

```
1 HOL : 외부 파일을 이용한 설정
2 -----
3 Task1. Lab
4 1. In Package Explorer > right-click > New > Java Project
5   1)Project Name : EnvironmentDemo
6
7 2. src > right-click > New > Package
8   2)Package name : com.example
9
10 3. Java Project를 Spring Project로 변환
11   1)EnvironmentDemo Project > right-click > Configure > Convert to Maven Project
12     -Project : /EnvironmentDemo
13     -Group Id : EnvironmentDemo
14     -Artifact Id : EnvironmentDemo
15     -version : 0.0.1-SNAPSHOT
16     -Packaging : jar
17     -Finish
18
19   2)EnvironmentDemo Project > right-click > Spring > Add Spring Project Natur
20
21   3)pom.xml file에 Spring Context Dependency 추가하기
22     <version>0.0.1-SNAPSHOT</version>
23     <dependencies>
24       <dependency>
25         <groupId>org.springframework</groupId>
26         <artifactId>spring-context</artifactId>
27         <version>5.2.0.RELEASE</version>
28       </dependency>
29     </dependencies>
30
31   4)pom.xml > right-click > Run As > Maven install
32     [INFO] BUILD SUCCESS 확인
33
34 4. EnvironmentDemo/resources folder 생성
35   1)EnvironmentDemo project > right-click > Build Path > Configure Build Path
36   2)Source Tab > Add Folder
37   3)EnvironmentDemo click
38   4>Create New Folder > Folder name : resources > Finish > OK
39   5)EnvironmentDemo/resources(new) 확인
40   6)Apply and Close
41
42 5. resources > right-click > New > File
43   1)File name : admin.properties
44
45     admin.id=javaexpert
46     admin.pwd=12345678
47
48 6. com.example.AdminConnection.java 생성
49   package com.example;
50
51   import org.springframework.beans.factory.DisposableBean;
52   import org.springframework.beans.factory.InitializingBean;
```

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

```

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

```

```
155
156 10. 결과
157     javaexpert
158     12345678
159     setEnvironment()
160     afterPropertiesSet()
161     admin ID : javaexpert
162     admin PWD : 12345678
163     destroy()
164
165
166 -----
167 Task2. Lab
168 1. In Package Explorer > right-click > New > Java Project
169     1)Project Name : PropertyDemo
170
171 2. src > right-click > New > Package
172     1)Package name : com.example
173
174 3. POJO Class 작성
175     1)com.example.Hello
176
177     package com.example;
178
179     public class Hello{
180         private String name;
181         private Printer printer;
182         private List<String> names;
183
184         public Hello(){
185
186         public void setName(String name){
187             this.name = name;
188         }
189
190         public void setPrinter(Printer printer){
191             this.printer = printer;
192         }
193
194         public void setNames(List<String> list){
195             this.names = list;
196         }
197
198         public List<String> getNames(){
199             return this.names;
200         }
201
202         public String sayHello(){
203             return "Hello " + name;
204         }
205
206         public void print(){
```

```
207         this.printer.print(sayHello());
208     }
209 }
210
211 2)com.example > right-click > New > Interface
212 3)Interface name : Printer
213
214     package com.example;
215
216     public interface Printer{
217         void print(String message);
218     }
219
220 4)com.example > right-click > New > Class
221 5)Class Name : StringPrinter
222 6)Interfaces : com.example.Printer
223     package com.example;
224
225     public class StringPrinter implements Printer{
226         private StringBuffer buffer = new StringBuffer();
227
228         @Override
229         public void print(String message){
230             this.buffer.append(message);
231         }
232
233         public String toString(){
234             return this.buffer.toString();
235         }
236     }
237
238 7)com.example > right-click > New > Class
239 8)Class Name : ConsolePrinter
240 9)Interfaces : com.example.Printer
241     package com.example;
242
243     public class ConsolePrinter implements Printer{
244
245         @Override
246         public void print(String message){
247             System.out.println(message);
248         }
249     }
250
251 4. Java Project를 Spring Project로 변환
252 1)PropertyDemo Project > right-click > Configure > Convert to Maven Project
253     -Project : /PropertyDemo
254     -Group Id : PropertyDemo
255     -Artifact Id : PropertyDemo
256     -version : 0.0.1-SNAPSHOT
257     -Packaging : jar
258     -Finish
```

```
259
260 2)PropertyDemo Project > right-click > Spring > Add Spring Project Nature
261
262 3)pom.xml file에 Spring Context Dependency 추가하기
263     <version>0.0.1-SNAPSHOT</version>
264     <dependencies>
265         <dependency>
266             <groupId>org.springframework</groupId>
267             <artifactId>spring-context</artifactId>
268             <version>5.2.0.RELEASE</version>
269         </dependency>
270     </dependencies>
271
272 4pom.xml > right-click > Run As > Maven install
273     [INFO] BUILD SUCCESS 확인
274
275 5. PropertyDemo/resources folder 생성
276 1)PropertyDemo project > right-click > Build Path > Configure Build Path
277 2)Source Tab > Add Folder
278 3)PropertyDemo click
279 4)Create New Folder > Folder name : resources > Finish > OK
280 5)PropertyDemo/resources(new) 확인
281 6)Apply and Close
282
283 6. Bean Configuration XML 작성
284 1)PropertyDemo/resources > right-click > New > Other > Spring > Spring Bean Configuration File
285 2)File name : beans.xml > Finish
286
287     <?xml version="1.0" encoding="UTF-8"?>
288     <beans xmlns="http://www.springframework.org/schema/beans"
289         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
290         xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans.xsd">
291
292         <bean id="hello" class="com.example.Hello">
293             <property name="name" value="Spring" />
294             <property name="printer" ref="printer" />
295             <property name="names">
296                 <list>
297                     <value>AOP</value>
298                     <value>Spring</value>
299                     <value>DI</value>
300                 </list>
301             </property>
302         </bean>
303
304         <bean id="printer" class="com.example.StringPrinter" />
305         <bean id="consolePrinter" class="com.example.ConsolePrinter" />
306
307     </beans>
308
309 7. com.example.MainClass
```

```
310
311 package com.example;
312
313 import java.util.List;
314
315 import org.springframework.context.ApplicationContext;
316 import org.springframework.context.support.GenericXmlApplicationContext;
317
318 public class MainClass {
319     public static void main(String [] args){
320         ApplicationContext ctx = new GenericXmlApplicationContext("classpath:beans.xml");
321
322         Hello hello = (Hello)ctx.getBean("hello");
323         System.out.println(hello.sayHello());
324         hello.print();
325
326         Printer printer = ctx.getBean("printer", StringPrinter.class);
327         System.out.println(printer.toString());
328
329         List<String> list = hello.getNames();
330         for(String value : list){
331             System.out.println(value);
332         }
333     }
334 }
```

#### 336 8. 실행

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

#### 339 9. 결과

340 Hello Spring

341 Hello Spring

342 AOP

343 Spring

344 DI

#### 346 10. JUnit & Spring TestContext Framework으로 Test

347 1)src/test package 생성

348 2)/src/test/ > right-click > New > JUnit Test Case > Select [New JUnit 4 test]

349 3)Name : HelloTest > Finish

350 4)New JUnit Test Case창에서, Not now 선택 > OK

351 5)<https://mvnrepository.com>에서 'junit'로 검색

352 6)JUnit에서 4.12로 들어가서

353 7)복사해서 pom.xml로 붙여넣기

```
354 <!-- https://mvnrepository.com/artifact/junit/junit -->
355 <dependency>
356     <groupId>junit</groupId>
357     <artifactId>junit</artifactId>
358     <version>4.12</version>
359     <scope>test</scope>
360 </dependency>
```

361

```
362 8)https://mvnrepository.com에서 'spring-test'로 검색
363 9)Spring TestContext Framework에서
364 10)5.2.0.RELEASE로 들어가서
365 11)복사해서 pom.xml로 붙여넣기
366 <!-- https://mvnrepository.com/artifact/org.springframework/spring-test -->
367 <dependency>
368     <groupId>org.springframework</groupId>
369     <artifactId>spring-test</artifactId>
370     <version>5.2.0.RELEASE</version>
371     <scope>test</scope>
372 </dependency>
373
374 12)pom.xml > right-click > Run As > Maven install
375 [INFO] BUILD SUCCESS 확인
376 -만일 Error 발생하면 PropertyDemo > right-click > Maven > Update Project > OK
377 -pom.xml > right-click > Run As > Maven install
378 -[INFO] BUILD SUCCESS 확인
379
380 11. com.example.test.HelloTest.java
381 import static org.junit.Assert.assertEquals;
382 import static org.junit.Assert.assertSame;
383
384 import java.util.List;
385
386 import org.junit.Test;
387 import org.junit.runner.RunWith;
388 import org.springframework.beans.factory.annotation.Autowired;
389 import org.springframework.context.ApplicationContext;
390 import org.springframework.test.context.ContextConfiguration;
391 import org.springframework.test.context.junit4.SpringJUnit4ClassRunner;
392
393 import com.example.Hello;
394 import com.example.Printer;
395
396 @RunWith(SpringJUnit4ClassRunner.class)
397 @ContextConfiguration(locations="classpath:beans.xml")
398 public class HelloTest {
399     @Autowired
400     ApplicationContext ctx;
401
402     @Test
403     public void test() {
404         Hello hello = (Hello)ctx.getBean("hello");
405         assertEquals("Hello Spring", hello.sayHello());
406         hello.print();
407
408         Printer printer = (Printer)ctx.getBean("printer");
409         assertEquals("Hello Spring", printer.toString());
410     }
411
412     @Test
413     public void test2(){
```



```

414     Hello hello = (Hello)ctx.getBean("hello");
415
416     Hello hello2 = ctx.getBean("hello", Hello.class);
417     assertSame(hello, hello2);
418
419     assertEquals(3, hello2.getNames().size());
420 }
421 }
422
423 1)만일 해당 객체를 찾을 수 없다는 오류가 계속 발생하면
424     -해당 Project > right-click > Build Path > Libraries tab
425     -spring-test-5.2.0.RELEASE.jar 선택 후 [Remove] 로 삭제
426     -[Add External JARs...] Click
427     -%M2_HOME%\repository\org\springframework\spring-test\5.2.0.RELEASE 선택할 것
428     -[Order and Export] tab에서 spring-test-5.2.0.RELEASE.jar 선택 후 [Up] button을 클릭
429     -해당 Project/src 바로 아래까지 올리고 [Apply and Close] Click
430
431 2)right-click > Run As > Junit Test
432 3)결과 -> Junit View에 초록색 bar
433
434 12. resources/value.properties 생성
435
436 myname=Spring
437 myprinter=printer
438 value1=HTML5
439 value2=CSS3
440 value3=JavaScript
441
442 13. ./resources/beans.xml 수정하기
443 1)beans.xml에서 [Namespaces] tab
444 2)목록에서 'context-http://www.springframework.org/schema/context' check
445 3)<context:property-placeholder />를 사용하기 위해서 다음과 같이 수정
446
447 <context:property-placeholder
448     location="classpath:value.properties" />
449
450 <bean id="hello" class="com.example.Hello">
451     <property name="name" value="${myname}" />
452     <property name="printer" ref="${myprinter}" />
453     <property name="names">
454         <list>
455             <value>${value1}</value>
456             <value>${value2}</value>
457             <value>${value3}</value>
458         </list>
459     </property>
460 </bean>
461
462 <bean id="printer" class="com.example.StringPrinter" />
463 <bean id="consolePrinter" class="com.example.ConsolePrinter" />
464
465 14. Test

```

```
466 1)com.example.MainClass.java
467 2)right-click > Run As > Java Application
468     Hello Spring
469     Hello Spring
470     HTML5
471     CSS3
472     JavaScript
473
474 3)/src/test/java/HelloTest.java
475     -right-click > Run As > JUnit Test
476     -Green Bar
477
478 15. resources/value.properties 수정
479     myname=Spring
480     myprinter=printer
481     value1=JUnit
482     value2=AOP
483     value3=DI
484     printer1=stringPrinter
485     printer2=consolePrinter
486
487 16. Hello.java code 수정
488     1)com.example/Hello.java
489
490     package com.example;
491
492     import java.util.List;
493
494     import org.springframework.beans.factory.annotation.Value;
495     import org.springframework.stereotype.Component;
496     import javax.annotation.Resource;
497
498     @Component("hello")
499     public class Hello {
500         @Value("${myname}")
501         private String name;
502
503         @Resource(name="${printer1}")
504         private Printer printer;
505
506         @Value("${value1}, ${value2}, ${value3}")
507         private List<String> names;
508
509         public List<String> getNames(){
510             return names;
511         }
512         public String sayHello(){
513             return "Hello " + name;
514         }
515
516         public void print(){
517             this.printer.print(sayHello());
```

```
518     }
519 }
520
521 17. StringPrinter.java 수정
522 package com.example;
523
524 import org.springframework.stereotype.Component;
525
526 @Component("stringPrinter")
527 public class StringPrinter implements Printer {
528     private StringBuffer buffer = new StringBuffer();
529
530     @Override
531     public void print(String message) {
532         this.buffer.append(message);
533     }
534
535     @Override
536     public String toString(){
537         return this.buffer.toString();
538     }
539 }
540
541 18. beans.xml 수정하기
542
543 <?xml version="1.0" encoding="UTF-8"?>
544 <beans xmlns="http://www.springframework.org/schema/beans"
545     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
546     xmlns:context="http://www.springframework.org/schema/context"
547     xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans.xsd
548     http://www.springframework.org/schema/context
http://www.springframework.org/schema/context/spring-context-3.2.xsd">
549
550     <context:property-placeholder location="classpath:value.properties "/>
551     <context:component-scan base-package="com.example" />
552 </beans>
553
554 19. MainClass.java 수정하기
555
556 package com.example;
557
558 import java.util.List;
559
560 import org.springframework.context.ApplicationContext;
561 import org.springframework.context.support.GenericXmlApplicationContext;
562
563 public class MainClass {
564     public static void main(String [] args){
565         ApplicationContext ctx = new GenericXmlApplicationContext("classpath:beans.xml");
566
567         Hello hello = (Hello)ctx.getBean("hello");
```

```
568     System.out.println(hello.sayHello());
569     hello.print();
570
571     Printer printer = ctx.getBean("stringPrinter", StringPrinter.class);
572     System.out.println(printer.toString());
573
574     List<String> list = hello.getNames();
575     for(String value : list){
576         System.out.println(value);
577     }
578 }
579 }
```

## 581 20. 실행

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

583

584 2) 결과

585 Hello Spring

586 Hello Spring

587 JUnit, AOP, DI

588

589

590 -----

## 591 Task 3. Lab

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

593 1) Project Name : PropertyDemo1

594

595 2. /src/ right-click > New > Package

596 1) Package name : com.example

597

598 3. /src/com.example.AdminConnection.java 생성

599 package com.example;

600

601 public class AdminConnection {

602 private String adminId;

603 private String adminPwd;

604 private String subAdminId;

605 private String subAdminPwd;

606

607 public String getAdminId() {

608 return adminId;

609 }

610

611 public void setAdminId(String adminId) {

612 this.adminId = adminId;

613 }

614

615 public String getAdminPwd() {

616 return adminPwd;

617 }

618

619 public void setAdminPwd(String adminPwd) {

```
620     this.adminPwd = adminPwd;
621 }
622
623 public String getSubAdminId() {
624     return subAdminId;
625 }
626
627 public void setSubAdminId(String subAdminId) {
628     this.subAdminId = subAdminId;
629 }
630
631 public String getSubAdminPwd() {
632     return subAdminPwd;
633 }
634
635 public void setSubAdminPwd(String subAdminPwd) {
636     this.subAdminPwd = subAdminPwd;
637 }
638 }
```

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

641 1)PropertyDemo1 Project > right-click > Configure > Convert to Maven Project

642 -Project : /PropertyDemo1

643 -Group Id : PropertyDemo1

644 -Artifact Id : PropertyDemo1

645 -version : 0.0.1-SNAPSHOT

646 -Packaging : jar

647 -Finish

648

649 2)PropertyDemo1 Project > right-click > Spring > Add Spring Project Nature

650

651 3)pom.xml file에 Spring Context Dependency 추가하기

652 <version>0.0.1-SNAPSHOT</version>

653 <dependencies>

654 <dependency>

655 <groupId>org.springframework</groupId>

656 <artifactId>spring-context</artifactId>

657 <version>5.2.0.RELEASE</version>

658 </dependency>

659 </dependencies>

660

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

662 [INFO] BUILD SUCCESS 확인

663

#### 664 5. PropertyDemo/resources folder 생성

665 1)PropertyDemo1 project > right-click > Build Path > Configure Build Path

666 2)Source Tab > Add Folder

667 3)PropertyDemo1 click

668 4>Create New Folder > Folder name : resources > Finish > OK

669 5)PropertyDemo1/resources(new) 확인

670 6)Apply and Close

671

```
672 6. ./resources 두 개의 properties file 생성
673
674 <admin.properties>
675     admin.id=javaexpert
676     admin.pwd=12345678
677
678 <sub.admin.properties>
679     sub.admin.id=javasoft
680     sub.admin.pwd=987654321
681
682 7. Bean Configuration XML 작성
683 1)PropertyDemo1/resources > right-click > New > Other > Spring > Spring Bean Configuration File
684 2)File name : beans.xml > Finish
685 3)Namespace tab에서 context - http://www.springframework.org/schema/context check
686
687 <?xml version="1.0" encoding="UTF-8"?>
688 <beans xmlns="http://www.springframework.org/schema/beans"
689     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
690     xmlns:context="http://www.springframework.org/schema/context"
691     xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans.xsd
692     http://www.springframework.org/schema/context
http://www.springframework.org/schema/context/spring-context-3.2.xsd">
693
694     <context:property-placeholder location="classpath:admin.properties,
        classpath:sub.admin.properties" />
695
696     <bean id="adminConnection" class="com.example.AdminConnection">
697         <property name="adminId">
698             <value>${admin.id}</value>
699         </property>
700         <property name="adminPwd">
701             <value>${admin.pwd}</value>
702         </property>
703         <property name="subAdminId">
704             <value>${sub.admin.id}</value>
705         </property>
706         <property name="subAdminPwd">
707             <value>${sub.admin.pwd}</value>
708         </property>
709     </bean>
710 </beans>
711
712 8. ./src/com.example.MainClass.java 생성
713 package com.example;
714
715 import org.springframework.context.support.AbstractApplicationContext;
716 import org.springframework.context.support.GenericXmlApplicationContext;
717
718 public class MainClass {
719     public static void main(String [] args){
720         AbstractApplicationContext ctx =
```

```
721         new GenericXmlApplicationContext("classpath:beans.xml");
722         AdminConnection connection = ctx.getBean("adminConnection", AdminConnection.class);
723         System.out.println("admin ID : " + connection.getAdminId());
724         System.out.println("admin PWD : " + connection.getAdminPwd());
725         System.out.println("sub admin ID : " + connection.getSubAdminId());
726         System.out.println("sub admin PWD : " + connection.getSubAdminPwd());
727
728         ctx.close();
729     }
730 }
```

## 731

### 732 9. 결과

```
733 1)MainClass.java > right-click > Run As > Java Application
734     admin ID : javaexpert
735     admin PWD : 12345678
736     sub admin ID : javasoft
737     sub admin PWD : 987654321
```

```
738
739
740 -----
```

### 741 Task 4. Lab

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

743 1)Project Name : PropertyDemo2

744

745 2. /src/ right-click > New > Package

746 1)Package name : com.example

747

748 3. /src/com.example.AdminConnection.java 생성

749 package com.example;

750

751 public class AdminConnection {

752 private String adminId;

753 private String adminPwd;

754 private String subAdminId;

755 private String subAdminPwd;

756

757 public String getAdminId() {

758 return adminId;

759 }

760

761 public void setAdminId(String adminId) {

762 this.adminId = adminId;

763 }

764

765 public String getAdminPwd() {

766 return adminPwd;

767 }

768

769 public void setAdminPwd(String adminPwd) {

770 this.adminPwd = adminPwd;

771 }

772

```
773     public String getSubAdminId() {  
774         return subAdminId;  
775     }  
776  
777     public void setSubAdminId(String subAdminId) {  
778         this.subAdminId = subAdminId;  
779     }  
780  
781     public String getSubAdminPwd() {  
782         return subAdminPwd;  
783     }  
784  
785     public void setSubAdminPwd(String subAdminPwd) {  
786         this.subAdminPwd = subAdminPwd;  
787     }  
788 }  
789
```

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

```
791 1)PropertyDemo1 Project > right-click > Configure > Convert to Maven Project  
792 -Project : /PropertyDemo2  
793 -Group Id : PropertyDemo2  
794 -Artifact Id : PropertyDemo2  
795 -version : 0.0.1-SNAPSHOT  
796 -Packaging : jar  
797 -Finish  
798
```

```
799 2)PropertyDemo2 Project > right-click > Spring > Add Spring Project Nature  
800
```

#### 801 3)pom.xml 파일에 Spring Context Dependency 추가하기

```
802 <version>0.0.1-SNAPSHOT</version>  
803 <dependencies>  
804 <dependency>  
805 <groupId>org.springframework</groupId>  
806 <artifactId>spring-context</artifactId>  
807 <version>5.2.0.RELEASE</version>  
808 </dependency>  
809 </dependencies>  
810
```

#### 811 4pom.xml > right-click > Run As > Maven install

```
812 [INFO] BUILD SUCCESS 확인  
813
```

#### 814 5. PropertyDemo/resources folder 생성

```
815 1)PropertyDemo2 project > right-click > Build Path > Configure Build Path
```

```
816 2)Source Tab > Add Folder
```

```
817 3)PropertyDemo2 click
```

```
818 4)Create New Folder > Folder name : resources > Finish > OK
```

```
819 5)PropertyDemo2/resources(new) 확인
```

```
820 6)Apply and Close  
821
```

#### 822 6. ./resources 두 개의 properties file 생성

```
823  
824 <admin.properties>
```



```
825     admin.id=javaexpert
826     admin.pwd=12345678
827
828     <sub.admin.properties>
829         sub.admin.id=javasoft
830         sub.admin.pwd=987654321
831
832 7. /src/com.example.ApplicationConfig.java>
833     package com.example;
834
835     import org.springframework.beans.factory.annotation.Value;
836     import org.springframework.context.annotation.Bean;
837     import org.springframework.context.annotation.Configuration;
838     import org.springframework.context.support.PropertySourcesPlaceholderConfigurer;
839     import org.springframework.core.io.ClassPathResource;
840     import org.springframework.core.io.Resource;
841
842     @Configuration
843     public class ApplicationConfig {
844         @Value("${admin.id}")
845         private String adminId;
846         @Value("${admin.pwd}")
847         private String adminPwd;
848         @Value("${sub.admin.id}")
849         private String subAdminId;
850         @Value("${sub.admin.pwd}")
851         private String subAdminPwd;
852
853         @Bean
854         public static PropertySourcesPlaceholderConfigurer Properties(){
855             PropertySourcesPlaceholderConfigurer configurer =
856                 new PropertySourcesPlaceholderConfigurer();
857
858             Resource [] locations = new Resource[2];
859             locations[0] = new ClassPathResource("admin.properties");
860             locations[1] = new ClassPathResource("sub.admin.properties");
861             configurer.setLocations(locations);
862
863             return configurer;
864         }
865
866         @Bean
867         public AdminConnection adminConfig(){
868             AdminConnection adminConnection = new AdminConnection();
869             adminConnection.setAdminId(adminId);
870             adminConnection.setAdminPwd(adminPwd);
871             adminConnection.setSubAdminId(subAdminId);
872             adminConnection.setSubAdminPwd(subAdminPwd);
873             return adminConnection;
874         }
875     }
876
```

```
877 8. /src/com.example.MainClass.java
878 package com.example;
879
880 import org.springframework.context.annotation.AnnotationConfigApplicationContext;
881
882 public class MainClass {
883     public static void main(String[] args) {
884         AnnotationConfigApplicationContext ctx =
885             new AnnotationConfigApplicationContext(ApplicationConfig.class);
886         AdminConnection conn = ctx.getBean("adminConfig", AdminConnection.class);
887
888         System.out.println("admin ID : " + conn.getAdminId());
889         System.out.println("admin PWD : " + conn.getAdminPwd());
890         System.out.println("sub admin ID : " + conn.getSubAdminId());
891         System.out.println("sub admin PWD : " + conn.getSubAdminPwd());
892     }
893 }
```

#### 895 9. 결과

```
896 1)MainClass.java > right-click > Run As > Java Application
897     admin ID : javaexpert
898     admin PWD : 12345678
899     sub admin ID : javasoft
900     sub admin PWD : 987654321
901
902
903 -----
```

#### 904 Task 5. Lab

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

906 1)Project Name : ProfileDemo

#### 908 2. Package 생성

909 1)/src/ > right-click > New > Package

910 2)Package name : com.example

#### 912 3. Java Project를 Spring Project로 변환

913 1)ProfileDemo Project > right-click > Configure > Convert to Maven Project

914 -Project : /ProfileDemo

915 -Group Id : ProfileDemo

916 -Artifact Id : ProfileDemo

917 -version : 0.0.1-SNAPSHOT

918 -Packaging : jar

919 -Finish

921 2)ProfileDemo Project > right-click > Spring > Add Spring Project Nature

923 3)pom.xml file에 Spring Context Dependency 추가하기

924 <version>0.0.1-SNAPSHOT</version>

925 <dependencies>

926 <dependency>

927 <groupId>org.springframework</groupId>

928 <artifactId>spring-context</artifactId>

```
929         <version>5.2.0.RELEASE</version>
930     </dependency>
931 </dependencies>
932
933 4)pom.xml > right-click > Run As > Maven install
934 [INFO] BUILD SUCCESS 확인
935
936 4. ProfileDemo/resources folder 생성
937 1)ProfileDemo project > right-click > Build Path > Configure Build Path
938 2)Source Tab > Add Folder
939 3)ProfileDemo click
940 4>Create New Folder > Folder name : resources > Finish > OK
941 5)ProfileDemo/resources(new) 확인
942 6)Apply and Close
943
944 5. ServerInfo.java 생성
945 1)/src/com.example.ServerInfo.java
946
947     package com.example;
948
949     public class ServerInfo {
950         private String ipNum;
951         private String portNum;
952         public String getIpNum() {
953             return ipNum;
954         }
955         public void setIpNum(String ipNum) {
956             this.ipNum = ipNum;
957         }
958         public String getPortNum() {
959             return portNum;
960         }
961         public void setPortNum(String portNum) {
962             this.portNum = portNum;
963         }
964     }
965
966 6. XML 설정 file 2개 생성
967 1)/resource > right-click > New > Other > Spring > Spring Bean Configuration File
968 2)File name : run.xml
969
970     <?xml version="1.0" encoding="UTF-8"?>
971     <beans xmlns="http://www.springframework.org/schema/beans"
972         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
973         xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans.xsd"
974         profile="run"> <---이것이 핵심
975
976         <bean id="serverInfo" class="com.example.ServerInfo">
977             <property name="ipNum" value="192.168.56.5" />
978             <property name="portNum" value="80" />
979         </bean>
```

```

980     </beans>
981
982 3)/resource > right-click > New > Other > Spring > Spring Bean Configuration File
983 4)File name : dev.xml
984
985     <?xml version="1.0" encoding="UTF-8"?>
986     <beans xmlns="http://www.springframework.org/schema/beans"
987         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
988         xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans.xsd"
989         profile="dev">     <---이것이 핵심
990
991         <bean id="serverInfo" class="com.example.ServerInfo">
992             <property name="ipNum" value="localhost" />
993             <property name="portNum" value="8080" />
994         </bean>
995     </beans>
996
997 7. MainClass 생성
998 1)/src/com.example.MainClass.java
999
1000     package com.example;
1001     import java.util.Scanner;
1002
1003     import org.springframework.context.support.GenericXmlApplicationContext;
1004
1005     public class MainClass {
1006         public static void main(String[] args) {
1007             Scanner scan = new Scanner(System.in);
1008             System.out.print("Select dev or run : ");
1009             String config = scan.next(); //"dev" or "run"
1010
1011             GenericXmlApplicationContext ctx = new GenericXmlApplicationContext();
1012             ctx.getEnvironment().setActiveProfiles(config);
1013             ctx.load("dev.xml", "run.xml");
1014             ctx.refresh();
1015
1016             ServerInfo info = ctx.getBean("serverInfo", ServerInfo.class);
1017             System.out.println("IP : " + info.getIpNum());
1018             System.out.println("Port : " + info.getPortNum());
1019             ctx.close();
1020         }
1021     }
1022
1023 8. 결과
1024 1)MainClass.java > right-click > Run As > Java Application
1025     Select dev or run :
1026 2)입력시 dev를 넣으면 dev환경인 localhost/8080이 나오고, 만일 run이라고 넣으면 192.168.56.5/80이 나
    온다.
1027
1028
1029 -----

```

```
1030 Task 6. Lab
1031 1. In Package Explorer > right-click > New > Java Project
1032 1)Project Name : ProfileDemo1
1033
1034 2. Package 생성
1035 1)/src/ > right-click > New > Package
1036 2)Package name : com.example
1037
1038 3. Java Project를 Spring Project로 변환
1039 1)ProfileDemo1 Project > right-click > Configure > Convert to Maven Project
1040 -Project : /ProfileDemo1
1041 -Group Id : ProfileDemo1
1042 -Artifact Id : ProfileDemo1
1043 -version : 0.0.1-SNAPSHOT
1044 -Packaging : jar
1045 -Finish
1046
1047 2)ProfileDemo1 Project > right-click > Spring > Add Spring Project Nature
1048
1049 3)pom.xml file에 Spring Context Dependency 추가하기
1050 <version>0.0.1-SNAPSHOT</version>
1051 <dependencies>
1052 <dependency>
1053 <groupId>org.springframework</groupId>
1054 <artifactId>spring-context</artifactId>
1055 <version>5.2.0.RELEASE</version>
1056 </dependency>
1057 </dependencies>
1058
1059 4)pom.xml > right-click > Run As > Maven install
1060 [INFO] BUILD SUCCESS 확인
1061
1062 4. ServerInfo.java 생성
1063 1)/src/com.example.ServerInfo.java
1064
1065 package com.example;
1066
1067 public class ServerInfo {
1068     private String ipNum;
1069     private String portNum;
1070     public String getIpNum() {
1071         return ipNum;
1072     }
1073     public void setIpNum(String ipNum) {
1074         this.ipNum = ipNum;
1075     }
1076     public String getPortNum() {
1077         return portNum;
1078     }
1079     public void setPortNum(String portNum) {
1080         this.portNum = portNum;
1081     }
1081 }
```

```
1082     }
1083
1084 5. Java 설정 file 2개 생성
1085 1)/src/com.example.ApplicationConfigDev.java
1086
1087     package com.example;
1088
1089     import org.springframework.context.annotation.Bean;
1090     import org.springframework.context.annotation.Configuration;
1091     import org.springframework.context.annotation.Profile;
1092
1093     @Configuration
1094     @Profile("dev")
1095     public class ApplicationConfigDev {
1096
1097         @Bean
1098         public ServerInfo serverInfo(){
1099             ServerInfo info = new ServerInfo();
1100             info.setIpNum("localhost");
1101             info.setPortNum("8080");
1102             return info;
1103         }
1104     }
1105
1106 2)/src/com.example.ApplicationConfigRun.java
1107
1108     package com.example;
1109
1110     import org.springframework.context.annotation.Bean;
1111     import org.springframework.context.annotation.Configuration;
1112     import org.springframework.context.annotation.Profile;
1113
1114     @Configuration
1115     @Profile("run")
1116     public class ApplicationConfigRun {
1117
1118         @Bean
1119         public ServerInfo serverInfo(){
1120             ServerInfo info = new ServerInfo();
1121             info.setIpNum("192.168.56.5");
1122             info.setPortNum("80");
1123             return info;
1124         }
1125     }
1126
1127 6. MainClass 생성
1128 1)/src/com.example.MainClass.java
1129
1130     package com.example;
1131     import java.util.Scanner;
1132
1133     import org.springframework.context.annotation.AnnotationConfigApplicationContext;
```

```
1134
1135     public class MainClass {
1136         public static void main(String[] args) {
1137             Scanner scan = new Scanner(System.in);
1138             System.out.print("Select dev or run : ");
1139             String config = scan.next(); //"dev" or "run"
1140
1141             AnnotationConfigApplicationContext ctx = new AnnotationConfigApplicationContext();
1142             ctx.getEnvironment().setActiveProfiles(config);
1143             ctx.register(ApplicationConfigDev.class, ApplicationConfigRun.class);
1144             ctx.refresh();
1145
1146             ServerInfo info = ctx.getBean("serverInfo", ServerInfo.class);
1147             System.out.println("IP : " + info.getIpNum());
1148             System.out.println("Port : " + info.getPortNum());
1149             ctx.close();
1150         }
1151     }
1152
1153 7. 결과
1154     1)MainClass.java > right-click > Run As > Java Application
1155         Select dev or run :
1156     2)입력시 dev를 넣으면 dev환경인 localhost/8080이 나오고, 만일 run이라고 넣으면 192.168.56.5/80이 나
        온다.
```