```
 1  1. Environment 객체
 2    1)Environment 객체를 이용해서 Spring bean 설정을 할 수 있다.
 3      Context  --------------------> Environment ------------------------> PropertySources
 4            ctx.getEnvironment()           env.getPropertySource()   property 추가 및 추출
 5                                                              추가 : propertySources.addLast()
 6                                                              추출 : env.getProperty()
 7
 8
 9  2. Lab
10    1)In Package Explorer > right-click > New > Java Project
11      -Project Name : EnvironmentDemo
12
13    2)src > right-click > New > Package
14      -Package name : com.example
15
16    3)Java Project를 Spring Project로 변환
17      -EnvironmentDemo Project > right-click > Configuration > Convert to Maven Project
18        --Project : /EnvironmentDemo
19        --Group Id : EnvironmentDemo
20        --Artifact Id : EnvironmentDemo
21        --version : 0.0.1-SNAPSHOT
22        --Packaging : jar
23        --Finish
24
25      -EnvironmentDemo Project > right-click > Spring > Add Spring Project Natur
26
27      -pom.xml file에 Spring Context Dependency 추가하기
28        <version>0.0.1-SNAPSHOT</version>
29        <dependencies>
30          <dependency>
31          <groupId>org.springframework</groupId>
32          <artifactId>spring-context</artifactId>
33          <version>4.3.24.RELEASE</version>
34          </dependency>
35        </dependencies>
36
37      -pom.xml > right-click > Run As > Maven install
38        [INFO] BUILD SUCCESS 확인
39
40    4)EnvironmentDemo/resources folder 생성
41      -EnvironmentDemo project > right-click > Build Path > Configure Build Path
42      -Source Tab > Add Folder
43      -EnvironmentDemo click
44      -Create New Folder > Folder name : resources > Finish > OK
45      -EnvironmentDemo/resources(new) 확인
46      -Apply and Close
47
48    5)resources/admin.properties file 생성
49
50      admin.id=javaexpert
51      admin.pwd=12345678
52
53    6)com.example.AdminConnection.java 생성
54
```

```
55      package com.example;
56
57      import org.springframework.beans.factory.DisposableBean;
58      import org.springframework.beans.factory.InitializingBean;
59      import org.springframework.context.EnvironmentAware;
60      import org.springframework.core.env.Environment;
61
62      public class AdminConnection implements EnvironmentAware, InitializingBean,
        DisposableBean{
63        private Environment env;
64        private String adminId;
65        private String adminPwd;
66
67        public void setEnv(Environment env) {
68          this.env = env;
69        }
70
71        public void setAdminId(String adminId) {
72          this.adminId = adminId;
73        }
74
75        public void setAdminPwd(String adminPwd) {
76          this.adminPwd = adminPwd;
77        }
78
79        public String getAdminId() {
80          return adminId;
81        }
82
83        public String getAdminPwd() {
84          return adminPwd;
85        }
86
87        @Override
88        public void destroy() throws Exception {
89          System.out.println("destroy()");
90        }
91
92        @Override
93        public void afterPropertiesSet() throws Exception {
94          System.out.println("afterPropertiesSet()");
95          setAdminId(env.getProperty("admin.id"));
96          setAdminPwd(env.getProperty("admin.pwd"));
97        }
98
99        //bean이 생성되기 전에 callback 으로 호출됨. 가장 먼저 호출됨.
100       //MainClass에서 사용하는 env 정보가 넘어옴.
101       @Override
102       public void setEnvironment(Environment env) {
103         System.out.println("setEnvironment()");
104         setEnv(env);
105       }
106     }
107
```

```
108    5)resources/beans.xml 생성
109
110      <?xml version="1.0" encoding="UTF-8"?>
111      <beans xmlns="http://www.springframework.org/schema/beans"
112        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
113        xsi:schemaLocation="http://www.springframework.org/schema/beans
         http://www.springframework.org/schema/beans/spring-beans.xsd">
114
115        <bean id="adminConnection" class="com.example.AdminConnection" />
116
117      </beans>
118
119    6)com.example.MainClass.java 생성
120
121      package com.example;
122
123      import java.io.IOException;
124
125      import org.springframework.context.ConfigurableApplicationContext;
126      import org.springframework.context.support.GenericXmlApplicationContext;
127      import org.springframework.core.env.ConfigurableEnvironment;
128      import org.springframework.core.env.MutablePropertySources;
129      import org.springframework.core.io.support.ResourcePropertySource;
130
131      public class MainClass {
132        public static void main(String [] args){
133          ConfigurableApplicationContext ctx = new GenericXmlApplicationContext();
134          ConfigurableEnvironment env = ctx.getEnvironment();
135
136          MutablePropertySources propertySouces = env.getPropertySources();
137          //내가 원하는 정보를 얻을 때까지 모든 propertySources를 앞에서 부터 차례로 모두 검색함.
138          try{
139            propertySouces.addLast(new ResourcePropertySource("classpath:admin.properties"));
               //property 추가
140
141            System.out.println(env.getProperty("admin.id"));    //property 추출
142            System.out.println(env.getProperty("admin.pwd"));
143          }catch(IOException ex){}
144
145          GenericXmlApplicationContext gCtx = (GenericXmlApplicationContext)ctx;
146          gCtx.load("classpath:beans.xml");
147          gCtx.refresh();
148
149          AdminConnection adminConnection = gCtx.getBean("adminConnection",
               AdminConnection.class);
150          System.out.println("admin ID : " + adminConnection.getAdminId());
151          System.out.println("admin PWD : " + adminConnection.getAdminPwd());
152
153          gCtx.close();
154          ctx.close();
155        }
156      }
157
158    7)실행
```

```
159        -MainClass > right-click > Run As > Java Application
160
161    8)결과
162        setEnvironment()
163        afterPropertiesSet()
164        admin ID : javaexpert
165        admin PWD : 12345678
166        ...
167        destroy()
168
169
170  3. Property file을 이용한 설정
171    1)환경에 따라 자주 변경되는 내용의 분리
172    2)XML의 Bean 설정 meta정보는 application 구조가 바뀌지 않으면 자주 변경되지 않는다.
173    3)반면에 property 값으로 제공되는 일부 설정 정보(예-DataSource Bean이 사용하는 DB 연결정보)는
       application이 동작하는 환경(개발, test, stageing, 운영)에 따라서 자주 바뀔 수 있다.
174    4)변경되는 이유와 시점이 다르다면 분리하는 것이 객체지향 설계의 기본 원칙이기에 설정에도 동일한 원칙을 적용할 수 있
       다.
175    5)환경에 따라 자주 변경될 수 있는 내용은 properties file로 분리하는 것이 가장 깔끔하다.
176    6)XML 처럼 복잡한 구성이 필요없고 키와 값의 쌍(key=value)으로 구성하면 된다.
177    7)환경에 따라 자주 변경되는 내용의 분리의 예시
178        -value속성에 설정된 값들은 환경에 따라 변경될 수 있는 내용이다.
179        -자주 변경되는 값들은 properties file에 넣어 분리하는 것이 좋다.
180
181        <beans.xml>
182        <bean id="dataSource"
183              class="org.springframework.jdbc.datasource.SimpleDriverDataSource">
184           <property name="driverClass" value="com.mysql.jdbc.Driver" />
185           <property name="url" value="jdbc:mysql://localhost/testdb" />
186           <property name="username" value="spring" />
187           <property name="password" value="book" />
188        </bean>
189
190    -properties file로 분리한 정보는 ${}(property 치환자)을 이용하여 설정한다.
191    -${} 값을 치환해주는 기능은 <context:property-placeholder> tag에 의해 자동으로 등록되는
       PropertyPlaceHolderConfigurer Bean이 담당한다.
192
193        <database.properties>
194           db.driverClass=com.mysql.jdbc.Driver
195           db.url=jdbc:mysql://localhost/testdb
196           db.username=spring
197           db.password=book
198
199        <beans.xml>
200          <context:property-placeholder
201              location="classpath:config/database.properties" />
202          <bean id="dataSource"
203              class="org.springframework.jdbc.datasource.SimpleDriverDataSource">
204             <property name="driverClass" value="${db.driverClass}" />
205             <property name="url" value="${db.url}" />
206             <property name="username" value="${db.username}" />
207             <property name="password" value="${db.password}" />
208          </bean>
209
```

```
210
211   4. Lab
212     1)In Package Explorer > right-click > New > Java Project
213       -Project Name : PropertyDemo
214
215     2)src > right-click > New > Package
216       -Package name : com.example
217
218     3)POJO class 작성
219       -com.example.Hello
220
221         package com.example;
222
223         public class Hello{
224           private String name;
225           private Printer printer;
226           private List<String> names;
227
228           public Hello(){}
229
230           public void setName(String name){
231             this.name = name;
232           }
233
234           public void setPrinter(Printer printer){
235             this.printer = printer;
236           }
237
238           public void setNames(List<String> list){
239             this.names = list;
240           }
241
242           public List<String> getNames(){
243             return this.names;
244           }
245
246           public String sayHello(){
247             return "Hello " + name;
248           }
249
250           public void print(){
251             this.printer.print(sayHello());
252           }
253         }
254
255       -com.example > right-click > New > Interface
256         Interface name : Printer
257
258         package com.example;
259
260         public interface Printer{
261           void print(String message);
262         }
263
```

```
264        -com.example > right-click > New > Class
265          Class Name : StringPrinter
266
267             package com.example;
268
269             public class StringPrinter implements Printer{
270               private StringBuffer buffer = new StringBuffer();
271
272               @Override
273               public void print(String message){
274                 this.buffer.append(message);
275               }
276
277               public String toString(){
278                 return this.buffer.toString();
279               }
280             }
281
282        -com.example > right-click > New > Class
283          Class Name : ConsolePrinter
284
285             package com.example;
286
287             public class ConsolePrinter implements Printer{
288
289               @Override
290               public void print(String message){
291                 System.out.println(message);
292               }
293             }
294
295      4)Java Project를 Spring Project로 변환
296        -PropertyDemo Project > right-click > Configuration > Convert to Maven Project
297          --Project : /PropertyDemo
298          --Group Id : PropertyDemo
299          --Artifact Id : PropertyDemo
300          --version : 0.0.1-SNAPSHOT
301          --Packaging : jar
302          --Finish
303
304        -PropertyDemo Project > right-click > Spring > Add Spring Project Nature
305
306        -pom.xml file에 Spring Context Dependency 추가하기
307          <version>0.0.1-SNAPSHOT</version>
308          <dependencies>
309            <dependency>
310              <groupId>org.springframework</groupId>
311              <artifactId>spring-context</artifactId>
312              <version>4.3.24.RELEASE</version>
313            </dependency>
314          </dependencies>
315
316        -pom.xml > right-click > Run As > Maven install
317          [INFO] BUILD SUCCESS 확인
```

```
318
319    5)PropertyDemo/resources folder 생성
320      -PropertyDemo project > right-click > Build Path > Configure Build Path
321      -Source Tab > Add Folder
322      -PropertyDemo click
323      -Create New Folder > Folder name : resources > Finish > OK
324      -PropertyDemo/resources(new) 확인
325      -Apply and Close
326
327    6)Bean Configuration XML 작성
328      -PropertyDemo/resources > right-click > New > Other > Spring > Spring Bean Configuration
       File
329      -File name : beans.xml > Finish
330
331        <?xml version="1.0" encoding="UTF-8"?>
332        <beans xmlns="http://www.springframework.org/schema/beans"
333          xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
334          xsi:schemaLocation="http://www.springframework.org/schema/beans
         http://www.springframework.org/schema/beans/spring-beans.xsd">
335
336          <bean id="hello" class="com.example.Hello">
337            <property name="name" value="Spring" />
338            <property name="printer" ref="printer" />
339            <property name="names">
340              <list>
341                <value>AOP</value>
342                <value>Spring</value>
343                <value>DI</value>
344              </list>
345            </property>
346          </bean>
347
348          <bean id="printer" class="com.example.StringPrinter" />
349          <bean id="consolePrinter" class="com.example.ConsolePrinter" />
350
351        </beans>
352
353    7)com.example.MainClass
354
355        package com.example;
356
357        import java.util.List;
358
359        import org.springframework.context.ApplicationContext;
360        import org.springframework.context.support.GenericXmlApplicationContext;
361
362        public class MainClass {
363          public static void main(String [] args){
364            ApplicationContext ctx = new GenericXmlApplicationContext("classpath:beans.xml");
365
366            Hello hello = (Hello)ctx.getBean("hello");
367            System.out.println(hello.sayHello());
368            hello.print();
369
```

```
370             Printer printer = ctx.getBean("printer", StringPrinter.class);
371             System.out.println(printer.toString());
372
373             List<String> list = hello.getNames();
374             for(String value : list){
375                 System.out.println(value);
376             }
377         }
378     }
379
```

380   8)실행
381   -MainClass > right-click > Run As > Java Application
382
383   9)결과
384   Hello Spring
385   Hello Spring
386   AOP
387   Spring
388   DI
389
390   10)jUnit으로 test
391   -src/test package 생성
392   -/src/test/ > right-click > New > JUnit Test Case > HelloTest > Finish
393   -New JUnit Test Case창에서, Not now 선택 > OK
394   -https://mvnrepository.com에서 'junit'로 검색
395   -JUnit에서
396   -4.12로 들어가서
397   -복사해서 pom.xml로 붙여넣기
398   -https://mvnrepository.com에서 'spring-test'로 검색
399   -Spring TestContext Framework에서
400   -4.3.24.RELEASE로 들어가서
401   -복사해서 pom.xml로 붙여넣기
402   -pom.xml > right-click > Run As > Maven install
403   [INFO] BUILD SUCCESS 확인

```
405         import static org.junit.Assert.assertEquals;
406         import static org.junit.Assert.assertSame;
407
408         import java.util.List;
409
410         import org.junit.Test;
411         import org.junit.runner.RunWith;
412         import org.springframework.beans.factory.annotation.Autowired;
413         import org.springframework.context.ApplicationContext;
414         import org.springframework.test.context.ContextConfiguration;
415         import org.springframework.test.context.junit4.SpringJUnit4ClassRunner;
416
417         import com.example.Hello;
418         import com.example.Printer;
419
420         @RunWith(SpringJUnit4ClassRunner.class)
421         @ContextConfiguration(locations="classpath:beans.xml")
422         public class HelloTest {
423             @Autowired
```

```
424            ApplicationContext ctx;
425
426            @Test
427            public void test() {
428               Hello hello = (Hello)ctx.getBean("hello");
429               assertEquals("Hello Spring", hello.sayHello());
430               hello.print();
431
432               Printer printer = (Printer)ctx.getBean("printer");
433               assertEquals("Hello Spring", printer.toString());
434            }
435
436            @Test
437            public void test2(){
438               Hello hello = (Hello)ctx.getBean("hello");
439
440               Hello hello2 = ctx.getBean("hello", Hello.class);
441               assertSame(hello, hello2);
442
443               assertEquals(3, hello2.getNames().size());
444            }
445         }
446
447      -right-click > Run As > Junit Test
448      -결과 -> Junit View에 초록색 bar
449
450   11)resources/value.properties 생성
451
452      myname=Spring
453      myprinter=printer
454      value1=HTML5
455      value2=CSS3
456      value3=JavaScript
457
458   12)/resources/beans.xml 에서 [Namespaces] tab
459      -목록에서 'context-http://www.springframework.org/schema/context' check
460      -<context:property-placeholder />를 사용하기 위해서
461
462         <context:property-placeholder
463               location="classpath:value.properties" />
464
465         <bean id="hello" class="com.example.Hello">
466            <property name="name" value="${myname}" />
467            <property name="printer" ref="${myprinter}" />
468            <property name="names">
469              <list>
470                <value>${value1}</value>
471                <value>${value2}</value>
472                <value>${value3}</value>
473              </list>
474            </property>
475         </bean>
476
477         <bean id="printer" class="com.example.StringPrinter" />
```

```
478          <bean id="consolePrinter" class="com.example.ConsolePrinter" />
479
480    13)Test
481      -com.example.MainClass.java
482        --right-click > Run As > Java Application
483          Hello Spring
484          Hello Spring
485          HTML5
486          CSS3
487          JavaScript
488
489      -/src/test/java/HelloTest.java
490        --right-click > Run As > JUnit Test
491        --Green Bar
492
493    14)resources/value.properties 수정
494        myname=Spring
495        myprinter=printer
496        value1=JUnit
497        value2=AOP
498        value3=DI
499        printer1=stringPrinter
500        printer2=consolePrinter
501
502    15)Hello.java code 수정
503      -com.example/Hello.java
504
505        package com.example;
506
507        import java.util.List;
508
509        import org.springframework.beans.factory.annotation.Value;
510        import org.springframework.stereotype.Component;
511        import javax.annotation.Resource;
512
513        @Component("hello")
514        public class Hello {
515          @Value("${myname}")
516          private String name;
517
518          @Resource(name="${printer1}")
519          private Printer printer;
520
521          @Value("${value1}, ${value2}, ${value3}")
522          private List<String> names;
523
524          public List<String> getNames(){
525            return names;
526          }
527          public String sayHello(){
528            return "Hello " + name;
529          }
530
531          public void print(){
```

```
532              this.printer.print(sayHello());
533          }
534       }
535
536    16)StringPrinter.java 수정
537        package com.example;
538
539        import org.springframework.stereotype.Component;
540
541        @Component("stringPrinter")
542        public class StringPrinter implements Printer {
543          private StringBuffer buffer =  new StringBuffer();
544
545          @Override
546          public void print(String message) {
547            this.buffer.append(message);
548          }
549
550          @Override
551          public String toString(){
552            return this.buffer.toString();
553          }
554       }
555
556    17)beans.xml 수정하기
557
558        <?xml version="1.0" encoding="UTF-8"?>
559        <beans xmlns="http://www.springframework.org/schema/beans"
560          xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
561          xmlns:context="http://www.springframework.org/schema/context"
562          xsi:schemaLocation="http://www.springframework.org/schema/beans
            http://www.springframework.org/schema/beans/spring-beans.xsd
563            http://www.springframework.org/schema/context
            http://www.springframework.org/schema/context/spring-context-3.2.xsd">
564
565        <context:property-placeholder location="classpath:value.properties "/>
566        <context:component-scan base-package="com.example" />
567        </beans>
568
569    18)MainClass.java 수정하기
570
571        package com.example;
572
573        import java.util.List;
574
575        import org.springframework.context.ApplicationContext;
576        import org.springframework.context.support.GenericXmlApplicationContext;
577
578        public class MainClass {
579          public static void main(String [] args){
580            ApplicationContext ctx = new GenericXmlApplicationContext("classpath:beans.xml");
581
582            Hello hello = (Hello)ctx.getBean("hello");
583            System.out.println(hello.sayHello());
```

```
584              hello.print();
585
586              Printer printer = ctx.getBean("stringPrinter", StringPrinter.class);
587              System.out.println(printer.toString());
588
589              List<String> list = hello.getNames();
590              for(String value : list){
591                 System.out.println(value);
592              }
593           }
594        }
595
596     19)실행
597        -MainClass > right-click > Run As > Java Application
598
599     20)결과
600        Hello Spring
601        Hello Spring
602        JUnit, AOP, DI
603
604
605  5. Lab
606     1)In Package Explorer > right-click > New > Java Project
607        -Project Name : PropertyDemo1
608
609     2)/src/ right-click > New > Package
610        -Package name : com.example
611
612     3)/src/com.example.AdminConnection.java 생성
613        package com.example;
614
615        public class AdminConnection {
616           private String adminId;
617           private String adminPwd;
618           private String subAdminId;
619           private String subAdminPwd;
620
621           public String getAdminId() {
622              return adminId;
623           }
624
625           public void setAdminId(String adminId) {
626              this.adminId = adminId;
627           }
628
629           public String getAdminPwd() {
630              return adminPwd;
631           }
632
633           public void setAdminPwd(String adminPwd) {
634              this.adminPwd = adminPwd;
635           }
636
637           public String getSubAdminId() {
```

```
638              return subAdminId;
639          }
640
641          public void setSubAdminId(String subAdminId) {
642              this.subAdminId = subAdminId;
643          }
644
645          public String getSubAdminPwd() {
646              return subAdminPwd;
647          }
648
649          public void setSubAdminPwd(String subAdminPwd) {
650              this.subAdminPwd = subAdminPwd;
651          }
652      }
653
654      4)Java Project를 Spring Project로 변환
655        -PropertyDemo1 Project > right-click > Configuration > Convert to Maven Project
656          --Project : /PropertyDemo1
657          --Group Id : PropertyDemo1
658          --Artifact Id : PropertyDemo1
659          --version : 0.0.1-SNAPSHOT
660          --Packaging : jar
661          --Finish
662
663        -PropertyDemo1 Project > right-click > Spring > Add Spring Project Nature
664
665        -pom.xml file에 Spring Context Dependency 추가하기
666          <version>0.0.1-SNAPSHOT</version>
667          <dependencies> <--- dependencies element 추가
668            <dependency> <---여기에 paste
669              <groupId>org.springframework</groupId>
670              <artifactId>spring-context</artifactId>
671              <version>4.3.24.RELEASE</version>
672            </dependency>
673          </dependencies>
674
675        -pom.xml > right-click > Run As > Maven install
676          [INFO] BUILD SUCCESS 확인
677
678      5)PropertyDemo/resources folder 생성
679        -PropertyDemo1 project > right-click > Build Path > Configure Build Path
680        -Source Tab > Add Folder
681        -PropertyDemo1 click
682        -Create New Folder > Folder name : resources > Finish > OK
683        -PropertyDemo1/resources(new) 확인
684        -Apply and Close
685
686      6)  /resources 두 개의 properties file 생성
687
688        <admin.properties>
689          admin.id=javaexpert
690          admin.pwd=12345678
691
```

```
692        <sub.admin.properties>
693          sub.admin.id=javasoft
694          sub.admin.pwd=987654321
695
696    7)Bean Configuration XML 작성
697      -PropertyDemo1/resources > right-click > New > Other > Spring > Spring Bean
        Configuration File
698      -File name : beans.xml > Finish
699      -Namespace tab에서 context - http://www.springframework.org/schema/context check
700
701        <?xml version="1.0" encoding="UTF-8"?>
702        <beans xmlns="http://www.springframework.org/schema/beans"
703          xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
704          xmlns:context="http://www.springframework.org/schema/context"
705          xsi:schemaLocation="http://www.springframework.org/schema/beans
          http://www.springframework.org/schema/beans/spring-beans.xsd
706            http://www.springframework.org/schema/context
            http://www.springframework.org/schema/context/spring-context-3.2.xsd">
707
708        <context:property-placeholder location="classpath:admin.properties,
          classpath:sub.admin.properties" />
709
710        <bean id="adminConnection" class="com.example.AdminConnection">
711          <property name="adminId">
712            <value>${admin.id}</value>
713          </property>
714          <property name="adminPwd">
715            <value>${admin.pwd}</value>
716          </property>
717          <property name="subAdminId">
718            <value>${sub.admin.id}</value>
719          </property>
720          <property name="subAdminPwd">
721            <value>${sub.admin.pwd}</value>
722          </property>
723        </bean>
724      </beans>
725
726    8)/src/com.example.MainClass.java 생성
727      package com.example;
728
729      import org.springframework.context.support.AbstractApplicationContext;
730      import org.springframework.context.support.GenericXmlApplicationContext;
731
732      public class MainClass {
733        public static void main(String [] args){
734          AbstractApplicationContext ctx =
735              new GenericXmlApplicationContext("classpath:beans.xml");
736          AdminConnection connection = ctx.getBean("adminConnection",
          AdminConnection.class);
737          System.out.println("admin ID : " + connection.getAdminId());
738          System.out.println("admin PWD : " + connection.getAdminPwd());
739          System.out.println("sub admin ID : " + connection.getSubAdminId());
740          System.out.println("sub admin PWD : " + connection.getSubAdminPwd());
```

```
741
742          ctx.close();
743        }
744      }
745
746    9)결과
747      -MainClass.java > right-click > Run As > Java Application
748        admin ID : javaexpert
749        admin PWD : 12345678
750        sub admin ID : javasoft
751        sub admin PWD : 987654321
752
753
754  6. Lab
755    1)In Package Explorer > right-click > New > Java Project
756      -Project Name : PropertyDemo2
757
758    2)/src/ right-click > New > Package
759      -Package name : com.example
760
761    3)/src/com.example.AdminConnection.java 생성
762      package com.example;
763
764      public class AdminConnection {
765        private String adminId;
766        private String adminPwd;
767        private String subAdminId;
768        private String subAdminPwd;
769
770        public String getAdminId() {
771          return adminId;
772        }
773
774        public void setAdminId(String adminId) {
775          this.adminId = adminId;
776        }
777
778        public String getAdminPwd() {
779          return adminPwd;
780        }
781
782        public void setAdminPwd(String adminPwd) {
783          this.adminPwd = adminPwd;
784        }
785
786        public String getSubAdminId() {
787          return subAdminId;
788        }
789
790        public void setSubAdminId(String subAdminId) {
791          this.subAdminId = subAdminId;
792        }
793
794        public String getSubAdminPwd() {
```

```
795            return subAdminPwd;
796          }
797
798          public void setSubAdminPwd(String subAdminPwd) {
799            this.subAdminPwd = subAdminPwd;
800          }
801        }
802
803    4)Java Project를 Spring Project로 변환
804      -PropertyDemo1 Project > right-click > Configuration > Convert to Maven Project
805        --Project : /PropertyDemo2
806        --Group Id : PropertyDemo2
807        --Artifact Id : PropertyDemo2
808        --version : 0.0.1-SNAPSHOT
809        --Packaging : jar
810        --Finish
811
812      -PropertyDemo2 Project > right-click > Spring > Add Spring Project Nature
813
814      -pom.xml 파일에 Spring Context Dependency 추가하기
815        <version>0.0.1-SNAPSHOT</version>
816        <dependencies>
817          <dependency>
818            <groupId>org.springframework</groupId>
819            <artifactId>spring-context</artifactId>
820            <version>4.3.24.RELEASE</version>
821          </dependency>
822        </dependencies>
823
824      -pom.xml > right-click > Run As > Maven install
825        [INFO] BUILD SUCCESS 확인
826
827    5)PropertyDemo/resources folder 생성
828      -PropertyDemo2 project > right-click > Build Path > Configure Build Path
829      -Source Tab > Add Folder
830      -PropertyDemo2 click
831      -Create New Folder > Folder name : resources > Finish > OK
832      -PropertyDemo2/resources(new) 확인
833      -Apply and Close
834
835    6)  /resources 두 개의 properties file 생성
836
837      <admin.properties>
838        admin.id=javaexpert
839        admin.pwd=12345678
840
841      <sub.admin.properties>
842        sub.admin.id=javasoft
843        sub.admin.pwd=987654321
844
845    7)/src/com.example.ApplicationConfig.java>
846        package com.example;
847
848        import org.springframework.beans.factory.annotation.Value;
```

```
849        import org.springframework.context.annotation.Bean;
850        import org.springframework.context.annotation.Configuration;
851        import org.springframework.context.support.PropertySourcesPlaceholderConfigurer;
852        import org.springframework.core.io.ClassPathResource;
853        import org.springframework.core.io.Resource;
854
855        @Configuration
856        public class ApplicationConfig {
857          @Value("${admin.id}")
858          private String adminId;
859          @Value("${admin.pwd}")
860          private String adminPwd;
861          @Value("${sub.admin.id}")
862          private String subAdminId;
863          @Value("${sub.admin.pwd}")
864          private String subAdminPwd;
865
866          @Bean
867          public static PropertySourcesPlaceholderConfigurer Properties(){
868            PropertySourcesPlaceholderConfigurer configurer =
869                new PropertySourcesPlaceholderConfigurer();
870
871            Resource [] locations = new Resource[2];
872            locations[0] = new ClassPathResource("admin.properties");
873            locations[1] = new ClassPathResource("sub.admin.properties");
874            configurer.setLocations(locations);
875
876            return configurer;
877          }
878
879        @Bean
880        public AdminConnection adminConfig(){
881          AdminConnection adminConnection = new AdminConnection();
882          adminConnection.setAdminId(adminId);
883          adminConnection.setAdminPwd(adminPwd);
884          adminConnection.setSubAdminId(subAdminId);
885          adminConnection.setSubAdminPwd(subAdminPwd);
886          return adminConnection;
887        }
888      }
889
890    8)/src/com.example.MainClass.java
891        package com.example;
892
893        import org.springframework.context.annotation.AnnotationConfigApplicationContext;
894
895        public class MainClass {
896          public static void main(String[] args) {
897            AnnotationConfigApplicationContext ctx =
898                new AnnotationConfigApplicationContext(ApplicationConfig.class);
899            AdminConnection conn = ctx.getBean("adminConfig", AdminConnection.class);
900
901            System.out.println("admin ID : " + conn.getAdminId());
902            System.out.println("admin PWD : " + conn.getAdminPwd());
```

```
903              System.out.println("sub admin ID : " + conn.getSubAdminId());
904              System.out.println("sub admin PWD : " + conn.getSubAdminPwd());
905          }
906       }
907
908    9)결과
909      -MainClass.java > right-click > Run As > Java Application
910        admin ID : javaexpert
911        admin PWD : 12345678
912        sub admin ID : javasoft
913        sub admin PWD : 987654321
914
915
916  7. Profile 속성을 이용한 설정
917    -동일한 Spring Bean을 여러 개 만들어 놓고 상황(환경)에 따라서 적절한 Spring bean을 사용할 수 있다.
918    -profile 속성을 사용한다.
919    -역시 Java file을 이용하는 방법과 XML 설정 file을 이용하는 방법이 있다.
920
921
922  8. Lab
923    1)In Package Explorer > right-click > New > Java Project
924      -Project Name : ProfileDemo
925
926    2)Package 생성
927      -/src/ > right-click > New > Package
928      -Package name : com.example
929
930    3)Java Project를 Spring Project로 변환
931      -ProfileDemo Project > right-click > Configuration > Convert to Maven Project
932        --Project : /ProfileDemo
933        --Group Id : ProfileDemo
934        --Artifact Id : ProfileDemo
935        --version : 0.0.1-SNAPSHOT
936        --Packaging : jar
937        --Finish
938      -ProfileDemo Project > right-click > Spring > Add Spring Project Nature
939
940      -pom.xml file에 Spring Context Dependency 추가하기
941        <version>0.0.1-SNAPSHOT</version>
942        <dependencies> <--- dependencies element 추가
943          <dependency> <---여기에 paste
944            <groupId>org.springframework</groupId>
945            <artifactId>spring-context</artifactId>
946            <version>4.3.24.RELEASE</version>
947          </dependency>
948        </dependencies>
949
950      -pom.xml > right-click > Run As > Maven install
951        [INFO] BUILD SUCCESS 확인
952
953    4)ProfileDemo/resources folder 생성
954      -ProfileDemo project > right-click > Build Path > Configure Build Path
955      -Source Tab > Add Folder
956      -ProfileDemo click
```

```
957         -Create New Folder > Folder name : resources > Finish > OK
958         -ProfileDemo/resources(new) 확인
959         -Apply and Close
960
961     5)ServerInfo.java 생성
962        -/src/com.example.ServerInfo.java
963
964          package com.example;
965
966          public class ServerInfo {
967             private String ipNum;
968             private String portNum;
969             public String getIpNum() {
970                return ipNum;
971             }
972             public void setIpNum(String ipNum) {
973                this.ipNum = ipNum;
974             }
975             public String getPortNum() {
976                return portNum;
977             }
978             public void setPortNum(String portNum) {
979                this.portNum = portNum;
980             }
981          }
982
983     6)   XML 설정 file 2개 생성
984        -/resource > right-click > New > Spring Bean Configuration File
985        -File name : run.xml
986
987           <?xml version="1.0" encoding="UTF-8"?>
988           <beans xmlns="http://www.springframework.org/schema/beans"
989             xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
990             xsi:schemaLocation="http://www.springframework.org/schema/beans
                http://www.springframework.org/schema/beans/spring-beans.xsd"
991             profile="run">    <---이것이 핵심
992
993             <bean id="serverInfo" class="com.example.ServerInfo">
994                <property name="ipNum" value="192.168.56.5" />
995                <property name="portNum" value="80" />
996             </bean>
997           </beans>
998
999        -/resource > right-click > New > Spring Bean Configuration File
1000       -File name : dev.xml
1001
1002          <?xml version="1.0" encoding="UTF-8"?>
1003          <beans xmlns="http://www.springframework.org/schema/beans"
1004            xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
1005            xsi:schemaLocation="http://www.springframework.org/schema/beans
                http://www.springframework.org/schema/beans/spring-beans.xsd"
1006            profile="dev">    <---이것이 핵심
1007
1008            <bean id="serverInfo" class="com.example.ServerInfo">
```

```
1009                <property name="ipNum" value="localhost" />
1010                <property name="portNum" value="8080" />
1011              </bean>
1012          </beans>
1013
1014    7)MainClass 생성
1015      -/src/com.example.MainClass.java
1016
1017        package com.example;
1018        import java.util.Scanner;
1019
1020        import org.springframework.context.support.GenericXmlApplicationContext;
1021
1022        public class MainClass {
1023          public static void main(String[] args) {
1024            Scanner scan = new Scanner(System.in);
1025            System.out.print("Select dev or run : ");
1026            String config = scan.next();  //"dev" or "run"
1027
1028            GenericXmlApplicationContext ctx = new GenericXmlApplicationContext();
1029            ctx.getEnvironment().setActiveProfiles(config);
1030            ctx.load("dev.xml", "run.xml");
1031            ctx.refresh();
1032
1033            ServerInfo info = ctx.getBean("serverInfo", ServerInfo.class);
1034            System.out.println("IP : " + info.getIpNum());
1035            System.out.println("Port : " + info.getPortNum());
1036            ctx.close();
1037          }
1038        }
1039
1040    8)결과
1041      --MainClass.java > right-click > Run As > Java Application
1042      Select dev or run :
1043      -입력시 dev를 넣으면 dev환경인 localhost/8080이 나오고, 만일 run이라고 넣으면 192.168.56.5/80이 나온
         다.
1044
1045
1046  9. Lab
1047    1)In Package Explorer > right-click > New > Java Project
1048      -Project Name : ProfileDemo1
1049
1050    2)Package 생성
1051      -/src/ > right-click > New > Package
1052      -Package name : com.example
1053
1054    3)Java Project를 Spring Project로 변환
1055      -ProfileDemo1 Project > right-click > Configuration > Convert to Maven Project
1056        --Project : /ProfileDemo1
1057        --Group Id : ProfileDemo1
1058        --Artifact Id : ProfileDemo1
1059        --version : 0.0.1-SNAPSHOT
1060        --Packaging : jar
1061        --Finish
```

```
1062
1063      -ProfileDemo1 Project > right-click > Spring > Add Spring Project Nature
1064
1065      -pom.xml file에 Spring Context Dependency 추가하기
1066        <version>0.0.1-SNAPSHOT</version>
1067        <dependencies>
1068          <dependency>
1069            <groupId>org.springframework</groupId>
1070            <artifactId>spring-context</artifactId>
1071            <version>4.3.24.RELEASE</version>
1072          </dependency>
1073        </dependencies>
1074
1075      -pom.xml > right-click > Run As > Maven install
1076        [INFO] BUILD SUCCESS 확인
1077
1078    4)ServerInfo.java 생성
1079      -/src/com.example.ServerInfo.java
1080
1081        package com.example;
1082
1083        public class ServerInfo {
1084          private String ipNum;
1085          private String portNum;
1086          public String getIpNum() {
1087            return ipNum;
1088          }
1089          public void setIpNum(String ipNum) {
1090            this.ipNum = ipNum;
1091          }
1092          public String getPortNum() {
1093            return portNum;
1094          }
1095          public void setPortNum(String portNum) {
1096            this.portNum = portNum;
1097          }
1098        }
1099
1100    5)Java 설정 file 2개 생성
1101      -/src/com.example.ApplicationConfigDev.java
1102
1103        package com.example;
1104
1105        import org.springframework.context.annotation.Bean;
1106        import org.springframework.context.annotation.Configuration;
1107        import org.springframework.context.annotation.Profile;
1108
1109        @Configuration
1110        @Profile("dev")
1111        public class ApplicationConfigDev {
1112
1113          @Bean
1114          public ServerInfo serverInfo(){
1115            ServerInfo info = new ServerInfo();
```

```
1116                info.setIpNum("localhost");
1117                info.setPortNum("8080");
1118                return info;
1119            }
1120        }
1121
1122      -/src/com.example.ApplicationConfigRun.java
1123
1124        package com.example;
1125
1126        import org.springframework.context.annotation.Bean;
1127        import org.springframework.context.annotation.Configuration;
1128        import org.springframework.context.annotation.Profile;
1129
1130        @Configuration
1131        @Profile("run")
1132        public class ApplicationConfigRun {
1133
1134          @Bean
1135          public ServerInfo serverInfo(){
1136            ServerInfo info = new ServerInfo();
1137            info.setIpNum("192.168.56.5");
1138            info.setPortNum("80");
1139            return info;
1140          }
1141        }
1142
1143    6)MainClass 생성
1144      -/src/com.example.MainClass.java
1145
1146        package com.example;
1147        import java.util.Scanner;
1148
1149        import org.springframework.context.annotation.AnnotationConfigApplicationContext;
1150
1151        public class MainClass {
1152          public static void main(String[] args) {
1153            Scanner scan = new Scanner(System.in);
1154            System.out.print("Select dev or run : ");
1155            String config = scan.next();  //"dev" or "run"
1156
1157            AnnotationConfigApplicationContext ctx = new AnnotationConfigApplicationContext();
1158            ctx.getEnvironment().setActiveProfiles(config);
1159            ctx.register(ApplicationConfigDev.class, ApplicationConfigRun.class);
1160            ctx.refresh();
1161
1162            ServerInfo info = ctx.getBean("serverInfo", ServerInfo.class);
1163            System.out.println("IP : " + info.getIpNum());
1164            System.out.println("Port : " + info.getPortNum());
1165            ctx.close();
1166          }
1167        }
1168
1169    7)결과
```

1170   -MainClass.java > right-click > Run As > Java Application
1171   Select dev or run :
1172   입력시 dev를 넣으면 dev환경인 localhost/8080이 나오고, 만일 run이라고 넣으면 192.168.56.5/80이 나온
              다.