```
HOL : 외부 파일을 이용한 설정
-----------------------------------------
Task1. Lab
1. In Package Explorer > right-click > New > Java Project
  1)Project Name : EnvironmentDemo
  2)JRE
    -Select [Use default JRE 'jdk-13.0.2' and workspace compiler preferences]
  3)Next
  4)Uncheck [Create module-info.java file]
  5)Finish


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


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

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

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

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


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

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

```
59          <version>1.18.12</version>
60          <scope>provided</scope>
61        </dependency>
62      </dependencies>
63
64    5)pom.xml > right-click > Run As > Maven install
65      [INFO] BUILD SUCCESS 확인
66
67
68  5. EnvironmentDemo/resources folder 생성
69    1)EnvironmentDemo project > right-click > New > Source Folder
70    2)Folder name : resources
71    3)Finish
72
73
74  6. resources > right-click > New > File
75    1)File name : admin.properties
76    2)Finish
77
78      admin.id=javaexpert
79      admin.pwd=12345678
80
81
82  7. com.example.AdminConnection 생성
83    1)com.example > right-click > New > Class
84    2)Name : AdminConnection
85    3)Finish
86
87      package com.example;
88
89      import org.springframework.beans.factory.DisposableBean;
90      import org.springframework.beans.factory.InitializingBean;
91      import org.springframework.context.EnvironmentAware;
92      import org.springframework.core.env.Environment;
93
94      import lombok.Getter;
95      import lombok.Setter;
96
97      public class AdminConnection implements EnvironmentAware, InitializingBean,
        DisposableBean {
98        @Setter
99        private Environment env;
100       @Getter
101       @Setter
102       private String adminId;
103       @Getter
104       @Setter
105       private String adminPwd;
106
107       @Override
108       public void destroy() throws Exception {
109         System.out.println("destroy()");
110       }
111
112       @Override
113       public void afterPropertiesSet() throws Exception {
114         System.out.println("afterPropertiesSet()");
115         setAdminId(env.getProperty("admin.id"));
```

```
        setAdminPwd(env.getProperty("admin.pwd"));
      }

      // bean이 생성되기 전에 callback 으로 호출됨. 가장 먼저 호출됨.
      // MainClass에서 사용하는 env 정보가 넘어옴.
      @Override
      public void setEnvironment(Environment env) {
        System.out.println("setEnvironment()");
        setEnv(env);
      }
    }


8. resources > right-click > New > Spring Bean Configuration File
  1)File name : beans.xml
  2)Finish

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

      <bean id="adminConnection" class="com.example.AdminConnection" />

  </beans>


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

    package com.example;

    import java.io.IOException;

    import org.springframework.context.ConfigurableApplicationContext;
    import org.springframework.context.support.GenericXmlApplicationContext;
    import org.springframework.core.env.ConfigurableEnvironment;
    import org.springframework.core.env.MutablePropertySources;
    import org.springframework.core.io.support.ResourcePropertySource;

    public class MainClass {
      public static void main(String [] args){
        ConfigurableApplicationContext ctx = new GenericXmlApplicationContext();
        ConfigurableEnvironment env = ctx.getEnvironment();

        MutablePropertySources propertySouces = env.getPropertySources();
        //내가 원하는 정보를 얻을 때까지 모든 propertySources를 앞에서 부터 차례로 모두 검색함.
        try{
          propertySouces.addLast(new
          ResourcePropertySource("classpath:admin.properties"));  //property 추가

          System.out.println(env.getProperty("admin.id"));    //property 추출
          System.out.println(env.getProperty("admin.pwd"));
        }catch(IOException ex){}
```

```
172           GenericXmlApplicationContext gCtx = (GenericXmlApplicationContext)ctx;
173           gCtx.load("classpath:beans.xml");
174           gCtx.refresh();
175
176           AdminConnection adminConnection = gCtx.getBean("adminConnection",
              AdminConnection.class);
177           System.out.println("admin ID : " + adminConnection.getAdminId());
178           System.out.println("admin PWD : " + adminConnection.getAdminPwd());
179
180           gCtx.close();
181           ctx.close();
182         }
183       }
184
185
186   10. 실행
187     1)MainClass > right-click > Run As > Java Application
188
189       javaexpert
190       12345678
191       setEnvironment()
192       afterPropertiesSet()
193       admin ID : javaexpert
194       admin PWD : 12345678
195       destroy()
196
197
198
199   ------------------------------------------------
200   Task2. Lab
201   1. In Package Explorer > right-click > New > Java Project
202     1)Project Name : PropertyDemo
203     2)JRE
204       -Select [Use default JRE 'jdk-13.0.2' and workspace compiler preferences]
205     3)Next
206     4)Uncheck [Create module-info.java file]
207     5)Finish
208
209
210   2. src > right-click > New > Package
211     1)Name : com.example
212     2)Finish
213
214
215   3. POJO Class 작성
216     1)com.example > right-click > New > Class
217     2)Name : Hello
218     3)Finish
219
220       package com.example;
221
222       import java.util.List;
223
224       public class Hello {
225         private String name;
226         private Printer printer;
227         private List<String> names;
228
```

```
        public String sayHello(){
          return "Hello " + name;
        }

        public void print(){
          this.printer.print(sayHello());
        }
      }

  4)com.example > right-click > New > Interface
  5)Name : Printer
  6)Finish

    package com.example;

    public interface Printer{
      void print(String message);
    }

  7)com.example > right-click > New > Class
  8)Name : StringPrinter
  9)Interfaces : com.example.Printer
  10)Finish

    package com.example;

    public class StringPrinter implements Printer{
      private StringBuffer buffer = new StringBuffer();

      @Override
      public void print(String message){
        this.buffer.append(message);
      }

      public String toString(){
        return this.buffer.toString();
      }
    }

  11)com.example > right-click > New > Class
  12)Name : ConsolePrinter
  13)Interfaces : com.example.Printer
  14)Finish

    package com.example;

    public class ConsolePrinter implements Printer{

      @Override
      public void print(String message){
        System.out.println(message);
      }
    }


4. Java Project를 Spring Project로 변환
  1)PropertyDemo Project > right-click > Configure > Convert to Maven Project
    -Project : /PropertyDemo
```

```
    -Group Id : PropertyDemo
    -Artifact Id : PropertyDemo
    -version : 0.0.1-SNAPSHOT
    -Packaging : jar
    -Finish

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

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

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


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

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

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


6. Hello.java에 lombok Annotation 붙이기
  1)Hello.java

    package com.example;

    import java.util.List;

    import lombok.Getter;
    import lombok.NoArgsConstructor;
    import lombok.Setter;

```

```
345     @NoArgsConstructor
346     @Setter
347     public class Hello {
348       private String name;
349       private Printer printer;
350       @Getter private List<String> names;
351
352       public String sayHello(){
353         return "Hello " + name;
354       }
355
356       public void print(){
357         this.printer.print(sayHello());
358       }
359     }
360
361
362 7. PropertyDemo/resources folder 생성
363   1)PropertyDemo project > right-click > New > Source Folder
364   2)Folder name : resources
365   3)Finish
366
367
368 8. Bean Configuration XML 작성
369   1)PropertyDemo/resources > right-click > New > Spring Bean Configuration File
370   2)File name : beans.xml
371   3)Finish
372
373     <?xml version="1.0" encoding="UTF-8"?>
374     <beans xmlns="http://www.springframework.org/schema/beans"
375       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
376       xsi:schemaLocation="http://www.springframework.org/schema/beans
        http://www.springframework.org/schema/beans/spring-beans.xsd">
377
378       <bean id="hello" class="com.example.Hello">
379         <property name="name" value="Spring" />
380         <property name="printer" ref="printer" />
381         <property name="names">
382           <list>
383             <value>AOP</value>
384             <value>Spring</value>
385             <value>DI</value>
386           </list>
387         </property>
388       </bean>
389
390       <bean id="printer" class="com.example.StringPrinter" />
391       <bean id="consolePrinter" class="com.example.ConsolePrinter" />
392
393     </beans>
394
395
396 9. com.example.MainClass Class 생성
397   1)com.example > right-click > New > Class
398   2)Name : MainClass
399   3)Finish
400
401   package com.example;
```

```
402
403   import org.springframework.context.ApplicationContext;
404   import org.springframework.context.support.GenericXmlApplicationContext;
405
406   public class MainClass {
407     public static void main(String [] args){
408       ApplicationContext ctx = new GenericXmlApplicationContext("classpath:beans.xml");
409
410       Hello hello = (Hello)ctx.getBean("hello");
411       System.out.println(hello.sayHello());
412       hello.print();
413
414       Printer printer = ctx.getBean("printer", StringPrinter.class);
415       System.out.println(printer.toString());
416
417       hello.getNames().forEach(value -> System.out.println(value));
418     }
419   }
420
421
422 10. 실행
423   1)MainClass > right-click > Run As > Java Application
424
425     Hello Spring
426     Hello Spring
427     AOP
428     Spring
429     DI
430
431
432 11. JUnit 5 & Spring TestContext Framework으로 Test
433   1)src > right-click > New > Package
434   2)Name : com.example.test
435   3)Finish
436   4)com.example.test > right-click > New > JUnit Test Case > Select [New JUnit Jupiter test]
437   5)Name : HelloJUnitTest
438   6)Finish
439   7)New JUnit Test Case창에서, [Perform the following action: Add JUnit 5 library to the build
      path]
440   8)OK
441   9)https://mvnrepository.com에서 'spring-test'로 검색
442   10)Spring TestContext Framework에서
443   11)5.2.5.RELEASE로 들어가서
444   12)복사해서 pom.xml로 붙여넣기
445     <!-- https://mvnrepository.com/artifact/org.springframework/spring-test -->
446     <dependency>
447       <groupId>org.springframework</groupId>
448       <artifactId>spring-test</artifactId>
449       <version>5.2.5.RELEASE</version>
450       <scope>test</scope>
451     </dependency>
452
453   13)pom.xml > right-click > Run As > Maven install
454     [INFO] BUILD SUCCESS 확인
455     -만일 Error 발생하면 PropertyDemo > right-click > Maven > Update Project > OK
456     -pom.xml > right-click > Run As > Maven install
457     -[INFO] BUILD SUCCESS 확인
458
```

12. com.example.test.HelloJUnitTest

```
package com.example.test;

import static org.junit.jupiter.api.Assertions.assertEquals;
import static org.junit.jupiter.api.Assertions.assertSame;

import org.junit.jupiter.api.Test;
import org.junit.jupiter.api.extension.ExtendWith;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.context.ApplicationContext;
import org.springframework.test.context.ContextConfiguration;
import org.springframework.test.context.junit.jupiter.SpringExtension;

import com.example.Hello;
import com.example.Printer;

@ExtendWith(SpringExtension.class)
@ContextConfiguration(locations = "classpath:beans.xml")
class HelloJUnitTest {
  @Autowired
  ApplicationContext ctx;

  @Test
  public void test() {
    Hello hello = (Hello) ctx.getBean("hello");
    assertEquals("Hello Spring", hello.sayHello());
    hello.print();

    Printer printer = (Printer) ctx.getBean("printer");
    assertEquals("Hello Spring", printer.toString());
  }

  @Test
  public void test2() {
    Hello hello = (Hello) ctx.getBean("hello");

    Hello hello2 = ctx.getBean("hello", Hello.class);
    assertSame(hello, hello2);

    assertEquals(3, hello2.getNames().size());
  }
}
```

1)만일 해당 객체를 찾을 수 없다는 오류가 계속 발생하면
  -해당 Project > right-click > Build Path > Libraries tab
  -spring-test-5.2.5.RELEASE.jar 선택 후 [Remove] 로 삭제
  -[Add External JARs...] Click
  -%M2_HOME%\repository\org\springframework\spring-test\5.2.5.RELEASE 선택할 것
  -[Order and Export] tab에서 spring-test-5.2.5.RELEASE.jar 선택 후 [Up] button을 클릭
  -해당 Project/src 바로 아래까지 올리고 [Apply and Close] Click

2)right-click > Run As > Junit Test
3)결과 -> Junit View에 초록색 bar

13. resources/value.properties 생성

```
myname=Spring
myprinter=printer
value1=HTML5
value2=CSS3
value3=JavaScript
```

14. /resources/beans.xml 수정하기

1)beans.xml에서 [Namespaces] tab

2)목록에서 'context-http://www.springframework.org/schema/context' check

3)<context:property-placeholder />를 사용하기 위해서 다음과 같이 수정

```
<context:property-placeholder
      location="classpath:value.properties" />

<bean id="hello" class="com.example.Hello">
  <property name="name" value="${myname}" />
  <property name="printer" ref="${myprinter}" />
  <property name="names">
    <list>
      <value>${value1}</value>
      <value>${value2}</value>
      <value>${value3}</value>
    </list>
  </property>
</bean>

<bean id="printer" class="com.example.StringPrinter" />
<bean id="consolePrinter" class="com.example.ConsolePrinter" />
```

15. Test

1)com.example.MainClass.java

2)right-click > Run As > Java Application

```
Hello Spring
Hello Spring
HTML5
CSS3
JavaScript
```

3)/src/test/java/HelloJUnitTest.java

-right-click > Run As > JUnit Test

-Green Bar

16. resources/value.properties 수정

```
myname=Spring
myprinter=printer
value1=HTML5
value2=CSS3
value3=JavaScript
printer1=stringPrinter
printer2=consolePrinter
```

17. pom.xml에 javax.annotation API dependency 추가하기
  1)https://mvnrepository.com에서 'javax annotation'로 검색
  2)Javax Annotation API click
  3)1.3.2. click
  4)dependency 복사해서 pom.xml로 붙여넣기

```
    <!-- https://mvnrepository.com/artifact/javax.annotation/javax.annotation-api -->
    <dependency>
        <groupId>javax.annotation</groupId>
        <artifactId>javax.annotation-api</artifactId>
        <version>1.3.2</version>
    </dependency>
```

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

18. Hello.java code 수정
  1)com.example/Hello.java

```
package com.example;

import java.util.List;

import javax.annotation.Resource;

import org.springframework.beans.factory.annotation.Value;
import org.springframework.stereotype.Component;

import lombok.Getter;
import lombok.NoArgsConstructor;
import lombok.Setter;

@NoArgsConstructor
@Setter
@Component
public class Hello {
  @Value("${myname}")
  private String name;

  @Resource(name="${printer1}")
  private Printer printer;

  @Value("${value1},${value2},${value3}")
  @Getter private List<String> names;

  public String sayHello(){
    return "Hello " + name;
  }

  public void print(){
    this.printer.print(sayHello());
  }
}
```

19. StringPrinter.java 수정

```
package com.example;
```

```
import org.springframework.stereotype.Component;

@Component("stringPrinter")
public class StringPrinter implements Printer {
  private StringBuffer buffer =  new StringBuffer();

  @Override
  public void print(String message) {
    this.buffer.append(message);
  }

  @Override
  public String toString(){
    return this.buffer.toString();
  }
}
```

20. beans.xml 수정하기

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:context="http://www.springframework.org/schema/context"
  xsi:schemaLocation="http://www.springframework.org/schema/beans
  http://www.springframework.org/schema/beans/spring-beans.xsd
    http://www.springframework.org/schema/context
    http://www.springframework.org/schema/context/spring-context-3.2.xsd">

  <context:property-placeholder location="classpath:value.properties "/>
  <context:component-scan base-package="com.example" />  <---추가하기
  <!-- 나머지 코드 삭제 -->
</beans>
```

21. MainClass.java 수정하기

```
package com.example;

import org.springframework.context.ApplicationContext;
import org.springframework.context.support.GenericXmlApplicationContext;

public class MainClass {
  public static void main(String[] args) {
    ApplicationContext ctx = new GenericXmlApplicationContext("classpath:beans.xml");

    Hello hello = (Hello) ctx.getBean("hello");
    System.out.println(hello.sayHello());
    hello.print();

    Printer printer = ctx.getBean("stringPrinter", StringPrinter.class);
    System.out.println(printer.toString());

    hello.getNames().forEach(value -> System.out.println(value));
  }
}
```

22. 실행
  1)MainClass > right-click > Run As > Java Application

  2)결과
    Hello Spring
    Hello Spring
    HTML5,CSS3,JavaScript

Task 3. Lab
1. In Package Explorer > right-click > New > Java Project
  1)Project Name : PropertyDemo1
  2)JRE : Use default JRE 'jdk-13.0.2' and workspace compiler preferences
  3)Next
  4)Uncheck [Create module-info.java file]
  5)Finish

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

3. /src/com.example.AdminConnection.java 생성
  1)com.example > right-click > New > Class
  2)Name : AdminConnection
  3)Finish

```
package com.example;

public class AdminConnection {
  private String adminId;
  private String adminPwd;
  private String subAdminId;
  private String subAdminPwd;
}
```

4. Java Project를 Spring Project로 변환
  1)PropertyDemo1 Project > right-click > Configure > Convert to Maven Project
    -Project : /PropertyDemo1
    -Group Id : PropertyDemo1
    -Artifact Id : PropertyDemo1
    -version : 0.0.1-SNAPSHOT
    -Packaging : jar
    -Finish

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

  3)pom.xml file에 Spring Context Dependency 추가하기

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

```
        </dependency>
      </dependencies>

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


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

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

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


6. AdminConnection.java 수정

      package com.example;

      import lombok.Getter;
      import lombok.Setter;

      @Setter
      @Getter
      public class AdminConnection {
        private String adminId;
        private String adminPwd;
        private String subAdminId;
        private String subAdminPwd;
      }


7. PropertyDemo/resources folder 생성
   1)PropertyDemo1 project > right-click > New > Source Folder
   2)Folder name : resources
   3)Finish


8./resources 두 개의 properties file 생성

   <admin.properties>
```

```
805     admin.id=javaexpert
806     admin.pwd=12345678
807
808   <sub.admin.properties>
809     sub.admin.id=javasoft
810     sub.admin.pwd=987654321
811
812
813 9. Bean Configuration XML 작성
814   1)PropertyDemo1/resources > right-click > New > Spring Bean Configuration File
815   2)File name : beans.xml
816   3)Finish
817   4)Namespace tab에서 context - http://www.springframework.org/schema/context check
818
819     <?xml version="1.0" encoding="UTF-8"?>
820     <beans xmlns="http://www.springframework.org/schema/beans"
821       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
822       xmlns:context="http://www.springframework.org/schema/context"
823       xsi:schemaLocation="http://www.springframework.org/schema/beans
          http://www.springframework.org/schema/beans/spring-beans.xsd
824         http://www.springframework.org/schema/context
            http://www.springframework.org/schema/context/spring-context-3.2.xsd">
825
826       <context:property-placeholder location="classpath:admin.properties,
          classpath:sub.admin.properties" />
827
828       <bean id="adminConnection" class="com.example.AdminConnection">
829         <property name="adminId">
830           <value>${admin.id}</value>
831         </property>
832         <property name="adminPwd">
833           <value>${admin.pwd}</value>
834         </property>
835         <property name="subAdminId">
836           <value>${sub.admin.id}</value>
837         </property>
838         <property name="subAdminPwd">
839           <value>${sub.admin.pwd}</value>
840         </property>
841       </bean>
842     </beans>
843
844
845 10. /src/com.example.MainClass.java 생성
846   1)com.example > right-click > New > Class
847   2)Name : MainClass
848   3)Finish
849
850   package com.example;
851
852   import org.springframework.context.support.AbstractApplicationContext;
853   import org.springframework.context.support.GenericXmlApplicationContext;
854
855   public class MainClass {
856     public static void main(String[] args) {
857       AbstractApplicationContext ctx = new
          GenericXmlApplicationContext("classpath:beans.xml");
858       AdminConnection connection = ctx.getBean("adminConnection",
```

```
        AdminConnection.class);
859       System.out.println("admin ID : " + connection.getAdminId());
860       System.out.println("admin PWD : " + connection.getAdminPwd());
861       System.out.println("sub admin ID : " + connection.getSubAdminId());
862       System.out.println("sub admin PWD : " + connection.getSubAdminPwd());
863
864       ctx.close();
865     }
866   }
867
868
869 11. 결과
870   1)MainClass.java > right-click > Run As > Java Application
871
872     admin ID : javaexpert
873     admin PWD : 12345678
874     sub admin ID : javasoft
875     sub admin PWD : 987654321
876
877
878
879 ----------------------------------------
880 Task 4. Lab
881 1. In Package Explorer > right-click > New > Java Project
882   1)Project Name : PropertyDemo2
883   2)JRE : Use default JRE 'jdk-13.0.2' and workspace compiler preferences
884   3)Next
885   4)Uncheck [Create module-info.java file]
886   5)Finish
887
888
889 2. /src/ right-click > New > Package
890   1)Package name : com.example
891   2)Finish
892
893
894 3. /src/com.example.AdminConnection.java 생성
895   1)com.example > right-click > New > Class
896   2)Name : AdminConnection
897   3)Finish
898
899     package com.example;
900
901     public class AdminConnection {
902       private String adminId;
903       private String adminPwd;
904       private String subAdminId;
905       private String subAdminPwd;
906     }
907
908
909 4. Java Project를 Spring Project로 변환
910   1)PropertyDemo1 Project > right-click > Configure > Convert to Maven Project
911     -Project : /PropertyDemo2
912     -Group Id : PropertyDemo2
913     -Artifact Id : PropertyDemo2
914     -version : 0.0.1-SNAPSHOT
915     -Packaging : jar
```

-Finish

2)PropertyDemo2 Project > right-click > Spring > Add Spring Project Nature

3)pom.xml 파일에 Spring Context Dependency 추가하기

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

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

[INFO] BUILD SUCCESS 확인

5. Lombok library 추가

1)https://mvnrepository.com/에서 'lombok'으로 검색

2)'Project Lombok' click

3)1.18.12 click

4)depency copy해서 pom.xml에 붙여넣기

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

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

[INFO] BUILD SUCCESS 확인

6. AdminConnection.java 수정

```
package com.example;

import lombok.Getter;
import lombok.Setter;

@Setter
@Getter
public class AdminConnection {
  private String adminId;
  private String adminPwd;
  private String subAdminId;
  private String subAdminPwd;
}
```

7. PropertyDemo/resources folder 생성
  1)PropertyDemo1 project > right-click > New > Source Folder
  2)Folder name : resources
  3)Finish

8. /resources 두 개의 properties file 생성

  <admin.properties>
    admin.id=javaexpert
    admin.pwd=12345678

  <sub.admin.properties>
    sub.admin.id=javasoft
    sub.admin.pwd=987654321

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

```
package com.example;

import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.ComponentScan;
import org.springframework.context.annotation.Configuration;
import org.springframework.context.support.PropertySourcesPlaceholderConfigurer;
import org.springframework.core.io.ClassPathResource;

@Configuration
@ComponentScan(basePackages = "com.example")
public class ApplicationConfig {
  @Bean
  public static PropertySourcesPlaceholderConfigurer properties() {
    PropertySourcesPlaceholderConfigurer configurer = new
    PropertySourcesPlaceholderConfigurer();
    configurer.setLocations(new ClassPathResource("admin.properties"), new
    ClassPathResource("sub.admin.properties"));
    return configurer;
  }
}
```

10. com.example.AdminConnection.java 수정

```
package com.example;

import org.springframework.beans.factory.annotation.Value;
import org.springframework.stereotype.Component;

import lombok.Getter;

@Getter
@Component
public class AdminConnection {
```

```
    @Value("${admin.id}")
    private String adminId;
    @Value("${admin.pwd}")
    private String adminPwd;
    @Value("${sub.admin.id}")
    private String subAdminId;
    @Value("${sub.admin.pwd}")
    private String subAdminPwd;
  }


11. /src/com.example.MainClass.java
  1)com.example > right-click > New > Class
  2)Name : MainClass1
  3)Finish

    package com.example;

    import org.springframework.context.ApplicationContext;
    import org.springframework.context.annotation.AnnotationConfigApplicationContext;

    public class MainClass1 {
      public static void main(String[] args) {
        ApplicationContext context = new
        AnnotationConfigApplicationContext(ApplicationConfig.class);
        AdminConnection connection = (AdminConnection)
        context.getBean("adminConnection");
        System.out.println("admin ID : " + connection.getAdminId());
        System.out.println("admin PWD : " + connection.getAdminPwd());
        System.out.println("sub admin ID : " + connection.getSubAdminId());
        System.out.println("sub admin PWD : " + connection.getSubAdminPwd());
      }
    }


12. 결과
  1)MainClass.java > right-click > Run As > Java Application
    admin ID : javaexpert
    admin PWD : 12345678
    sub admin ID : javasoft
    sub admin PWD : 987654321



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


2. Package 생성
  1)/src/ > right-click > New > Package
  2)Package name : com.example
```

```
1086    3)Finish
1087
1088
1089  3. Java Project를 Spring Project로 변환
1090    1)ProfileDemo Project > right-click > Configure > Convert to Maven Project
1091      -Project : /ProfileDemo
1092      -Group Id : ProfileDemo
1093      -Artifact Id : ProfileDemo
1094      -version : 0.0.1-SNAPSHOT
1095      -Packaging : jar
1096      -Finish
1097
1098    2)ProfileDemo Project > right-click > Spring > Add Spring Project Nature
1099
1100    3)pom.xml file에 Spring Context Dependency 추가하기
1101      <version>0.0.1-SNAPSHOT</version>
1102      <dependencies>
1103        <dependency>
1104          <groupId>org.springframework</groupId>
1105          <artifactId>spring-context</artifactId>
1106          <version>5.2.5.RELEASE</version>
1107        </dependency>
1108      </dependencies>
1109
1110    4)pom.xml > right-click > Run As > Maven install
1111      [INFO] BUILD SUCCESS 확인
1112
1113
1114  4. Lombok library 추가
1115    1)https://mvnrepository.com/에서 'lombok'으로 검색
1116    2)'Project Lombok' click
1117    3)1.18.12 click
1118    4)depency copy해서 pom.xml에 붙여넣기
1119
1120      <dependencies>
1121        <dependency>
1122          <groupId>org.springframework</groupId>
1123          <artifactId>spring-context</artifactId>
1124          <version>5.2.5.RELEASE</version>
1125        </dependency>
1126        <!-- https://mvnrepository.com/artifact/org.projectlombok/lombok -->
1127        <dependency>
1128          <groupId>org.projectlombok</groupId>
1129          <artifactId>lombok</artifactId>
1130          <version>1.18.12</version>
1131          <scope>provided</scope>
1132        </dependency>
1133      </dependencies>
1134
1135    5)pom.xml > right-click > Run As > Maven install
1136      [INFO] BUILD SUCCESS 확인
1137
1138
1139  5. ProfileDemo/resources folder 생성
1140    1)ProfileDemo project > New > Source Folder
1141    2)Folder name : resources
1142    3)Finish
1143
```

6. ServerInfo.java 생성
  1)com.example > right-click > New >Class
  2)Name : ServerInfo
  3)Finish

```
package com.example;

import lombok.Getter;
import lombok.Setter;

@Getter
@Setter
public class ServerInfo {
  private String ipNum;
  private String portNum;
}
```

7. XML 설정 file 2개 생성
  1)resource > right-click > New > Spring Bean Configuration File
  2)File name : run.xml
  3)Finish

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

  <bean id="serverInfo" class="com.example.ServerInfo">
    <property name="ipNum" value="192.168.56.5" />
    <property name="portNum" value="80" />
  </bean>
</beans>
```

  3)/resource > right-click > New > Spring Bean Configuration File
  4)File name : dev.xml
  5)Finish

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

  <bean id="serverInfo" class="com.example.ServerInfo">
    <property name="ipNum" value="localhost" />
    <property name="portNum" value="8080" />
  </bean>
</beans>
```

8. MainClass 생성
  1)com.example > right-click > New > Class
  2)Name : MainClass.java

3)Finish

```
package com.example;

import java.util.Scanner;

import org.springframework.context.support.GenericXmlApplicationContext;

public class MainClass {
  public static void main(String[] args) {
    Scanner scan = new Scanner(System.in);
    System.out.print("Select dev or run : ");
    String config = scan.next();  //"dev" or "run"

    GenericXmlApplicationContext ctx = new GenericXmlApplicationContext();
    ctx.getEnvironment().setActiveProfiles(config);
    ctx.load("dev.xml", "run.xml");
    ctx.refresh();

    ServerInfo info = ctx.getBean("serverInfo", ServerInfo.class);
    System.out.println("IP : " + info.getIpNum());
    System.out.println("Port : " + info.getPortNum());
    ctx.close();
  }
}
```

9. 결과
1)MainClass.java > right-click > Run As > Java Application
Select dev or run :
2)입력시 dev를 넣으면 dev환경인 localhost/8080이 나오고, 만일 run이라고 넣으면 192.168.56.5/80이 나온다.

Task 6. Lab

1. In Package Explorer > right-click > New > Java Project
1)Project Name : ProfileDemo1
2)JRE
-Select [Use default JRE 'jdk-13.0.2' and workspace compiler preferences]
3)Next
4)Uncheck [Create module-info.java file]
5)Finish

2. Package 생성
1)/src/ > right-click > New > Package
2)Package name : com.example
3)Finish

3. Java Project를 Spring Project로 변환
1)ProfileDemo1 Project > right-click > Configure > Convert to Maven Project
-Project : /ProfileDemo1
-Group Id : ProfileDemo1
-Artifact Id : ProfileDemo1
-version : 0.0.1-SNAPSHOT

```
1257      -Packaging : jar
1258      -Finish
1259
1260    2)ProfileDemo1 Project > right-click > Spring > Add Spring Project Nature
1261
1262    3)pom.xml file에 Spring Context Dependency 추가하기
1263      <version>0.0.1-SNAPSHOT</version>
1264      <dependencies>
1265        <dependency>
1266          <groupId>org.springframework</groupId>
1267          <artifactId>spring-context</artifactId>
1268          <version>5.2.5.RELEASE</version>
1269        </dependency>
1270      </dependencies>
1271
1272    4)pom.xml > right-click > Run As > Maven install
1273      [INFO] BUILD SUCCESS 확인
1274
1275
1276  4. Lombok library 추가
1277    1)https://mvnrepository.com/에서 'lombok'으로 검색
1278    2)'Project Lombok' click
1279    3)1.18.12 click
1280    4)depency copy해서 pom.xml에 붙여넣기
1281
1282      <dependencies>
1283        <dependency>
1284          <groupId>org.springframework</groupId>
1285          <artifactId>spring-context</artifactId>
1286          <version>5.2.5.RELEASE</version>
1287        </dependency>
1288        <!-- https://mvnrepository.com/artifact/org.projectlombok/lombok -->
1289        <dependency>
1290          <groupId>org.projectlombok</groupId>
1291          <artifactId>lombok</artifactId>
1292          <version>1.18.12</version>
1293          <scope>provided</scope>
1294        </dependency>
1295      </dependencies>
1296
1297    5)pom.xml > right-click > Run As > Maven install
1298      [INFO] BUILD SUCCESS 확인
1299
1300
1301  5. ServerInfo.java 생성
1302    1)com.example > right-click > New  >Class
1303    2)Name : ServerInfo
1304    3)Finish
1305
1306      package com.example;
1307
1308      import lombok.Getter;
1309      import lombok.Setter;
1310
1311      @Getter
1312      @Setter
1313      public class ServerInfo {
1314        private String ipNum;
```

```
1315        private String portNum;
1316      }
1317
1318
1319  6. Java 설정 file 2개 생성
1320    1)com.example > right-click > New > Class
1321    2)Name : ApplicationConfigDev
1322    3)Finish
1323
1324      package com.example;
1325
1326      import org.springframework.context.annotation.Bean;
1327      import org.springframework.context.annotation.Configuration;
1328      import org.springframework.context.annotation.Profile;
1329
1330      @Configuration
1331      @Profile("dev")
1332      public class ApplicationConfigDev {
1333
1334        @Bean
1335        public ServerInfo serverInfo(){
1336          ServerInfo info = new ServerInfo();
1337          info.setIpNum("localhost");
1338          info.setPortNum("8080");
1339          return info;
1340        }
1341      }
1342
1343    4)com.example > right-click > New > Class
1344    5)Name : ApplicationConfigRun.java
1345    6)Finish
1346
1347      package com.example;
1348
1349      import org.springframework.context.annotation.Bean;
1350      import org.springframework.context.annotation.Configuration;
1351      import org.springframework.context.annotation.Profile;
1352
1353      @Configuration
1354      @Profile("run")
1355      public class ApplicationConfigRun {
1356
1357        @Bean
1358        public ServerInfo serverInfo(){
1359          ServerInfo info = new ServerInfo();
1360          info.setIpNum("192.168.56.5");
1361          info.setPortNum("80");
1362          return info;
1363        }
1364      }
1365
1366
1367  7. MainClass 생성
1368    1)com.example > right-click > New > Class
1369    2)Name : MainClass.java
1370
1371      package com.example;
1372      import java.util.Scanner;
```

```
1373
1374      import org.springframework.context.annotation.AnnotationConfigApplicationContext;
1375
1376      public class MainClass {
1377        public static void main(String[] args) {
1378          Scanner scan = new Scanner(System.in);
1379          System.out.print("Select dev or run : ");
1380          String config = scan.next();  //"dev" or "run"
1381
1382          AnnotationConfigApplicationContext ctx = new AnnotationConfigApplicationContext();
1383          ctx.getEnvironment().setActiveProfiles(config);
1384          ctx.register(ApplicationConfigDev.class, ApplicationConfigRun.class);
1385          ctx.refresh();
1386
1387          ServerInfo info = ctx.getBean("serverInfo", ServerInfo.class);
1388          System.out.println("IP : " + info.getIpNum());
1389          System.out.println("Port : " + info.getPortNum());
1390          ctx.close();
1391        }
1392      }
1393
1394
1395  8. 결과
1396    1)MainClass.java > right-click > Run As > Java Application
1397          Select dev or run :
1398    2)입력시 dev를 넣으면 dev환경인 localhost/8080이 나오고, 만일 run이라고 넣으면 192.168.56.5/80이 나
      온다.
```