```
1. Lifecycle
 1
 2
       1)Spring Container 생성
 3
         GenericXmlApplicationContext context = new GenericXmlApplicationContext();
 4
 5
       2)Spring Container 설정
         context.load("classpath:ApplicationContext.xml");
 6
 7
         context.refresh(); //생성자를 사용하지 않을 때는 반드시 refresh() 할 것
 8
 9
       3)Spring Container 사용
         Student student = context.getBean("student", Student.class);
10
         System.out.println(student);
11
12
13
       4)Spring Container 종료
14
         context.close();
15
16
17
    2. Lab
18
       1)New > Spring Legacy Project > Simple Projects > Simple Spring Maven
19
        Project Name: SpringLifecycle
20
       2)Create Package: src/main/java/info.javaexpert
21
22
23
       3)info.javaexpert.Student.java
24
25
         package info.javaexpert;
26
27
         import java.util.ArrayList;
28
29
         public class Student {
30
            private String name;
31
            private int age;
            private ArrayList<String> hobbys;
32
33
            private double height;
34
            private double weight;
35
36
            public Student(String name, int age, ArrayList<String> hobbys) {
37
               this.name = name;
38
               this.age = age;
39
               this.hobbys = hobbys;
40
            }
41
42
            public void setName(String name) {
43
               this.name = name;
44
45
46
            public void setAge(int age) {
47
               this.age = age;
48
            }
49
            public void setHobbys(ArrayList<String> hobbys) {
50
51
               this.hobbys = hobbys;
52
            }
53
54
            public void setHeight(double height) {
55
               this.height = height;
56
            }
57
58
            public void setWeight(double weight) {
59
               this.weight = weight;
60
61
            @Override
62
63
            public String toString() {
               return String.format("Student [name=%s, age=%s, hobbys=%s, height=%s,
64
               weight=%s]", name, age, hobbys, height,
65
                    weight);
66
            }
```

```
}
 67
 68
 69
       4)src/main/resources/ApplicationContext.xml 생성
 70
          -/src/main/resources > right-click > New > Spring Bean Configuration File
             File name: ApplicationContext.xml
 71
 72
          <?xml version="1.0" encoding="UTF-8"?>
 73
 74
          <beans xmlns="http://www.springframework.org/schema/beans"</pre>
             xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 75
             xsi:schemaLocation="http://www.springframework.org/schema/beans
 76
             http://www.springframework.org/schema/beans/spring-beans.xsd">
 77
             <bean id="student1" class="info.javaexpert.Student">
 78
               <constructor-arg value="한지민"/>
 79
               <constructor-arg value="25" />
 80
               <constructor-arg>
 81
 82
                  t>
 83
                    <value>독서</value>
 84
                    <value>영화감상</value>
                    <value>요리</value>
 85
                  </list>
 86
 87
               </constructor-arg>
 88
               property name="height" value="165" />
               property name="weight">
 89
 90
                  <value>45</value>
 91
               </property>
 92
             </bean>
 93
          </beans>
 94
 95
       5)info.javaexpert.MainClass.java
 96
          package info.javaexpert;
 97
 98
          import org.springframework.context.support.GenericXmlApplicationContext;
 99
100
          public class MainClass {
             public static void main(String[] args) {
101
102
               GenericXmlApplicationContext context = new GenericXmlApplicationContext();
103
104
               context.load("classpath:ApplicationContext.xml");
105
               context.refresh();
106
               Student student1 = context.getBean("student1", Student.class);
107
108
               System.out.println(student1);
109
110
               context.close();
111
            }
112
          }
113
114
       6)실행
115
          -MainClass > right-click > Run As > Java Application
116
117
118
     3. Spring Bean Lifecycle
        1)context.refresh() 할 때 bean 초기화 과정에서는 InitializingBean interface의 afterPropertiesSet()
119
        또는 @PostConstruct annotation 을 호출
120
          -ctx.load("classpath:applicationContext.xml"); 에서 Bean이 생성되는 것이고,
121
          -ctx.refresh() 할 때 Bean이 생성되는 것이다.
122
          -즉 Bean이 초기화될 때 afterPropertiesSet()이 호출된다.
123
124
       2)context.close() 할 때 bean 소멸 과정에서는 DisposableBean interface의 destory() 또는
       @PreDestroy() 를 호출
125
126
       3)만일 close()할 때 컨테이더가 소멸되면 그 안의 모든 bean은 자동 소멸된다.
127
          -반면 bean만 소멸하고자 한다면 student.destory()를 이용하면 된다.
128
129
130 4. Lab
```

```
131
        1)New > Spring Legacy Project > Simple Projects > Simple Spring Maven
132
           Project Name: SpringLifecycle1
133
        2)Create Package: src/main/java/info.javaexpert
134
        3)InitializingBean, DisposableBean interface 이용하기
135
           -info.javaexpert.Student.java
136
137
             package info.javaexpert;
138
139
             import java.util.ArrayList;
140
141
             import org.springframework.beans.factory.DisposableBean;
142
             import org.springframework.beans.factory.InitializingBean;
143
144
             public class Student implements InitializingBean, DisposableBean{
145
                private String name;
146
                private int age;
147
148
                public Student(String name, int age) {
149
                   this.name = name;
150
                   this.age = age;
151
                }
152
153
                @Override
154
                public String toString() {
155
                   return String.format("Student [name=%s, age=%s]", name, age);
156
157
158
                @Override
159
                public void destroy() throws Exception {
160
                   System.out.println("Called destroy()");
161
                @Override
162
163
                public void afterPropertiesSet() throws Exception {
164
                   System.out.println("Called afterPropertiesSet()");
165
                }
             }
166
167
168
          -src/main/resources/ApplicationContext.xml
169
             <?xml version="1.0" encoding="UTF-8"?>
170
             <beans xmlns="http://www.springframework.org/schema/beans"</p>
171
                xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
172
                xsi:schemaLocation="http://www.springframework.org/schema/beans
                http://www.springframework.org/schema/beans/spring-beans.xsd">
173
174
                <bean id="student" class="info.javaexpert.Student">
175
                   <constructor-arg value="한지민"/>
176
                   <constructor-arg value="25" />
177
                </bean>
178
179
             </beans>
180
181
          -info.javaexpert.MainClass.java
182
183
             package info.javaexpert;
184
             import org.springframework.context.support.GenericXmlApplicationContext;
185
186
             public class MainClass {
187
                public static void main(String[] args) {
188
                   GenericXmlApplicationContext context = new GenericXmlApplicationContext();
189
                   context.load("classpath:ApplicationContext.xml");
190
                   context.refresh();
191
192
                   Student student = context.getBean("student", Student.class);
193
                   System.out.println(student);
194
                   context.close();
195
196
             }
```

```
197
198
        7)@PostConstruct, @PreDestroy 이용하기
199
          -info.javaexpert.Student2.java
200
201
             package info.javaexpert;
202
203
             import javax.annotation.PostConstruct;
204
             import javax.annotation.PreDestroy;
205
206
             public class Student2 {
207
                private String name;
208
               private int age;
209
210
               public Student2(String name, int age) {
211
                  this.name = name;
                  this.age = age;
212
213
               }
214
215
                @Override
216
               public String toString() {
                  return String.format("Student [name=%s, age=%s]", name, age);
217
218
               }
219
220
                @PostConstruct <--bean이 생성단계에서 해야할 일 기술
221
               public void initMethod(){
                  System.out.println("Called initMethod()");
222
223
224
225
                @PreDestroy
                               <--bean이 소멸할 때 해야할 일 기술
226
               public void destroyMethod(){
227
                  System.out.println("Called destoryMethod()");
228
229
             }
230
231
          -src/main/resources/ApplicationContext.xml
232
             <?xml version="1.0" encoding="UTF-8"?>
233
             <beans xmlns="http://www.springframework.org/schema/beans"</p>
234
               xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
235
               xsi:schemaLocation="http://www.springframework.org/schema/beans
               http://www.springframework.org/schema/beans/spring-beans.xsd">
236
237
                <bean
               class="org.springframework.context.annotation.CommonAnnotationBeanPostProcessor" />
                <--필수
238
239
                <bean id="student2" class="info.javaexpert.Student2">
240
                  <constructor-arg value="설운도" />
241
                  <constructor-arg value="50" />
242
                </bean>
243
             </beans>
244
245
          -info.javaexpert.MainClass.java
246
247
             package info.javaexpert;
248
249
             import org.springframework.context.support.GenericXmlApplicationContext;
250
251
             public class MainClass {
252
               public static void main(String[] args) {
253
                  GenericXmlApplicationContext context = new GenericXmlApplicationContext();
254
                  context.load("classpath:ApplicationContext.xml");
255
                  context.refresh();
256
257
                  Student2 student2 = context.getBean("student2", Student2.class);
258
                  System.out.println(student2);
259
                  context.close();
260
               }
```

```
262
263
264
     5. Spring Bean Scope
265
       1)Spring Container가 생성되고, Spring Bean이 생성될 때, 생성된 Spring Bean은 Scope를 가지고 있다.
          -scope이란 쉽게 생각해서 해당하는 객체가 어디까지 영향을 미치는지 결정하는 것이라고 생각하면 된다.
266
267
          -singleton : 스프핑 컨테이너에 단 한 개의 빈 객체만 존재, 기본값
          -prototype: Bean을 사용할 때마다 객체를 생성
268
269
          -request: Http 요청마다 빈 객체를 생성 WebApplicationContext 에서만 적용 가능
270
          -session: Http Session 마다 빈 객체를 생성한다. WebApplicationContext에서만 적용 가능
271
272
273
     6. Lab
274
       1)New > Spring Legacy Project > Simple Projects > Simple Spring Maven
275
          Project Name: SpringScopeDemo
276
277
       2)Create Package: src/main/java/info.javaexpert
278
279
       3)info.javaexpert.Student.java
280
281
          package info.javaexpert;
282
283
          import java.util.ArrayList;
284
          import org.springframework.beans.factory.DisposableBean;
285
          import org.springframework.beans.factory.InitializingBean;
286
287
          public class Student{
288
            private String name;
289
            private int age;
290
291
            public Student(String name, int age) {
292
               this.name = name;
293
               this.age = age;
294
            }
295
296
            public void setName(String name) {
297
               this.name = name;
298
            }
299
300
            public void setAge(int age) {
301
               this.age = age;
302
303
304
            @Override
305
            public String toString() {
306
               return String.format("Student [name=%s, age=%s]", name, age);
307
            }
308
          }
309
310
       4)src/main/resources/ApplicationContext.xml
          <?xml version="1.0" encoding="UTF-8"?>
311
312
          <beans xmlns="http://www.springframework.org/schema/beans"</pre>
313
            xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
314
            xsi:schemaLocation="http://www.springframework.org/schema/beans
            http://www.springframework.org/schema/beans/spring-beans.xsd">
315
316
            <bean id="student" class="info.javaexpert.Student" scope="singleton">
317
               <constructor-arg value="한지민"/>
318
               <constructor-arg value="25" />
319
            </bean>
320
321
          </beans>
322
323
       6)info.javaexpert.MainClass.java
324
325
          package info.javaexpert;
326
```

261

}

```
327
          import org.springframework.context.support.AbstractApplicationContext;
328
          import org.springframework.context.support.GenericXmlApplicationContext;
329
330
          public class MainClass {
331
             public static void main(String[] args) {
               AbstractApplicationContext context = new
332
               GenericXmlApplicationContext("classpath:ApplicationContext.xml");
333
334
               Student student = context.getBean("student", Student.class);
335
               System.out.println(student);
               System.out.println("-----");
336
337
               Student student1 = context.getBean("student", Student.class);
338
               student1.setName("설운도");
339
               student1.setAge(55);
340
               System.out.println(student1);
341
               System.out.println("-----");
342
343
               if(student.equals(student1)) System.out.println("Equals"); //Print Equals
344
345
               else System.out.println("Different");
               context.close();
346
347
            }
          }
348
```