

```
1 HOL : Spring Object Lifecycle
2 -----
3 Task 1. Lab
4 1. In Package Explorer > right-click > New > Java Project
5 1)Project Name : SpringLifecycle
6
7 2. src > right-click > New > Package
8 2)Package name : com.example
9
10 3. POJO 객체 생성
11 1)Class name : com.example.Student
12 package com.example;
13
14 import java.util.ArrayList;
15
16 public class Student {
17     private String name;
18     private int age;
19     private ArrayList<String> hobbies;
20     private double height;
21     private double weight;
22
23     public Student(String name, int age, ArrayList<String> hobbies) {
24         this.name = name;
25         this.age = age;
26         this.hobbies = hobbies;
27     }
28
29     public void setName(String name) {
30         this.name = name;
31     }
32
33     public void setAge(int age) {
34         this.age = age;
35     }
36
37     public void setHobbys(ArrayList<String> hobbies) {
38         this.hobbies = hobbies;
39     }
40
41     public void setHeight(double height) {
42         this.height = height;
43     }
44
45     public void setWeight(double weight) {
46         this.weight = weight;
47     }
48
49     @Override
50     public String toString() {
51         return String.format("Student [name=%s, age=%s, hobbies=%s, height=%s, weight=%s]", name,
            age, hobbies, height,
```

```
52         weight);
53     }
54 }
```

55

56 4. Java Project를 Spring Project로 변환

57 1)SpringLifecycle Project > right-click > Configure > Convert to Maven Project

58 -Project : /SpringLifecycle

59 -Group Id : SpringLifecycle

60 -Artifact Id : SpringLifecycle

61 -version : 0.0.1-SNAPSHOT

62 -Packaging : jar

63 -Finish

64

65 2)SpringLifecycle Project > right-click > Spring > Add Spring Project Nature

66

67 3)pom.xml 파일에 Spring Context Dependency 추가하기

68 <version>0.0.1-SNAPSHOT</version>

69 <dependencies>

70 <dependency>

71 <groupId>org.springframework</groupId>

72 <artifactId>spring-context</artifactId>

73 <version>5.2.0.RELEASE</version>

74 </dependency>

75 </dependencies>

76

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

78 [INFO] BUILD SUCCESS 확인

79

80 5. SpringLifecycle/resources folder 생성

81 1)SpringLifecycle project > right-click > Build Path > Configure Build Path

82 2)Source Tab > Add Folder

83 3)SpringLifecycle click

84 4)Create New Folder > Folder name : resources > Finish > OK

85 5)SpringLifecycle/resources(new) 확인

86 6)Apply and Close

87

88 6. Bean Configuration XML 작성

89 1)SpringLifecycle/resources > right-click > New > Other > Spring > Spring Bean Configuration File

90 2)File name : applicationContext.xml > Finish

91

92 <?xml version="1.0" encoding="UTF-8"?>

93 <beans xmlns="<http://www.springframework.org/schema/beans>"94 xmlns:xsi="<http://www.w3.org/2001/XMLSchema-instance>"95 xsi:schemaLocation="<http://www.springframework.org/schema/beans><http://www.springframework.org/schema/beans/spring-beans.xsd>">

96

97 <bean id="student1" class="com.example.Student">

98 <constructor-arg value="한지민" />

99 <constructor-arg value="25" />

100 <constructor-arg>

101 <list>

102 <value>독서</value>

```
103         <value>영화감상</value>
104         <value>요리</value>
105     </list>
106 </constructor-arg>
107 <property name="height" value="165" />
108 <property name="weight">
109     <value>45</value>
110 </property>
111 </bean>
112 </beans>
113
```

114 7. com.example.MainClass.java

```
115 package com.example;
116
117 import org.springframework.context.support.GenericXmlApplicationContext;
118
119 public class MainClass {
120     public static void main(String[] args) {
121         GenericXmlApplicationContext context = new GenericXmlApplicationContext();
122
123         context.load("classpath:applicationContext.xml");
124         context.refresh();
125
126         Student student1 = context.getBean("student1", Student.class);
127         System.out.println(student1);
128
129         context.close();
130     }
131 }
132
```

133 8. 실행

134 1)MainClass > right-click > Run As > Java Application

135 9. 결과

137 Student [name=한지민, age=25, hobbies=[독서, 영화감상, 요리], height=165.0,weight=45.0]

138

139 10. Java Annotation 방식 사용하기

```
140 1)com.example.ApplicationConfig.java
141 package com.example;
142
143 import java.util.ArrayList;
144
145 import org.springframework.context.annotation.Bean;
146 import org.springframework.context.annotation.Configuration;
147
148 @Configuration
149 public class ApplicationConfig {
150
151     @Bean
152     public Student student1() {
153         ArrayList<String> hobbies = new ArrayList<String>();
154         hobbies.add("독서");
155     }
156 }
```

```
155     hobbies.add("영화감상");
156     hobbies.add("요리");
157     Student student = new Student("한지민", 25, hobbies);
158     student.setHeight(165.0);
159     student.setWeight(45.0);
160     return student;
161 }
162 }
163
164 2)com.example.MainClass2.java
165     package com.example;
166
167     import org.springframework.context.annotation.AnnotationConfigApplicationContext;
168
169     public class MainClass2 {
170     public static void main(String[] args) {
171         AnnotationConfigApplicationContext ctx = new
            AnnotationConfigApplicationContext(ApplicationConfig.class);
172         Student student1 = ctx.getBean("student1", Student.class);
173         System.out.println(student1);
174
175         Student student2 = ctx.getBean("student1", Student.class);
176         System.out.println(student1 == student2);
177         ctx.close();
178     }
179 }
180
181 11. 실행 결과
182     Student [name=한지민, age=25, hobbies=[독서, 영화감상, 요리], height=165.0, weight=45.0]
183
184 12. junit을 사용한 DI test class(HelloBeanJUnitTest.java) 작성
185     1)/src < right-click > New > Package
186     2)Package name : com.example.test
187     3)/src/com.example.test/HelloBeanJUnitTest.java
188     package com.example.test;
189     import static org.junit.Assert.assertEquals;
190     import org.junit.Before;
191     import org.junit.Test;
192     import org.springframework.context.annotation.AnnotationConfigApplicationContext;
193     import com.example.ApplicationConfig;
194     import com.example.Student;
195
196     public class HelloBeanJUnitTest {
197         AnnotationConfigApplicationContext context;
198
199         @Before
200         public void init() {
201             this.context = new AnnotationConfigApplicationContext(ApplicationConfig.class);
202         }
203
204         @Test
205         public void test1() {
```

```
206         Student student1 = this.context.getBean("student1", Student.class);
207         assertEquals(3, student1.getSize());
208     }
209 }
```

210

211 4)right-click > Run As > Junit Test

212 5)결과 -> Junit View에 초록색 bar

213

214

215 -----

216 Task 2. Lab

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

218 1)Project Name : SpringLifecycle1

219

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

221 1)Package name : com.example

222

223 3. POJO 객체 생성

224 1)InitializingBean, DisposableBean interface 이용하기

225 2)com.example.Student.java

226 package com.example;

227

228 import java.util.ArrayList;

229

230 import org.springframework.beans.factory.DisposableBean;

231 import org.springframework.beans.factory.InitializingBean;

232

233 public class Student implements InitializingBean, DisposableBean{

234 private String name;

235 private int age;

236

237 public Student(String name, int age) {

238 this.name = name;

239 this.age = age;

240 }

241

242 @Override

243 public String toString() {

244 return String.format("Student [name=%s, age=%s]", name, age);

245 }

246

247 @Override

248 public void destroy() throws Exception {

249 System.out.println("방금 bean이 소멸됐습니다.");

250 }

251

252 @Override

253 public void afterPropertiesSet() throws Exception {

254 System.out.println("방금 bean이 생성됐습니다.");

255 }

256

257 4. Java Project를 Spring Project로 변환

```
258 1)SpringLifecycle1 Project > right-click > Configure > Convert to Maven Project
259 -Project : /SpringLifecycle1
260 -Group Id : SpringLifecycle1
261 -Artifact Id : SpringLifecycle1
262 -version : 0.0.1-SNAPSHOT
263 -Packaging : jar
264 -Finish
265
266 2)SpringLifecycle1 Project > right-click > Spring > Add Spring Project Nature
267
268 3)pom.xml 파일에 Spring Context Dependency 추가하기
269 <version>0.0.1-SNAPSHOT</version>
270 <dependencies>
271 <dependency>
272 <groupId>org.springframework</groupId>
273 <artifactId>spring-context</artifactId>
274 <version>5.2.0.RELEASE</version>
275 </dependency>
276 </dependencies>
277
278 4)pom.xml > right-click > Run As > Maven install
279 [INFO] BUILD SUCCESS 확인
280
281 5. SpringLifecycle1/resources folder 생성
282 1)SpringLifecycle1 project > right-click > Build Path > Configure Build Path
283 2)Source Tab > Add Folder
284 3)SpringLifecycle1 click
285 4>Create New Folder > Folder name : resources > Finish > OK
286 5)SpringLifecycle1/resources(new) 확인
287 6)Apply and Close
288
289 6. Bean Configuration XML 작성
290 1)SpringLifecycle1/resources > right-click > New > Other > Spring > Spring Bean Configuration File
291 2)File name : applicationContext.xml > Finish
292
293 <?xml version="1.0" encoding="UTF-8"?>
294 <beans xmlns="http://www.springframework.org/schema/beans"
295 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
296 xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans.xsd">
297
298 <bean id="student" class="com.example.Student">
299 <constructor-arg value="한지민" />
300 <constructor-arg value="25" />
301 </bean>
302
303 </beans>
304
305 3)com.example.MainClass.java
306
307 package com.example;
308 import org.springframework.context.support.GenericXmlApplicationContext;
```

```
309
310     public class MainClass {
311         public static void main(String[] args) {
312             GenericXmlApplicationContext context = new GenericXmlApplicationContext();
313             context.load("classpath:applicationContext.xml");
314             context.refresh();
315
316             Student student = context.getBean("student", Student.class);
317             System.out.println(student);
318             context.close();
319         }
320     }
```

322 7. 실행

```
323     1)MainClass > right-click > Run As > Java Application
324         방금 bean이 생성됐습니다.
325         Student [name=한지민, age=25]
326         방금 bean이 소멸됐습니다.
```

328 8. @PostConstruct, @PreDestroy 이용하기

```
329     1)com.example.Student2.java
330         package com.example;
331
332         import javax.annotation.PostConstruct;
333         import javax.annotation.PreDestroy;
334
335         public class Student2 {
336             private String name;
337             private int age;
338
339             public Student2(String name, int age) {
340                 this.name = name;
341                 this.age = age;
342             }
343
344             @Override
345             public String toString() {
346                 return String.format("Student [name=%s, age=%s]", name, age);
347             }
348
349             @PostConstruct //bean이 생성단계에서 해야할 일 기술
350             public void initTest(){
351                 System.out.println("방금 객체가 생성됐습니다.");
352             }
353
354             @PreDestroy //bean이 소멸할 때 해야할 일 기술
355             public void destroyTest(){
356                 System.out.println("방금 객체가 소멸됐습니다.");
357             }
358         }
```

360 9. resources/applicationContext.xml 수정하기

```
361 1)Namespaces Tab > context check 할것
362 <?xml version="1.0" encoding="UTF-8"?>
363 <beans xmlns="http://www.springframework.org/schema/beans"
364   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
365   xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans.xsd">
366
367   <!-- 첫번째 방법 -->
368   <context:annotation-config/>
369   <bean id="student2" class="com.example.Student2">
370     <constructor-arg value="설운도" />
371     <constructor-arg value="50" />
372   </bean>
373
374   <!-- 두번째 방법-->
375   <bean class="org.springframework.context.annotation.CommonAnnotationBeanPostProcessor" />
376
377   <bean id="student2" class="com.example.Student2">
378     <constructor-arg value="설운도" />
379     <constructor-arg value="50" />
380   </bean>
381
382   <!-- 세번째 방법 -->
383   <bean id="student2" class="com.example.Student2" init-method="initTest"
destroy-method="destroyTest">
384     <constructor-arg value="설운도" />
385     <constructor-arg value="50" />
386   </bean>
387 </beans>
388
389 10. MainClass 수정
390 1)com.example.MainClass.java
391
392   package com.example;
393
394   import org.springframework.context.support.GenericXmlApplicationContext;
395
396   public class MainClass {
397     public static void main(String[] args) {
398       GenericXmlApplicationContext context = new GenericXmlApplicationContext();
399       context.load("classpath:applicationContext.xml");
400       context.refresh();
401
402       Student2 student2 = context.getBean("student2", Student2.class);
403       System.out.println(student2);
404       context.close();
405     }
406   }
407
408 11. 실행
409 1)MainClass > right-click > Run As > Java Application
410   방금 bean이 생성됐습니다.
```



```
411      Student [name=설운도, age=50]
412      방금 bean이 소멸됐습니다.
413
414
415 -----
416 Task 3. Lab
417 1. In Package Explorer > right-click > New > Java Project
418    1)Project Name : SpringScopeDemo
419
420 2. src > right-click > New > Package
421    1)Package name : com.example
422
423 3. com.example.Student.java
424
425     package com.example;
426
427     public class Student{
428         private String name;
429         private int age;
430
431         public Student(String name, int age) {
432             this.name = name;
433             this.age = age;
434         }
435
436         public void setName(String name) {
437             this.name = name;
438         }
439
440         public void setAge(int age) {
441             this.age = age;
442         }
443
444         @Override
445         public String toString() {
446             return String.format("Student [name=%s, age=%s]", name, age);
447         }
448     }
449
450 4. Java Project를 Spring Project로 변환
451    1)SpringScopeDemo Project > right-click > Configure > Convert to Maven Project
452      -Project : /SpringScopeDemo
453      -Group Id : SpringScopeDemo
454      -Artifact Id : SpringScopeDemo
455      -version : 0.0.1-SNAPSHOT
456      -Packaging : jar
457      -Finish
458
459    2)SpringScopeDemo Project > right-click > Spring > Add Spring Project Nature
460
461    3)pom.xml 파일에 Spring Context Dependency 추가하기
462      <version>0.0.1-SNAPSHOT</version>
```

```
463     <dependencies>
464         <dependency>
465             <groupId>org.springframework</groupId>
466             <artifactId>spring-context</artifactId>
467             <version>5.2.0.RELEASE</version>
468         </dependency>
469     </dependencies>
470
471 4)pom.xml > right-click > Run As > Maven install
472 [INFO] BUILD SUCCESS 확인
473
474 5. SpringScopeDemo/resources folder 생성
475 1)SpringScopeDemo project > right-click > Build Path > Configure Build Path
476 2)Source Tab > Add Folder
477 3)SpringScopeDemo click
478 4>Create New Folder > Folder name : resources > Finish > OK
479 5)SpringScopeDemo/resources(new) 확인
480 6)Apply and Close
481
482 6. Bean Configuration XML 작성
483 1)SpringScopeDemo/resources > right-click > New > Other > Spring > Spring Bean Configuration File
484 2)File name : applicationContext.xml > Finish
485
486 <?xml version="1.0" encoding="UTF-8"?>
487 <beans xmlns="http://www.springframework.org/schema/beans"
488     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
489     xsi:schemaLocation="http://www.springframework.org/schema/beans
490         http://www.springframework.org/schema/beans/spring-beans.xsd">
491
492     <bean id="student" class="com.example.Student" scope="singleton">
493         <constructor-arg value="한지민" />
494         <constructor-arg value="25" />
495     </bean>
496 </beans>
497
498 7. com.example.MainClass.java
499
500     package com.example;
501
502     import org.springframework.context.support.AbstractApplicationContext;
503     import org.springframework.context.support.GenericXmlApplicationContext;
504
505     public class MainClass {
506         public static void main(String[] args) {
507             AbstractApplicationContext context = new
508                 GenericXmlApplicationContext("classpath:applicationContext.xml");
509
510             Student student = context.getBean("student", Student.class);
511             System.out.println(student);
512             System.out.println("-----");
```

```
513     Student student1 = context.getBean("student", Student.class);
514     student1.setName("설운도");
515     student1.setAge(55);
516     System.out.println(student1);
517     System.out.println("-----");
518
519     if(student.equals(student1)) System.out.println("Equals"); //Print Equals
520     else System.out.println("Different");
521     context.close();
522 }
523 }
524
525 8. Java Application 실행 결과
526 Student [name=한지민, age=25]
527 -----
528 Student [name=설운도, age=55]
529 -----
530 Equals
```