Assert.assertEquals;
446         import static org.junit.Assert.assertSame;
447
448         public class HelloBeanJUnitTest {
449             ApplicationContext context;
450
451             @Before
452             public void init(){
453                 //항상 먼저 ApplicationContext를 생성해야 하기 때문에
454                 //1. IoC Container 생성
455                 context = new GenericXmlApplicationContext("classpath:beans.xml");
456             }
457
458             @Test
459             public void test1(){
460                 //2. Hello Beans 가져오기
461                 Hello hello = (Hello)context.getBean("hello");
462                 assertEquals("Hello Spring", hello.sayHello());

```



```

463         hello.print();
464
465         //3. SpringPrinter 가져오기
466         Printer printer = (Printer)context.getBean("printer");
467         assertEquals("Hello Spring", printer.toString());
468     }
469
470     @Test
471     public void test2(){
472         Hello hello = (Hello)context.getBean("hello");
473
474         Hello hello2 = context.getBean("hello", Hello.class);
475         assertEquals(hello, hello2);
476     }
477 }
478
479
480 2. @Before에 mouse를 올려놓으면 Fix project setup... click
481 1)Add archive 'junit-4.12.jar ... > OK
482 -import org.junit...에 mouse를 올려놓으면 Fix project setup... click
483 -Add JUnit 4 library to the build path > OK
484
485
486 3. right-click > Run As > JUnit Test
487 1)결과 -> JUnit View에 초록색 bar
488 2)만일, test1() method를 junit에서 제외하고 싶을 때에는 @Test 옆에 @Ignore를 선언한다.
489
490     import org.junit.Ignore;
491     ...
492     @Test @Ignore
493     public void test1(){
494     ...
495
496 3)right-click > Run As > JUnit Test
497 -JUnit Test 목록에서 test1()는 실행되지 않는다.
498
499
500
501 -----
502 Task 5. Spring TestContext Framework
503 1. Spring-Test library 설치
504 1)http://mvnrepository.com에서 'spring test'로 검색
505 2)검색 결과 목록에서 'Spring TestContext Framework' Click
506 3)version 목록에서 5.2.5.RELEASE Click
507
508
509 2. dependency 복사해서 pom.xml에 붙여넣기
510 <!-- https://mvnrepository.com/artifact/org.springframework/spring-test -->
511 <dependency>
512     <groupId>org.springframework</groupId>
513     <artifactId>spring-test</artifactId>
514     <version>5.2.5.RELEASE</version>
515     <scope>test</scope>
516 </dependency>
517
518
519 3. pom.xml > right-click > Maven Install
520 [INFO] BUILD SUCCESS

```

```
521
522
523 4. Spring-Test를 사용할 DI test class-HelloBeanJUnitSpringTest.java 작성하기
524 1)src/com.example.test > New > Class
525 2)Name : HelloBeanJUnitSpringTest
526 package com.example.test;
527
528 import static org.junit.Assert.assertEquals;
529 import static org.junit.Assert.assertSame;
530
531 import org.junit.Test;
532 import org.junit.runner.RunWith;
533 import org.springframework.beans.factory.annotation.Autowired;
534 import org.springframework.context.ApplicationContext;
535 import org.springframework.test.context.ContextConfiguration;
536 import org.springframework.test.context.junit4.SpringJUnit4ClassRunner;
537
538 import com.example.Hello;
539 import com.example.Printer;
540
541 @RunWith(SpringJUnit4ClassRunner.class)
542 //JUnit 4.x에서 사용
543 @ContextConfiguration(locations="classpath:beans.xml")
544 public class HelloBeanJUnitSpringTest {
545     @Autowired
546     ApplicationContext context;
547
548     @Test
549     public void test1(){
550         Hello hello = (Hello)context.getBean("hello");
551         assertEquals("Hello Spring", hello.sayHello());
552         hello.print();
553
554         Printer printer = (Printer)context.getBean("printer");
555         assertEquals("Hello Spring", printer.toString());
556     }
557
558     @Test
559     public void test2(){
560         Hello hello = (Hello)context.getBean("hello");
561
562         Hello hello2 = context.getBean("hello", Hello.class);
563         assertSame(hello, hello2);
564     }
565 }
566
567 -right-click > Run As > JUnit Test
568 -결과 -> JUnit View에 초록색 bar
569
570 4)만일 해당 객체를 찾을 수 없다는 오류가 계속 발생하면
571 -해당 Project > right-click > Build Path > Configure Build Path > Libraries tab
572 -spring-test-5.2.5.RELEASE.jar 선택 후 [Remove] 로 삭제
573 -Classpath 선택
574 -[Add External JARs...] Click
575 -Local M2 Repository(e.g
576 C:\Users\bluee\.m2\repository\org\springframework\spring-test\5.2.5.RELEASE)에서 직접
577 jar(spring-test-5.2.5.RELEASE.jar)를 선택할 것
578
579 -[Order and Export] tab에서 spring-test-5.2.5.RELEASE.jar 선택 후 [Up] button을 클릭
```

```
577 -해당 DIDemo/src 바로 아래까지 올리고 [Apply and Close] Click
578
579
580
581 -----
582 Task 6. Java Annotation을 이용하여 setter를 이용한 의존주입하기 실습
583 1. In Package Explorer > right-click > New > Java Project
584 1)Project name : DIDemo1
585 2)JRE
586 -Select [Use default JRE 'jdk-13.0.2' and workspace compiler preferences]
587 3)Next
588 4)Uncheck [Create module-info.java file]
589 5)Finish
590
591
592 2. src > right-click > New > Package
593 1)Package name : com.example
594 2)Finish
595
596
597 3. POJO class 작성
598 1)com.example > right-click > New > Class
599 2)Name : Hello
600
601 package com.example;
602
603 public class Hello {
604     private String name;
605     private Printer printer;
606
607     public Hello(){ }
608
609     public void setName(String name){
610         this.name = name;
611     }
612
613     public void setPrinter(Printer printer){
614         this.printer = printer;
615     }
616
617     public String sayHello(){
618         return "Hello " + name;
619     }
620
621     public void print(){
622         this.printer.print(sayHello());
623     }
624 }
625
626 3)com.example > right-click > New > Interface
627 4)Name : Printer
628
629 package com.example;
630
631 public interface Printer{
632     void print(String message);
633 }
634
```

```
635 5)com.example > right-click > New > Class
636 6)Name : StringPrinter
637 7)Interfaces : com.example.Printer
638
639 package com.example;
640
641 public class StringPrinter implements Printer{
642     private StringBuffer buffer = new StringBuffer();
643
644     @Override
645     public void print(String message){
646         this.buffer.append(message);
647     }
648
649     public String toString(){
650         return this.buffer.toString();
651     }
652 }
653
654 8)com.example > right-click > New > Class
655 9)Name : ConsolePrinter
656 10)Interfaces : com.example.Printer
657
658 package com.example;
659
660 public class ConsolePrinter implements Printer{
661
662     @Override
663     public void print(String message){
664         System.out.println(message);
665     }
666 }
667
668
669 4. Java Project를 Spring Project로 변환
670 1)DIDemo1 Project > right-click > Configure > Convert to Maven Project
671     -Project : /DIDemo1
672     -Group Id : DIDemo1
673     -Artifact Id : DIDemo1
674     -version : 0.0.1-SNAPSHOT
675     -Packaging : jar
676     -Finish
677
678 2)DIDemo1 Project > right-click > Spring > Add Spring Project Nature
679
680 3)pom.xml file에 Spring Context Dependency 추가하기
681     <version>0.0.1-SNAPSHOT</version>
682     <dependencies>
683         <dependency>
684             <groupId>org.springframework</groupId>
685             <artifactId>spring-context</artifactId>
686             <version>5.2.5.RELEASE</version>
687         </dependency>
688     </dependencies>
689
690 4)pom.xml > right-click > Run As > Maven install
691     [INFO] BUILD SUCCESS
692
```

```
693
694 5. config package 생성
695   1)com.example > right-click > New > Package > com.example.config
696   2)Finish
697
698
699 6. AppCtx Class 생성
700   1)com.example.config > right-click > New > Class
701   2)Name : AppCtx.java
702
703   package com.example.config;
704
705   import org.springframework.context.annotation.Bean;
706   import org.springframework.context.annotation.Configuration;
707
708   import com.example.ConsolePrinter;
709   import com.example.Hello;
710   import com.example.StringPrinter;
711
712   @Configuration
713   public class AppCtx {
714
715       @Bean
716       public Hello hello() {
717           Hello hello = new Hello();
718           hello.setName("Spring");
719           hello.setPrinter(this.printer());
720           return hello;
721       }
722
723       @Bean
724       public StringPrinter printer() {
725           return new StringPrinter();
726       }
727
728       @Bean
729       public ConsolePrinter consolePrinter() {
730           return new ConsolePrinter();
731       }
732   }
733
734
735 7. DI Test class 작성
736   1)src > right-click > New > Package
737   2)Package Name : com.example.test
738   3)Finish
739   4)com.example.test > right-click > New > Class
740   5)Name : HelloBeanTest
741
742   package com.example.test;
743
744   import org.springframework.context.ApplicationContext;
745   import org.springframework.context.annotation.AnnotationConfigApplicationContext;
746
747   import com.example.Hello;
748   import com.example.Printer;
749   import com.example.config.AppCtx;
750
```

```

751     public class HelloBeanTest {
752         public static void main(String[] args) {
753             // 1. IoC Container 생성
754             ApplicationContext ctx = new AnnotationConfigApplicationContext(AppCtx.class);
755
756             // 2. Hello Beans 가져오기
757             Hello hello = (Hello)ctx.getBean("hello");
758             System.out.println(hello.sayHello());
759             hello.print();
760
761             // 3. SpringPrinter 가져오기
762             Printer printer = (Printer) ctx.getBean("printer");
763             System.out.println(printer.toString());
764             Hello hello2 = ctx.getBean("hello", Hello.class);
765             hello2.print();
766             System.out.println(hello == hello2);
767         }
768     }
769
770

```

### 771 8. Test

```

772     1)/src/com.example.test/HelloBeanTest.java > right-click > Run As > Java Application
773     Hello Spring
774     Hello Spring
775     true
776
777
778
779

```

### 780 Task 7. setter를 이용한 의존주입하기 실습

```

781     1. In Package Explorer > right-click > New > Java Project

```

```

782     1)Project Name : SpringDemo
783     2)JRE
784     -Select [Use default JRE 'jdk-13.0.2' and workspace compiler preferences]
785     3)Next
786     4)Uncheck [Create module-info.java file]
787     5)Finish
788
789

```

```

790     2. src > right-click > New > Package

```

```

791     1)Package name : com.example
792
793

```

### 794 3. POJO class 작성

```

795     1)com.example > right-click > New > Class

```

```

796     2)Class Name : BmiCalculator
797

```

```

798     package com.example;
799
800     public class BmiCalculator {
801         private double lowWeight;
802         private double normal;
803         private double overWeight;
804         private double obesity;
805
806         public void setLowWeight(double lowWeight) {
807             this.lowWeight = lowWeight;
808         }

```

```
809
810     public void setNormal(double normal) {
811         this.normal = normal;
812     }
813
814     public void setOverWeight(double overWeight) {
815         this.overWeight = overWeight;
816     }
817
818     public void setObesity(double obesity) {
819         this.obesity = obesity;
820     }
821     public void bmiCalcu(double weight, double height){
822         double h = height * 0.01;
823         double result = weight / (h * h);
824
825         System.out.println("BMI 지수 : " + (int)result);
826
827         if(result > obesity)
828             System.out.println("비만입니다.");
829         else if(result > overWeight)
830             System.out.println("과체중입니다.");
831         else if(result > normal)
832             System.out.println("정상입니다.");
833         else
834             System.out.println("저체중입니다.");
835     }
836 }
837
838 3)com.example > right-click > New > Class
839 4)Class Name : MyInfo.java
840
841     package com.example;
842
843     import java.util.ArrayList;
844
845     public class MyInfo {
846         private String name;
847         private double height;
848         private double weight;
849         private ArrayList<String> hobby;
850         private BmiCalculator bmiCalculator;
851
852         public void setBmiCalculator(BmiCalculator bmiCalculator) {
853             this.bmiCalculator = bmiCalculator;
854         }
855         public void setName(String name) {
856             this.name = name;
857         }
858         public void setHeight(double height) {
859             this.height = height;
860         }
861         public void setWeight(double weight) {
862             this.weight = weight;
863         }
864         public void setHobby(ArrayList<String> hobby) {
865             this.hobby = hobby;
866         }
867     }
```

```
867     public void getInfo(){
868         System.out.println("Name : " + this.name);
869         System.out.println("Height : " + this.height);
870         System.out.println("Weight : " + this.weight);
871         System.out.println("Hobby : " + this.hobby);
872         this.bmiCalcu();
873     }
874     public void bmiCalcu(){
875         this.bmiCalculator.bmiCalcu(this.weight, this.height);
876     }
877 }
```

878  
879

#### 880 4. Java Project를 Spring Project로 변환

```
881 1)SpringDemo Project > right-click > Configue > Convert to Maven Project
882   -Project : /SpringDemo
883   -Group Id : SpringDemo
884   -Artifact Id : SpringDemo
885   -version : 0.0.1-SNAPSHOT
886   -Packaging : jar
887   -Finish
```

888  
889

```
889 2)SpringDemo Project > right-click > Spring > Add Spring Project Nature
```

890  
891

```
891 3)pom.xml file에 Spring Context Dependency 추가하기
```

```
892     <version>0.0.1-SNAPSHOT</version>
893     <dependencies>
894         <dependency>
895             <groupId>org.springframework</groupId>
896             <artifactId>spring-context</artifactId>
897             <version>5.2.5.RELEASE</version>
898         </dependency>
899     </dependencies>
```

900  
901

```
901 4)pom.xml > right-click > Run As > Maven install
```

```
902     [INFO] BUILD SUCCESS 확인
```

903  
904

#### 905 5. SpringDemo/resources folder 생성

```
906 1)SpringDemo project > right-click > Build Path > Configure Build Path
907 2)Source Tab > Add Folder
908 3)SpringDemo 선택 확인
909 4)Create New Folder > Folder name : resources > Finish > OK
910 5)SpringDemo/resources(new) 확인
911 6)Apply and Close
```

912  
913

#### 914 6. Bean Configuration XML 작성

```
915 1)SpringDemo/resources > right-click > New > Other > Spring > Spring Bean Configuration
    File
916 2)File name : applicationContext.xml
917 3)Finish
```

918  
919

```
919     <bean id="bmiCalculator" class="com.example.BmiCalculator">
920         <property name="lowWeight" value="18.5" />
921         <property name="normal" value="23" />
922         <property name="overWeight" value="25" />
923         <property name="obesity">
```



```

924         <value>30</value>
925     </property>
926 </bean>
927 <bean id="myInfo" class="com.example.MyInfo">
928     <property name="name" value="한지민" />
929     <property name="height" value="170.5" />
930     <property name="weight" value="67" />
931     <property name="hobby">
932         <list>
933             <value>수영</value>
934             <value>요리</value>
935             <value>독서</value>
936         </list>
937     </property>
938     <property name="bmiCalculator">
939         <ref bean="bmiCalculator" />
940     </property>
941 </bean>
942
943
944 7. MainClass 생성하기
945 1)com.example.MainClass.java
946 package com.example;
947
948 import org.springframework.context.AbstractApplicationContext;
949 import org.springframework.context.support.GenericXmlApplicationContext;
950
951 public class MainClass {
952     public static void main(String[] args) {
953         String configFile = "classpath:applicationContext.xml";
954
955         //Spring Container 생성
956         AbstractApplicationContext context = new GenericXmlApplicationContext(configFile);
957
958         //Spring Container 에서 객체를 가져옴
959         MyInfo myInfo = context.getBean("myInfo", MyInfo.class);
960
961         myInfo.getInfo();
962         context.close();
963     }
964 }
965
966
967 8. Java Application 실행
968 Name : 한지민
969 Height : 170.5
970 Weight : 67.0
971 Hobby : [수영, 요리, 독서]
972 BMI 지수 : 23
973 정상입니다.
974
975
976
977 -----
978 Task 8. 생성자 이용하여 의존 주입하기 실습
979 1. In Package Explorer > right-click > New > Java Project
980 1)Project name : DIDemo2
981 2)JRE

```

```
982     -Select [Use default JRE 'jdk-13.0.2' and workspace compiler preferences]
983     3)Next
984     4)Uncheck [Create module-info.java file]
985     5)Finish
986
987
988 2. src > right-click > New > Package
989     1)Package name : com.example
990     2)Finish
991
992
993 3. POJO class 작성
994     1)com.example > right-click > New > Class
995     2)Class Name : Hello
996         package com.example;
997
998         public class Hello{
999             private String name;
1000             private Printer printer;
1001
1002             public Hello(){ }
1003
1004             public void setName(String name){
1005                 this.name = name;
1006             }
1007
1008             public void setPrinter(Printer printer){
1009                 this.printer = printer;
1010             }
1011
1012             public String sayHello(){
1013                 return "Hello " + name;
1014             }
1015
1016             public void print(){
1017                 this.printer.print(sayHello());
1018             }
1019         }
1020
1021 3)com.example > right-click > New > Interface
1022 4)interface name : Printer
1023
1024     package com.example;
1025
1026     public interface Printer{
1027         void print(String message);
1028     }
1029
1030 5)com.example > right-click > New > Class
1031 6)Class Name : StringPrinter
1032 7)Interfaces : com.example.Printer
1033
1034     package com.example;
1035
1036     public class StringPrinter implements Printer{
1037         private StringBuffer buffer = new StringBuffer();
1038
1039         @Override
```

```
1040     public void print(String message){
1041         this.buffer.append(message);
1042     }
1043
1044     public String toString(){
1045         return this.buffer.toString();
1046     }
1047 }
```

1048 8)com.example > right-click > New > Class

1049 9)Class Name : ConsolePrinter

1050 10)Interfaces : com.example.Printer

```
1051
1052
1053     package com.example;
1054
1055     public class ConsolePrinter implements Printer{
1056
1057         @Override
1058         public void print(String message){
1059             System.out.println(message);
1060         }
1061     }
```

1062
1063 4. Java Project를 Spring Project로 변환

1064 1)DIDemo2 Project > right-click > Configure > Convert to Maven Project

1065 -Project : /DIDemo2

1066 -Group Id : DIDemo2

1067 -Artifact Id : DIDemo2

1068 -version : 0.0.1-SNAPSHOT

1069 -Packaging : jar

1070 -Finish

1071
1072 2)DIDemo2 Project > right-click > Spring > Add Spring Project Nature

1073 3)pom.xml file에 Spring Context Dependency 추가하기

1074 <version>0.0.1-SNAPSHOT</version>

1075 <dependencies>

1076 <dependency>

1077 <groupId>org.springframework</groupId>

1078 <artifactId>spring-context</artifactId>

1079 <version>5.2.5.RELEASE</version>

1080 </dependency>

1081 </dependencies>

1082
1083 4)pom.xml > right-click > Run As > Maven install

1084 [INFO] BUILD SUCCESS 확인

1085

1086

1087 5. DIDemo2/resources folder 생성

1088 1)DIDemo2 project > right-click > Build Path > Configure Build Path

1089 2)Source Tab > Add Folder

1090 3)DIDemo2 선택확인

1091 4>Create New Folder > Folder name : resources > Finish > OK

1092 5)DIDemo2/resources(new) 확인

1093 6)Apply and Close

1094

1095

1096

1097

## 1098 6. Bean Configuration XML 작성

1099 1)DIDemo2/resources &gt; right-click &gt; New &gt; Other &gt; Spring &gt; Spring Bean Configuration File

1100 -File name : beans.xml &gt; Finish

1101

1102 &lt;?xml version="1.0" encoding="UTF-8"?&gt;

1103 &lt;beans xmlns="http://www.springframework.org/schema/beans"

1104 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

1105 xsi:schemaLocation="http://www.springframework.org/schema/beans  
http://www.springframework.org/schema/beans/spring-beans.xsd">

1106

1107 &lt;bean id="hello" class="com.example.Hello"&gt;

1108 &lt;property name="name" value="Spring" /&gt;

1109 &lt;property name="printer" ref="printer" /&gt;

1110 &lt;/bean&gt;

1111 &lt;bean id="printer" class="com.example.StringPrinter" /&gt;

1112 &lt;bean id="consolePrinter" class="com.example.ConsolePrinter" /&gt;

1113

1114 &lt;/beans&gt;

1115

1116

## 1117 7. Test class 작성

1118 1)/src &gt; right-click &gt; New &gt; Package

1119 2)Package Name : com.example.test

1120 3)/src/com.example/test/HelloBeanTest.java

1121

1122 package com.example.test;

1123

1124 import org.springframework.context.ApplicationContext;

1125 import org.springframework.context.support.GenericXmlApplicationContext;

1126

1127 import com.example.Hello;

1128 import com.example.Printer;

1129

1130 public class HelloBeanTest {

1131 public static void main(String [] args){

1132 //1. IoC Container 생성

1133 ApplicationContext context =

1134 new GenericXmlApplicationContext("classpath:beans.xml");

1135

1136 //2. Hello Beans 가져오기

1137 Hello hello = (Hello)context.getBean("hello");

1138 System.out.println(hello.sayHello());

1139 hello.print();

1140

1141 //3. StringPrinter 가져오기

1142 Printer printer = (Printer)context.getBean("printer");

1143 System.out.println(printer.toString());

1144

1145 Hello hello2 = context.getBean("hello", Hello.class);

1146 hello2.print();

1147

1148 System.out.println(hello == hello2); //Singleton Pattern

1149 }

1150 }

1151

1152

## 1153 8. Test

```
1154 1)/src/com.example.test/HelloBeanTest.java > right-click > Run As > Java Application
1155 Hello Spring
1156 Hello Spring
1157 true
1158
1159
1160 9. /src/com.example.Hello 생성자 추가
1161
1162 public Hello(String name, Printer printer) {
1163     this.name = name;
1164     this.printer = printer;
1165 }
1166
1167
1168 10. /resources/beans.xml에 아래 Code 추가
1169
1170 <bean id="hello2" class="com.example.Hello">
1171     <constructor-arg index="0" value="Spring" />
1172     <constructor-arg index="1" ref="printer" />
1173 </bean>
1174
1175
1176 11. /src/com.example.test/HelloBeanTest.java 수정
1177
1178 ...
1179 //2. Hello Beans 가져오기
1180 Hello hello = (Hello)context.getBean("hello2");
1181 ...
1182 Hello hello2 = context.getBean("hello2", Hello.class);
1183 ...
1184
1185
1186 12. Test
1187 1)/src/com.example.test/HelloBeanTest.java > right-click > Run As > Java Application
1188 Hello Spring
1189 Hello Spring
1190 true
1191
1192
1193
1194 -----
1195 Task 9. Java Annotation을 이용한 생성자 이용하여 의존 주입하기 실습
1196 1. In Package Explorer > right-click > New > Java Project
1197 1)Project Name : SpringDemo1
1198 2)JRE > Select [Use default JRE 'jdk-13.0.2' and workspace compiler preferences]
1199 3)Next
1200 4)Uncheck [Create module-info.java file]
1201 5)Finish
1202
1203
1204 2. src > right-click > New > Package
1205 1)Package name : com.example
1206 2)Finish
1207
1208
1209 3. POJO Class 생성
1210 1)com.example.Student.java
1211 package com.example;
```

```
1212
1213     public class Student {
1214         private String name;
1215         private int age;
1216         private int grade;
1217         private int classNum;
1218     }
1219
1220 2)com.example.StudentInfo.java
1221     package com.example;
1222
1223     public class StudentInfo {
1224         private Student student;
1225     }
1226
1227
1228 4. Java Project를 Spring Project로 변환
1229 1)SpringDemo1 Project > right-click > Configure > Convert to Maven Project
1230     -Project : /SpringDemo1
1231     -Group Id : SpringDemo1
1232     -Artifact Id : SpringDemo1
1233     -version : 0.0.1-SNAPSHOT
1234     -Packaging : jar
1235     -Finish
1236
1237 2)SpringDemo1 Project > right-click > Spring > Add Spring Project Nature
1238
1239 3)pom.xml file에 Spring Context Dependency 추가하기
1240     <version>0.0.1-SNAPSHOT</version>
1241     <dependencies>
1242         <dependency>
1243             <groupId>org.springframework</groupId>
1244             <artifactId>spring-context</artifactId>
1245             <version>5.2.5.RELEASE</version>
1246         </dependency>
1247     </dependencies>
1248
1249 4)pom.xml > right-click > Run As > Maven install
1250     [INFO] BUILD SUCCESS 확인
1251
1252
1253 5. Lombok library 추가
1254 1)https://mvnrepository.com/에서 'lombok'으로 검색
1255 2)'Project Lombok' click
1256 3)1.18.12 click
1257 4)dependency copy해서 pom.xml에 붙여넣기
1258
1259     <dependencies>
1260         <dependency>
1261             <groupId>org.springframework</groupId>
1262             <artifactId>spring-context</artifactId>
1263             <version>5.2.5.RELEASE</version>
1264         </dependency>
1265         <!-- https://mvnrepository.com/artifact/org.projectlombok/lombok -->
1266         <dependency>
1267             <groupId>org.projectlombok</groupId>
1268             <artifactId>lombok</artifactId>
1269             <version>1.18.12</version>
```

```
1270         <scope>provided</scope>
1271     </dependency>
1272 </dependencies>
1273
1274 5)pom.xml > right-click > Run As > Maven install
1275 [INFO] BUILD SUCCESS 확인
1276
1277
1278 6. Student.java와 StudentInfo.java 수정
1279 1)Student.java
1280
1281     package com.example;
1282
1283     import lombok.Getter;
1284     import lombok.Setter;
1285     import lombok.ToString;
1286     import lombok.AllArgsConstructor;
1287
1288     @Getter
1289     @Setter
1290     @ToString
1291     @AllArgsConstructor
1292     public class Student {
1293         private String name;
1294         private int age;
1295         private int grade;
1296         private int classNum;
1297     }
1298
1299 2)StudentInfo.java
1300
1301     package com.example;
1302
1303     import lombok.Setter;
1304     import lombok.AllArgsConstructor;
1305
1306     @Setter
1307     @AllArgsConstructor
1308     public class StudentInfo {
1309         private Student student;
1310
1311         public void printInfo(){
1312             if(this.student != null){
1313                 System.out.println("Name : " + this.student.getName());
1314                 System.out.println("Age : " + this.student.getAge());
1315                 System.out.println("Grade : " + this.student.getGrade());
1316                 System.out.println("Class : " + this.student.getClassNum());
1317                 System.out.println("-----");
1318             }
1319         }
1320     }
1321
1322
1323 7. 환경설정을 위해 config package 생성
1324 1)com.example package > right-click > New > Package
1325 2)Name : com.example.config
1326 3)Finish
1327
```

```
1328
1329 8. ApplicationContext.java 생성
1330 1)com.example.config > right-click > New > Class
1331 2)Name : ApplicationCtx
1332 3)Finish
1333
1334     package com.example.config;
1335
1336     import org.springframework.context.annotation.Bean;
1337     import org.springframework.context.annotation.Configuration;
1338
1339     import com.example.Student;
1340     import com.example.StudentInfo;
1341
1342     @Configuration
1343     public class ApplicationCtx {
1344         @Bean
1345         public Student student1() {
1346             return new Student("한지민", 15, 2, 5);
1347         }
1348
1349         @Bean
1350         public Student student2() {
1351             return new Student("김지민", 16, 3, 7);
1352         }
1353
1354         @Bean
1355         public StudentInfo studentInfo() {
1356             return new StudentInfo(this.student1());
1357         }
1358     }
1359
1360
1361 9. com.example.MainClass.java
1362
1363     package com.example;
1364
1365     import org.springframework.context.ApplicationContext;
1366     import org.springframework.context.annotation.AnnotationConfigApplicationContext;
1367
1368     import com.example.config.ApplicationCtx;
1369
1370     public class MainClass {
1371         public static void main(String[] args) {
1372             ApplicationContext ctx = new
1373                 AnnotationConfigApplicationContext(ApplicationCtx.class);
1374
1375             StudentInfo studentInfo = ctx.getBean("studentInfo", StudentInfo.class);
1376             studentInfo.printInfo();
1377
1378             Student student2 = ctx.getBean("student2", Student.class);
1379             studentInfo.setStudent(student2);
1380             studentInfo.printInfo();
1381         }
1382     }
1383
1384 10. Java Application 실행
```



```
1385 Name : 한지민
1386 Age : 15
1387 Grade : 2
1388 Class : 5
1389 -----
1390 Name : 김지민
1391 Age : 16
1392 Grade : 3
1393 Class : 7
1394 -----
1395
1396
1397
1398 -----
1399 Task 10. Context file 여러개 사용하기
1400 1. In Package Explorer > right-click > New > Java Project
1401    1)Project Name : SpringDemo2
1402    2)JRE
1403       -Select [Use default JRE 'jdk-13.0.2' and workspace compiler preferences]
1404    3)Next
1405    4)Uncheck [Create module-info.java file]
1406    5)Finish
1407
1408
1409 2)src > right-click > New > Package
1410    2)Package name : com.example
1411
1412
1413 3. POJO Class 생성
1414    1)com.example.Student.java
1415
1416        package com.example;
1417
1418        import java.util.ArrayList;
1419
1420        public class Student {
1421            private String name;
1422            private int age;
1423            private ArrayList<String> hobbies;
1424            private double height;
1425            private double weight;
1426        }
1427
1428    2)com.example.StudentInfo.java
1429
1430        package com.example;
1431        public class StudentInfo {
1432            private Student student;
1433        }
1434
1435    3)com.example.Product.java
1436
1437        package com.example;
1438        public class Product {
1439            private String pName;
1440            private int pPrice;
1441            private String maker;
1442            private String color;
```

```
1443     }
1444
1445
1446 4. Java Project를 Spring Project로 변환
1447 1)SpringDemo2 Project > right-click > Configure > Convert to Maven Project
1448     -Project : /SpringDemo2
1449     -Group Id : SpringDemo2
1450     -Artifact Id : SpringDemo2
1451     -version : 0.0.1-SNAPSHOT
1452     -Packaging : jar
1453     -Finish
1454
1455 2)SpringDemo2 Project > right-click > Spring > Add Spring Project Nature
1456
1457 3)pom.xml file에 Spring Context Dependency 추가하기
1458     <version>0.0.1-SNAPSHOT</version>
1459     <dependencies>
1460         <dependency>
1461             <groupId>org.springframework</groupId>
1462             <artifactId>spring-context</artifactId>
1463             <version>5.2.5.RELEASE</version>
1464         </dependency>
1465     </dependencies>
1466
1467 4)pom.xml > right-click > Run As > Maven install
1468     [INFO] BUILD SUCCESS 확인
1469
1470
1471 5. Lombok library 추가
1472 1)https://mvnrepository.com/에서 'lombok'으로 검색
1473 2)'Project Lombok' click
1474 3)1.18.12 click
1475 4)dependency copy해서 pom.xml에 붙여넣기
1476
1477     <dependencies>
1478         <dependency>
1479             <groupId>org.springframework</groupId>
1480             <artifactId>spring-context</artifactId>
1481             <version>5.2.5.RELEASE</version>
1482         </dependency>
1483         <!-- https://mvnrepository.com/artifact/org.projectlombok/lombok -->
1484         <dependency>
1485             <groupId>org.projectlombok</groupId>
1486             <artifactId>lombok</artifactId>
1487             <version>1.18.12</version>
1488             <scope>provided</scope>
1489         </dependency>
1490     </dependencies>
1491
1492 5)pom.xml > right-click > Run As > Maven install
1493     [INFO] BUILD SUCCESS 확인
1494
1495
1496 6. SpringDemo2/resources folder 생성
1497 1)SpringDemo2 project > right-click > new > Source Folder
1498 2)Folder Name : resources
1499 3)Finish
1500
```

```
1501
1502 7. Bean Configuration XML 작성
1503     1)resources Folder > right-click > New > Spring Bean Configuration File
1504     2)File name : applicationContext.xml > Finish
1505     3)resources Folder > right-click > New > Spring Bean Configuration File
1506     4)File name : applicationContext2.xml > Finish
1507
1508
1509 8. Student.java, StudentInfo.java 그리고 Product.java에 lombok Annotation 붙이기
1510     1)Student.java
1511
1512         package com.example;
1513
1514         import java.util.ArrayList;
1515
1516         import lombok.AllArgsConstructor;
1517         import lombok.Data;
1518         import lombok.NonNull;
1519         import lombok.RequiredArgsConstructor;
1520
1521         @Data
1522         @RequiredArgsConstructor
1523         @AllArgsConstructor
1524         public class Student {
1525             private @NonNull String name;
1526             private @NonNull int age;
1527             private @NonNull ArrayList<String> hobbies;
1528             private double height;
1529             private double weight;
1530         }
1531
1532     2)StudentInfo.java
1533
1534         package com.example;
1535
1536         import lombok.Setter;
1537         import lombok.Getter;
1538
1539         @Setter
1540         @Getter
1541         public class StudentInfo {
1542             private Student student;
1543         }
1544
1545     3)Product.java
1546
1547         package com.example;
1548
1549         import lombok.AllArgsConstructor;
1550         import lombok.NoArgsConstructor;
1551         import lombok.NonNull;
1552         import lombok.RequiredArgsConstructor;
1553         import lombok.Setter;
1554         import lombok.ToString;
1555
1556         @NoArgsConstructor
1557         @AllArgsConstructor
1558         @RequiredArgsConstructor
```

```

1559     @Setter
1560     @ToString
1561     public class Product {
1562         private @NonNull String pName;
1563         private @NonNull int pPrice;
1564         private String maker;
1565         private String color;
1566     }
1567
1568
1569 9. applicationContext.xml
1570 <?xml version="1.0" encoding="UTF-8"?>
1571 <beans xmlns="http://www.springframework.org/schema/beans"
1572       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
1573       xsi:schemaLocation="http://www.springframework.org/schema/beans
1574       http://www.springframework.org/schema/beans/spring-beans.xsd">
1575
1576     <bean id="student1" class="com.example.Student">
1577         <constructor-arg value="한지민" />
1578         <constructor-arg value="25" />
1579         <constructor-arg>
1580             <list>
1581                 <value>독서</value>
1582                 <value>영화감상</value>
1583                 <value>요리</value>
1584             </list>
1585         </constructor-arg>
1586         <property name="height" value="165" />
1587         <property name="weight">
1588             <value>45</value>
1589         </property>
1590     </bean>
1591
1592     <bean id="studentInfo1" class="com.example.StudentInfo">
1593         <property name="student">
1594             <ref bean="student1" />
1595         </property>
1596     </bean>
1597 </beans>
1598
1599 10. applicationContext2.xml
1600 1) Namespace tab을 선택하여 c, p를 선택한다.
1601 <?xml version="1.0" encoding="UTF-8"?>
1602 <beans xmlns="http://www.springframework.org/schema/beans"
1603       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
1604       xmlns:c="http://www.springframework.org/schema/c"
1605       xmlns:p="http://www.springframework.org/schema/p"
1606       xsi:schemaLocation="http://www.springframework.org/schema/beans
1607       http://www.springframework.org/schema/beans/spring-beans.xsd">
1608
1609     <bean id="student3" class="com.example.Student">
1610         <constructor-arg value="김지민" />
1611         <constructor-arg value="50" />
1612         <constructor-arg>
1613             <list>
1614                 <value>노래부르기</value>
1615                 <value>게임</value>

```

```

1615         </list>
1616     </constructor-arg>
1617     <property name="height" value="175" />
1618     <property name="weight">
1619         <value>75</value>
1620     </property>
1621 </bean>
1622
1623     <bean id="product" class="com.example.Product" c:pName="Computer"
1624         c:pPrice="2000000" p:maker="Samsung">
1625         <property name="color" value="Yellow" />
1626     </bean>
1627 </beans>
1628
1629 11. com.example.MainClass
1630 package com.example;
1631
1632 import org.springframework.context.support.AbstractApplicationContext;
1633 import org.springframework.context.support.GenericXmlApplicationContext;
1634
1635 public class MainClass {
1636     public static void main(String[] args) {
1637         String configFile = "classpath:applicationContext.xml";
1638         String configFile1 = "classpath:applicationContext2.xml";
1639         AbstractApplicationContext context = new GenericXmlApplicationContext(configFile,
1640             configFile1);
1641         Student student1 = context.getBean("student1", Student.class);
1642         System.out.println(student1);
1643
1644         StudentInfo studentInfo = context.getBean("studentInfo1", StudentInfo.class);
1645         Student student2 = studentInfo.getStudent();
1646         System.out.println(student2);
1647         if(student1.equals(student2)) System.out.println("Equals");
1648         else System.out.println("Different");
1649
1650         Student student3 = context.getBean("student3", Student.class);
1651         System.out.println(student3);
1652
1653         if(student1.equals(student3)) System.out.println("Equals");
1654         else System.out.println("Different");
1655
1656         Product product = context.getBean("product", Product.class);
1657         System.out.println(product);
1658         context.close();
1659     }
1660 }
1661
1662 12. Java Application 실행
1663 Student [name=한지민, age=25, hobbies=[독서, 영화감상, 요리], height=165.0,weight=45.0]
1664 Student [name=한지민, age=25, hobbies=[독서, 영화감상, 요리], height=165.0,weight=45.0]
1665 Equals
1666 Student [name=김지민, age=50, hobbies=[노래부르기, 게임], height=175.0,weight=75.0]
1667 Different
1668 Product [pName=Computer, pPrice=2000000, maker=Samsung, color=Yellow]
1669
1670

```

```
1671 -----
1672 Task 11. Java Annotation을 이용하여 두 개 이상의 설정 파일로 DI 설정하기
1673 1. In Package Explorer > right-click > New > Java Project
1674     1)Project Name : SpringDemo3
1675     2)JRE
1676         -Select [Use default JRE 'jdk-13.0.2' and workspace compiler preferences]
1677     3)Next
1678     4)Uncheck [Create module-info.java file]
1679     5)Finish
1680
1681
1682 2)src > right-click > New > Package
1683     1)Package name : com.example
1684     2)Finish
1685
1686
1687 3. POJO Class 생성
1688     1)com.example.Student.java
1689
1690         package com.example;
1691
1692         import java.util.ArrayList;
1693
1694         public class Student {
1695             private String name;
1696             private int age;
1697             private ArrayList<String> hobbies;
1698             private double height;
1699             private double weight;
1700         }
1701
1702     2)com.example.StudentInfo.java
1703
1704         package com.example;
1705         public class StudentInfo {
1706             private Student student;
1707         }
1708
1709     3)com.example.Product.java
1710
1711         package com.example;
1712         public class Product {
1713             private String pName;
1714             private int pPrice;
1715             private String maker;
1716             private String color;
1717         }
1718
1719 4. Java Project를 Spring Project로 변환
1720     1)SpringDemo3 Project > right-click > Configure > Convert to Maven Project
1721         -Project : /SpringDemo3
1722         -Group Id : SpringDemo3
1723         -Artifact Id : SpringDemo3
1724         -version : 0.0.1-SNAPSHOT
1725         -Packaging : jar
1726         -Finish
1727
1728     2)SpringDemo3 Project > right-click > Spring > Add Spring Project Nature
```

```
1729
1730 3)pom.xml file에 Spring Context Dependency 추가하기
1731     <version>0.0.1-SNAPSHOT</version>
1732     <dependencies>
1733         <dependency>
1734             <groupId>org.springframework</groupId>
1735             <artifactId>spring-context</artifactId>
1736             <version>5.2.5.RELEASE</version>
1737         </dependency>
1738     </dependencies>
1739
1740 4)pom.xml > right-click > Run As > Maven install
1741     [INFO] BUILD SUCCESS 확인
1742
1743 5. Lombok library 추가
1744 1) https://mvnrepository.com/에서 'lombok'으로 검색
1745 2)'Project Lombok' click
1746 3)1.18.12 click
1747 4)dependency copy해서 pom.xml에 붙여넣기
1748
1749     <dependencies>
1750     <dependency>
1751         <groupId>org.springframework</groupId>
1752         <artifactId>spring-context</artifactId>
1753         <version>5.2.5.RELEASE</version>
1754     </dependency>
1755     <!-- https://mvnrepository.com/artifact/org.projectlombok/lombok -->
1756     <dependency>
1757         <groupId>org.projectlombok</groupId>
1758         <artifactId>lombok</artifactId>
1759         <version>1.18.12</version>
1760         <scope>provided</scope>
1761     </dependency>
1762 </dependencies>
1763
1764 5)pom.xml > right-click > Run As > Maven install
1765     [INFO] BUILD SUCCESS 확인
1766
1767 6. com.example.config package 생성
1768 1)com.example > right-click > new > Package
1769 2)Name : com.example.config
1770 3)Finish
1771
1772 7. 2개의 Config Class 작성
1773 1)com.example.config > right-click > New > Class
1774 2)Name : AppConfig1
1775 3)Finish
1776 4)com.example.config > right-click > New > Class
1777 5)Name : AppConfig2
1778 6)Finish
1779
1780 8. Student.java, StudentInfo.java 그리고 Product.java에 lombok Annotation 붙이기
1781 1)Student.java
1782
1783
1784
1785
1786
```

```
1787     package com.example;
1788
1789     import java.util.List;
1790
1791     import lombok.AllArgsConstructor;
1792     import lombok.Data;
1793     import lombok.NonNull;
1794     import lombok.RequiredArgsConstructor;
1795
1796     @Data
1797     @RequiredArgsConstructor
1798     @AllArgsConstructor
1799     public class Student {
1800         private @NonNull String name;
1801         private @NonNull int age;
1802         private @NonNull List<String> hobbies;
1803         private double height;
1804         private double weight;
1805     }
```

#### 2)StudentInfo.java

```
1809     package com.example;
1810
1811     import lombok.Setter;
1812     import lombok.Getter;
1813
1814     @Setter
1815     @Getter
1816     public class StudentInfo {
1817         private Student student;
1818     }
```

#### 3)Product.java

```
1822     package com.example;
1823
1824     import lombok.AllArgsConstructor;
1825     import lombok.NoArgsConstructor;
1826     import lombok.NonNull;
1827     import lombok.RequiredArgsConstructor;
1828     import lombok.Setter;
1829     import lombok.ToString;
1830
1831     @NoArgsConstructor
1832     @AllArgsConstructor
1833     @RequiredArgsConstructor
1834     @Setter
1835     @ToString
1836     public class Product {
1837         private @NonNull String pName;
1838         private @NonNull int pPrice;
1839         private @NonNull String maker;
1840         private String color;
1841     }
```

#### 9. AppConfig1.java



```
1845 package com.example.config;
1846
1847 import java.util.Arrays;
1848 import java.util.List;
1849
1850 import org.springframework.context.annotation.Bean;
1851 import org.springframework.context.annotation.Configuration;
1852
1853 import com.example.Student;
1854 import com.example.StudentInfo;
1855
1856 @Configuration
1857 public class AppConfig1 {
1858     @Bean
1859     public Student student1() {
1860         List<String> list = Arrays.asList("독서", "영화감상", "요리");
1861         Student student1 = new Student("한지민", 25, list);
1862         student1.setHeight(165);
1863         student1.setWeight(45);
1864         return student1;
1865     }
1866
1867     @Bean
1868     public StudentInfo studentInfo1() {
1869         StudentInfo studentInfo1 = new StudentInfo();
1870         studentInfo1.setStudent(this.student1());
1871         return studentInfo1;
1872     }
1873 }
1874
1875
1876 10. AppConfig2.java
1877 package com.example.config;
1878
1879 import java.util.Arrays;
1880 import java.util.List;
1881
1882 import org.springframework.context.annotation.Bean;
1883 import org.springframework.context.annotation.Configuration;
1884
1885 import com.example.Product;
1886 import com.example.Student;
1887
1888 @Configuration
1889 public class AppConfig2 {
1890     @Bean
1891     public Student student3() {
1892         List<String> list = Arrays.asList("노래부르기", "게임");
1893         Student student3 = new Student("김지민", 50, list);
1894         student3.setHeight(175);
1895         student3.setWeight(75);
1896         return student3;
1897     }
1898
1899     @Bean
1900     public Product product() {
1901         Product product = new Product("Computer", 2000000, "Samsung");
1902         product.setColor("Yellow");
```

```

1903     return product;
1904 }
1905 }
1906
1907
1908 11. com.example.MainClass
1909     package com.example;
1910
1911     import org.springframework.context.ApplicationContext;
1912     import org.springframework.context.annotation.AnnotationConfigApplicationContext;
1913
1914     import com.example.config.AppConfig1;
1915     import com.example.config.AppConfig2;
1916
1917     public class MainClass {
1918         public static void main(String[] args) {
1919             ApplicationContext context = new
1920                 AnnotationConfigApplicationContext(AppConfig1.class, AppConfig2.class);
1921             Student student1 = context.getBean("student1", Student.class);
1922             System.out.println(student1);
1923
1924             StudentInfo studentInfo = context.getBean("studentInfo1", StudentInfo.class);
1925             Student student2 = studentInfo.getStudent();
1926             System.out.println(student2);
1927             if(student1.equals(student2)) System.out.println("Equals");
1928             else System.out.println("Different");
1929
1930             Student student3 = context.getBean("student3", Student.class);
1931             System.out.println(student3);
1932
1933             if(student1.equals(student3)) System.out.println("Equals");
1934             else System.out.println("Different");
1935
1936             Product product = context.getBean("product", Product.class);
1937             System.out.println(product);
1938         }
1939     }
1940
1941 12. Java Application 실행
1942     Student [name=한지민, age=25, hobbies=[독서, 영화감상, 요리], height=165.0,weight=45.0]
1943     Student [name=한지민, age=25, hobbies=[독서, 영화감상, 요리], height=165.0,weight=45.0]
1944     Equals
1945     Student [name=김지민, age=50, hobbies=[노래부르기, 게임], height=175.0,weight=75.0]
1946     Different
1947     Product [pName=Computer, pPrice=2000000, maker=Samsung, color=Yellow]
1948
1949
1950
1951 -----
1952 Task 12. Java Annotation과 XML 을 이용한 DI 설정 방법 : XML file에 Java file을 포함시켜 사용하는 방법
1953 1. In Package Explorer > right-click > New > Java Project
1954     1)Project Name : SpringDemo4
1955     2)JRE
1956         -Select [Use default JRE 'jdk-13.0.2' and workspace compiler preferences]
1957     3)Next
1958     4)Uncheck [Create module-info.java file]
1959     5)Finish

```

1960  
1961  
1962 2. src > right-click > New > Package  
1963 1)Package name : com.example  
1964  
1965  
1966 3. POJO 생성  
1967 1)com.example.Student.java  
1968 package com.example;  
1969  
1970 import java.util.ArrayList;  
1971  
1972 public class Student {  
1973 private String name;  
1974 private int age;  
1975 private ArrayList<String> hobbies;  
1976 private double height;  
1977 private double weight;  
1978 }  
1979  
1980  
1981 4. Java Project를 Spring Project로 변환  
1982 1)SpringDemo4 Project > right-click > Configure > Convert to Maven Project  
1983 -Project : /SpringDemo4  
1984 -Group Id : SpringDemo4  
1985 -Artifact Id : SpringDemo4  
1986 -version : 0.0.1-SNAPSHOT  
1987 -Packaging : jar  
1988 -Finish  
1989  
1990 2)SpringDemo4 Project > right-click > Spring > Add Spring Project Nature  
1991  
1992 3)pom.xml file에 Spring Context Dependency 추가하기  
1993 <version>0.0.1-SNAPSHOT</version>  
1994 <dependencies>  
1995 <dependency>  
1996 <groupId>org.springframework</groupId>  
1997 <artifactId>spring-context</artifactId>  
1998 <version>5.2.5.RELEASE</version>  
1999 </dependency>  
2000 </dependencies>  
2001  
2002 4)pom.xml > right-click > Run As > Maven install  
2003 [INFO] BUILD SUCCESS 확인  
2004  
2005  
2006 5. Lombok library 추가  
2007 1)<https://mvnrepository.com/>에서 'lombok'으로 검색  
2008 2)'Project Lombok' click  
2009 3)1.18.12 click  
2010 4)dependency copy해서 pom.xml에 붙여넣기  
2011  
2012 <dependencies>  
2013 <dependency>  
2014 <groupId>org.springframework</groupId>  
2015 <artifactId>spring-context</artifactId>  
2016 <version>5.2.5.RELEASE</version>  
2017 </dependency>

```
2018      <!-- https://mvnrepository.com/artifact/org.projectlombok/lombok -->
2019      <dependency>
2020          <groupId>org.projectlombok</groupId>
2021          <artifactId>lombok</artifactId>
2022          <version>1.18.12</version>
2023          <scope>provided</scope>
2024      </dependency>
2025  </dependencies>
```

```
2026
2027 5) pom.xml > right-click > Run As > Maven install
2028 [INFO] BUILD SUCCESS 확인
2029
```

```
2030
2031 6. Student.java lombok Annotation 붙이기
2032 1) Student.java
```

```
2033
2034     package com.example;
2035
2036     import java.util.List;
2037
2038     import lombok.AllArgsConstructor;
2039     import lombok.Data;
2040     import lombok.NonNull;
2041     import lombok.RequiredArgsConstructor;
2042
2043     @Data
2044     @RequiredArgsConstructor
2045     @AllArgsConstructor
2046     public class Student {
2047         private @NonNull String name;
2048         private @NonNull int age;
2049         private @NonNull List<String> hobbies;
2050         private double height;
2051         private double weight;
2052     }
```

```
2053
2054
2055 7. com.example.ApplicationConfig.java
```

```
2056     package com.example;
2057
2058     import java.util.Arrays;
2059     import java.util.List;
2060
2061     import org.springframework.context.annotation.Bean;
2062     import org.springframework.context.annotation.Configuration;
2063
2064     @Configuration
2065     public class ApplicationConfig {
2066         @Bean
2067         public Student student1(){
2068             List<String> list = Arrays.asList("독서", "영화감상", "요리");
2069
2070             Student student1 = new Student("한지민", 25, list);
2071             student1.setHeight(165);
2072             student1.setWeight(45);
2073
2074             return student1;
2075         }
```

```
2076     }
2077
2078
2079 8. SpringDemo4/resources folder 생성
2080 1)SpringDemo4 project > right-click > Build Path > Configure Build Path
2081 2)Source Tab > Add Folder
2082 3)SpringDemo4 선택 확인
2083 4)Create New Folder > Folder name : resources > Finish > OK
2084 5)SpringDemo4/resources(new) 확인
2085 6)Apply and Close
2086
2087
2088 9. Bean Configuration XML 작성
2089 1)SpringDemo4/resources > right-click > New > Spring Bean Configuration File
2090 2)File name : applicationContext.xml > Finish
2091
2092
2093 10. /resources/applicationContext.xml
2094 <?xml version="1.0" encoding="UTF-8"?>
2095 <beans xmlns="http://www.springframework.org/schema/beans"
2096     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
2097     xmlns:context="http://www.springframework.org/schema/context"
2098     xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans.xsd">
2099
2100     <bean class="org.springframework.context.annotation.ConfigurationClassPostProcessor"
2101     />
2102     <bean class="com.example.ApplicationConfig" />
2103     <bean id="student3" class="com.example.Student">
2104         <constructor-arg value="박지민" />
2105         <constructor-arg value="50" />
2106         <constructor-arg>
2107             <list>
2108                 <value>노래부르기</value>
2109                 <value>게임</value>
2110             </list>
2111         </constructor-arg>
2112         <property name="height" value="175" />
2113         <property name="weight">
2114             <value>75</value>
2115         </property>
2116     </bean>
2117 </beans>
2118
2119 11. com.example.MainClass.java
2120 package com.example;
2121
2122 import org.springframework.context.support.AbstractApplicationContext;
2123 import org.springframework.context.support.GenericXmlApplicationContext;
2124
2125 public class MainClass {
2126     public static void main(String[] args) {
2127         String configFile = "classpath:applicationContext.xml";
2128         AbstractApplicationContext context = new GenericXmlApplicationContext(configFile);
2129         Student student1 = context.getBean("student1", Student.class);
2130         System.out.println(student1);
2131     }
```

```
2132     Student student3 = context.getBean("student3", Student.class);
2133     System.out.println(student3);
2134 }
2135 }
2136
2137
2138 12. Java Application 실행
2139     Student [name=한지민, age=25, hobbies=[독서, 영화감상, 요리], height=165.0,weight=45.0]
2140     Student [name=박지민, age=50, hobbies=[노래부르기, 게임], height=175.0,weight=75.0]
2141
2142
2143 13. JUnit을 사용한 DI test class 작성하기
2144     1)com.example > right-click > New > JUnit Test Case
2145     2)Select [New JUnit 4 test]
2146     3)Name : HelloBeanJUnitTest
2147     4)Finish
2148     5)[New JUnit Test Case] 창에서 Select [Perform the follwing action:] > Add JUnit 4 library to
2149     the build path
2150     6)OK
2151
2152 14. JUnit을 사용한 Test
2153     1)src/com.example > New > Class
2154     -Name : HelloBeanJUnitTest.java
2155
2156     package com.example;
2157
2158     import static org.junit.Assert.assertEquals;
2159     import static org.junit.Assert.assertSame;
2160
2161     import org.junit.Before;
2162     import org.junit.Test;
2163     import org.springframework.context.ApplicationContext;
2164     import org.springframework.context.support.GenericXmlApplicationContext;
2165
2166     public class HelloBeanJUnitTest {
2167         ApplicationContext context;
2168
2169         @Before
2170         public void init(){
2171             context = new GenericXmlApplicationContext("classpath:applicationContext.xml");
2172         }
2173
2174         @Test
2175         public void test1(){
2176             Student student1 = (Student)context.getBean("student1");
2177             assertEquals("한지민", student1.getName());
2178             System.out.println(student1);
2179         }
2180
2181         @Test
2182         public void test2(){
2183             Student student3 = context.getBean("student3", Student.class);
2184             System.out.println(student3);
2185
2186             Student student4 = (Student)context.getBean("student3");
2187             assertEquals(student3, student4);
2188         }
2189     }
```

```

2189     }
2190
2191 2)HelloBeanJUnitTest.java > right-click > Run As > JUnit Test
2192     -JUnit 창에 Green Bar
2193         Student(name=한지민, age=25, hobbies=[독서, 영화감상, 요리], height=165.0,
2194             weight=45.0)
2195         Student(name=박지민, age=50, hobbies=[노래부르기, 게임], height=175.0, weight=75.0)
2196
2197 15. Spring TestContext Framework을 이용한 Test
2198 1)Spring-Test library 설치
2199     -http://mvnrepository.com에서 'spring test'로 검색
2200     -검색 결과 목록에서 'Spring TestContext Framework' Click
2201     -version 목록에서 5.2.5.RELEASE Click
2202
2203 2)dependency 복사해서 pom.xml에 붙여넣기
2204     <!-- https://mvnrepository.com/artifact/org.springframework/spring-test -->
2205     <dependency>
2206         <groupId>org.springframework</groupId>
2207         <artifactId>spring-test</artifactId>
2208         <version>5.2.5.RELEASE</version>
2209         <scope>test</scope>
2210     </dependency>
2211
2212 3)pom.xml > right-click > Maven Install
2213     [INFO] BUILD SUCCESS
2214
2215 4)Spring-Test를 사용할 HelloBeanJUnitSpringTest.java 작성
2216     -src/com.example > New > Class
2217     -Name : HelloBeanJUnitSpringTest
2218     -Finish
2219
2220     package com.example;
2221
2222     import static org.junit.Assert.assertEquals;
2223     import static org.junit.Assert.assertSame;
2224
2225     import org.junit.Test;
2226     import org.junit.runner.RunWith;
2227     import org.springframework.beans.factory.annotation.Autowired;
2228     import org.springframework.context.ApplicationContext;
2229     import org.springframework.test.context.ContextConfiguration;
2230     import org.springframework.test.context.junit4.SpringJUnit4ClassRunner;
2231
2232     @RunWith(SpringJUnit4ClassRunner.class)
2233     @ContextConfiguration(locations="classpath:applicationContext.xml")
2234     public class HelloBeanJUnitSpringTest {
2235         @Autowired
2236         ApplicationContext context;
2237
2238         @Test
2239         public void test1() {
2240             Student student1 = this.context.getBean("student1", Student.class);
2241             assertEquals(25, student1.getAge());
2242             System.out.println(student1);
2243         }
2244
2245         @Test

```

```

2246     public void test2() {
2247         Student student3 = (Student)this.context.getBean("student3");
2248         Student student4 = this.context.getBean("student3", Student.class);
2249         assertEquals(student3, student4);
2250         System.out.println(student4);
2251     }
2252 }
2253
2254 -right-click > Run As > JUnit Test
2255 -결과 -> JUnit View에 초록색 bar
2256
2257 5)만일 해당 객체를 찾을 수 없다는 오류가 계속 발생하면
2258 -해당 Project > right-click > Build Path > Libraries tab
2259 -spring-test-5.2.5.RELEASE.jar 선택 후 [Remove] 로 삭제
2260 -Classpath 선택
2261 -[Add External JARs...] Click
2262 -Local M2 Repository(e.g
    C:\Users\bluee\.m2\repository\org\springframework\spring-test\5.2.5.RELEASE)에서 직접
    jar(spring-test-5.2.5.RELEASE.jar)를 선택할 것
2263 -[Order and Export] tab에서 spring-test-5.2.5.RELEASE.jar 선택 후 [Up] button을 클릭
2264 -해당 DIdemo/src 바로 아래까지 올리고 [Apply and Close] Click
2265
2266
2267
2268 -----
2269 Task 13. Java Annotation과 XML 을 이용한 DI 설정 방법 : Java file에 XML file을 포함시켜 사용하는 방법
2270 1. In Package Explorer > right-click > New > Java Projectn
2271     1)Project Name : SpringDemo5
2272     2)JRE
2273         -Select [Use default JRE 'jdk-13.0.2' and workspace compiler preferences]
2274     3)Next
2275     4)Uncheck [Create module-info.java file]
2276     5)Finish
2277
2278
2279 2. src > right-click > New > Package
2280     1)Package name : com.example
2281     2)Finish
2282
2283
2284 3. com.example.Student.java
2285     package com.example;
2286
2287     import java.util.List;
2288
2289     public class Student {
2290         private String name;
2291         private int age;
2292         private List<String> hobbies;
2293         private double height;
2294         private double weight;
2295     }
2296
2297
2298 4. Java Project를 Spring Project로 변환
2299     1)SpringDemo5 Project > right-click > Configure > Convert to Maven Project
2300         -Project : /SpringDemo5
2301         -Group Id : SpringDemo5

```



```
2302     -Artifact Id : SpringDemo5
2303     -version : 0.0.1-SNAPSHOT
2304     -Packaging : jar
2305     -Finish
2306
2307 2)SpringDemo5 Project > right-click > Spring > Add Spring Project Nature
2308
2309 3)pom.xml file에 Spring Context Dependency 추가하기
2310     <version>0.0.1-SNAPSHOT</version>
2311     <dependencies>
2312         <dependency>
2313             <groupId>org.springframework</groupId>
2314             <artifactId>spring-context</artifactId>
2315             <version>5.2.5.RELEASE</version>
2316         </dependency>
2317     </dependencies>
2318
2319 4)pom.xml > right-click > Run As > Maven install
2320     [INFO] BUILD SUCCESS 확인
2321
2322
2323 5. Lombok library 추가
2324 1)<a href="https://mvnrepository.com/">https://mvnrepository.com/에서 'lombok'으로 검색
2325 2)'Project Lombok' click
2326 3)1.18.12 click
2327 4)dependency copy해서 pom.xml에 붙여넣기
2328
2329     <dependencies>
2330         <dependency>
2331             <groupId>org.springframework</groupId>
2332             <artifactId>spring-context</artifactId>
2333             <version>5.2.5.RELEASE</version>
2334         </dependency>
2335         <!-- <a href="https://mvnrepository.com/artifact/org.projectlombok/lombok">https://mvnrepository.com/artifact/org.projectlombok/lombok -->
2336         <dependency>
2337             <groupId>org.projectlombok</groupId>
2338             <artifactId>lombok</artifactId>
2339             <version>1.18.12</version>
2340             <scope>provided</scope>
2341         </dependency>
2342     </dependencies>
2343
2344 5)pom.xml > right-click > Run As > Maven install
2345     [INFO] BUILD SUCCESS 확인
2346
2347
2348 6. Student.java lombok Annotation 붙이기
2349 1)Student.java
2350
2351     package com.example;
2352
2353     import java.util.List;
2354
2355     import lombok.AllArgsConstructor;
2356     import lombok.Data;
2357     import lombok.NonNull;
2358     import lombok.RequiredArgsConstructor;
2359
```

```
2360     @Data
2361     @RequiredArgsConstructor
2362     @AllArgsConstructor
2363     public class Student {
2364         private @NonNull String name;
2365         private @NonNull int age;
2366         private @NonNull List<String> hobbies;
2367         private double height;
2368         private double weight;
2369     }
2370
2371
2372 7. SpringDemo5/resources folder 생성
2373 1)SpringDemo5 project > right-click > Build Path > Configure Build Path
2374 2)Source Tab > Add Folder
2375 3)SpringDemo5 선택 확인
2376 4)Create New Folder > Folder name : resources > Finish > OK
2377 5)SpringDemo5/resources(new) 확인
2378 6)Apply and Close
2379
2380
2381 8. Bean Configuration XML 작성
2382 1)SpringDemo5/resources > right-click > New > Spring Bean Configuration File
2383 2)File name : applicationContext.xml > Finish
2384
2385
2386 9. ./resources/applicationContext.xml
2387 <?xml version="1.0" encoding="UTF-8"?>
2388 <beans xmlns="http://www.springframework.org/schema/beans"
2389     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
2390     xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans.xsd">
2391
2392     <bean id="student3" class="com.example.Student">
2393         <constructor-arg value="홍지민" />
2394         <constructor-arg value="30" />
2395         <constructor-arg>
2396             <list>
2397                 <value>등산</value>
2398                 <value>게임</value>
2399                 <value>독서</value>
2400             </list>
2401         </constructor-arg>
2402         <property name="height" value="165" />
2403         <property name="weight">
2404             <value>49</value>
2405         </property>
2406     </bean>
2407 </beans>
2408
2409
2410 10. com.example.ApplicationConfig.java
2411 package com.example;
2412
2413 import java.util.Arrays;
2414 import java.util.List;
2415
2416 import org.springframework.context.annotation.Bean;
```

```
2417 import org.springframework.context.annotation.Configuration;
2418 import org.springframework.context.annotation.ImportResource;
2419
2420 @Configuration
2421 @ImportResource("classpath:ApplicationContext.xml")
2422 public class ApplicationConfig {
2423     @Bean
2424     public Student student1(){
2425         List<String> hobbies = Arrays.asList("독서", "영화감상", "요리");
2426
2427         Student student = new Student("한지민", 25, hobbies);
2428         student.setHeight(165);
2429         student.setWeight(45);
2430
2431         return student;
2432     }
2433 }
2434
2435
2436 11. com.example.MainClass.java
2437 package com.example;
2438
2439 import org.springframework.context.annotation.AnnotationConfigApplicationContext;
2440
2441 public class MainClass {
2442     public static void main(String[] args) {
2443         AnnotationConfigApplicationContext context = new
2444             AnnotationConfigApplicationContext(ApplicationConfig.class);
2445         Student student1 = context.getBean("student1", Student.class);
2446         System.out.println(student1);
2447
2448         Student student3 = context.getBean("student3", Student.class);
2449         System.out.println(student3);
2450
2451         context.close();
2452     }
2453 }
2454
2455 12. Java Application 실행
2456 Student(name=한지민, age=25, hobbies=[독서, 영화감상, 요리], height=165.0, weight=45.0)
2457 Student(name=홍지민, age=30, hobbies=[등산, 게임, 독서], height=165.0, weight=49.0)
2458
2459
2460 13. JUnit 5를 사용한 DI test class 작성하기
2461 1)com.example > right-click > New > JUnit Test Case
2462 2)Select [New JUnit Jupiter test]
2463 3)Name : HelloBeanJUnitTest
2464 4)Finish
2465 5)[New JUnit Test Case] 창에서 Select [Perform the following action:] > Add JUnit 5 library to
the build path
2466 6)OK
2467
2468
2469 14. pom.xml에 dependency 추가
2470 1)JUnit 5 설치
2471 -http://mvnrepository.com에서 'junit'로 검색
2472 -검색 결과 목록에서 'JUnit Jupiter API' Click
```

```
2473 -version 목록에서 5.6.2 click
2474
2475 2)dependency 복사해서 pom.xml에 붙여넣기
2476 <!-- https://mvnrepository.com/artifact/org.junit.jupiter/junit-jupiter-api -->
2477 <dependency>
2478     <groupId>org.junit.jupiter</groupId>
2479     <artifactId>junit-jupiter-api</artifactId>
2480     <version>5.6.2</version>
2481     <scope>test</scope>
2482 </dependency>
2483
2484 3)pom.xml > right-click > Maven Install
2485 [INFO] BUILD SUCCESS
2486 -만일 ERROR 발생하면 다음과 같이 조치한다.
2487 -SpringDemo5 > right-click > Maven > Update Project
2488 -SpringDemo5가 check되어 있음을 확인하고 OK
2489 -다시 pom.xml > right-click > Maven Install
2490 [INFO] BUILD SUCCESS
2491
2492
2493 15. JUnit 5를 사용한 Test
2494 1)com.example.HelloBeanJUnitTest.java
2495
2496 package com.example;
2497
2498 import static org.junit.jupiter.api.Assertions.assertEquals;
2499 import static org.junit.jupiter.api.Assertions.assertSame;
2500
2501 import org.junit.jupiter.api.BeforeEach;
2502 import org.junit.jupiter.api.Test;
2503 import org.springframework.context.ApplicationContext;
2504 import org.springframework.context.annotation.AnnotationConfigApplicationContext;
2505
2506 class HelloBeanJUnitTest {
2507     ApplicationContext context;
2508
2509     @BeforeEach
2510     public void init() {
2511         this.context = new AnnotationConfigApplicationContext(ApplicationConfig.class);
2512     }
2513
2514     @Test
2515     public void test1(){
2516         Student student1 = (Student)context.getBean("student1");
2517         assertEquals("한지민", student1.getName());
2518         System.out.println(student1);
2519     }
2520
2521     @Test
2522     public void test2() {
2523         Student student3 = context.getBean("student3", Student.class);
2524         Student student4 = (Student)context.getBean("student3");
2525         assertEquals(student3, student4);
2526         System.out.println(student3);
2527     }
2528 }
2529
2530 2)HelloBeanJUnitTest.java > right-click > Run As > JUnit Test
```

```
2531 -JUnit 창에 Green Bar
2532 Student(name=한지민, age=25, hobbies=[독서, 영화감상, 요리], height=165.0,
2533 weight=45.0)
2534 Student(name=홍지민, age=30, hobbies=[등산, 게임, 독서], height=165.0, weight=49.0)
2535
2536
2537 -----
2538 Task 14. Lab
2539 1. In Package Explorer > right-click > New > Java Project
2540 1)Project name : DIDemo3
2541 2)JRE
2542 -Select [Use default JRE 'jdk-13.0.2' and workspace compiler preferences]
2543 3)Next
2544 4)Uncheck [Create module-info.java file]
2545 5)Finish
2546
2547
2548 2. src > right-click > New > Package
2549 1)Package name : com.example
2550 2)Finish
2551
2552
2553 3. POJO class 작성
2554 1)com.example > right-click > New > Class
2555 2)Class Name : Hello
2556
2557 package com.example;
2558
2559 public class Hello{
2560     private String name;
2561     private Printer printer;
2562
2563     public String sayHello(){
2564         return "Hello " + name;
2565     }
2566
2567     public void print(){
2568         this.printer.print(sayHello());
2569     }
2570 }
2571
2572 3)com.example > right-click > New > Interface
2573 4)interface name : Printer
2574
2575 package com.example;
2576
2577 public interface Printer{
2578     void print(String message);
2579 }
2580
2581 5)com.example > right-click > New > Class
2582 6)Class Name : StringPrinter
2583 7)Interfaces : com.example.Printer
2584
2585 package com.example;
2586
2587 public class StringPrinter implements Printer{
```

```
2588     private StringBuffer buffer = new StringBuffer();
2589
2590     @Override
2591     public void print(String message){
2592         this.buffer.append(message);
2593     }
2594
2595     public String toString(){
2596         return this.buffer.toString();
2597     }
2598 }
```

2600 8)com.example > right-click > New > Class

2601 9)Class Name : ConsolePrinter

2602 10)Interfaces : com.example.Printer

```
2603
2604     package com.example;
2605
2606     public class ConsolePrinter implements Printer{
2607
2608         @Override
2609         public void print(String message){
2610             System.out.println(message);
2611         }
2612     }
```

2613

2614

2615 4. Java Project를 Spring Project로 변환

2616 1)DIDemo3 Project > right-click > Configure > Convert to Maven Project

2617 -Project : /DIDemo3

2618 -Group Id : DIDemo3

2619 -Artifact Id : DIDemo3

2620 -version : 0.0.1-SNAPSHOT

2621 -Packaging : jar

2622 -Finish

2623

2624 2)DIDemo3 Project > right-click > Spring > Add Spring Project Nature

2625

2626 3)pom.xml file에 Spring Context Dependency 추가하기

2627 <version>0.0.1-SNAPSHOT</version>

2628 <dependencies>

2629 <dependency>

2630 <groupId>org.springframework</groupId>

2631 <artifactId>spring-context</artifactId>

2632 <version>5.2.5.RELEASE</version>

2633 </dependency>

2634 </dependencies>

2635

2636 4)pom.xml > right-click > Run As > Maven install

2637 [INFO] BUILD SUCCESS 확인

2638

2639

2640 5. Lombok library 추가

2641 1)<https://mvnrepository.com/>에서 'lombok'으로 검색

2642 2)'Project Lombok' click

2643 3)1.18.12 click

2644 4)dependency copy해서 pom.xml에 붙여넣기

2645

```
2646 <dependencies>
2647 <dependency>
2648 <groupId>org.springframework</groupId>
2649 <artifactId>spring-context</artifactId>
2650 <version>5.2.5.RELEASE</version>
2651 </dependency>
2652 <!-- https://mvnrepository.com/artifact/org.projectlombok/lombok -->
2653 <dependency>
2654 <groupId>org.projectlombok</groupId>
2655 <artifactId>lombok</artifactId>
2656 <version>1.18.12</version>
2657 <scope>provided</scope>
2658 </dependency>
2659 </dependencies>
```

2660  
2661 5) pom.xml > right-click > Run As > Maven install  
2662 [INFO] BUILD SUCCESS 확인

#### 2663 6. Hello.java에 lombok Annotation으로 수정하기

```
2664  
2665 package com.example;
2666  
2667 import lombok.NoArgsConstructor;
2668 import lombok.Setter;
2669  
2670 @Setter
2671 @NoArgsConstructor
2672 public class Hello {
2673     private String name;
2674     private Printer printer;
2675  
2676     public String sayHello(){
2677         return "Hello " + name;
2678     }
2679  
2680     public void print(){
2681         this.printer.print(sayHello());
2682     }
2683 }
2684  
2685 }
```

#### 2686 7. src/config folder 생성

- 2687 1) /src > right-click > New > Folder
- 2688 2) Folder name : config

#### 2689 8. Bean Configuration XML 작성

- 2690 1) /src/config > right-click > New > Spring Bean Configuration File
- 2691 2) File name : beans.xml > Finish

```
2692  
2693 <?xml version="1.0" encoding="UTF-8"?>
2694 <beans xmlns="http://www.springframework.org/schema/beans"
2695     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
2696     xsi:schemaLocation="http://www.springframework.org/schema/beans
2697     http://www.springframework.org/schema/beans/spring-beans.xsd">
2698  
2699     <bean id="hello" class="com.example.Hello">
```

```

2703     <property name="name" value="Spring" />
2704     <property name="printer" ref="printer" />
2705 </bean>
2706 <bean id="printer" class="com.example.StringPrinter" />
2707 <bean id="consolePrinter" class="com.example.ConsolePrinter" />
2708 </beans>
2709
2710
2711 9. DI Test class 작성
2712 1)/src > right-click > New > Package
2713 2)Name : com.example.test
2714 3)Finish
2715 4)/src/com.example.test > right-click > New > Class
2716 5)Name : HelloBeanTest
2717
2718 package com.example.test;
2719
2720 import org.springframework.context.ApplicationContext;
2721 import org.springframework.context.support.GenericXmlApplicationContext;
2722
2723 import com.example.Hello;
2724 import com.example.Printer;
2725
2726 public class HelloBeanTest {
2727     public static void main(String [] args){
2728         ApplicationContext context = new
2729             GenericXmlApplicationContext("config/beans.xml");
2730
2731         Hello hello = (Hello)context.getBean("hello");
2732         System.out.println(hello.sayHello());
2733         hello.print();
2734
2735         Printer printer = (Printer)context.getBean("printer");
2736         System.out.println(printer.toString());
2737
2738         Hello hello2 = context.getBean("hello", Hello.class);
2739         hello2.print();
2740
2741         System.out.println(hello == hello2); //Singleton Pattern
2742     }
2743 }
2744
2745 6)Java Application 실행
2746 Hello Spring
2747 Hello Spring
2748 true
2749
2750 10. JUnit 5 Library 설치
2751 1)JUnit 5 설치
2752 -http://mvnrepository.com에서 'junit'로 검색
2753 -검색 결과 목록에서 'JUnit Jupiter API' Click
2754 -version 목록에서 5.6.2 click
2755
2756 2)dependency 복사해서 pom.xml에 붙여넣기
2757 <!-- https://mvnrepository.com/artifact/org.junit.jupiter/junit-jupiter-api -->
2758 <dependency>
2759     <groupId>org.junit.jupiter</groupId>

```



```
2760     <artifactId>junit-jupiter-api</artifactId>
2761     <version>5.6.2</version>
2762     <scope>test</scope>
2763 </dependency>
2764
2765 3)pom.xml > right-click > Maven Install
2766 [INFO] BUILD SUCCESS
2767 -만일 ERROR 발생하면 다음과 같이 조치한다.
2768 -SpringDemo5 > right-click > Maven > Update Project
2769 -SpringDemo5가 check되어 있음을 확인하고 OK
2770 -다시 pom.xml > right-click > Maven Install
2771 [INFO] BUILD SUCCESS
2772
2773
2774 11. JUnit 5를 사용한 Test
2775 1)com.example.test > right-click > New > Class
2776 2)Name : HelloBeanJUnitTest
2777
2778 package com.example.test;
2779
2780 import static org.junit.jupiter.api.Assertions.assertEquals;
2781 import static org.junit.jupiter.api.Assertions.assertSame;
2782
2783 import org.junit.jupiter.api.BeforeEach;
2784 import org.junit.jupiter.api.Test;
2785 import org.springframework.context.ApplicationContext;
2786 import org.springframework.context.support.GenericXmlApplicationContext;
2787
2788 import com.example.Hello;
2789
2790 public class HelloBeanJUnitTest {
2791     ApplicationContext context;
2792
2793     @BeforeEach
2794     public void init() {
2795         this.context = new GenericXmlApplicationContext("config/beans.xml");
2796     }
2797
2798     @Test
2799     public void test1(){
2800         Hello hello = (Hello)context.getBean("hello");
2801         assertEquals("Hello Spring", hello.sayHello());
2802         hello.print();
2803     }
2804
2805     @Test
2806     public void test2(){
2807         Hello hello = (Hello)context.getBean("hello");
2808         Hello hello2 = context.getBean("hello", Hello.class);
2809         assertSame(hello, hello2);
2810     }
2811 }
2812
2813
2814 3)HelloBeanJUnitTest.java > right-click > Run As > JUnit Test
2815 -JUnit 창에 Green Bar
2816
2817
```

## 2818 12. Spring TestContext Framework

2819 1)Spring-Test library 설치

2820 2)pom.xml 수정

2821

2822 &lt;dependency&gt;

2823 &lt;groupId&gt;org.springframework&lt;/groupId&gt;

2824 &lt;artifactId&gt;spring-test&lt;/artifactId&gt;

2825 &lt;version&gt;5.2.5.RELEASE&lt;/version&gt;

2826 &lt;scope&gt;test&lt;/scope&gt;

2827 &lt;/dependency&gt;

2828

2829 3)pom.xml &gt; right-click &gt; Maven Install

2830 -만일 Error 발생시 DIDemo3 &gt; right-click &gt; Maven &gt; Update Project... &gt; Ok

2831 -다시 Maven Install 실행

2832

2833 4)Spring-Test를 사용할 DI test class-HelloBeanJUnitSpringTest.java 작성하기

2834 -/src/com.example.test &gt; New &gt; Class

2835 -Name : HelloBeanJUnitSpringTest

2836 -Finish

2837

2838 package com.example.test;

2839

2840 import static org.junit.jupiter.api.Assertions.assertEquals;

2841 import static org.junit.jupiter.api.Assertions.assertSame;

2842

2843 import org.junit.jupiter.api.Test;

2844 import org.junit.jupiter.api.extension.ExtendWith;

2845 import org.springframework.beans.factory.annotation.Autowired;

2846 import org.springframework.context.ApplicationContext;

2847 import org.springframework.test.context.ContextConfiguration;

2848 import org.springframework.test.context.junit.jupiter.SpringExtension;

2849

2850 import com.example.Hello;

2851

2852 @ExtendWith(SpringExtension.class)

2853 //JUnit 5.x에서 사용

2854 @ContextConfiguration(locations="classpath:config/beans.xml")

2855 public class HelloBeanJUnitSpringTest {

2856 @Autowired

2857 ApplicationContext context;

2858

2859 @Test

2860 public void test1(){

2861 Hello hello = (Hello)context.getBean("hello");

2862 assertEquals("Hello Spring", hello.sayHello());

2863 hello.print();

2864 }

2865

2866 @Test

2867 public void test2(){

2868 Hello hello = (Hello)context.getBean("hello");

2869 Hello hello2 = context.getBean("hello", Hello.class);

2870 assertSame(hello, hello2);

2871 }

2872 }

2873

2874 5)right-click &gt; Run As &gt; Junit Test

2875 6)결과 -&gt; Junit View에 초록색 bar

2876 7)만일 해당 객체를 찾을 수 없다는 오류가 계속 발생하면  
2877 -해당 Project > right-click > Build Path > Configure Build Path > Libraries tab  
2878 -spring-test-5.2.5.RELEASE.jar 선택 후 [Remove] 로 삭제  
2879 -Classpath 선택  
2880 -[Add External JARs...] Click  
2881 -Local M2 Repository(e.g C:\Users\사용자아이디  
\.m2\repository\org\springframework\spring-test\5.2.5.RELEASE)에서 직접  
jar(spring-test-5.2.5.RELEASE.jar)를 선택할 것  
2882 -[Order and Export] tab에서 spring-test-5.2.5.RELEASE.jar 선택 후 [Up] button을 클릭  
2883 -해당 Project/src 바로 아래까지 올리고 [Apply and Close] Click  
2884  
2885

2886 13. src/com.example/StringPrinter.java 수정  
2887 package com.example;  
2888  
2889 import org.springframework.stereotype.Component;  
2890  
2891 @Component("stringPrinter")  
2892 public class StringPrinter implements Printer{  
2893 private StringBuffer buffer = new StringBuffer();  
2894 ...  
2895  
2896

2897 14. src/com.example/ConsolePrinter.java 수정  
2898  
2899 package com.example;  
2900  
2901 import org.springframework.stereotype.Component;  
2902  
2903 @Component("consolePrinter")  
2904 public class ConsolePrinter implements Printer{  
2905 ...  
2906  
2907

2908 15. /src/com.example/Hello.java 수정  
2909 package com.example;  
2910  
2911 import org.springframework.beans.factory.annotation.Autowired;  
2912 import org.springframework.beans.factory.annotation.Qualifier;  
2913 import org.springframework.beans.factory.annotation.Value;  
2914 import org.springframework.stereotype.Component;  
2915  
2916 import lombok.NoArgsConstructor;  
2917 import lombok.Setter;  
2918  
2919 @Setter  
2920 @NoArgsConstructor  
2921 @Component  
2922 public class Hello {  
2923 @Value("Spring")  
2924 private String name;  
2925  
2926 @Autowired  
2927 @Qualifier("stringPrinter")  
2928 private Printer printer;  
2929  
2930 public String sayHello(){  
2931 return "Hello " + name;

```
2932     }
2933
2934     public void print(){
2935         this.printer.print(sayHello());
2936     }
2937 }
2938
2939
2940 16. 기존의 설정file과 충돌이 발생하기 때문에 /src/config/beans.xml 삭제
2941
2942
2943 17. 새로운 설정 file 생성
2944 1)src/config > right-click > New > Spring Bean Configuration File
2945 2)File name : annos.xml > Finish
2946 3)Namespace tab > context Check
2947
2948 <?xml version="1.0" encoding="UTF-8"?>
2949 <beans xmlns="http://www.springframework.org/schema/beans"
2950     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
2951     xmlns:context="http://www.springframework.org/schema/context"
2952     xsi:schemaLocation="http://www.springframework.org/schema/beans
2953         http://www.springframework.org/schema/beans/spring-beans.xsd
2954         http://www.springframework.org/schema/context
2955         http://www.springframework.org/schema/context/spring-context-4.3.xsd">
2956
2957     <context:component-scan base-package="com.example" />
2958 </beans>
2959
2960 18. /src/com.example.test/HelloBeanJUnitSpringTest.java 수정하기
2961 package com.example.test;
2962
2963 import static org.junit.jupiter.api.Assertions.assertEquals;
2964 import static org.junit.jupiter.api.Assertions.assertSame;
2965
2966 import org.junit.jupiter.api.Test;
2967 import org.junit.jupiter.api.extension.ExtendWith;
2968 import org.springframework.beans.factory.annotation.Autowired;
2969 import org.springframework.context.ApplicationContext;
2970 import org.springframework.test.context.ContextConfiguration;
2971 import org.springframework.test.context.junit.jupiter.SpringExtension;
2972
2973 import com.example.Hello;
2974
2975 @ExtendWith(SpringExtension.class)
2976 @ContextConfiguration(locations="classpath:config/annos.xml")
2977 public class HelloBeanJUnitSpringTest {
2978     @Autowired
2979     ApplicationContext context;
2980
2981     @Test
2982     public void test1(){
2983         Hello hello = (Hello)context.getBean("hello");
2984         assertEquals("Hello Spring", hello.sayHello());
2985         hello.print();
2986     }
2987
2988     @Test
```

```
2988     public void test2(){
2989         Hello hello = (Hello)context.getBean("hello");
2990         Hello hello2 = context.getBean("hello", Hello.class);
2991         assertSame(hello, hello2);
2992     }
2993 }
2994
2995 1)right-click > Run As > Junit Test
2996 2)결과 -> Junit View에 초록색 bar
2997
2998
2999
3000 -----
3001 Task 15. Lab with JUnit 5 Jupiter
3002 1. In Package Explorer > right-click > New > Java Project
3003     1)Project Name : DIDemo4
3004     2)JRE
3005         -Select [Use default JRE 'jdk-13.0.2' and workspace compiler preferences]
3006     3)Next
3007     4)Uncheck [Create module-info.java file]
3008     5)Finish
3009
3010
3011 2. src > right-click > New > Package
3012     1)Package name : com.example
3013     2)Finish
3014
3015
3016 3. com.example.Student.java, com.example.StudentInfo.java
3017     1)Student.java
3018         package com.example;
3019
3020         import java.util.List;
3021
3022         public class Student {
3023             private String name;
3024             private int age;
3025             private List<String> hobbies;
3026             private double height;
3027             private double weight;
3028         }
3029
3030     2)StudentInfo.java
3031         package com.example;
3032
3033         public class StudentInfo {
3034             private Student student;
3035         }
3036
3037
3038 4. Java Project를 Spring Project로 변환
3039     1)DIDemo4 Project > right-click > Configure > Convert to Maven Project
3040         -Project : /DIDemo4
3041         -Group Id : DIDemo4
3042         -Artifact Id : DIDemo4
3043         -version : 0.0.1-SNAPSHOT
3044         -Packaging : jar
3045         -Finish
```

```
3046
3047 2)DIDemo4 Project > right-click > Spring > Add Spring Project Nature
3048
3049 3)pom.xml file에 Spring Context Dependency 추가하기
3050     <version>0.0.1-SNAPSHOT</version>
3051     <dependencies>
3052         <dependency>
3053             <groupId>org.springframework</groupId>
3054             <artifactId>spring-context</artifactId>
3055             <version>5.2.5.RELEASE</version>
3056         </dependency>
3057     </dependencies>
3058
3059 4)pom.xml > right-click > Run As > Maven install
3060     [INFO] BUILD SUCCESS 확인
3061
3062
3063 5. Lombok library 추가
3064 1)<a href="https://mvnrepository.com/">https://mvnrepository.com/</a>에서 'lombok'으로 검색
3065 2)'Project Lombok' click
3066 3)1.18.12 click
3067 4)dependency copy해서 pom.xml에 붙여넣기
3068
3069     <dependencies>
3070         <dependency>
3071             <groupId>org.springframework</groupId>
3072             <artifactId>spring-context</artifactId>
3073             <version>5.2.5.RELEASE</version>
3074         </dependency>
3075         <!-- <a href="https://mvnrepository.com/artifact/org.projectlombok/lombok">https://mvnrepository.com/artifact/org.projectlombok/lombok</a> -->
3076         <dependency>
3077             <groupId>org.projectlombok</groupId>
3078             <artifactId>lombok</artifactId>
3079             <version>1.18.12</version>
3080             <scope>provided</scope>
3081         </dependency>
3082     </dependencies>
3083
3084 5)pom.xml > right-click > Run As > Maven install
3085     [INFO] BUILD SUCCESS 확인
3086
3087
3088 6. Student.java, StudentInfo.java lombok Annotation 붙이기
3089 1)Student.java
3090
3091     package com.example;
3092
3093     import java.util.List;
3094
3095     import lombok.AllArgsConstructor;
3096     import lombok.Data;
3097     import lombok.NonNull;
3098     import lombok.RequiredArgsConstructor;
3099
3100     @Data
3101     @RequiredArgsConstructor
3102     @AllArgsConstructor
3103     public class Student {
```

```
3104     private @NonNull String name;
3105     private @NonNull int age;
3106     private @NonNull List<String> hobbies;
3107     private double height;
3108     private double weight;
3109 }
3110
3111 2)StudentInfo.java
3112
3113     package com.example;
3114
3115     import lombok.AllArgsConstructor;
3116     import lombok.Setter;
3117
3118     @Setter
3119     @AllArgsConstructor
3120     public class StudentInfo {
3121         private Student student;
3122
3123         public void printInfo(){
3124             if(this.student != null){
3125                 System.out.println("Name : " + this.student.getName());
3126                 System.out.println("Age : " + this.student.getAge());
3127                 System.out.println("Hobbies");
3128                 this.student.getHobbies().forEach(hobby -> System.out.println(hobby));
3129                 System.out.println("Height : " + this.student.getHeight());
3130                 System.out.println("Weight : " + this.student.getWeight());
3131             }
3132         }
3133     }
3134
3135
3136 7. com.example.config package 생성
3137 1)com.example > right-click > New > Package
3138 2)Name : com.example.config
3139 3)Finish
3140
3141
3142 8. com.example.config.ApplicationConfig.java 생성
3143 1)com.example.config > right-click > New > Click
3144 2)Name : ApplicationConfig
3145 3)Finish
3146
3147     package com.example.config;
3148
3149     import java.util.Arrays;
3150     import java.util.List;
3151
3152     import org.springframework.context.annotation.Bean;
3153     import org.springframework.context.annotation.Configuration;
3154
3155     import com.example.Student;
3156     import com.example.StudentInfo;
3157
3158     @Configuration
3159     public class ApplicationConfig {
3160         @Bean
3161         public Student student1() {
```

```
3162     List<String> list = Arrays.asList("독서", "영화감상", "요리");
3163     Student student1 = new Student("한지민", 25, list);
3164     student1.setHeight(165);
3165     student1.setWeight(45);
3166     return student1;
3167 }
3168
3169 @Bean
3170 public StudentInfo studentInfo() {
3171     return new StudentInfo(this.student1());
3172 }
3173 }
3174
3175
```

### 3176 9. com.example.MainClass.java

```
3177
3178 package com.example;
3179
3180 import org.springframework.context.annotation.AnnotationConfigApplicationContext;
3181
3182 import com.example.config.ApplicationConfig;
3183
3184 public class MainClass {
3185     public static void main(String[] args) {
3186         AnnotationConfigApplicationContext context = new
3187             AnnotationConfigApplicationContext(ApplicationConfig.class);
3188         Student student1 = context.getBean("student1", Student.class);
3189         System.out.println(student1);
3190
3191         StudentInfo studentInfo = context.getBean("studentInfo", StudentInfo.class);
3192         studentInfo.setStudent(student1);
3193         studentInfo.printInfo();
3194
3195         context.close();
3196     }
3197 }
3198
```

### 3199 10. Java Application 실행

```
3200 Student(name=한지민, age=25, hobbies=[독서, 영화감상, 요리], height=165.0, weight=45.0)
3201 Name : 한지민
3202 Age : 25
3203 Hobbies
3204 독서
3205 영화감상
3206 요리
3207 Height : 165.0
3208 Weight : 45.0
3209
```

### 3211 11. Student.java 수정

```
3212
3213 package com.example;
3214
3215 import java.util.List;
3216
3217 import org.springframework.beans.factory.annotation.Value;
3218 import org.springframework.stereotype.Component;
```



```
3219
3220 import lombok.Getter;
3221 import lombok.Setter;
3222
3223 @Component
3224 @Setter
3225 @Getter
3226 public class Student {
3227     @Value("한지민")
3228     private String name;
3229     @Value("25")
3230     private int age;
3231     @Value("등산, 게임, 독서")
3232     private List<String> hobbies;
3233     @Value("162.5")
3234     private double height;
3235     @Value("49.2")
3236     private double weight;
3237 }
3238
3239
3240 12. StudentInfo.java 수정
3241
3242 package com.example;
3243
3244 import org.springframework.beans.factory.annotation.Autowired;
3245 import org.springframework.stereotype.Component;
3246
3247 import lombok.NoArgsConstructor;
3248 import lombok.Setter;
3249
3250 @NoArgsConstructor
3251 @Component
3252 public class StudentInfo {
3253     @Setter(onMethod_ = @Autowired)
3254     private Student student;
3255
3256     public void printInfo(){
3257         if(this.student != null){
3258             System.out.println("Name : " + this.student.getName());
3259             System.out.println("Age : " + this.student.getAge());
3260             System.out.println("Hobbies");
3261             this.student.getHobbys().forEach(hobby -> System.out.println(hobby));
3262             System.out.println("Height : " + this.student.getHeight());
3263             System.out.println("Weight : " + this.student.getWeight());
3264         }else {
3265             System.out.println("Null");
3266         }
3267     }
3268 }
3269
3270
3271 13. ApplicationConfig.java 수정
3272
3273 package com.example.config;
3274
3275 import org.springframework.context.annotation.Bean;
3276 import org.springframework.context.annotation.ComponentScan;
```

```
3277 import org.springframework.context.annotation.Configuration;
3278
3279 import com.example.StudentInfo;
3280
3281 @Configuration
3282 @ComponentScan(basePackages = {"com.example"})
3283 public class ApplicationConfig {
3284     @Bean
3285     public StudentInfo studentInfo() {
3286         return new StudentInfo();
3287     }
3288 }
3289
3290
3291 14. MainClass.java 수정
3292
3293 package com.example;
3294
3295 import org.springframework.context.annotation.AnnotationConfigApplicationContext;
3296
3297 import com.example.config.ApplicationConfig;
3298
3299 public class MainClass {
3300     public static void main(String[] args) {
3301         AnnotationConfigApplicationContext context = new
3302             AnnotationConfigApplicationContext(ApplicationConfig.class);
3303         StudentInfo info = context.getBean("studentInfo", StudentInfo.class);
3304         info.printInfo();
3305         context.close();
3306     }
3307 }
3308
3309 15. MainClass 실행
3310
3311 Name : 한지민
3312 Age : 25
3313 Hobbies
3314 등산, 게임, 독서
3315 Height : 162.5
3316 Weight : 49.2
```