HOL : Spring DI

Task 1. Non-DI Java Project
1. Project 유형 : Java Project
2. Project Name : BeforeSpring
3. Package Name : com.example
4. Calculator Class
com.example.Calculator.java

```
package com.example;

public class Calculator {
    public void addAction(int a, int b){
        System.out.println("Called addAction()");
        System.out.printf("%d + %d = %d₩n", a, b, (a + b));
    }
    public void subAction(int a, int b){
        System.out.println("Called subAction()");
        System.out.printf("%d - %d = %d₩n", a, b, (a - b));
    }
    public void multiAction(int a, int b){
        System.out.println("Called multiAction()");
        System.out.printf("%d x %d = %d₩n", a, b, (a * b));
    }
    public void divAction(int a, int b){
        System.out.println("Called divAction()");
        System.out.printf("%d / %d = %d₩n", a, b, (a / b));
    }
}
```

5. MyCalculator Class
com.example.MyCalculator.java

```
package com.example;

public class MyCalculator {
    private Calculator calculator;
    private int firstNum;
    private int secondNum;

    public void setFirstNum(int firstNum) {
        this.firstNum = firstNum;
    }
    public void setSecondNum(int secondNum) {
        this.secondNum = secondNum;
    }
    public void setCalculator(Calculator calculator){
        this.calculator = calculator;
    }

    public void add(){
        this.calculator.addAction(firstNum, secondNum);
    }
    public void sub(){
```

```
        this.calculator.subAction(firstNum, secondNum);
    }
    public void multi(){
        this.calculator.multiAction(firstNum, secondNum);
    }
    public void div(){
        this.calculator.divAction(firstNum, secondNum);
    }
}

6. MainClass Class
  com.example.MainClass
  package com.example;

  public class MainClass {
    public static void main(String[] args) {
        MyCalculator myCalculator = new MyCalculator();
        myCalculator.setCalculator(new Calculator());

        myCalculator.setFirstNum(10);
        myCalculator.setSecondNum(2);

        myCalculator.add();
        myCalculator.sub();
        myCalculator.multi();
        myCalculator.div();
    }
  }

7. Result
  Called addAction()
  10 + 2 = 12
  Called subAction()
  10 - 2 = 8
  Called multiAction()
  10 x 2 = 20
  Called divAction()
  10 / 2 = 5


------------------------------
Task 2. DI Demo in Spring
1. New > Java Project
  1)Project Name : StartSpring
  2)JRE : Use default JRE (currently 'jdk1.8.0_221')
  3)Finish
2. Create package to src : com.example
3. Copy MyCalculator.java, Calculator.java from BeforeSpring project to StartSpring's package
4. Create class : com.example.MainClass.java
  package com.example;

  public class MainClass {
```

```
105         public static void main(String[] args) {
106
107         }
108     }
109
110 5. Java Project를 Spring Project로 변환
111     1)StartSpring Project > right-click > Configure > Convert to Maven Project
112         -Project : /StartSpring
113         -Group Id : StartSpring
114         -Artifact Id : StartSpring
115         -version : 0.0.1-SNAPSHOT
116         -Packaging : jar
117         -Finish
118         -Package Explorer에서 보이는 Project icon에 Maven의 'M'자가 보임.
119
120     2)StartSpring Project > right-click > Spring > Add Spring Project Nature
121         -Package Explorer에서 보이는 Project icon에 'M'자와 Spring의 'S'가 보임.
122
123     3)pom.xml file에 Spring Context Dependency 추가하기
124         --https://mvnrepository.com에서 spring context로 검색
125         --현재 Spring 5.x의 현재 version인 5.2.0.RELEASE click
126         --Copy하여 pom.xml에 paste
127
128       <version>0.0.1-SNAPSHOT</version>
129         <dependencies>    <--- dependencies element 추가
130             <dependency>    <---여기에 paste
131                 <groupId>org.springframework</groupId>
132                 <artifactId>spring-context</artifactId>
133                 <version>5.2.0.RELEASE</version>
134             </dependency>
135         </dependencies>
136
137     4)pom.xml > right-click > Run As > Maven install
138         [INFO] BUILD SUCCESS 확인
139
140 6. src/config folder 생성
141     1)/src > right-click > New > Folder
142         Folder name : config
143
144 7. Bean Configuration XML 작성
145     -src/config > right-click >New > Other > Spring > Spring Bean Configuration File > Next
146     -Name : applicationContext.xml > Finish
147     <?xml version="1.0" encoding="UTF-8"?>
148     <beans xmlns="http://www.springframework.org/schema/beans"
149         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
150         xsi:schemaLocation="http://www.springframework.org/schema/beans
        http://www.springframework.org/schema/beans/spring-beans.xsd">
151
152         <bean id="calculator" class="com.example.Calculator" />
153
154         <bean id="myCalculator" class="com.example.MyCalculator">
155             <property name="calculator">
```

```
            <ref bean="calculator" />
        </property>
        <property name="firstNum" value="10" />
        <property name="secondNum" value="2" />
    </bean>
</beans>

8. MainClass.java
package com.javasoft;

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

public class MainClass {
    public static void main(String[] args) {
        String configFile = "config/applicationContext.xml";
        AbstractApplicationContext ctx = new GenericXmlApplicationContext(configFile);
        MyCalculator myCalculator = ctx.getBean("myCalculator", MyCalculator.class);

        myCalculator.add();
        myCalculator.sub();
        myCalculator.multi();
        myCalculator.div();

        ctx.close();
    }
}

9. Result
BeforeSpring과 같음.


------------------------------
Task 3. 간단한 DI Project
1. In Package Explorer > right-click > New > Java Project
   Project name : DIDemo

2. src > right-click > New > Package
   Package name : com.example

3. Interface 작성
   1)com.example > right-click > New > Interface
   2)Interface name : Printer

   3)Printer.java
      package com.example;

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

4. POJO class 작성
```

1)com.example > right-click > New > Class
2)Class name : Hello
3)Hello.java

```
package com.example;

public class Hello{
   private String name;
   private Printer printer;

   public Hello(){}

   public void setName(String name){
      this.name = name;
   }

   public void setPrinter(Printer printer){
      this.printer = printer;
   }

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

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

5. Printer interface의 child class 작성하기
1)com.example > right-click > New > Class
-Class Name : StringPrinter
-Interfaces : com.example.Printer

2)StringPrinter.java

```
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();
   }
}
```

3)om.example > right-click > New > Class
-Class Name : ConsolePrinter
-Interface : com.example.Printer

```
260
261    4)ConsolePrinter.java
262       package com.example;
263
264       public class ConsolePrinter implements Printer{
265
266          @Override
267          public void print(String message){
268             System.out.println(message);
269          }
270       }
271
272 6. Java Project를 Spring Project로 변환
273    1)DIDemo Project > right-click > Configure > Convert to Maven Project
274       -Project : /DIDemo
275       -Group Id : DIDemo
276       -Artifact Id : DIDemo
277       -version : 0.0.1-SNAPSHOT
278       -Packaging : jar
279       -Finish
280
281    2)DIDemo Project > right-click > Spring > Add Spring Project Nature
282
283    3)pom.xml file에 Spring Context Dependency 추가하기
284        <version>0.0.1-SNAPSHOT</version>
285       <dependencies>
286          <dependency>
287             <groupId>org.springframework</groupId>
288             <artifactId>spring-context</artifactId>
289             <version>5.2.0.RELEASE</version>
290          </dependency>
291       </dependencies>
292
293    4)pom.xml > right-click > Run As > Maven install
294    [INFO] BUILD SUCCESS 확인
295
296 7. src/config folder 생성
297       -/src > right-click > New > Folder
298          Folder name : config
299
300 8. Bean Configuration XML 작성
301    1)/src/config > right-click > New > Other > Spring > Spring Bean Configuration File > Next
302    2)File name : beans.xml > Next
303    3)Finish
304
305       <?xml version="1.0" encoding="UTF-8"?>
306       <beans xmlns="http://www.springframework.org/schema/beans"
307          xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
308          xsi:schemaLocation="http://www.springframework.org/schema/beans
          http://www.springframework.org/schema/beans/spring-beans.xsd">
309
310          <bean id="hello" class="com.example.Hello">
```

```
            <property name="name" value="Spring" />
            <property name="printer" ref="printer" />
        </bean>
        <bean id="printer" class="com.example.StringPrinter" />
        <bean id="consolePrinter" class="com.example.ConsolePrinter" />

    </beans>
```

9. Beans Graph 사용하기
  1)Window menu > Show View > Other > Spring > Spring Explorer > Open
  2)n Spring Explorer
    -DIDemo > Beans > beans.xml > right-click > Open Beans Graphs

10. DI Test class 작성
  1)/src/com.example > right-click > New > Package
    Package Name : test
  2)/src/com.example/test/HelloBeanTest.java

```
    package com.example.test;

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

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

    public class HelloBeanTest {
        public static void main(String [] args){
            //1. IoC Container 생성
            ApplicationContext context =
                    new GenericXmlApplicationContext("config/beans.xml");

            //2. Hello Beans 가져오기
            Hello hello = (Hello)context.getBean("hello");
            System.out.println(hello.sayHello());
            hello.print();

            //3. SpringPrinter 가져오기
            Printer printer = (Printer)context.getBean("printer");
            System.out.println(printer.toString());

            Hello hello2 = context.getBean("hello", Hello.class);
            hello2.print();

            System.out.println(hello == hello2);  //Singleton Pattern
        }
    }
```

11. Result
  Hello Spring
  Hello Spring
  true

```
363
364
365 --------------------------------
366 Task 4. jUnit을 사용한 DI test class 작성하기
367 1. jUnit을 사용한 DI test class(HelloBeanJunitTest.java) 작성
368   1)pom.xml에 아래 코드 붙여넣기
369     <dependency>
370       <groupId>junit</groupId>
371       <artifactId>junit</artifactId>
372       <version>4.12</version>
373       <scope>test</scope>
374     </dependency>
375
376   2)pom.xml > right-click > Run As > Maven install
377     [INFO] BUILD SUCCESS 확인
378
379   3)/src/com.example.test/HelloBeanTest.java 복사
380   4)/src/com.example.test/ 붙여넣고 이름변경 -> HelloBeanJunitTest.java
381
382     package com.example.test;
383
384     import org.junit.Before;
385     import org.junit.Test;
386     import org.springframework.context.ApplicationContext;
387     import org.springframework.context.support.GenericXmlApplicationContext;
388
389     import com.example.Hello;
390     import com.example.Printer;
391
392     import static org.junit.Assert.assertEquals;
393     import static org.junit.Assert.assertSame;
394
395     public class HelloBeanJunitTest {
396       ApplicationContext context;
397
398       @Before
399       public void init(){
400         //항상 먼저 ApplicationContext를 생성해야 하기 때문에
401         //1. IoC Container 생성
402         context = new GenericXmlApplicationContext("config/beans.xml");
403       }
404
405       @Test
406       public void test1(){
407         //2. Hello Beans 가져오기
408         Hello hello = (Hello)context.getBean("hello");
409         assertEquals("Hello Spring", hello.sayHello());
410         hello.print();
411
412         //3. SpringPrinter 가져오기
413         Printer printer = (Printer)context.getBean("printer");
414         assertEquals("Hello Spring", printer.toString());
```

```
        }

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

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

2. @Before에 mouse를 올려놓으면 Fix project setup... click
   1)Add archive 'junit-4.12.jar ... > OK
      -import org.junit...에 mouse를 올려놓으면 Fix project setup... click
      -Add JUnit 4 library to the build path > OK

3. right-click > Run As > Junit Test
   1)결과 -> Junit View에 초록색 bar
   2)만일, test1() method를 jUnit에서 제외하고 싶을 때에는 @Test 옆에 @Ignore를 선언한다.

      import import org.junit.Ignore;
      ...
      @Test @Ignore
      public void test1(){
      ...

   3)right-click > Run As > Junit Test
      -jUnit Test 목록에서 test1()는 실행되지 않는다.


----------------------------------------
Task 5. Spring TestContext Framework
1. Spring-Test library 설치
   1)http://mvnrepository.com에서 'spring-test'로 검색
   2)검색 결과 목록에서 'Spring TestContext Framework' 클릭
   3)version 목록에서 5.2.0.RELEASE 클릭
2. dependency 복사해서 pom.xml에 붙여넣기
   <!-- https://mvnrepository.com/artifact/org.springframework/spring-test -->
   <dependency>
      <groupId>org.springframework</groupId>
      <artifactId>spring-test</artifactId>
      <version>5.2.0.RELEASE</version>
      <scope>test</scope>
   </dependency>

3. pom.xml > right-click > Maven Install

4. Spring-Test를 사용할 DI test class-HelloBeanJunitSpringTest.java 작성하기
   1)/src/com.example.test/HelloBeanJunitTest.java 복사해서
   2)/src/com.example.test/HelloBeanJunitSpringTest.java 로 붙여넣기
      -ApplicationContext 생성하는 부분을 매번 수행하는 것이 아니라 이 부분을 자동으로 해주는 것은
      SpringTest Framework가 하게 한다.
```

-따라서 init()이 필요하지 않도록 설정한다.

```
import org.junit.runner.RunWith;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.context.ApplicationContext;
import org.springframework.context.annotation.Configuration;
import org.springframework.test.context.ContextConfiguration;
import org.springframework.test.context.junit4.SpringJUnit4ClassRunner;

...
@RunWith(SpringJUnit4ClassRunner.class)
@ContextConfiguration(locations="classpath:config/beans.xml")
//beans.xml경로를 수정한다. 경로 앞에 classpath:를 넣는다.
public class HelloBeanJunitSpringTest {

   @Autowired
   ApplicationContext context;
```

3)아래의 init()가 필요 없어짐으로 삭제한다.

```
/*
@Before
public void init(){
   //항상 먼저 ApplicationContext를 생성해야 하기 때문에
   //1. IoC Container 생성
   context = new GenericXmlApplicationContext("config/beans.xml");
}
*/
```

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

4)만일 해당 객체를 찾을 수 없다는 오류가 계속 발생하면
-해당 Project > right-click > Build Path > Libraries tab
-spring-test-5.2.0.RELEASE.jar 선택 후 [Remove] 로 삭제
-[Add External JARs...] Click
-Local M2 Repository(e.g C:₩Users₩bluee₩.m2₩repository₩org₩springframework₩spring-test₩5.2.0.RELEASE)에서 직접 jar를 선택할 것
-[Order and Export] tab에서 spring-test-5.2.0.RELEASE.jar 선택 후 [Up] button을 클릭
-해당 Project/src 바로 아래까지 올리고 [Apply and Close] Click

Task 6. setter를 이용한 의존주입하기 실습

1. In Package Explorer > right-click > New > Java Project
   Project name : DIDemo1

2. src > right-click > New > Package
   Package name : com.example

3. POJO class 작성
   1)com.example > right-click > New > Class

2)Class Name : Hello

```
package com.example;

public class Hello{
   private String name;
   private Printer printer;

   public Hello(){}

   public void setName(String name){
      this.name = name;
   }

   public void setPrinter(Printer printer){
      this.printer = printer;
   }

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

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

3)com.example > right-click > New > Interface

4)interface name : Printer

```
package com.example;

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

5)com.example > right-click > New > Class

6)Class Name : StringPrinter

7)Interfaces : com.example.Printer

```
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();
   }
}
```

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

```
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)DIDemo1 Project > right-click > Configure > Convert to Maven Project
-Project : /DIDemo1
-Group Id : DIDemo1
-Artifact Id : DIDemo1
-version : 0.0.1-SNAPSHOT
-Packaging : jar
-Finish

2)DIDemo1 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>5.2.0.RELEASE</version>
    </dependency>
  </dependencies>
```

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

5. src/config folder 생성
1)/src > right-click > New > Folder
2)Folder name : config

6. Bean Configuration XML 작성
1)/src/config > right-click > New > Other > Spring > Spring Bean Configuration File
2)File name : beans.xml > 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="hello" class="com.example.Hello">
```

```
            <property name="name" value="Spring" />
            <property name="printer" ref="printer" />
        </bean>
        <bean id="printer" class="com.example.StringPrinter" />
        <bean id="consolePrinter" class="com.example.ConsolePrinter" />

    </beans>
```

7. DI Test class 작성
   1)/src > right-click > New > Package
   2)Package Name : com.example.test
   3)com.example.test > right-click > New > Class
   4)Class Name : HelloBeanTest

```
    package com.example.test;

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

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

    public class HelloBeanTest {
        public static void main(String [] args){
            //1. IoC Container 생성
            ApplicationContext context =
                    new GenericXmlApplicationContext("config/beans.xml");

            //2. Hello Beans 가져오기
            Hello hello = (Hello)context.getBean("hello");
            System.out.println(hello.sayHello());
            hello.print();

            //3. SpringPrinter 가져오기
            Printer printer = (Printer)context.getBean("printer");
            System.out.println(printer.toString());

            Hello hello2 = context.getBean("hello", Hello.class);
            hello2.print();

            System.out.println(hello == hello2);
        }
    }
```

8. Test
   1)/src/com.example.test/HelloBeanTest.java > right-click > Run As > Java Application

```
    Hello Spring
    Hello Spring
    true
```

9. jUnit으로 test
   1)jUnit Library 설치

```
671     -jUnit 4.12 version을 pom.xml에 추가
672
673     <dependency>
674         <groupId>junit</groupId>
675         <artifactId>junit</artifactId>
676         <version>4.12</version>
677         <scope>test</scope>
678     </dependency>
679
680   2)pom.xml > right-click > Run As > Maven Install
681
682   3)jUnit을 사용한 DI test class(HelloBeanJunitTest.java) 작성
683     -/src/com.example.test/HelloBeanTest.java 복사
684     -/src/com.example.test/ 붙여넣고 이름변경 -> HelloBeanJunitTest.java
685
686     package com.example.test;
687
688     import org.junit.Before;
689     import org.junit.Test;
690     import org.springframework.context.ApplicationContext;
691     import org.springframework.context.support.GenericXmlApplicationContext;
692
693     import com.example.Hello;
694     import com.example.Printer;
695
696     import static org.junit.Assert.assertEquals;
697     import static org.junit.Assert.assertSame;
698
699     public class HelloBeanJunitTest {
700       ApplicationContext context;
701
702       @Before
703       public void init(){
704         context = new GenericXmlApplicationContext("config/beans.xml");
705       }
706
707       @Test
708       public void test1(){
709         Hello hello = (Hello)context.getBean("hello");
710         assertEquals("Hello Spring", hello.sayHello());
711         hello.print();
712
713         Printer printer = (Printer)context.getBean("printer");
714         assertEquals("Hello Spring", printer.toString());
715       }
716
717       @Test
718       public void test2(){
719         Hello hello = (Hello)context.getBean("hello");
720
721         Hello hello2 = context.getBean("hello", Hello.class);
722         assertSame(hello, hello2);
```

```
723         }
724       }
725
726    4)right-click > Run As > Junit Test
727    5)결과 -> Junit View에 초록색 bar
728
729  10. Spring-Test를 사용할 DI test class-HelloBeanJunitSpringTest.java 작성하기
730    1)Spring-Test library 설치
731    2)pom.xml code 추가
732       <dependency>
733          <groupId>org.springframework</groupId>
734          <artifactId>spring-test</artifactId>
735          <version>5.2.0.RELEASE</version>
736          <scope>test</scope>
737       </dependency>
738
739    3)pom.xml > right-click > Maven Install
740       -만일 Error 발생하면 pom.xml > right-click > Maven > Update Project...
741       -다시 Maven Install
742
743    4)Spring-Test를 사용할 DI test class-HelloBeanJunitSpringTest.java 작성하기
744       -/src/com.example.test/HelloBeanJunitTest.java 복사해서
745       -/src/com.example.test/HelloBeanJunitSpringTest.java로 이름 변경해서 붙여넣기
746
747          import org.junit.runner.RunWith;
748          import org.springframework.beans.factory.annotation.Autowired;
749          import org.springframework.test.context.ContextConfiguration;
750          import org.springframework.test.context.junit4.SpringJUnit4ClassRunner;
751          ...
752          @RunWith(SpringJUnit4ClassRunner.class)
753          @ContextConfiguration(locations="classpath:config/beans.xml")
754          public class HelloBeanJunitSpringTest {
755
756             @Autowired
757             ApplicationContext context;
758
759    5)만일 해당 객체를 찾을 수 없다는 오류가 계속 발생하면
760       -해당 Project > right-click > Build Path > Libraries tab
761       -spring-test-5.2.0.RELEASE.jar 선택 후 [Remove] 로 삭제
762       -[Add External JARs...] Click
763       -Local M2 Repository(e.g
          C:₩Users₩bluee₩.m2₩repository₩org₩springframework₩spring-test₩5.2.0.RELEASE)에서 직접 jar를
          선택할 것
764       -[Order and Export] tab에서 spring-test-5.2.0.RELEASE.jar 선택 후 [Up] button을 클릭
765       -해당 Project/src 바로 아래까지 올리고 [Apply and Close] Click
766
767       -아래 Code 삭제
768          @Before
769          public void init() {
770             context = new GenericXmlApplicationContext("config/beans.xml");
771          }
772
```

6)right-click > Run As > Junit Test
7)결과 -> Junit View에 초록색 bar

11. Hello class 수정

```
...
private List<String> names;
...
public void setNames(List<String> list){
   this.names = list;
}

public List<String> getNames(){
   return this.names;
}
...
```

12. beans.xml 수정
1)아래 코드 추가

```
<bean id="hello2" class="com.example.Hello">
   <property name="names">
      <list>
         <value>AOP</value>
         <value>Spring</value>
         <value>DI</value>
      </list>
   </property>
</bean>
```

13. HelloBeanJunitTest로 test하기
1)아래의 코드를 수정한다.

```
@Test <--@Ignore 붙여서 test하지 않는다.
public void test1(){
   Hello hello = (Hello)context.getBean("hello");
   assertEquals("Hello Spring", hello.sayHello());
   hello.print();

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

//아래 code로 수정할 것
@Test
public void test2(){
   Hello hello = (Hello) context.getBean("hello");

   Hello hello2 = context.getBean("hello2", Hello.class);

   assertEquals(3, hello2.getNames().size());
   List<String> list = hello2.getNames();
   for(String value : list){
      System.out.println(value);
```

```
825         }
826       }
827
828    2)right-click > Run As > Junit Test
829    3)결과 -> Junit View에 초록색 bar
830
831
832 ---------------------------------
833 Task 7. setter를 이용한 의존주입하기 실습
834 1. In Package Explorer > right-click > New > Java Project
835    1)Project Name : SpringDemo
836
837 2. src > right-click > New > Package
838    1)Package name : com.example
839
840 3. POJO class 작성
841    1)com.example > right-click > New > Class
842    2)Class Name : BmiCalculator
843       package com.example;
844
845       public class BmiCalculator {
846          private double lowWeight;
847          private double normal;
848          private double overWeight;
849          private double obesity;
850
851          public void setLowWeight(double lowWeight) {
852             this.lowWeight = lowWeight;
853          }
854
855          public void setNormal(double normal) {
856             this.normal = normal;
857          }
858
859          public void setOverWeight(double overWeight) {
860             this.overWeight = overWeight;
861          }
862
863          public void setObesity(double obesity) {
864             this.obesity = obesity;
865          }
866          public void bmiCalcu(double weight, double height){
867             double h = height * 0.01;
868             double result = weight / (h * h);
869
870             System.out.println("BMI 지수 : " + (int)result);
871
872             if(result > obesity)
873                System.out.println("비만입니다.");
874             else if(result > overWeight)
875                System.out.println("과체중입니다.");
876             else if(result > normal)
```

```
877                 System.out.println("정상입니다.");
878             else
879                 System.out.println("저체중입니다.");
880         }
881     }
882
883   3)com.example > right-click > New > Class
884   4)Class Name : MyInfo.java
885     package com.example;
886
887     import java.util.ArrayList;
888
889     public class MyInfo {
890         private String name;
891         private double height;
892         private double weight;
893         private ArrayList<String> hobby;
894         private BmiCalculator bmiCalculator;
895
896         public void setBmiCalculator(BmiCalculator bmiCalculator) {
897             this.bmiCalculator = bmiCalculator;
898         }
899         public void setName(String name) {
900             this.name = name;
901         }
902         public void setHeight(double height) {
903             this.height = height;
904         }
905         public void setWeight(double weight) {
906             this.weight = weight;
907         }
908         public void setHobby(ArrayList<String> hobby) {
909             this.hobby = hobby;
910         }
911         public void getInfo(){
912             System.out.println("Name : " + this.name);
913             System.out.println("Height : " + this.height);
914             System.out.println("Weight : " + this.weight);
915             System.out.println("Hobby : " + this.hobby);
916             this.bmiCalcu();
917         }
918         public void bmiCalcu(){
919             this.bmiCalculator.bmiCalcu(this.weight, this.height);
920         }
921     }
922
923 4. Java Project를 Spring Project로 변환
924   1)SpringDemo Project > right-click > Configue > Convert to Maven Project
925     -Project : /SpringDemo
926     -Group Id : SpringDemo
927     -Artifact Id : SpringDemo
928     -version : 0.0.1-SNAPSHOT
```

```
929       -Packaging : jar
930       -Finish
931
932    2)SpringDemo Project > right-click > Spring > Add Spring Project Nature
933
934    3)pom.xml file에 Spring Context Dependency 추가하기
935      <version>0.0.1-SNAPSHOT</version>
936       <dependencies>
937          <dependency>
938             <groupId>org.springframework</groupId>
939             <artifactId>spring-context</artifactId>
940             <version>5.2.0.RELEASE</version>
941          </dependency>
942       </dependencies>
943
944    4)pom.xml > right-click > Run As > Maven install
945       [INFO] BUILD SUCCESS 확인
946
947 5. SpringDemo/resources folder 생성
948    1)SpringDemo project > right-click > Build Path > Configure Build Path
949    2)Source Tab > Add Folder
950    3)SpringDemo 선택 확인
951    4)Create New Folder > Folder name : resources > Finish > OK
952    5)SpringDemo/resources(new) 확인
953    6)Apply and Close
954
955 6. Bean Configuration XML 작성
956    1)SpringDemo/resources > right-click > New > Other > Spring > Spring Bean Configuration File
957    -File name : applicationContext.xml > Finish
958
959    <bean id="bmiCalculator" class="com.example.BmiCalculator">
960       <property name="lowWeight" value="18.5" />
961       <property name="normal" value="23" />
962       <property name="overWeight" value="25" />
963       <property name="obesity">
964          <value>30</value>
965       </property>
966    </bean>
967    <bean id="myInfo" class="com.example.MyInfo">
968       <property name="name" value="한지민" />
969       <property name="height" value="170.5" />
970       <property name="weight" value="67" />
971       <property name="hobby">
972          <list>
973             <value>수영</value>
974             <value>요리</value>
975             <value>독서</value>
976          </list>
977       </property>
978       <property name="bmiCalculator">
979          <ref bean="bmiCalculator" />
980       </property>
```

```
    </bean>

7. MainClass 생성하기
   1)com.example.MainClass.java
      package com.example;

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

      public class MainClass {
         public static void main(String[] args) {
            String configFile = "classpath:applicationContext.xml";

            //Spring Container 생성
            AbstractApplicationContext context = new GenericXmlApplicationContext(configFile);

            //Spring Container 에서 객체를 가져옴
            MyInfo myInfo = context.getBean("myInfo", MyInfo.class);

            myInfo.getInfo();
            context.close();
         }
      }

8. Java Application 실행
   Name : 한지민
   Height : 170.5
   Weight : 67.0
   Hobby : [수영, 요리, 독서]
   BMI 지수 : 23
   정상입니다.


-----------------------------------------
Task 8. 생성자 이용하여 의존 주입하기 실습
1. In Package Explorer > right-click > New > Java Project
   Project name : DIDemo2

2. src > right-click > New > Package
   Package name : com.example

3. POJO class 작성
   1)com.example > right-click > New > Class
   2)Class Name : Hello
      package com.example;

      public class Hello{
         private String name;
         private Printer printer;

         public Hello(){}

```

```
1033        public void setName(String name){
1034            this.name = name;
1035        }
1036
1037        public void setPrinter(Printer printer){
1038            this.printer = printer;
1039        }
1040
1041        public String sayHello(){
1042            return "Hello " + name;
1043        }
1044
1045        public void print(){
1046            this.printer.print(sayHello());
1047        }
1048    }
1049
1050  3)com.example > right-click > New > Interface
1051  4)interface name : Printer
1052    package com.example;
1053
1054    public interface Printer{
1055        void print(String message);
1056    }
1057
1058  5)com.example > right-click > New > Class
1059  6)Class Name : StringPrinter
1060  7)Interfaces : com.example.Printer
1061    package com.example;
1062
1063    public class StringPrinter implements Printer{
1064        private StringBuffer buffer = new StringBuffer();
1065
1066        @Override
1067        public void print(String message){
1068            this.buffer.append(message);
1069        }
1070
1071        public String toString(){
1072            return this.buffer.toString();
1073        }
1074    }
1075
1076  8)com.example > right-click > New > Class
1077  9)Class Name : ConsolePrinter
1078  10)Intefaces : com.example.Printer
1079    package com.example;
1080
1081    public class ConsolePrinter implements Printer{
1082
1083        @Override
1084        public void print(String message){
```

```
            System.out.println(message);
        }
    }
```

4. Java Project를 Spring Project로 변환
  1)DIDemo2 Project > right-click > Configure > Convert to Maven Project
    -Project : /DIDemo2
    -Group Id : DIDemo2
    -Artifact Id : DIDemo2
    -version : 0.0.1-SNAPSHOT
    -Packaging : jar
    -Finish

  2)DIDemo2 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>5.2.0.RELEASE</version>
      </dependency>
   </dependencies>
```

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

5. DIDemo2/resources folder 생성
  1)DIDemo2 project > right-click > Build Path > Configure Build Path
  2)Source Tab > Add Folder
  3)DIDemo2 선택확인
  4)Create New Folder > Folder name : resources > Finish > OK
  5)DIDemo2/resources(new) 확인
  6)Apply and Close

6. Bean Configuration XML 작성
  1)DIDemo2/resources > right-click > New > Other > Spring > Spring Bean Configuration File
  -File name : beans.xml > 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="hello" class="com.example.Hello">
         <property name="name" value="Spring" />
         <property name="printer" ref="printer" />
      </bean>
      <bean id="printer" class="com.example.StringPrinter" />
      <bean id="consolePrinter" class="com.example.ConsolePrinter" />
```

```
1136
1137        </beans>
1138
1139  7. Test class 작성
1140     1)/src > right-click > New > Package
1141     2)Package Name : com.example.test
1142     3)/src/com.example/test/HelloBeanTest.java
1143
1144        package com.example.test;
1145
1146        import org.springframework.context.ApplicationContext;
1147        import org.springframework.context.support.GenericXmlApplicationContext;
1148
1149        import com.example.Hello;
1150        import com.example.Printer;
1151
1152        public class HelloBeanTest {
1153           public static void main(String [] args){
1154              //1. IoC Container 생성
1155              ApplicationContext context =
1156                    new GenericXmlApplicationContext("classpath:beans.xml");
1157
1158              //2. Hello Beans 가져오기
1159              Hello hello = (Hello)context.getBean("hello");
1160              System.out.println(hello.sayHello());
1161              hello.print();
1162
1163              //3. SpringPrinter 가져오기
1164              Printer printer = (Printer)context.getBean("printer");
1165              System.out.println(printer.toString());
1166
1167              Hello hello2 = context.getBean("hello", Hello.class);
1168              hello2.print();
1169
1170              System.out.println(hello == hello2);  //Singleton Pattern
1171           }
1172        }
1173
1174  8. Test
1175     1)/src/com.example.test/HelloBeanTest.java > right-click > Run As > Java Application
1176        Hello Spring
1177        Hello Spring
1178        true
1179
1180  9. /src/com.example.Hello 생성자 추가
1181
1182     public Hello(String name, Printer printer) {
1183        this.name = name;
1184        this.printer = printer;
1185     }
1186
1187  10. /resources/beans.xml에 아래 Code 추가
```

```
1188
1189    <bean id="hello2" class="com.example.Hello">
1190        <constructor-arg index="0" value="Spring" />
1191        <constructor-arg index="1" ref="printer" />
1192    </bean>
1193
1194  11. /src/com.example.test/HelloBeanTest.java 수정
1195
1196        ...
1197        //2. Hello Beans 가져오기
1198        Hello hello = (Hello)context.getBean("hello2");
1199        ...
1200        Hello hello2 = context.getBean("hello2", Hello.class);
1201        ...
1202
1203  12. Test
1204    1)/src/com.example.test/HelloBeanTest.java > right-click > Run As > Java Application
1205        Hello Spring
1206        Hello Spring
1207        true
1208
1209
1210  ----------------------------------------
1211  Task 9. 생성자 이용하여 의존 주입하기 실습
1212  1. In Package Explorer > right-click > New > Java Project
1213    1)Project Name : SpringDemo1
1214
1215  2. src > right-click > New > Package
1216    1)Package name : com.example
1217
1218  3. POJO Class 생성
1219    1)com.example.Student.java
1220        package com.example;
1221
1222        public class Student {
1223            private String name;
1224            private int age;
1225            private int grade;
1226            private int classNum;
1227            public Student(String name, int age, int grade, int classNum) {
1228                this.name = name;
1229                this.age = age;
1230                this.grade = grade;
1231                this.classNum = classNum;
1232            }
1233            public String getName() {
1234                return name;
1235            }
1236            public void setName(String name) {
1237                this.name = name;
1238            }
1239            public int getAge() {
```

```
            return age;
         }
         public void setAge(int age) {
            this.age = age;
         }
         public int getGrade() {
            return grade;
         }
         public void setGrade(int grade) {
            this.grade = grade;
         }
         public int getClassNum() {
            return classNum;
         }
         public void setClassNum(int classNum) {
            this.classNum = classNum;
         }
      }
```

2)com.example.StudentInfo.java

```
   package com.example;

   public class StudentInfo {
      private Student student;

      public StudentInfo(Student student) {
         this.student = student;
      }

      public void printInfo(){
         if(this.student != null){
            System.out.println("Name : " + this.student.getName());
            System.out.println("Age : " + this.student.getAge());
            System.out.println("Grade : " + this.student.getGrade());
            System.out.println("Class : " + this.student.getClassNum());
            System.out.println("------------------------");
         }
      }

      public void setStudent(Student student){
         this.student = student;
      }
   }
```

4. Java Project를 Spring Project로 변환

1)SpringDemo1 Project > right-click > Configure > Convert to Maven Project

-Project : /SpringDemo1
-Group Id : SpringDemo1
-Artifact Id : SpringDemo1
-version : 0.0.1-SNAPSHOT
-Packaging : jar
-Finish

2)SpringDemo1 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>5.2.0.RELEASE</version>
   </dependency>
 </dependencies>
```

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

5. SpringDemo1/resources folder 생성
1)SpringDemo1 project > right-click > Build Path > Configure Build Path
2)Source Tab > Add Folder
3)SpringDemo1 선택 확인
4)Create New Folder > Folder name : resources > Finish > OK
5)SpringDemo1/resources(new) 확인
6)Apply and Close

6. Bean Configuration XML 작성
1)SpringDemo1/resources > right-click > New > Other > Spring > Spring Bean Configuration File
2)File name : applicationContext.xml > 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="student1" class="com.example.Student">
    <constructor-arg>
      <value>한지민</value>
    </constructor-arg>
    <constructor-arg>
      <value>15</value>
    </constructor-arg>
    <constructor-arg>
      <value>2</value>
    </constructor-arg>
    <constructor-arg>
      <value>5</value>
    </constructor-arg>
  </bean>

  <bean id="student2" class="com.example.Student">
    <constructor-arg value="설운도" />
    <constructor-arg value="16" />
```

```
            <constructor-arg value="3" />
            <constructor-arg value="7" />
        </bean>

        <bean id="studentInfo" class="com.example.StudentInfo">
            <constructor-arg>
                <ref bean="student1"/>
            </constructor-arg>
        </bean>
    </beans>
```

7. com.example.MainClass.java

```
package com.example;

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

public class MainClass {
    public static void main(String[] args) {
        String configFile = "classpath:applicationContext.xml";
        AbstractApplicationContext context = new GenericXmlApplicationContext(configFile);
        StudentInfo studentInfo = context.getBean("studentInfo", StudentInfo.class);
        studentInfo.printInfo();

        Student student2 = context.getBean("student2", Student.class);
        studentInfo.setStudent(student2);
        studentInfo.printInfo();

        context.close();
    }
}
```

8. Java Application 실행

```
Name : 한지민
Age : 15
Grade : 2
Class : 5
------------------------
Name : 설운도
Age : 16
Grade : 3
Class : 7
------------------------
```

Task 10. Context file 여러개 사용하기

1. In Package Explorer > right-click > New > Java Project
   1)Project Name : SpringDemo2

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

```
1395
1396 3. POJO Class 생성
1397    1)com.example.Student.java
1398       package com.example;
1399
1400       import java.util.ArrayList;
1401
1402       public class Student {
1403          private String name;
1404          private int age;
1405          private ArrayList<String> hobbys;
1406          private double height;
1407          private double weight;
1408          public Student(String name, int age, ArrayList<String> hobbys) {
1409             this.name = name;
1410             this.age = age;
1411             this.hobbys = hobbys;
1412          }
1413          public void setName(String name) {
1414             this.name = name;
1415          }
1416          public void setAge(int age) {
1417             this.age = age;
1418          }
1419          public void setHobbys(ArrayList<String> hobbys) {
1420             this.hobbys = hobbys;
1421          }
1422          public void setHeight(double height) {
1423             this.height = height;
1424          }
1425          public void setWeight(double weight) {
1426             this.weight = weight;
1427          }
1428          @Override
1429          public String toString() {
1430             return String.format("Student [name=%s, age=%s, hobbys=%s, height=%s, weight=%s]", name,
                age, hobbys, height,
1431                   weight);
1432          }
1433       }
1434
1435    2)com.example.StudentInfo.java
1436       package com.example;
1437       public class StudentInfo {
1438          private Student student;
1439
1440          public Student getStudent() {
1441             return student;
1442          }
1443
1444          public void setStudent(Student student) {
1445             this.student = student;
```

```
        }
    }

3)com.example.Product.java
    package com.example;
    public class Product {
        private String pName;
        private int pPrice;
        private String maker;
        private String color;
        public Product(String pName, int pPrice) {
            this.pName = pName;
            this.pPrice = pPrice;
        }
        public void setpName(String pName) {
            this.pName = pName;
        }
        public void setpPrice(int pPrice) {
            this.pPrice = pPrice;
        }
        public void setMaker(String maker) {
            this.maker = maker;
        }
        public void setColor(String color) {
            this.color = color;
        }
        @Override
        public String toString() {
            return String.format("Product [pName=%s, pPrice=%s, maker=%s, color=%s]", pName, pPrice,
            maker, color);
        }
    }

4. Java Project를 Spring Project로 변환
  1)SpringDemo2 Project > right-click > Configure > Convert to Maven Project
    -Project : /SpringDemo2
    -Group Id : SpringDemo2
    -Artifact Id : SpringDemo2
    -version : 0.0.1-SNAPSHOT
    -Packaging : jar
    -Finish

  2)SpringDemo2 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>5.2.0.RELEASE</version>
        </dependency>
```

```
1497        </dependencies>
1498
1499     4)pom.xml > right-click > Run As > Maven install
1500        [INFO] BUILD SUCCESS 확인
1501
1502  5. SpringDemo2/resources folder 생성
1503     1)SpringDemo2 project > right-click > Build Path > Configure Build Path
1504     2)Source Tab > Add Folder
1505     3)SpringDemo2 선택 확인
1506     4)Create New Folder > Folder name : resources > Finish > OK
1507     5)SpringDemo2/resources(new) 확인
1508     6)Apply and Close
1509
1510  6. Bean Configuration XML 작성
1511     1)SpringDemo2/resources > right-click > New > Other > Spring > Spring Bean Configuration File
1512     2)File name : applicationContext.xml > Finish
1513     3)SpringDemo2/resources > right-click > New > Other > Spring > Spring Bean Configuration File
1514     4)File name : applicationContext2.xml > Finish
1515
1516  7. applicationContext.xml
1517     <?xml version="1.0" encoding="UTF-8"?>
1518     <beans xmlns="http://www.springframework.org/schema/beans"
1519        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
1520        xsi:schemaLocation="http://www.springframework.org/schema/beans
           http://www.springframework.org/schema/beans/spring-beans.xsd">
1521
1522        <bean id="student1" class="com.example.Student">
1523           <constructor-arg value="한지민" />
1524           <constructor-arg value="25" />
1525           <constructor-arg>
1526              <list>
1527                 <value>독서</value>
1528                 <value>영화감상</value>
1529                 <value>요리</value>
1530              </list>
1531           </constructor-arg>
1532           <property name="height" value="165" />
1533           <property name="weight">
1534              <value>45</value>
1535           </property>
1536        </bean>
1537
1538        <bean id="studentInfo1" class="com.example.StudentInfo">
1539           <property name="student">
1540              <ref bean="student1" />
1541           </property>
1542        </bean>
1543     </beans>
1544
1545  8. /resources/applicationContext2.xml
1546     1)Namespace tab을 선택하여 c, p를 선택한다.
1547        <?xml version="1.0" encoding="UTF-8"?>
```

```
1548        <beans xmlns="http://www.springframework.org/schema/beans"
1549           xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
1550           xmlns:c="http://www.springframework.org/schema/c"
1551           xmlns:p="http://www.springframework.org/schema/p"
1552           xsi:schemaLocation="http://www.springframework.org/schema/beans
               http://www.springframework.org/schema/beans/spring-beans.xsd">
1553
1554           <bean id="student3" class="com.example.Student">
1555              <constructor-arg value="설운도" />
1556              <constructor-arg value="50" />
1557              <constructor-arg>
1558                 <list>
1559                    <value>노래부르기</value>
1560                    <value>게임</value>
1561                 </list>
1562              </constructor-arg>
1563              <property name="height" value="175" />
1564              <property name="weight">
1565                 <value>75</value>
1566              </property>
1567           </bean>
1568
1569           <bean id="product" class="com.example.Product" c:pName="Computer" c:pPrice="2000000"
               p:maker="Samsung">
1570              <property name="color" value="Yellow" />
1571           </bean>
1572        </beans>
1573
1574  9. com.example.MainClass
1575     package com.example;
1576
1577     import org.springframework.context.support.AbstractApplicationContext;
1578     import org.springframework.context.support.GenericXmlApplicationContext;
1579
1580     public class MainClass {
1581        public static void main(String[] args) {
1582           String configFile = "classpath:applicationContext.xml";
1583           String configFile1 = "classpath:applicationContext2.xml";
1584           AbstractApplicationContext context = new GenericXmlApplicationContext(configFile, configFile1);
1585           Student student1 = context.getBean("student1", Student.class);
1586           System.out.println(student1);
1587
1588           StudentInfo studentInfo = context.getBean("studentInfo1", StudentInfo.class);
1589           Student student2 = studentInfo.getStudent();
1590           System.out.println(student2);
1591           if(student1.equals(student2)) System.out.println("Equals");
1592           else System.out.println("Different");
1593
1594           Student student3 = context.getBean("student3", Student.class);
1595           System.out.println(student3);
1596
1597           if(student1.equals(student3)) System.out.println("Equals");
```

```
1598            else System.out.println("Different");
1599
1600            Product product = context.getBean("product", Product.class);
1601            System.out.println(product);
1602            context.close();
1603        }
1604    }
1605
1606  10. Java Application 실행
1607    Student [name=한지민, age=25, hobbys=[독서, 영화감상, 요리], height=165.0,weight=45.0]
1608    Student [name=한지민, age=25, hobbys=[독서, 영화감상, 요리], height=165.0,weight=45.0]
1609    Equals
1610    Student [name=설운도, age=50, hobbys=[노래부르기, 게임], height=175.0,weight=75.0]
1611    Different
1612    Product [pName=Computer, pPrice=2000000, maker=Samsung, color=Yellow]
1613
1614
1615  ---------------------------------
1616  Task 11. Java Annotation을 이용한 DI 설정하기
1617  1. In Package Explorer > right-click > New > Java Project
1618    1)Project Name : SpringDemo3
1619
1620  2. src > right-click > New > Package
1621    1)Package name : com.example
1622
1623  3. POJO 생성
1624    1)com.example.Student.java
1625        package com.example;
1626
1627        import java.util.ArrayList;
1628
1629        public class Student {
1630            private String name;
1631            private int age;
1632            private ArrayList<String> hobbys;
1633            private double height;
1634            private double weight;
1635            public Student(String name, int age, ArrayList<String> hobbys) {
1636                this.name = name;
1637                this.age = age;
1638                this.hobbys = hobbys;
1639            }
1640            public void setName(String name) {
1641                this.name = name;
1642            }
1643            public void setAge(int age) {
1644                this.age = age;
1645            }
1646            public void setHobbys(ArrayList<String> hobbys) {
1647                this.hobbys = hobbys;
1648            }
1649            public void setHeight(double height) {
```

```
1650            this.height = height;
1651         }
1652         public void setWeight(double weight) {
1653            this.weight = weight;
1654         }
1655         @Override
1656         public String toString() {
1657            return String.format("Student [name=%s, age=%s, hobbys=%s, height=%s, weight=%s]", name,
               age, hobbys, height,
1658                  weight);
1659         }
1660      }
1661
1662 4. Java Project를 Spring Project로 변환
1663    1)SpringDemo3 Project > right-click > Configure > Convert to Maven Project
1664       -Project : /SpringDemo3
1665       -Group Id : SpringDemo3
1666       -Artifact Id : SpringDemo3
1667       -version : 0.0.1-SNAPSHOT
1668       -Packaging : jar
1669       -Finish
1670
1671    2)SpringDemo3 Project > right-click > Spring > Add Spring Project Nature
1672
1673    3)pom.xml file에 Spring Context Dependency 추가하기
1674          <version>0.0.1-SNAPSHOT</version>
1675          <dependencies>
1676             <dependency>
1677                <groupId>org.springframework</groupId>
1678                <artifactId>spring-context</artifactId>
1679                <version>5.2.0.RELEASE</version>
1680             </dependency>
1681          </dependencies>
1682
1683    4)pom.xml > right-click > Run As > Maven install
1684       [INFO] BUILD SUCCESS 확인
1685
1686 5. com.example.ApplicationConfig.java
1687    import org.springframework.context.annotation.Bean;
1688    import org.springframework.context.annotation.Configuration;
1689
1690    @Configuration
1691    public class ApplicationConfig {
1692
1693       @Bean
1694       public Student student1(){
1695          ArrayList<String> hobbys = new ArrayList<String>();
1696          hobbys.add("독서");
1697          hobbys.add("영화감상");
1698          hobbys.add("요리");
1699
1700          Student student = new Student("한지민", 25, hobbys);
```

```
1701            student.setHeight(165);
1702            student.setWeight(45);
1703
1704            return student;
1705        }
1706
1707        @Bean
1708        public Student student2(){
1709            ArrayList<String> hobbys = new ArrayList<String>();
1710            hobbys.add("노래부르기");
1711            hobbys.add("게임");
1712            Student student = new Student("설운도", 50, hobbys);
1713            student.setHeight(175);
1714            student.setWeight(75);
1715
1716            return student;
1717        }
1718    }
1719
1720  6. com.example.MainClass.java
1721     package com.example;
1722
1723     import org.springframework.context.annotation.AnnotationConfigApplicationContext;
1724
1725     public class MainClass {
1726        public static void main(String[] args) {
1727            AnnotationConfigApplicationContext context = new
                AnnotationConfigApplicationContext(ApplicationConfig.class);
1728            Student student1 = context.getBean("student1", Student.class);
1729            System.out.println(student1);
1730
1731            Student student2 = context.getBean("student2", Student.class);
1732            System.out.println(student2);
1733
1734            context.close();
1735        }
1736     }
1737
1738  7. Java Application 실행
1739     Student [name=한지민, age=25, hobbys=[독서, 영화감상, 요리], height=165.0,weight=45.0]
1740     Student [name=설운도, age=50, hobbys=[노래부르기, 게임], height=175.0,weight=75.0]
1741
1742
1743  ------------------------------------
1744  Task 12. Java Annotation과 XML 을 이용한 DI 설정 방법 : XML file에 Java file을 포함시켜 사용하는 방법
1745  1. In Package Explorer > right-click > New > Java Project
1746     1)Project Name : SpringDemo4
1747
1748  2. src > right-click > New > Package
1749     1)Package name : com.example
1750
1751  3. POJO 생성
```

```
1752    1)com.example.Student.java
1753      package com.example;
1754
1755      import java.util.ArrayList;
1756
1757      public class Student {
1758        private String name;
1759        private int age;
1760        private ArrayList<String> hobbys;
1761        private double height;
1762        private double weight;
1763        public Student(String name, int age, ArrayList<String> hobbys) {
1764          this.name = name;
1765          this.age = age;
1766          this.hobbys = hobbys;
1767        }
1768        public void setName(String name) {
1769          this.name = name;
1770        }
1771        public void setAge(int age) {
1772          this.age = age;
1773        }
1774        public void setHobbys(ArrayList<String> hobbys) {
1775          this.hobbys = hobbys;
1776        }
1777        public void setHeight(double height) {
1778          this.height = height;
1779        }
1780        public void setWeight(double weight) {
1781          this.weight = weight;
1782        }
1783        @Override
1784        public String toString() {
1785          return String.format("Student [name=%s, age=%s, hobbys=%s, height=%s, weight=%s]", name,
                 age, hobbys, height,
1786                 weight);
1787        }
1788      }
1789
1790  4. Java Project를 Spring Project로 변환
1791    1)SpringDemo4 Project > right-click > Configure > Convert to Maven Project
1792      -Project : /SpringDemo4
1793      -Group Id : SpringDemo4
1794      -Artifact Id : SpringDemo4
1795      -version : 0.0.1-SNAPSHOT
1796      -Packaging : jar
1797      -Finish
1798
1799    2)SpringDemo4 Project > right-click > Spring > Add Spring Project Nature
1800
1801    3)pom.xml file에 Spring Context Dependency 추가하기
1802      <version>0.0.1-SNAPSHOT</version>
```

```
1803        <dependencies>
1804          <dependency>
1805            <groupId>org.springframework</groupId>
1806            <artifactId>spring-context</artifactId>
1807            <version>5.2.0.RELEASE</version>
1808          </dependency>
1809        </dependencies>
1810
1811      4)pom.xml > right-click > Run As > Maven install
1812        [INFO] BUILD SUCCESS 확인
1813
1814
1815    5. com.example.ApplicationConfig.java
1816      package com.example;
1817
1818      import java.util.ArrayList;
1819
1820      import org.springframework.context.annotation.Bean;
1821      import org.springframework.context.annotation.Configuration;
1822
1823      @Configuration
1824      public class ApplicationConfig {
1825        @Bean
1826        public Student student1(){
1827          ArrayList<String> hobbys = new ArrayList<String>();
1828          hobbys.add("독서");
1829          hobbys.add("영화감상");
1830          hobbys.add("요리");
1831
1832          Student student = new Student("한지민", 25, hobbys);
1833          student.setHeight(165);
1834          student.setWeight(45);
1835
1836          return student;
1837        }
1838      }
1839
1840    6. SpringDemo4/resources folder 생성
1841      1)SpringDemo4 project > right-click > Build Path > Configure Build Path
1842      2)Source Tab > Add Folder
1843      3)SpringDemo4 선택 확인
1844      4)Create New Folder > Folder name : resources > Finish > OK
1845      5)SpringDemo4/resources(new) 확인
1846      6)Apply and Close
1847
1848    7. Bean Configuration XML 작성
1849      1)SpringDemo4/resources > right-click > New > Other > Spring > Spring Bean Configuration File
1850      2)File name : applicationContext.xml > Finish
1851
1852    8. /resources/applicationContext.xml
1853      <?xml version="1.0" encoding="UTF-8"?>
1854      <beans xmlns="http://www.springframework.org/schema/beans"
```

```
1855        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
1856        xmlns:context="http://www.springframework.org/schema/context"
1857        xsi:schemaLocation="http://www.springframework.org/schema/beans
            http://www.springframework.org/schema/beans/spring-beans.xsd">
1858
1859        <bean class="org.springframework.context.annotation.ConfigurationClassPostProcessor" />
1860        <bean class="com.example.ApplicationConfig" />
1861        <bean id="student3" class="com.example.Student">
1862           <constructor-arg value="설운도" />
1863           <constructor-arg value="50" />
1864           <constructor-arg>
1865              <list>
1866                 <value>노래부르기</value>
1867                 <value>게임</value>
1868              </list>
1869           </constructor-arg>
1870           <property name="height" value="175" />
1871           <property name="weight">
1872              <value>75</value>
1873           </property>
1874        </bean>
1875     </beans>
1876
1877  9. com.example.MainClass.java
1878     package com.example;
1879
1880     import org.springframework.context.support.AbstractApplicationContext;
1881     import org.springframework.context.support.GenericXmlApplicationContext;
1882
1883     public class MainClass {
1884        public static void main(String[] args) {
1885           String configFile = "classpath:applicationContext.xml";
1886           AbstractApplicationContext context = new GenericXmlApplicationContext(configFile);
1887           Student student1 = context.getBean("student1", Student.class);
1888           System.out.println(student1);
1889
1890           Student student3 = context.getBean("student3", Student.class);
1891           System.out.println(student3);
1892        }
1893     }
1894
1895  10. Java Application 실행
1896     Student [name=한지민, age=25, hobbys=[독서, 영화감상, 요리], height=165.0,weight=45.0]
1897     Student [name=설운도, age=50, hobbys=[노래부르기, 게임], height=175.0,weight=75.0]
1898
1899
1900  ------------------------------
1901  Task 13. Java Annotation과 XML 을 이용한 DI 설정 방법 : Java file에 XML file을 포함시켜 사용하는 방법
1902  1. In Package Explorer > right-click > New > Java Projectn
1903     1)Project Name : SpringDemo5
1904
1905  2. src > right-click > New > Package
```

1906 1)Package name : com.example
1907
1908 3. com.example.Student.java
1909 package com.example;
1910
1911 import java.util.ArrayList;
1912
1913 public class Student {
1914 private String name;
1915 private int age;
1916 private ArrayList<String> hobbys;
1917 private double height;
1918 private double weight;
1919 public Student(String name, int age, ArrayList<String> hobbys) {
1920 this.name = name;
1921 this.age = age;
1922 this.hobbys = hobbys;
1923 }
1924 public void setName(String name) {
1925 this.name = name;
1926 }
1927 public void setAge(int age) {
1928 this.age = age;
1929 }
1930 public void setHobbys(ArrayList<String> hobbys) {
1931 this.hobbys = hobbys;
1932 }
1933 public void setHeight(double height) {
1934 this.height = height;
1935 }
1936 public void setWeight(double weight) {
1937 this.weight = weight;
1938 }
1939 @Override
1940 public String toString() {
1941 return String.format("Student [name=%s, age=%s, hobbys=%s, height=%s, weight=%s]", name,
age, hobbys, height,
1942 weight);
1943 }
1944 }
1945
1946 4. Java Project를 Spring Project로 변환
1947 1)SpringDemo5 Project > right-click > Configure > Convert to Maven Project
1948 -Project : /SpringDemo5
1949 -Group Id : SpringDemo5
1950 -Artifact Id : SpringDemo5
1951 -version : 0.0.1-SNAPSHOT
1952 -Packaging : jar
1953 -Finish
1954
1955 2)SpringDemo5 Project > right-click > Spring > Add Spring Project Nature
1956

```
1957    3)pom.xml file에 Spring Context Dependency 추가하기
1958      <version>0.0.1-SNAPSHOT</version>
1959       <dependencies>
1960          <dependency>
1961             <groupId>org.springframework</groupId>
1962             <artifactId>spring-context</artifactId>
1963             <version>5.2.0.RELEASE</version>
1964          </dependency>
1965       </dependencies>
1966
1967    4)pom.xml > right-click > Run As > Maven install
1968       [INFO] BUILD SUCCESS 확인
1969
1970  5. SpringDemo5/resources folder 생성
1971    1)SpringDemo5 project > right-click > Build Path > Configure Build Path
1972    2)Source Tab > Add Folder
1973    3)SpringDemo5 선택 확인
1974    4)Create New Folder > Folder name : resources > Finish > OK
1975    5)SpringDemo5/resources(new) 확인
1976    6)Apply and Close
1977
1978  6. Bean Configuration XML 작성
1979    1)SpringDemo4/resources > right-click > New > Other > Spring > Spring Bean Configuration File
1980    2)File name : applicationContext.xml > Finish
1981
1982  7. /resources/applicationContext.xml
1983    <?xml version="1.0" encoding="UTF-8"?>
1984    <beans xmlns="http://www.springframework.org/schema/beans"
1985       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
1986       xsi:schemaLocation="http://www.springframework.org/schema/beans
          http://www.springframework.org/schema/beans/spring-beans.xsd">
1987
1988       <bean id="student3" class="com.example.Student">
1989          <constructor-arg value="설운도" />
1990          <constructor-arg value="50" />
1991          <constructor-arg>
1992             <list>
1993                <value>노래부르기</value>
1994                <value>게임</value>
1995             </list>
1996          </constructor-arg>
1997          <property name="height" value="175" />
1998          <property name="weight">
1999             <value>75</value>
2000          </property>
2001       </bean>
2002    </beans>
2003
2004  8. com.example.ApplicationConfig.java
2005    package com.example;
2006
2007    import java.util.ArrayList;
```

```
2008
2009    import org.springframework.context.annotation.Bean;
2010    import org.springframework.context.annotation.Configuration;
2011    import org.springframework.context.annotation.ImportResource;
2012
2013    @Configuration
2014    @ImportResource("classpath:ApplicationContext.xml")
2015    public class ApplicationConfig {
2016
2017       @Bean
2018       public Student student1(){
2019          ArrayList<String> hobbys = new ArrayList<String>();
2020          hobbys.add("독서");
2021          hobbys.add("영화감상");
2022          hobbys.add("요리");
2023
2024          Student student = new Student("한지민", 25, hobbys);
2025          student.setHeight(165);
2026          student.setWeight(45);
2027
2028          return student;
2029       }
2030    }
2031
2032  9. com.example.MainClass.java
2033    package com.example;
2034
2035    import org.springframework.context.annotation.AnnotationConfigApplicationContext;
2036
2037    public class MainClass {
2038       public static void main(String[] args) {
2039          AnnotationConfigApplicationContext context = new
           AnnotationConfigApplicationContext(ApplicationConfig.class);
2040          Student student1 = context.getBean("student1", Student.class);
2041          System.out.println(student1);
2042
2043          Student student3 = context.getBean("student3", Student.class);
2044          System.out.println(student3);
2045
2046          context.close();
2047       }
2048    }
2049
2050  10. Java Application 실행
2051    Student [name=한지민, age=25, hobbys=[독서, 영화감상, 요리], height=165.0,weight=45.0]
2052    Student [name=설운도, age=50, hobbys=[노래부르기, 게임], height=175.0,weight=75.0]
2053
2054
2055  ----------------------------------------
2056  Task 14. Lab
2057  1. In Package Explorer > right-click > New > Java Project
2058     1)Project name : DIDemo3
```

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

3. POJO class 작성
   1)com.example > right-click > New > Class
   2)Class Name : Hello

```
package com.example;

public class Hello{
   private String name;
   private Printer printer;

   public Hello(){}

   public void setName(String name){
      this.name = name;
   }

   public void setPrinter(Printer printer){
      this.printer = printer;
   }

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

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

   3)com.example > right-click > New > Interface
   4)interface name : Printer

```
package com.example;

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

   5)com.example > right-click > New > Class
   6)Class Name : StringPrinter
   7)Interfaces : com.example.Printer

```
package com.example;

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

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

```
2111
2112            public String toString(){
2113               return this.buffer.toString();
2114            }
2115         }
2116
2117      8)com.example > right-click > New > Class
2118      9)Class Name : ConsolePrinter
2119      10)Interfaces : com.example.Printer
2120         package com.example;
2121
2122         public class ConsolePrinter implements Printer{
2123
2124            @Override
2125            public void print(String message){
2126               System.out.println(message);
2127            }
2128         }
2129
2130   4. Java Project를 Spring Project로 변환
2131      1)DIDemo3 Project > right-click > Configure > Convert to Maven Project
2132         -Project : /DIDemo3
2133         -Group Id : DIDemo3
2134         -Artifact Id : DIDemo3
2135         -version : 0.0.1-SNAPSHOT
2136         -Packaging : jar
2137         -Finish
2138
2139      2)DIDemo3 Project > right-click > Spring > Add Spring Project Nature
2140
2141      3)pom.xml file에 Spring Context Dependency 추가하기
2142        <version>0.0.1-SNAPSHOT</version>
2143         <dependencies>
2144            <dependency>
2145               <groupId>org.springframework</groupId>
2146               <artifactId>spring-context</artifactId>
2147               <version>5.2.0.RELEASE</version>
2148            </dependency>
2149         </dependencies>
2150
2151      4)pom.xml > right-click > Run As > Maven install
2152         [INFO] BUILD SUCCESS 확인
2153
2154   5. src/config folder 생성
2155      1)/src > right-click > New > Folder
2156      2)Folder name : config
2157
2158   6. Bean Configuration XML 작성
2159      1)/src/config > right-click > New > Other > Spring > Spring Bean Configuration File
2160      2)File name : beans.xml > Finish
2161         <?xml version="1.0" encoding="UTF-8"?>
2162         <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-4.3.xsd">

        <bean id="hello" class="com.example.Hello">
            <property name="name" value="Spring" />
            <property name="printer" ref="printer" />
        </bean>
        <bean id="printer" class="com.example.StringPrinter" />
        <bean id="consolePrinter" class="com.example.ConsolePrinter" />

    </beans>
```

7. DI Test class 작성

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

2)Package Name : com.example.test

3)/src/com.example/test/HelloBeanTest.java

```
    package com.example.test;

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

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

    public class HelloBeanTest {
        public static void main(String [] args){
            //1. IoC Container 생성
            ApplicationContext context =
                    new GenericXmlApplicationContext("config/beans.xml");

            //2. Hello Beans 가져오기
            Hello hello = (Hello)context.getBean("hello");
            System.out.println(hello.sayHello());
            hello.print();

            //3. SpringPrinter 가져오기
            Printer printer = (Printer)context.getBean("printer");
            System.out.println(printer.toString());

            Hello hello2 = context.getBean("hello", Hello.class);
            hello2.print();

            System.out.println(hello == hello2);  //Singleton Pattern
        }
    }
```

4)Java Application 실행

```
    Hello Spring
    Hello Spring
    true
```

8. jUnit Library 설치
1)jUnit 4.12 버전을 pom.xml에 추가

```
<dependency>
    <groupId>junit</groupId>
    <artifactId>junit</artifactId>
    <version>4.12</version>
    <scope>test</scope>
</dependency>
```

2)pom.xml > right-click > Run As > Maven Install

9. jUnit을 사용한 DI test class(HelloBeanJunitTest.java) 작성
1)/src/com.example.test/HelloBeanTest.java 복사
2)/src/com.example.test/ 붙여넣고 이름변경 -> HelloBeanJunitTest.java

```
package com.example.test;

import org.junit.Before;
import org.junit.Test;
import org.springframework.context.ApplicationContext;
import org.springframework.context.support.GenericXmlApplicationContext;

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

import static org.junit.Assert.assertEquals;
import static org.junit.Assert.assertSame;

public class HelloBeanJunitTest {
   ApplicationContext context;

   @Before
   public void init(){
      //항상 먼저 ApplicationContext를 생성해야 하기 때문에
      //1. IoC Container 생성
      context = new GenericXmlApplicationContext("config/beans.xml");
   }

   @Test
   public void test1(){
      //2. Hello Beans 가져오기
      Hello hello = (Hello)context.getBean("hello");
      assertEquals("Hello Spring", hello.sayHello());
      hello.print();

      //3. SpringPrinter 가져오기
      Printer printer = (Printer)context.getBean("printer");
      assertEquals("Hello Spring", printer.toString());
   }

```

```
2266          @Test
2267          public void test2(){
2268             Hello hello = (Hello)context.getBean("hello");
2269
2270             Hello hello2 = context.getBean("hello", Hello.class);
2271             assertSame(hello, hello2);
2272          }
2273       }
2274
2275    3)right-click > Run As > Junit Test
2276    4)결과 -> Junit View에 초록색 bar
2277
2278 10. Spring TestContext Framework
2279    1)Spring-Test library 설치
2280    2)pom.xml 수정
2281
2282       <dependency>
2283          <groupId>org.springframework</groupId>
2284          <artifactId>spring-test</artifactId>
2285          <version>5.2.0.RELEASE</version>
2286          <scope>test</scope>
2287       </dependency>
2288
2289    3)pom.xml > right-click > Maven Install
2290       -만일 Error 발생시 DIDemo3 > right-click > Maven > Update Project... > Ok
2291       -다시 Maven Install 실행
2292
2293    4)Spring-Test를 사용할 DI test class-HelloBeanJunitSpringTest.java 작성하기
2294       -/src/com.example.test/HelloBeanJunitTest.java 복사해서
2295       -/src/com.example.test/HelloBeanJunitSpringTest.java 로 붙여넣기
2296
2297          import org.junit.runner.RunWith;
2298          import org.springframework.beans.factory.annotation.Autowired;
2299          import org.springframework.test.context.ContextConfiguration;
2300          import org.springframework.test.context.junit4.SpringJUnit4ClassRunner;
2301          ...
2302          @RunWith(SpringJUnit4ClassRunner.class)
2303          @ContextConfiguration(locations="classpath:config/beans.xml")
2304          public class HelloBeanJunitSpringTest {
2305
2306             @Autowired
2307             ApplicationContext context;
2308
2309    5)아래 코드 삭제
2310       @Before
2311       public void init() {
2312          context = new GenericXmlApplicationContext("config/beans.xml");
2313       }
2314
2315    6)right-click > Run As > Junit Test
2316    7)결과 -> Junit View에 초록색 bar
2317    8)만일 해당 객체를 찾을 수 없다는 오류가 계속 발생하면
```

-해당 Project > right-click > Build Path > Libraries tab
-spring-test-5.2.0.RELEASE.jar 선택 후 [Remove] 로 삭제
-[Add External JARs...] Click
-Local M2 Repository(e.g C:₩Users₩bluee₩.m2₩repository₩org₩springframework₩spring-test₩5.2.0.RELEASE)에서 직접 jar를 선택할 것
-[Order and Export] tab에서 spring-test-5.2.0.RELEASE.jar 선택 후 [Up] button을 클릭
-해당 Project/src 바로 아래까지 올리고 [Apply and Close] Click

11. src/com.example/StringPrinter.java 수정

```
package com.example;

import org.springframework.stereotype.Component;

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

12. src/com.example/ConsolePrinter.java 수정

```
package com.example;

import org.springframework.stereotype.Component;

@Component("consolePrinter")
public class ConsolePrinter implements Printer{
...
```

13. /src/com.example/Hello.java 수정

```
package com.example;

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

@Component
public class Hello {
   @Value("Spring")
   private String name;

   @Autowired
   @Qualifier("stringPrinter")
   private Printer printer;

   //setter method가 필요 없음.

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

```

```
2368        public void print(){
2369            this.printer.print(sayHello());
2370        }
2371    }
2372
2373 14. 기존의 설정file과 충돌이 발생하기 때문에 /src/config/beans.xml 삭제
2374
2375 15. 새로운 설정 file 생성
2376    1)/src/config/beans.xml 새로 생성
2377    2)/src/config > right-click > New > Spring Bean Configuration File
2378    3)File name : annos.xml > Finish
2379    4)Namespace tab > context  Check
2380
2381    <?xml version="1.0" encoding="UTF-8"?>
2382    <beans xmlns="http://www.springframework.org/schema/beans"
2383        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
2384        xmlns:context="http://www.springframework.org/schema/context"
2385        xsi:schemaLocation="http://www.springframework.org/schema/beans
           http://www.springframework.org/schema/beans/spring-beans.xsd
2386          http://www.springframework.org/schema/context
              http://www.springframework.org/schema/context/spring-context-4.3.xsd">
2387
2388        <context:component-scan base-package="com.example" />
2389    </beans>
2390
2391 16. /src/com.example.test/HelloBeanJunitSpringTest.java 수정하기
2392        package com.example.test;
2393
2394        import static org.junit.Assert.assertEquals;
2395
2396        import org.junit.Test;
2397        import org.junit.runner.RunWith;
2398        import org.springframework.beans.factory.annotation.Autowired;
2399        import org.springframework.context.ApplicationContext;
2400        import org.springframework.test.context.ContextConfiguration;
2401        import org.springframework.test.context.junit4.SpringJUnit4ClassRunner;
2402
2403        import com.example.Hello;
2404
2405        @RunWith(SpringJUnit4ClassRunner.class)
2406        @ContextConfiguration(locations="classpath:config/annos.xml")
2407        public class HelloBeanJunitSpringTest {
2408            @Autowired
2409            ApplicationContext context;
2410
2411            @Test
2412            public void test(){
2413                Hello hello = context.getBean("hello", Hello.class);
2414                assertEquals("Hello Spring", hello.sayHello());
2415            }
2416        }
2417
```

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