```
HOL : Spring Object Lifecycle
------------------------------
Task 1. Lab
1. In Package Explorer > right-click > New > Java Project
  1)Project Name : SpringLifecycle
  2)JRE
    -Select [Use default JRE 'jdk-13.0.2' and workspace compiler preferences]
  3)Next
  4)Uncheck [Create module-info.java file]
  5)Finish


2. src > right-click > New > Package
  1)Name : com.example
  2)Finish


3. POJO 객체 생성
  1)src/com.example > New > Class
  2)Name : Student

    package com.example;

    import java.util.List;

    public class Student {
      private String name;
      private int age;
      private List<String> hobbys;
      private double height;
      private double weight;
    }


4. Java Project를 Spring Project로 변환
  1)SpringLifecycle Project > right-click > Configure > Convert to Maven Project
    -Project : /SpringLifecycle
    -Group Id : SpringLifecycle
    -Artifact Id : SpringLifecycle
    -version : 0.0.1-SNAPSHOT
    -Packaging : jar
    -Finish

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

  3)pom.xml 파일에 Spring Context Dependency 추가하기
    <version>0.0.1-SNAPSHOT</version>
    <dependencies>
      <dependency>
        <groupId>org.springframework</groupId>
        <artifactId>spring-context</artifactId>
        <version>5.2.5.RELEASE</version>
      </dependency>
    </dependencies>

  4)pom.xml > right-click > Run As > Maven install
    [INFO] BUILD SUCCESS 확인

```

5. Lombok library 추가
  1)https://mvnrepository.com/에서 'lombok'으로 검색
  2)'Project Lombok' click
  3)1.18.12 click
  4)depency copy해서 pom.xml에 붙여넣기

```
  <dependencies>
    <dependency>
      <groupId>org.springframework</groupId>
      <artifactId>spring-context</artifactId>
      <version>5.2.5.RELEASE</version>
    </dependency>
    <!-- https://mvnrepository.com/artifact/org.projectlombok/lombok -->
    <dependency>
      <groupId>org.projectlombok</groupId>
      <artifactId>lombok</artifactId>
      <version>1.18.12</version>
      <scope>provided</scope>
    </dependency>
  </dependencies>
```

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

6. Student.java 수정
  1)Student.java

```
  package com.example;

  import java.util.List;

  import lombok.NonNull;
  import lombok.RequiredArgsConstructor;
  import lombok.Setter;
  import lombok.ToString;

  @RequiredArgsConstructor
  @ToString
  public class Student {
    @NonNull private String name;
    @NonNull private Integer age;
    @NonNull private List<String> hobbys;
    @Setter private double height;
    @Setter private double weight;
  }
```

7. SpringLifecycle/resources folder 생성
  1)SpringLifecycle project > right-click > New > Source Folder
  2)Folder name : resources
  3)Finish

8. Bean Configuration XML 작성
  1)SpringLifecycle/resources > right-click > New > Spring Bean Configuration File
  2)File name : applicationContext.xml

```
117   3)Finish
118
119   <?xml version="1.0" encoding="UTF-8"?>
120   <beans xmlns="http://www.springframework.org/schema/beans"
121     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
122     xsi:schemaLocation="http://www.springframework.org/schema/beans
      http://www.springframework.org/schema/beans/spring-beans.xsd">
123
124     <bean id="student1" class="com.example.Student">
125       <constructor-arg value="한지민" />
126       <constructor-arg value="25" />
127       <constructor-arg>
128         <list>
129           <value>독서</value>
130           <value>영화감상</value>
131           <value>요리</value>
132         </list>
133       </constructor-arg>
134       <property name="height" value="165" />
135       <property name="weight">
136         <value>45</value>
137       </property>
138     </bean>
139   </beans>
140
141
142 9. com.example.MainClass.java
143   package com.example;
144
145   import org.springframework.context.support.GenericXmlApplicationContext;
146
147   public class MainClass {
148     public static void main(String[] args) {
149       GenericXmlApplicationContext context = new GenericXmlApplicationContext();
150
151       context.load("classpath:applicationContext.xml");
152       context.refresh();
153
154       Student student1 = context.getBean("student1", Student.class);
155       System.out.println(student1);
156
157       context.close();
158     }
159   }
160
161 10. 실행
162   1)MainClass > right-click > Run As > Java Application
163     Student(name=한지민, age=25, hobbys=[독서, 영화감상, 요리], height=165.0, weight=45.0)
164
165
166 11. Java Annotation 방식 사용하기
167   1)src/com.example > right-click > New > Class
168   2)Name : ApplicationConfig
169
170     package com.example;
171
172     import java.util.Arrays;
173     import java.util.List;
```

```
174
175     import org.springframework.context.annotation.Bean;
176     import org.springframework.context.annotation.Configuration;
177
178     @Configuration
179     public class ApplicationConfig {
180       @Bean
181       public Student student1() {
182         List<String> list = Arrays.asList("독서", "영화감상", "요리");
183         Student student = new Student("한지민", 25, list);
184         student.setHeight(165.0);
185         student.setWeight(45.0);
186         return student;
187       }
188     }
189
190   3)com.example.MainClass2 생성
191     -src/com.example > right-click > New > Class
192     -Name : MainClass2
193
194       package com.example;
195
196       import org.springframework.context.annotation.AnnotationConfigApplicationContext;
197
198       public class MainClass2 {
199         public static void main(String[] args) {
200           AnnotationConfigApplicationContext ctx = new
              AnnotationConfigApplicationContext(ApplicationConfig.class);
201           Student student1 =  ctx.getBean("student1", Student.class);
202           System.out.println(student1);
203
204           Student student2 =  ctx.getBean("student1", Student.class);
205           System.out.println(student1 == student2);
206           ctx.close();
207         }
208       }
209
210   4)실행 결과
211     Student(name=한지민, age=25, hobbys=[독서, 영화감상, 요리], height=165.0, weight=45.0)
212     true
213
214
215 12. @Component 방식으로 Student 객체 변경
216   1)Student.java
217
218     package com.example;
219
220     import java.util.List;
221
222     import org.springframework.beans.factory.annotation.Value;
223     import org.springframework.stereotype.Component;
224
225     import lombok.Getter;
226     import lombok.ToString;
227
228     @ToString
229     @Getter
230     @Component
```

```
231     public class Student {
232       @Value("박지민")
233       private String name;
234       @Value("35")
235       private Integer age;
236       @Value("등산, 게임, 독서")
237       private List<String> hobbys;
238       @Value("162.5")
239       private double height;
240       @Value("49.2")
241       private double weight;
242     }
243
244   2)com.example.ApplicationConfig 변경
245
246     package com.example;
247
248     import org.springframework.context.annotation.ComponentScan;
249     import org.springframework.context.annotation.Configuration;
250
251     @Configuration
252     @ComponentScan(basePackages = {"com.example"})
253     public class ApplicationConfig {}
254
255   3)resources/applicationContext.xml 삭제
256
257
258 13. JUnit 5를 사용한 DI test class(JUnit5Test.java) 작성
259   1)src > right-click > New > Package
260   2)Name : com.example.test
261   3)Finish
262
263   4)com.example.test > right-click > New > JUnit Test Case
264   5)Select [New JUnit Jupiter test]
265   6)Name : JUnit5Test
266   7)Finish
267   8)Select [Perform the following action: Add Juni 5 library to the build path
268   9)OK
269
270     package com.example.test;
271
272     import static org.junit.jupiter.api.Assertions.assertEquals;
273
274     import org.junit.jupiter.api.BeforeEach;
275     import org.junit.jupiter.api.Test;
276     import org.springframework.context.annotation.AnnotationConfigApplicationContext;
277
278     import com.example.ApplicationConfig;
279     import com.example.Student;
280
281     class JUnit5Test {
282       private AnnotationConfigApplicationContext context;
283
284       @BeforeEach
285       public void init() {
286         this.context = new AnnotationConfigApplicationContext(ApplicationConfig.class);
287       }
288
```

```
289         @Test
290         public void test1() {
291           Student student = this.context.getBean("student", Student.class);
292           assertEquals("박지민", student.getName());
293         }
294       }
295
296     4)right-click > Run As > JUnit Test
297     5)결과 -> Junit View에 초록색 bar
298
299
300   ---------------------------------------
301   Task 2. Lab
302   1. In Package Explorer > right-click > New > Java Project
303     1)Project Name : SpringLifecycle1
304     2)JRE
305       -Select [Use default JRE 'jdk-13.0.2' and workspace compiler preferences]
306     3)Next
307     4)Uncheck [Create module-info.java file]
308     5)Finish
309
310
311   2. src > right-click > New > Package
312     1)Package name : com.example
313
314
315   3. POJO 객체 생성
316     1)com.example > right-click > New > Class
317     2)Name : Student.java
318
319       package com.example;
320
321       public class Student{
322         private String name;
323         private int age;
324       }
325
326
327   4. Java Project를 Spring Project로 변환
328     1)SpringLifecycle1 Project > right-click > Configure > Convert to Maven Project
329       -Project : /SpringLifecycle1
330       -Group Id : SpringLifecycle1
331       -Artifact Id : SpringLifecycle1
332       -version : 0.0.1-SNAPSHOT
333       -Packaging : jar
334       -Finish
335
336     2)SpringLifecycle1 Project > right-click > Spring > Add Spring Project Nature
337
338     3)pom.xml 파일에 Spring Context Dependency 추가하기
339       <version>0.0.1-SNAPSHOT</version>
340       <dependencies>
341         <dependency>
342           <groupId>org.springframework</groupId>
343           <artifactId>spring-context</artifactId>
344           <version>5.2.5.RELEASE</version>
345         </dependency>
346       </dependencies>
```

4)pom.xml > right-click > Run As > Maven install
[INFO] BUILD SUCCESS 확인

5. Lombok library 추가
1)https://mvnrepository.com/에서 'lombok'으로 검색
2)'Project Lombok' click
3)1.18.12 click
4)depency copy해서 pom.xml에 붙여넣기

```
<dependencies>
  <dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-context</artifactId>
    <version>5.2.5.RELEASE</version>
  </dependency>
  <!-- https://mvnrepository.com/artifact/org.projectlombok/lombok -->
  <dependency>
    <groupId>org.projectlombok</groupId>
    <artifactId>lombok</artifactId>
    <version>1.18.12</version>
    <scope>provided</scope>
  </dependency>
</dependencies>
```

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

6. Student.java lombok Annotation 붙이고, InitializingBean, DisposableBean interface 구현하기

```
package com.example;

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

import lombok.AllArgsConstructor;
import lombok.ToString;

@AllArgsConstructor
@ToString
public class Student implements InitializingBean, DisposableBean{
  private String name;
  private int age;

  @Override
  public void destroy() throws Exception {
    System.out.println("방금 Bean이 소멸됐습니다.");
  }
  @Override
  public void afterPropertiesSet() throws Exception {
    System.out.println("방금 Bean이 생성됐습니다.");
  }
}
```

7. SpringLifecycle1/resources folder 생성

1)SpringLifecycle1 project > right-click > New > Source Folder
2)Folder name : resources
3)Finish

8. Bean Configuration XML 작성
1)SpringLifecycle1/resources > right-click > New > Spring Bean Configuration File
2)File name : applicationContext.xml
3)Finish

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.springframework.org/schema/beans
    http://www.springframework.org/schema/beans/spring-beans.xsd">

  <bean id="student" class="com.example.Student">
    <constructor-arg value="한지민" />
    <constructor-arg value="25" />
  </bean>

</beans>
```

9. MainClass 작성
1)com.example > right-click > New > Class
2)Name : MainClass
3)Finish

```
package com.example;

import org.springframework.context.support.GenericXmlApplicationContext;

public class MainClass {
  public static void main(String[] args) {
    GenericXmlApplicationContext context = new GenericXmlApplicationContext();
    context.load("classpath:applicationContext.xml");
    context.refresh();

    Student student = context.getBean("student", Student.class);
    System.out.println(student);
    context.close();
  }
}
```

10. 실행
1)MainClass > right-click > Run As > Java Application
방금 Bean이 생성됐습니다.
Student (name=한지민, age=25)
방금 Bean이 소멸됐습니다.

11. @PostConstruct, @PreDestroy 이용하기
1)https://mvnrepository.com에서 'javax annotation'으로 검색
2)[Javax Annoation API] click
3)1.3.2 click
4)dependency copy하여 pom.xml에 paste

```
462
463       <!-- https://mvnrepository.com/artifact/javax.annotation/javax.annotation-api -->
464       <dependency>
465          <groupId>javax.annotation</groupId>
466          <artifactId>javax.annotation-api</artifactId>
467          <version>1.3.2</version>
468       </dependency>
469
470    5)pom.xml > right-click > Run As > Maven install
471      [INFO] BUILD SUCCESS 확인
472
473    6)Student2 class 생성하기
474      -com.example > right-click > New > Class
475      -Name : Student2
476
477         package com.example;
478
479         import javax.annotation.PostConstruct;
480         import javax.annotation.PreDestroy;
481
482         import lombok.AllArgsConstructor;
483         import lombok.ToString;
484
485         @AllArgsConstructor
486         @ToString
487         public class Student2 {
488           private String name;
489           private int age;
490
491           @PostConstruct // Bean이 생성단계에서 해야할 일 기술
492           public void initTest() {
493             System.out.println("방금 객체가 생성됐습니다.");
494           }
495
496           @PreDestroy // Bean이 소멸할 때 해야할 일 기술
497           public void destroyTest() {
498             System.out.println("방금 객체가 소멸됐습니다.");
499           }
500         }
501
502
503  12. resources/applicationContext.xml 수정하기
504    1)Namespaces Tab click
505    2)[context] check
506    <?xml version="1.0" encoding="UTF-8"?>
507    <beans xmlns="http://www.springframework.org/schema/beans"
508      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
509      xsi:schemaLocation="http://www.springframework.org/schema/beans
         http://www.springframework.org/schema/beans/spring-beans.xsd">
510
511      <!-- 첫번째 방법 -->
512      <context:annotation-config/>
513      <bean id="student2" class="com.example.Student2">
514        <constructor-arg value="한지민" />
515        <constructor-arg value="25" />
516      </bean>
517
518      <!-- 두번째 방법-->
```

```
519     <bean
        class="org.springframework.context.annotation.CommonAnnotationBeanPostProcessor" />
520
521     <bean id="student2" class="com.example.Student2">
522       <constructor-arg value="한지민" />
523       <constructor-arg value="25" />
524     </bean>
525
526     <!-- 세번째 방법 -->
527     <bean id="student2" class="com.example.Student2" init-method="initTest"
        destroy-method="destroyTest">
528       <constructor-arg value="한지민" />
529       <constructor-arg value="25" />
530     </bean>
531   </beans>
532
533
534 13. MainClass 수정
535   1)com.example.MainClass.java
536
537     package com.example;
538
539     import org.springframework.context.support.GenericXmlApplicationContext;
540
541     public class MainClass {
542       public static void main(String[] args) {
543         GenericXmlApplicationContext context = new GenericXmlApplicationContext();
544         context.load("classpath:applicationContext.xml");
545         context.refresh();
546
547         Student2 student2 = context.getBean("student2", Student2.class);
548         System.out.println(student2);
549         context.close();
550       }
551     }
552
553   2)실행
554     -MainClass > right-click > Run As > Java Application
555       방금 객체가 생성됐습니다.
556       Student2(name=한지민, age=25)
557       방금 객체가 소멸됐습니다.
558
559
560
561 --------------------------------
562 Task 3. Lab
563 1. In Package Explorer > right-click > New > Java Project
564   1)Project Name : SpringScopeDemo
565   2)JRE
566     -Select [Use default JRE 'jdk-13.0.2' and workspace compiler preferences]
567   3)Next
568   4)Uncheck [Create module-info.java file]
569   5)Finish
570
571 2. src > right-click > New > Package
572   1)Name : com.example
573   2)Finish
574
```

3. com.example.Student class 생성
  1)com.example > right-click > New > Class
  2)Name : Student
  3)Finish

```
package com.example;

public class Student{
  private String name;
  private int age;
}
```

4. Java Project를 Spring Project로 변환
  1)SpringScopeDemo Project > right-click > Configure > Convert to Maven Project
    -Project : /SpringScopeDemo
    -Group Id : SpringScopeDemo
    -Artifact Id : SpringScopeDemo
    -version : 0.0.1-SNAPSHOT
    -Packaging : jar
    -Finish

  2)SpringScopeDemo Project > right-click > Spring > Add Spring Project Nature

  3)pom.xml 파일에 Spring Context Dependency 추가하기

```
<version>0.0.1-SNAPSHOT</version>
<dependencies>
  <dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-context</artifactId>
    <version>5.2.5.RELEASE</version>
  </dependency>
</dependencies>
```

  4)pom.xml > right-click > Run As > Maven install
    [INFO] BUILD SUCCESS 확인

5. Lombok library 추가
  1)https://mvnrepository.com/에서 'lombok'으로 검색
  2)'Project Lombok' click
  3)1.18.12 click
  4)depency copy해서 pom.xml에 붙여넣기

```
<dependencies>
  <dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-context</artifactId>
    <version>5.2.5.RELEASE</version>
  </dependency>
  <!-- https://mvnrepository.com/artifact/org.projectlombok/lombok -->
  <dependency>
    <groupId>org.projectlombok</groupId>
    <artifactId>lombok</artifactId>
    <version>1.18.12</version>
    <scope>provided</scope>
  </dependency>
```

```
633       </dependencies>
634
635     5)pom.xml > right-click > Run As > Maven install
636       [INFO] BUILD SUCCESS 확인
637
638
639   6. Student.java lombok Annotation 붙이기
640
641     package com.example;
642
643     import lombok.AllArgsConstructor;
644     import lombok.Setter;
645     import lombok.ToString;
646
647     @AllArgsConstructor
648     @Setter
649     @ToString
650     public class Student {
651       private String name;
652       private int age;
653     }
654
655
656   7. SpringLifecycle1/resources folder 생성
657     1)SpringLifecycle1 project > right-click > New > Source Folder
658     2)Folder name : resources
659     3)Finish
660
661
662   8. Bean Configuration XML 작성
663     1)SpringLifecycle1/resources > right-click > New > Spring Bean Configuration File
664     2)File name : applicationContext.xml
665     3)Finish
666
667       <?xml version="1.0" encoding="UTF-8"?>
668       <beans xmlns="http://www.springframework.org/schema/beans"
669           xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
670           xsi:schemaLocation="http://www.springframework.org/schema/beans
          http://www.springframework.org/schema/beans/spring-beans.xsd">
671
672         <bean id="student" class="com.example.Student" scope="singleton">
673           <constructor-arg value="한지민" />
674           <constructor-arg value="25" />
675         </bean>
676
677       </beans>
678
679
680   9. MainClass 생성하기
681     1)com.example > right-click > New > Class
682     2)Name : MainClass
683
684       package com.example;
685
686       import org.springframework.context.support.AbstractApplicationContext;
687       import org.springframework.context.support.GenericXmlApplicationContext;
688
689       public class MainClass {
```

```
690         public static void main(String[] args) {
691           AbstractApplicationContext context = new
              GenericXmlApplicationContext("classpath:applicationContext.xml");
692
693           Student student = context.getBean("student", Student.class);
694           System.out.println(student);
695           System.out.println("--------------------------------");
696
697           Student student1 = context.getBean("student", Student.class);
698           student1.setName("설운도");
699           student1.setAge(55);
700           System.out.println(student1);
701           System.out.println("--------------------------------");
702
703           if(student.equals(student1)) System.out.println("Equals");  //Print Equals
704           else System.out.println("Different");
705           context.close();
706         }
707       }
708
709
710   10. Java Application 실행 결과
711     Student [name=한지민, age=25]
712     --------------------------------
713     Student [name=설운도, age=55]
714     --------------------------------
715     Equals
716
717
718   11. ApplicationConfig와 MainClass2 생성하기
719     1)com.example > right-click > New > Class
720     2)Name : ApplicationConfig
721     3)Finish
722
723       package com.example;
724
725       import org.springframework.context.annotation.Bean;
726       import org.springframework.context.annotation.Scope;
727       import org.springframework.stereotype.Component;
728
729       @Component
730       public class ApplicationConfig {
731         @Bean
732         @Scope("prototype")
733         public Student student() {
734           Student student = new Student("박지민", 35);
735           return student;
736         }
737       }
738
739     4)com.example > right-click > New > Class
740     5)Name : MainClass2
741     6)Finish
742
743       package com.example;
744
745       import org.springframework.context.annotation.AnnotationConfigApplicationContext;
746
```

```
747     public class MainClass2 {
748       public static void main(String[] args) {
749         AnnotationConfigApplicationContext context =
750               new AnnotationConfigApplicationContext(ApplicationConfig.class);
751
752         Student student = context.getBean("student", Student.class);
753         System.out.println(student);
754         System.out.println("---------------------------------");
755
756         Student student1 = context.getBean("student", Student.class);
757         student1.setName("설운도");
758         student1.setAge(55);
759         System.out.println(student1);
760         System.out.println("---------------------------------");
761
762         if(student.equals(student1)) System.out.println("Equals");  //Print Equals
763         else System.out.println("Different");
764         context.close();
765       }
766     }
767
768   7)실행결과
769
770     Student(name=박지민, age=35)
771     ---------------------------------
772     Student(name=설운도, age=55)
773     ---------------------------------
774     Different
```