



박 경 태

comsi.java@gmail.com

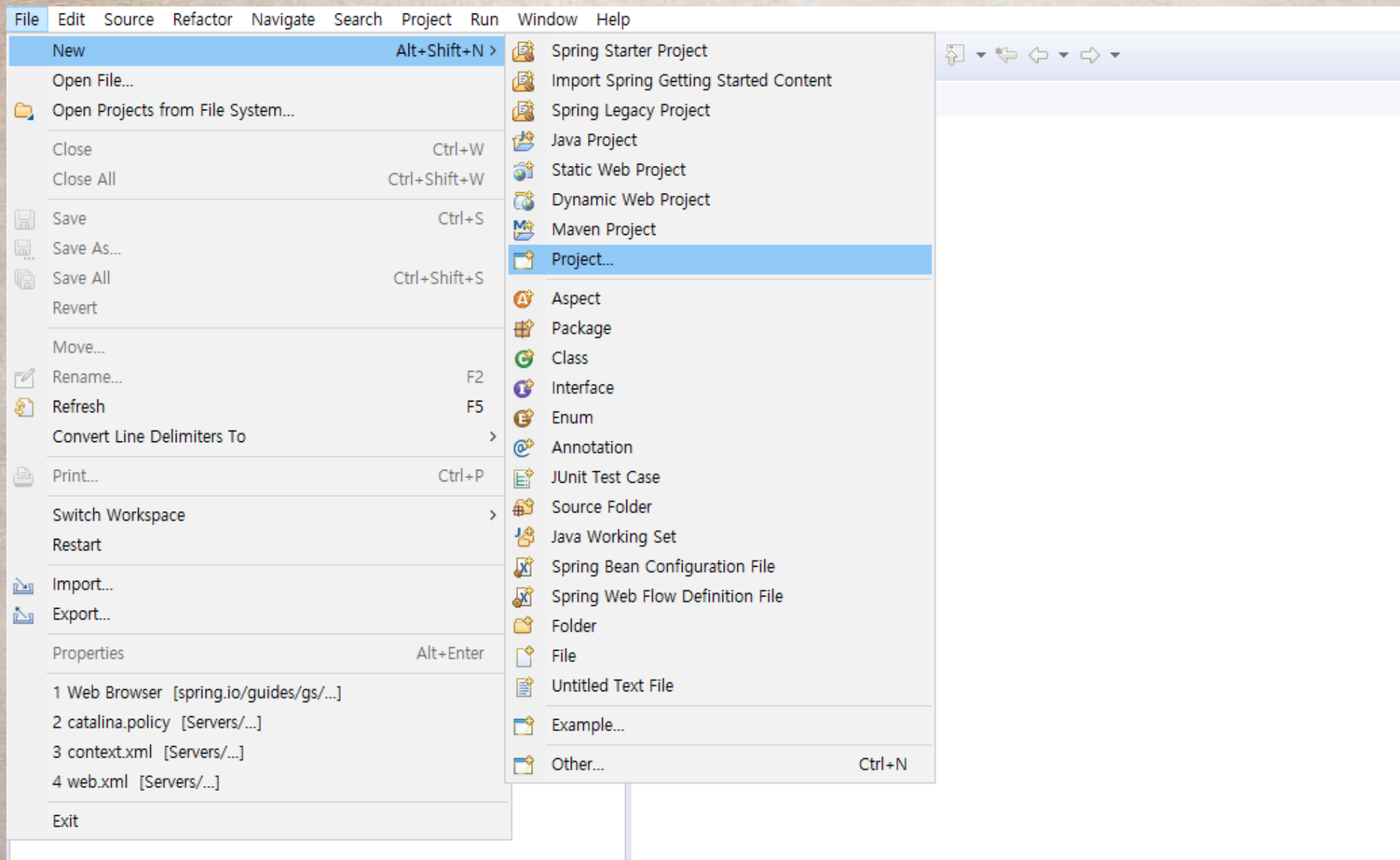
고급 자바 프로그래밍
: STS를 이용한 Spring 프로그래밍

강의 내용

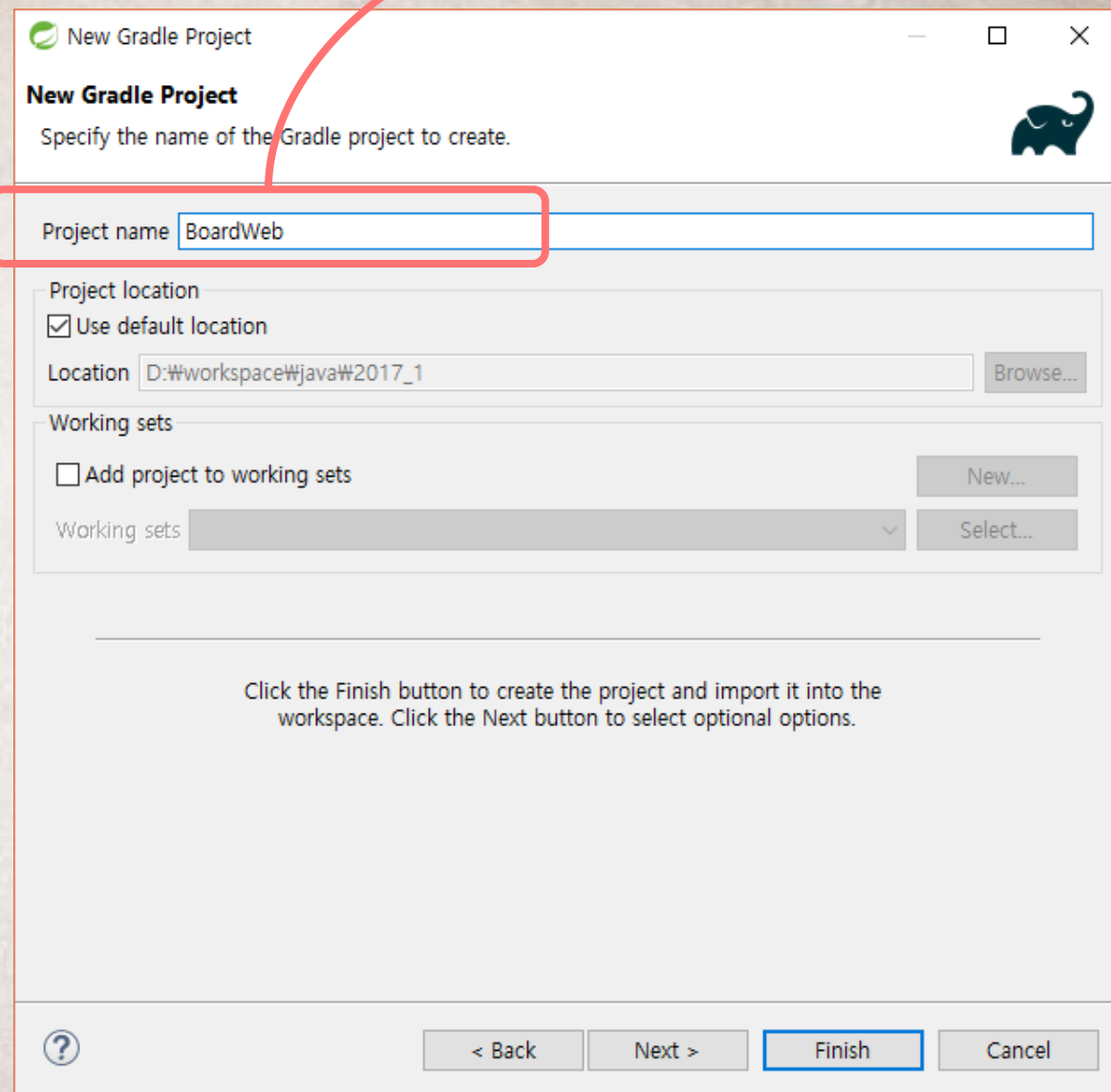
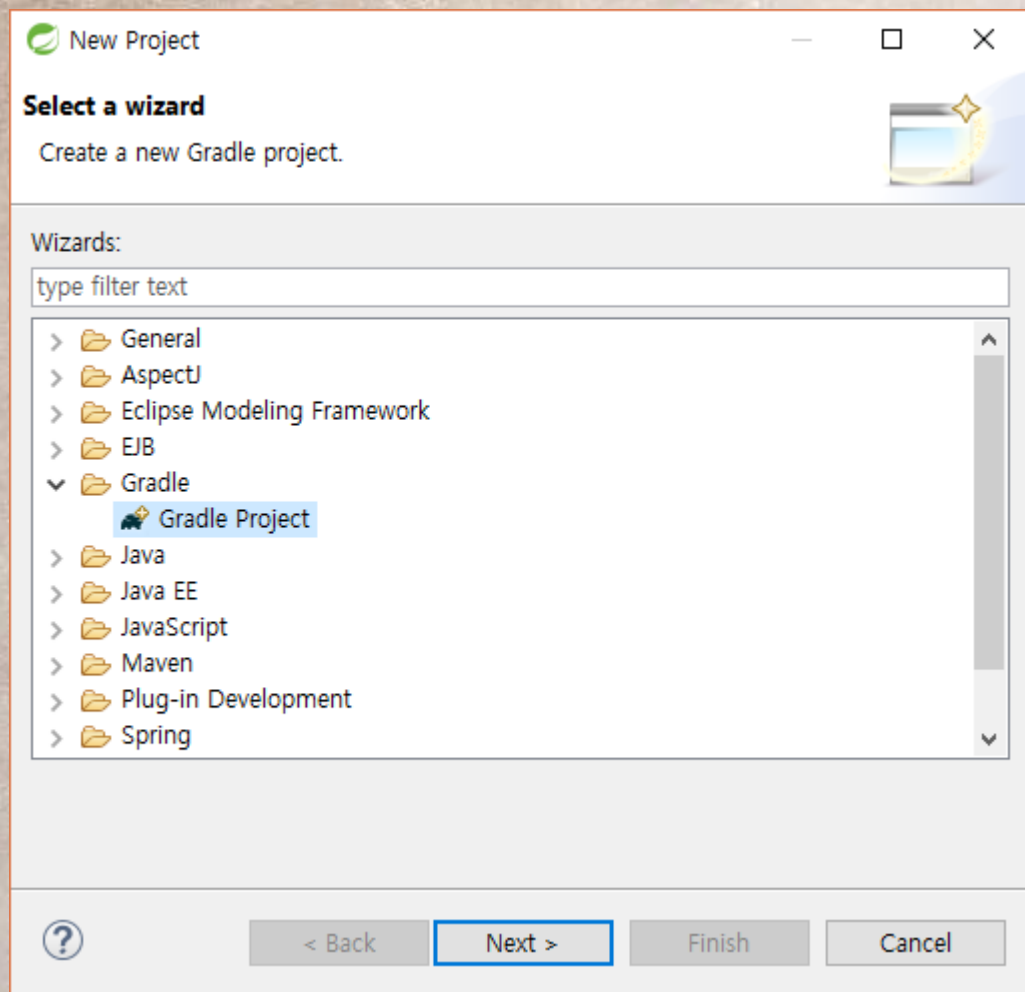
순서	내 용
1	<ul style="list-style-type: none">• Spring IoC를 이용한 비즈니스 컴포넌트 만들기
2	<ul style="list-style-type: none">• Spring AOP(Aspect Oriented Programming)를 이용한 공통 서비스 만들기• Spring DAO(Data Access Object)를 이용한 데이터베이스 연동 및 트랜잭션 처리
3	<ul style="list-style-type: none">• Spring MVC를 이용한 MVC 아키텍처 적용하기
4	<ul style="list-style-type: none">• Spring MVC의 부가 기능 사용하기(파일 업로드, 다국어, 예외 처리 등)
5	<ul style="list-style-type: none">• Spring과 MyBatis 연동하기• Spring과 JPA 연동하기

실습 프로젝트 생성

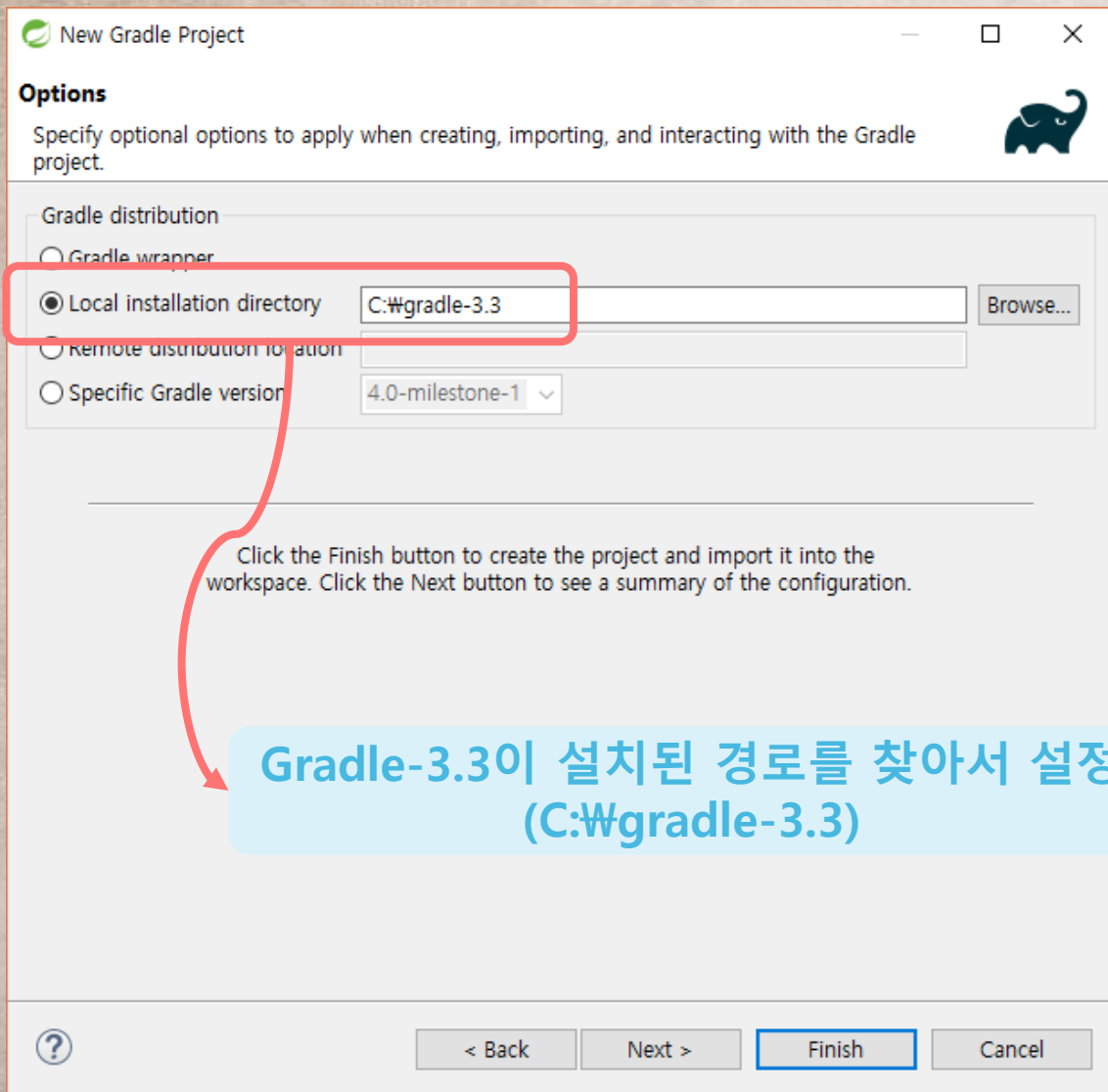
실습 프로젝트 생성 - Gradle Project



• Gradle Project 생성



• Options - Gradle distribution



New Gradle Project

Options

Specify optional options to apply when creating, importing, and interacting with the Gradle project.

Gradle distribution

☐ Gradle wrapper

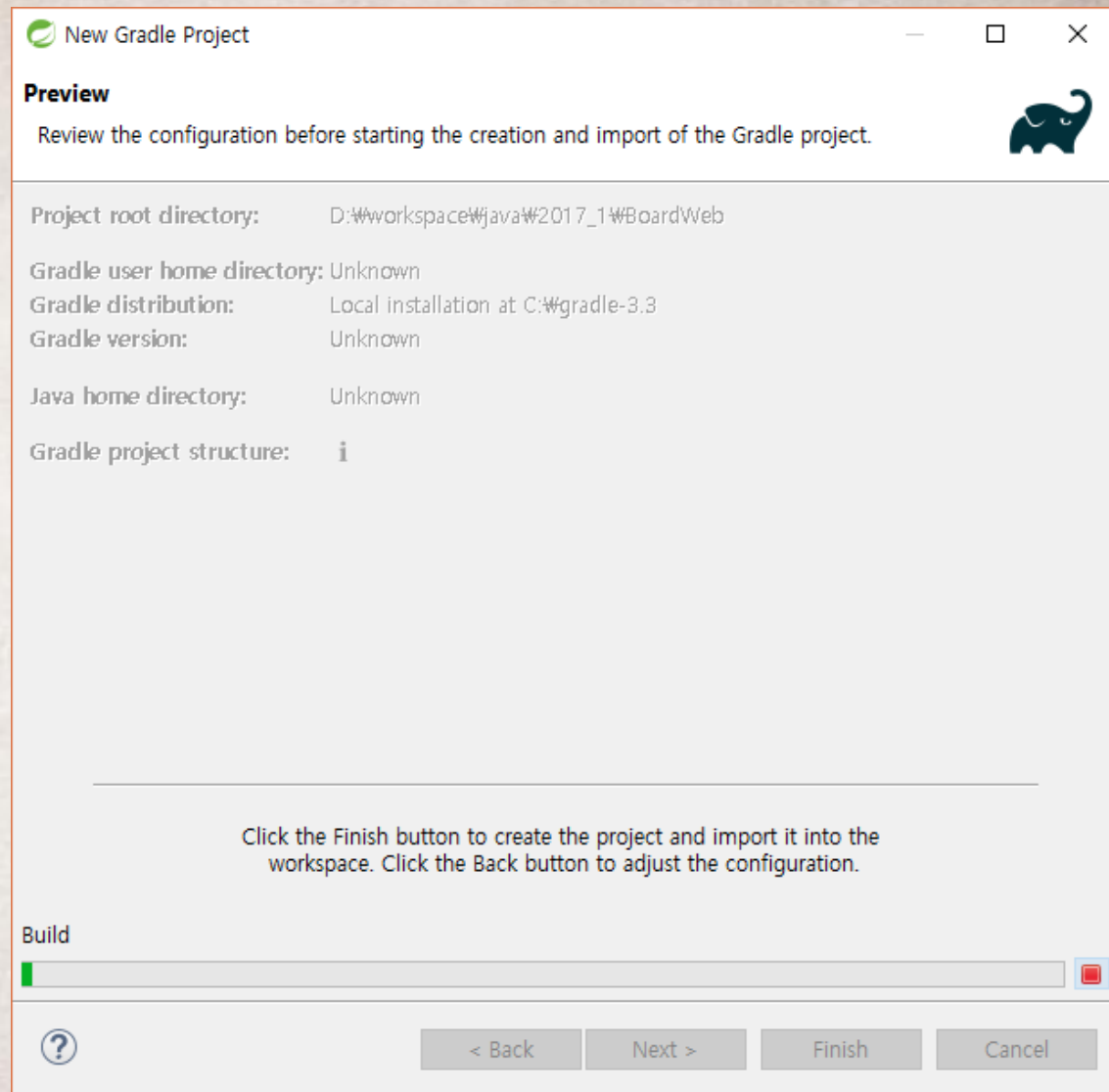
☒ Local installation directory

☐ Remote distribution location

☐ Specific Gradle version

Click the Finish button to create the project and import it into the workspace. Click the Next button to see a summary of the configuration.

Gradle-3.3이 설치된 경로를 찾아서 설정 (C:\gradle-3.3)



New Gradle Project

Preview

Review the configuration before starting the creation and import of the Gradle project.


Project root directory: D:\workspace\java\2017_1\BoardWeb

Gradle user home directory: Unknown

Gradle distribution: Local installation at C:\gradle-3.3

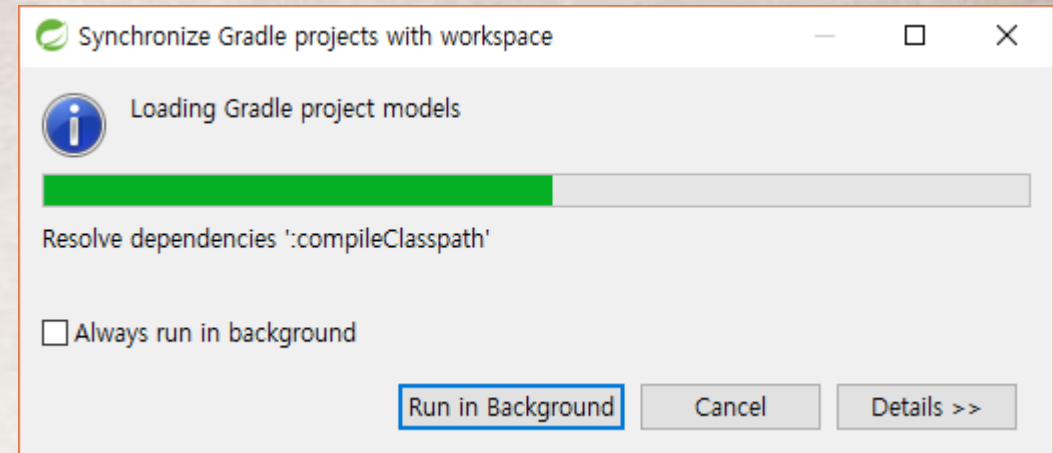
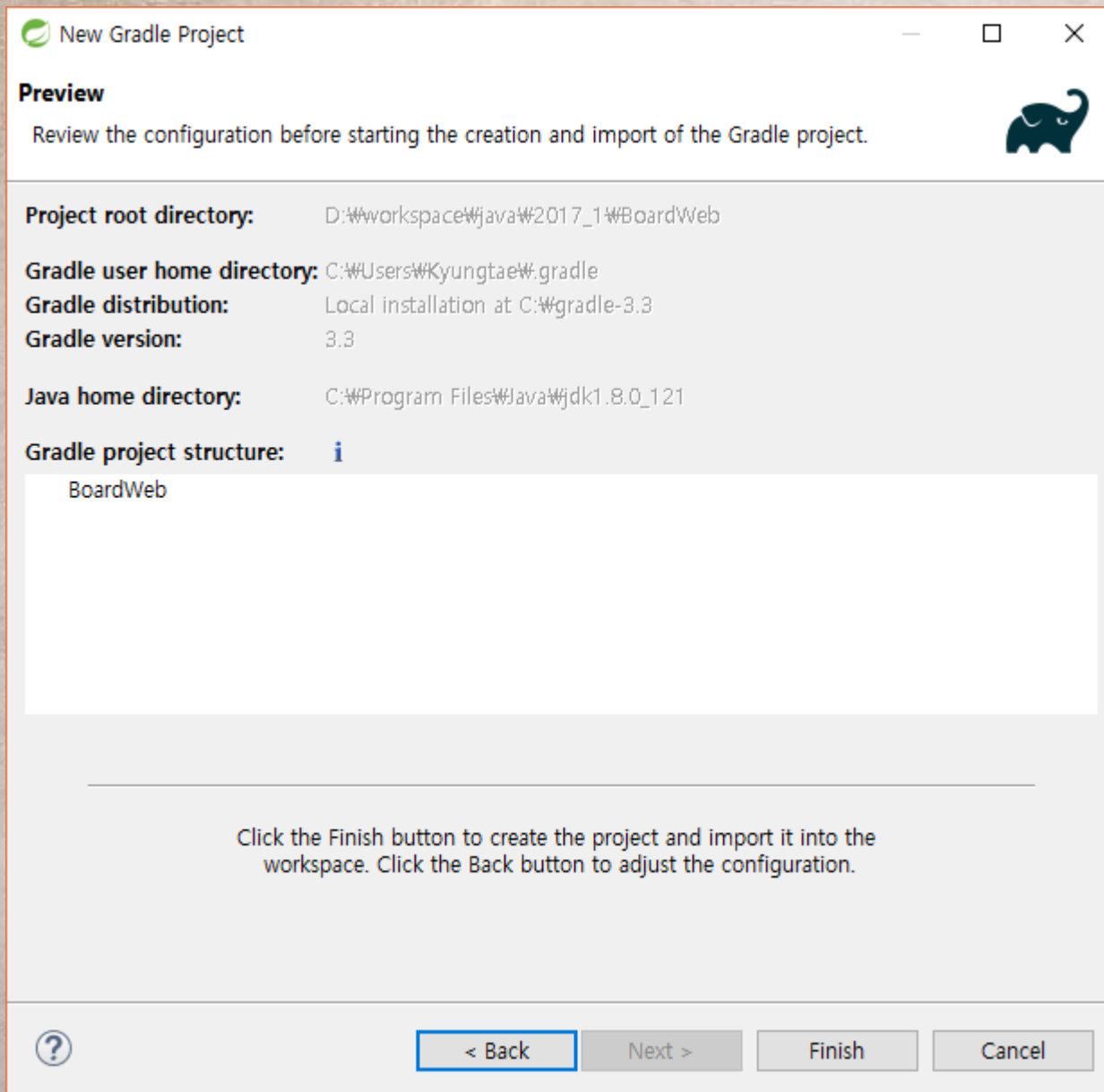
Gradle version: Unknown

Java home directory: Unknown

Gradle project structure: 

Click the Finish button to create the project and import it into the workspace. Click the Back button to adjust the configuration.

Build



• Gradle Project 생성 중...

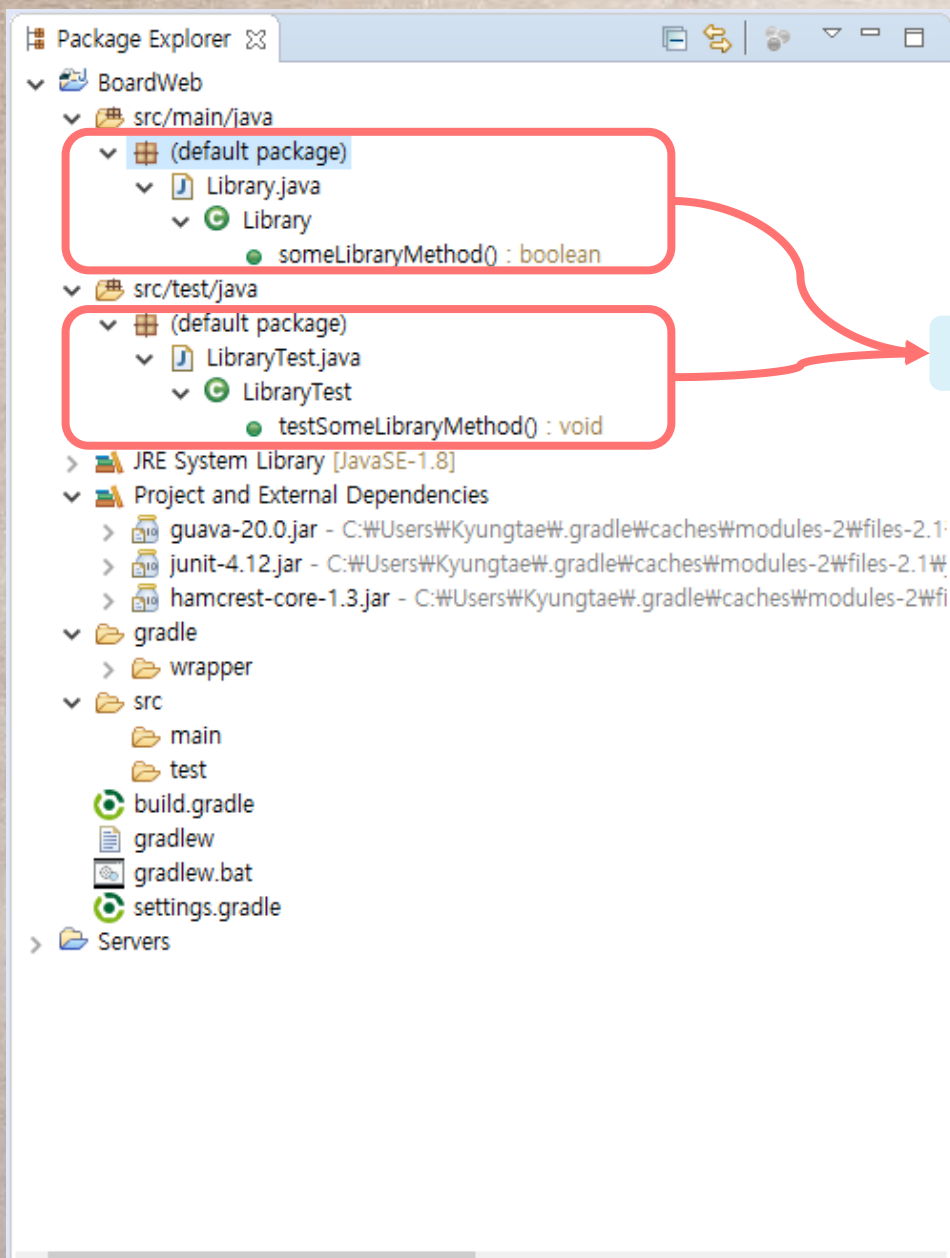
The screenshot shows the Spring Tool Suite interface during a Gradle project synchronization. A dialog box titled "Synchronize Gradle projects with workspace" is open, displaying a progress bar and the message "Loading Gradle project models". Below the progress bar, it shows a download link: "Download https://jcenter.bintray.com/com/...e/guava/guava/20.0/guava-20.0-sources.jar". There is an unchecked checkbox labeled "Always run in background" and three buttons: "Run in Background", "Cancel", and "Details >>".

In the bottom right corner, the "Console" view is visible, showing the following output:

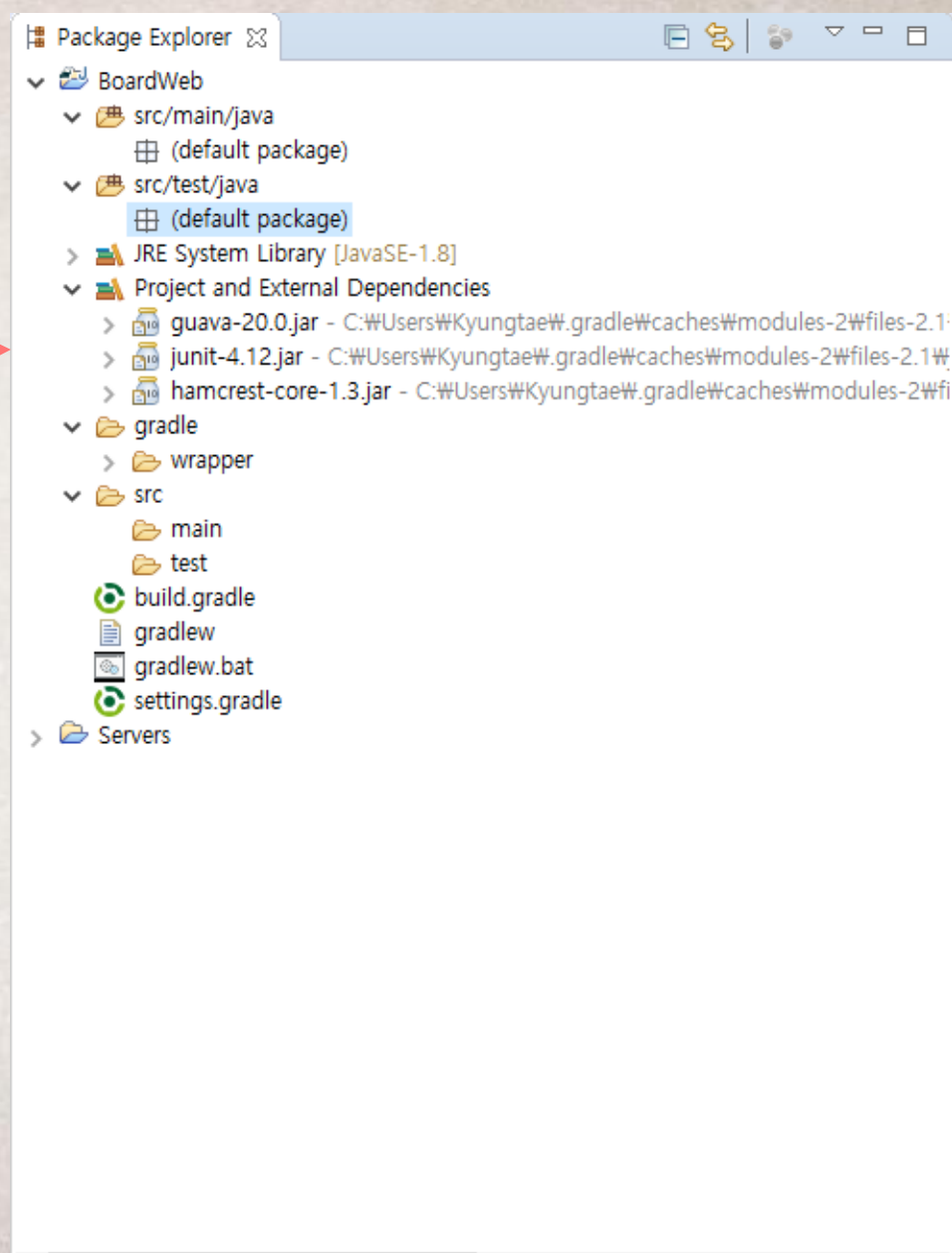
```
[Gradle Model Retrieval]  
CONFIGURE SUCCESSFUL  
  
Total time: 0.609 secs  
Download https://jcenter.bintray.com/com/google/guava/guava/20.0/guava-20.0.pom  
Download https://jcenter.bintray.com/com/google/guava/guava-parent/20.0/guava-parent-20.0.pom  
Download https://jcenter.bintray.com/org/sonatype/oss/oss-parent/7/oss-parent-7.pom  
Download https://jcenter.bintray.com/com/google/guava/guava/20.0/guava-20.0.jar  
Download https://jcenter.bintray.com/com/google/guava/guava/20.0/guava-20.0-sources.jar
```

The status bar at the bottom right indicates "Synchronize Gradle proj...pace: (52%)".

• 프로젝트 생성 완료와 초기 설정



삭제



다형성 - Interface를 이용한 결합도 낮추기

TV.Java 인터페이스

The screenshot illustrates the process of creating a new Java interface in the Eclipse IDE. The Package Explorer on the left shows the project structure with 'BoardWeb' and its sub-packages 'src/main/java' and 'src/test/java'. The 'New' context menu is open, and the 'Interface' option is selected. The 'New Java Interface' dialog box is displayed, showing the following configuration:

- Source folder: BoardWeb/src/main/java
- Package: polymorphism
- Enclosing type: (unchecked)
- Name: TV
- Modifiers: public (selected), package, private, protected
- Extended interfaces: (empty list)
- Do you want to add comments? (Generate comments unchecked)

Red arrows point from the 'polymorphism' package in the Package Explorer and the 'Interface' option in the 'New' menu to the 'Package' and 'Name' fields in the dialog box, respectively. A blue callout box at the bottom center contains the text:

Package: polymorphism
Name: TV

TV.java 인터페이스

TV.java

```
1 package polymorphism;
2
3 public interface TV {
4     public void powerOn();
5     public void powerOff();
6     public void volumeUp();
7     public void volumeDown();
8 }
9
10
```

SamungTV.Java 클래스

The screenshot shows an IDE interface with the Package Explorer on the left, a code editor in the center, and a 'New Java Class' dialog on the right. The Package Explorer shows a project named 'BoardWeb' with a package 'polymorphism' containing a file 'TV.java'. The code editor shows the start of the 'TV.java' file with the package declaration 'package polymorphism;'. The 'New Java Class' dialog is open, showing the 'Source folder' as 'BoardWeb/src/main/java', the 'Package' as 'polymorphism', and the 'Name' as 'SamsungTV'. The 'Class' option is selected in the 'New' menu. A red box highlights the 'Package' field in the dialog, and another red box highlights the 'Name' field. A red arrow points from a text box at the bottom to the 'Package' field.

Package Explorer

- BoardWeb
 - src/main/java
 - polymorphism
 - TV.java

TV.java

```
1 package polymorphism;  
2
```

New Java Class

Java Class

Create a new Java class.

Source folder: BoardWeb/src/main/java Browse...

Package: polymorphism Browse...

Enclosing type: Browse...

Name: SamsungTV

Modifiers: ☒ public ☐ package ☐ private ☐ protected
☐ abstract ☐ final ☐ static

Superclass: java.lang.Object Browse...

Interfaces: Add... Remove

Which method stubs would you like to create?

- ☐ public static void main(String[] args)
- ☐ Constructors from superclass
- ☒ Inherited abstract methods

Do you want to add comments? (Configure templates and default value [here](#))

- ☐ Generate comments

Finish Cancel

Package: polymorphism
Name: TV

SamungTV.Java 클래스

검색어: TV

New Java Class

Create a new Java class.

Source folder: BoardWeb/src/main/java Browse...

Package: polymorphism Browse...

☐ Enclosing type: Browse...

Name: SamsungTV

Modifiers: ☒ public ☐ package ☐ private ☐ protected
☐ abstract ☐ final ☐ static

Superclass: java.lang.Object Browse...

Interfaces: Add... Remove

Which method stubs would you like to create?

☐ public static void main(String[] args)

☐ Constructors from superclass

☒ Inherited abstract methods

Do you want to add comments? (Configure templates and default value [here](#))

☐ Generate comments

Finish Cancel

Implemented Interfaces Selection

Choose interfaces: TV

Matching items:

- TV - polymorphism
- Workspace matches -----
- TabableView
- TypeVariable - java.lang.reflect - [jre1.8.0_121]
- TypeVariable - javax.lang.model.type - [jre1.8.0_121]
- TypeVisitor

polymorphism - BoardWeb/src/main/java

Add OK Cancel

SamungTV.Java 클래스

New Java Class

Java Class

Create a new Java class.

Source folder: BoardWeb/src/main/java Browse...

Package: polymorphism Browse...

☐ Enclosing type: Browse...

Name: SamsungTV

Modifiers: ☒ public ☐ package ☐ private ☐ protected
☐ abstract ☐ final ☐ static

Superclass: java.lang.Object Browse...

Interfaces: polymorphism.TV Add...
Remove

Which method stubs would you like to create?

☐ public static void main(String[] args)
☐ Constructors from superclass
☒ Inherited abstract methods

Do you want to add comments? (Configure templates and default value [here](#))
☐ Generate comments

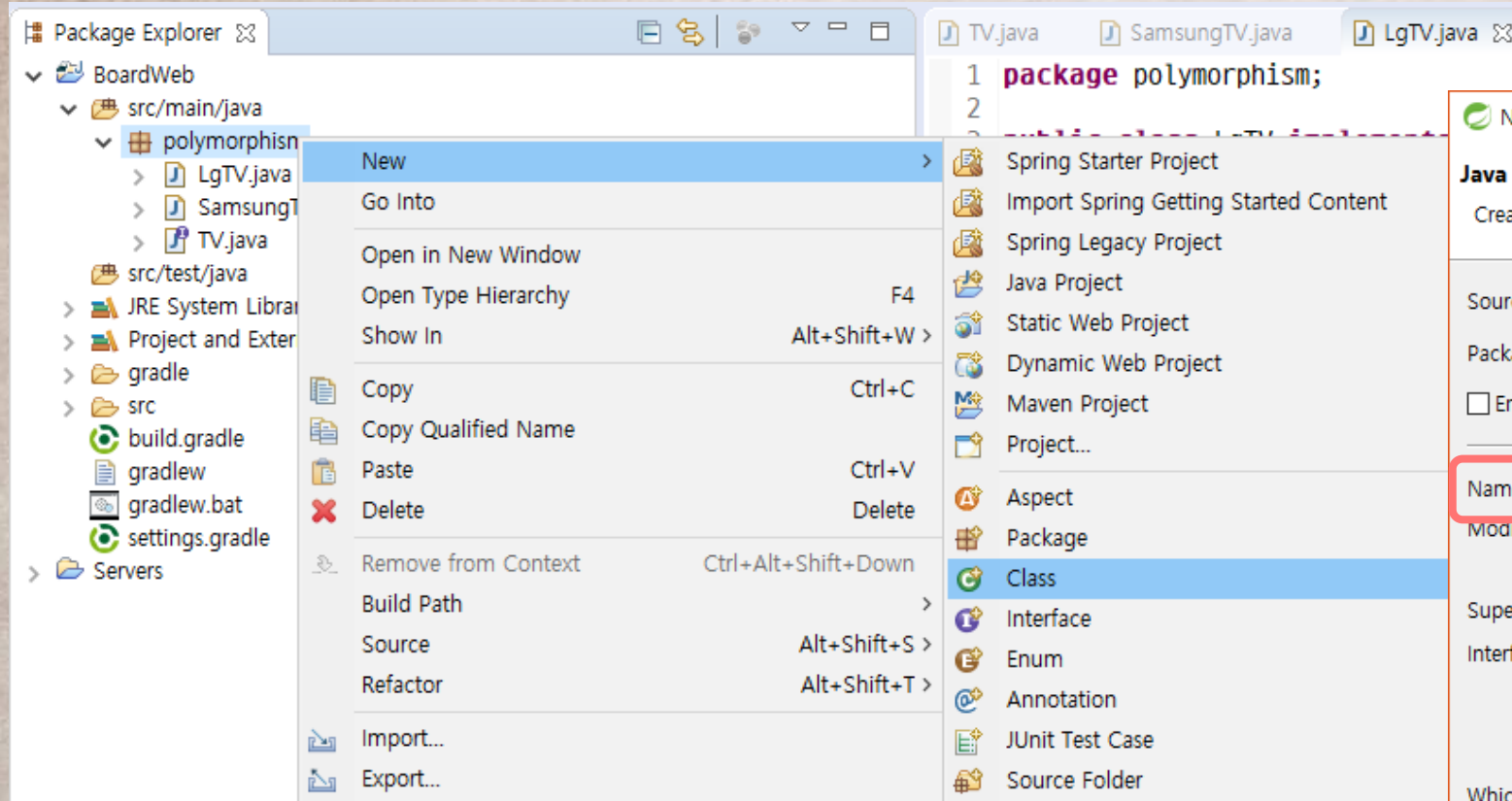
? Finish Cancel

```
TV.java SamsungTV.java
1 package polymorphism;
2
3 public class SamsungTV implements TV {
4
5     @Override
6     public void powerOn() {
7         // TODO Auto-generated method stub
8     }
9
10
11     @Override
12     public void powerOff() {
13         // TODO Auto-generated method stub
14     }
15
16
17     @Override
18     public void volumeUp() {
19         // TODO Auto-generated method stub
20     }
21
22
23     @Override
24     public void volumeDown() {
25         // TODO Auto-generated method stub
26     }
27
28
29 }
30
```

SamungTV.java 클래스와 LgTV.java 클래스

```
TVUser.java  build.gradle  SamsungTV.java  ✖
1 package polymorphism;
2
3 public class SamsungTV implements TV {
4
5     @Override
6     public void powerOn() {
7         System.out.println("SamungTV -- 전원을 켜다.");
8     }
9
10
11     @Override
12     public void powerOff() {
13         System.out.println("SamungTV -- 전원을 끈다.");
14     }
15
16
17     @Override
18     public void volumeUp() {
19         System.out.println("SamungTV -- 소리를 올린다.");
20     }
21
22
23     @Override
24     public void volumeDown() {
25         System.out.println("SamungTV -- 소리를 내린다.");
26     }
27
28 }
29
30
```

```
TV.java  SamsungTV.java  LgTV.java  ✖
1 package polymorphism;
2
3 public class LgTV implements TV {
4
5     @Override
6     public void powerOn() {
7         System.out.println("LgTV -- 전원을 켜다.");
8     }
9
10
11     @Override
12     public void powerOff() {
13         System.out.println("LgTV -- 전원을 끈다.");
14     }
15
16
17     @Override
18     public void volumeUp() {
19         System.out.println("LgTV -- 소리를 올린다.");
20     }
21
22
23     @Override
24     public void volumeDown() {
25         System.out.println("LgTV -- 소리를 내린다.");
26     }
27
28 }
29
30
```

Name: BeanFactory

The 'New Java Class' dialog box is shown. The 'Name' field is highlighted with a red box and contains the text 'BeanFactory'. A red arrow points from this field to the 'Name: BeanFactory' text above. The 'Package' field contains 'polymorphism'. The 'Modifiers' section has 'public' selected. The 'Superclass' field contains 'java.lang.Object'. The 'Which method stubs would you like to create?' section has 'Inherited abstract methods' checked. The 'Do you want to add comments?' section has 'Generate comments' unchecked. The 'Finish' button is highlighted.

New Java Class

Create a new Java class.

Source folder: BoardWeb/src/main/java Browse...

Package: polymorphism Browse...

☐ Enclosing type: Browse...

Name: **BeanFactory**

Modifiers: ☒ public ☐ package ☐ private ☐ protected
☐ abstract ☐ final ☐ static

Superclass: java.lang.Object Browse...

Interfaces: Add... Remove

Which method stubs would you like to create?
☐ public static void main(String[] args)
☐ Constructors from superclass
☒ Inherited abstract methods

Do you want to add comments? (Configure templates and default value [here](#))
☐ Generate comments

Finish Cancel

TVUser.java

*BeanFactory.java

```
1 package polymorphism;
2
3 public class BeanFactory {
4
5     public Object getBean(String beanName){
6         if (beanName.equals("samsung")){
7             return new SamsungTV();
8         }else if (beanName.equals("lg")){
9             return new LgTV();
10        }
11        return null;
12    }
13
14 }
15
16 |
```

TVUser.Java 클래스 - main 클래스

The screenshot shows an IDE interface with the Package Explorer on the left, a code editor in the center, and a 'New Java Class' dialog box on the right.

Package Explorer: The project structure shows a package named 'polymorphism' under 'src/main/java'. A context menu is open over this package, with 'New' selected, leading to a submenu where 'Class' is highlighted.

Code Editor: The file 'TV.java' is open, showing the package declaration: `package polymorphism;`

New Java Class Dialog: The dialog is titled 'New Java Class' and contains the following fields and options:

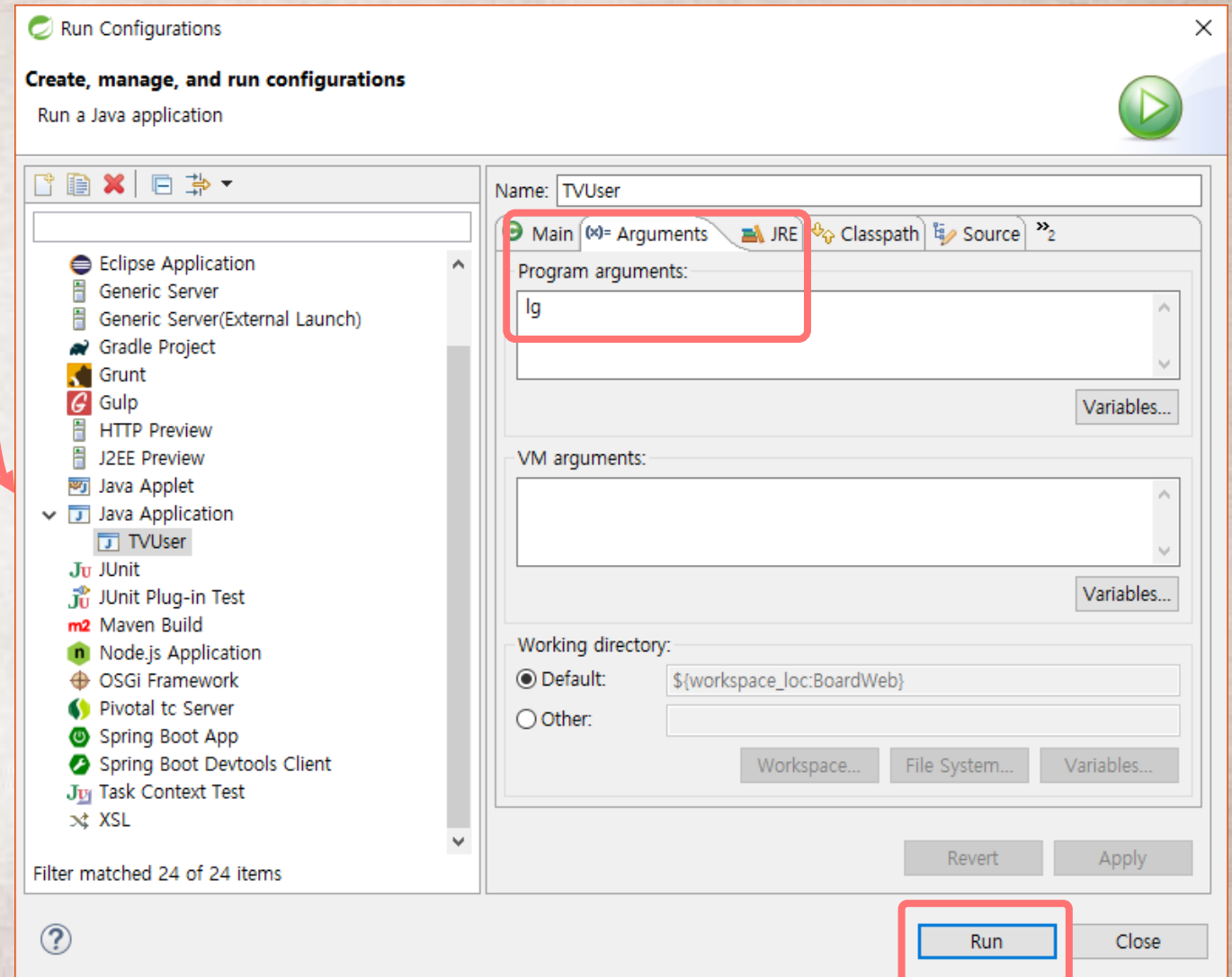
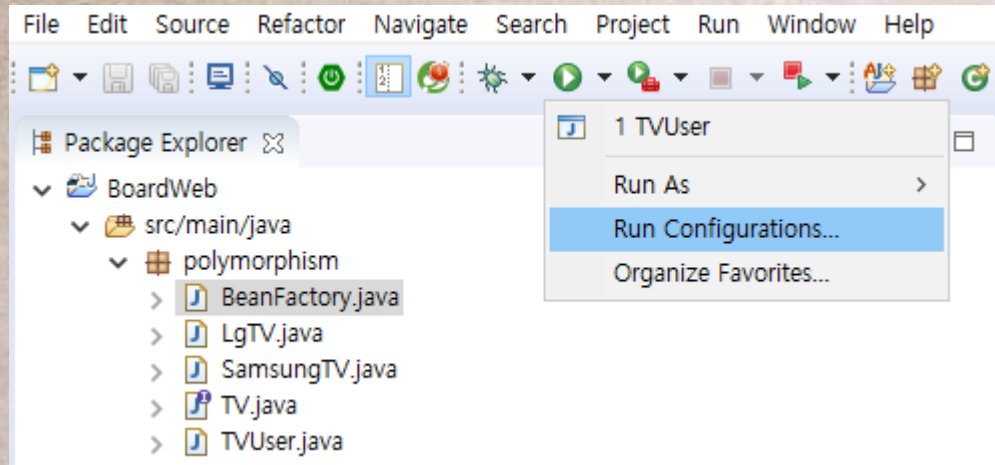
- Source folder:** BoardWeb/src/main/java
- Package:** polymorphism
- Enclosing type:** (unchecked)
- Name:** TVUser (highlighted with a red box and a callout bubble)
- Modifiers:** ☒ public, ☐ package, ☐ private, ☐ protected; ☐ abstract, ☐ final, ☐ static
- Superclass:** java.lang.Object
- Interfaces:** (empty list)
- Which method stubs would you like to create?**
 - ☐ public static void main(String[] args)
 - ☐ Constructors from superclass
 - ☒ Inherited abstract methods
- Do you want to add comments?** (Configure templates and default value [here](#))
 - ☐ Generate comments

Buttons at the bottom: Finish, Cancel.

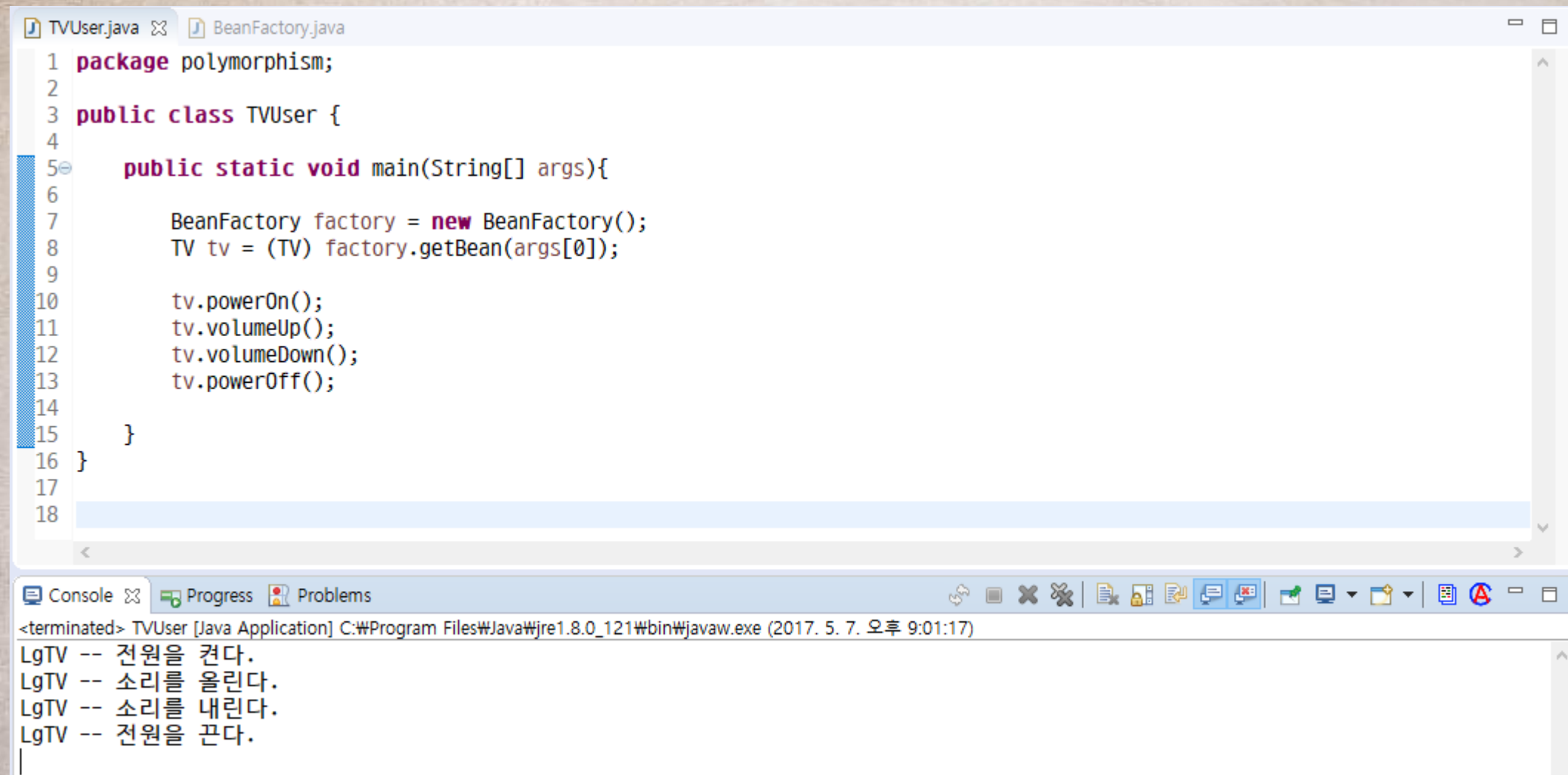
TVUser.java BeanFactory.java

```
1 package polymorphism;
2
3 public class TVUser {
4
5     public static void main(String[] args){
6
7         BeanFactory factory = new BeanFactory();
8         TV tv = (TV) factory.getBean(args[0]);
9
10        tv.powerOn();
11        tv.volumeUp();
12        tv.volumeDown();
13        tv.powerOff();
14
15    }
16 }
17
18 |
```

Run...



실행 결과



The screenshot shows an IDE with two tabs: TVUser.java and BeanFactory.java. The TVUser.java file contains the following code:

```
1 package polymorphism;
2
3 public class TVUser {
4
5     public static void main(String[] args){
6
7         BeanFactory factory = new BeanFactory();
8         TV tv = (TV) factory.getBean(args[0]);
9
10        tv.powerOn();
11        tv.volumeUp();
12        tv.volumeDown();
13        tv.powerOff();
14
15    }
16 }
17
18
```

The console output shows the execution of the TVUser application, displaying the following messages:

```
<terminated> TVUser [Java Application] C:\Program Files\Java\jre1.8.0_121\bin\javaw.exe (2017. 5. 7. 오후 9:01:17)
LgTV -- 전원을 켜다.
LgTV -- 소리를 올린다.
LgTV -- 소리를 내린다.
LgTV -- 전원을 끈다.
```


4. 스프링 컨테이너 및 설정 파일

IoC 컨테이너는 각 컨테이너에서 관리할 객체들을 위한 별도의 설정 파일이 있다.

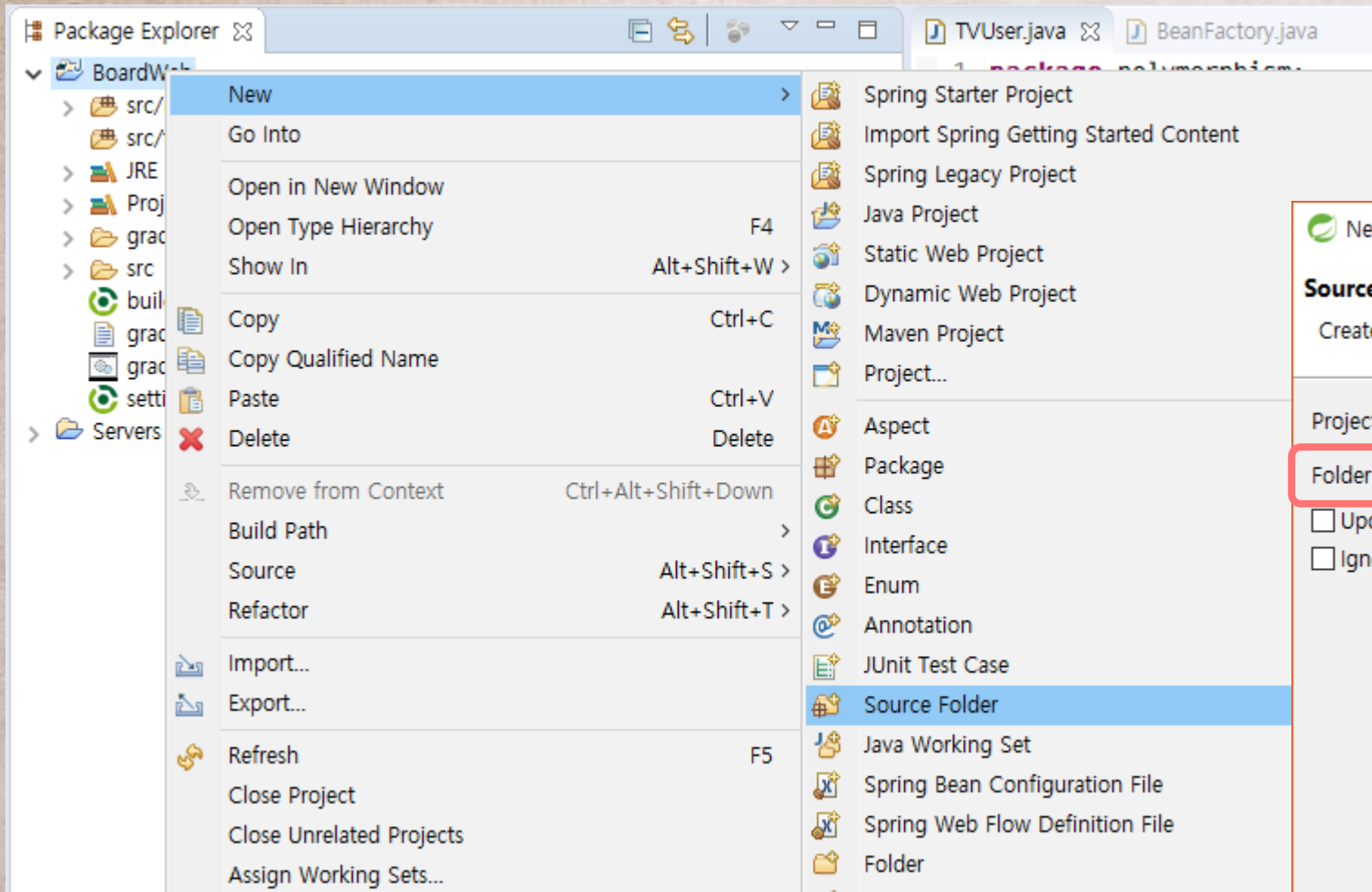
Servlet 컨테이너는 `web.xml`, EJB 컨테이너는 `ejb-jar.xml` 파일에 해당 컨테이너가 생성하고 관리할 클래스 등록

스프링도 자신이 관리할 클래스들이 등록된 XML 설정 파일이 필요

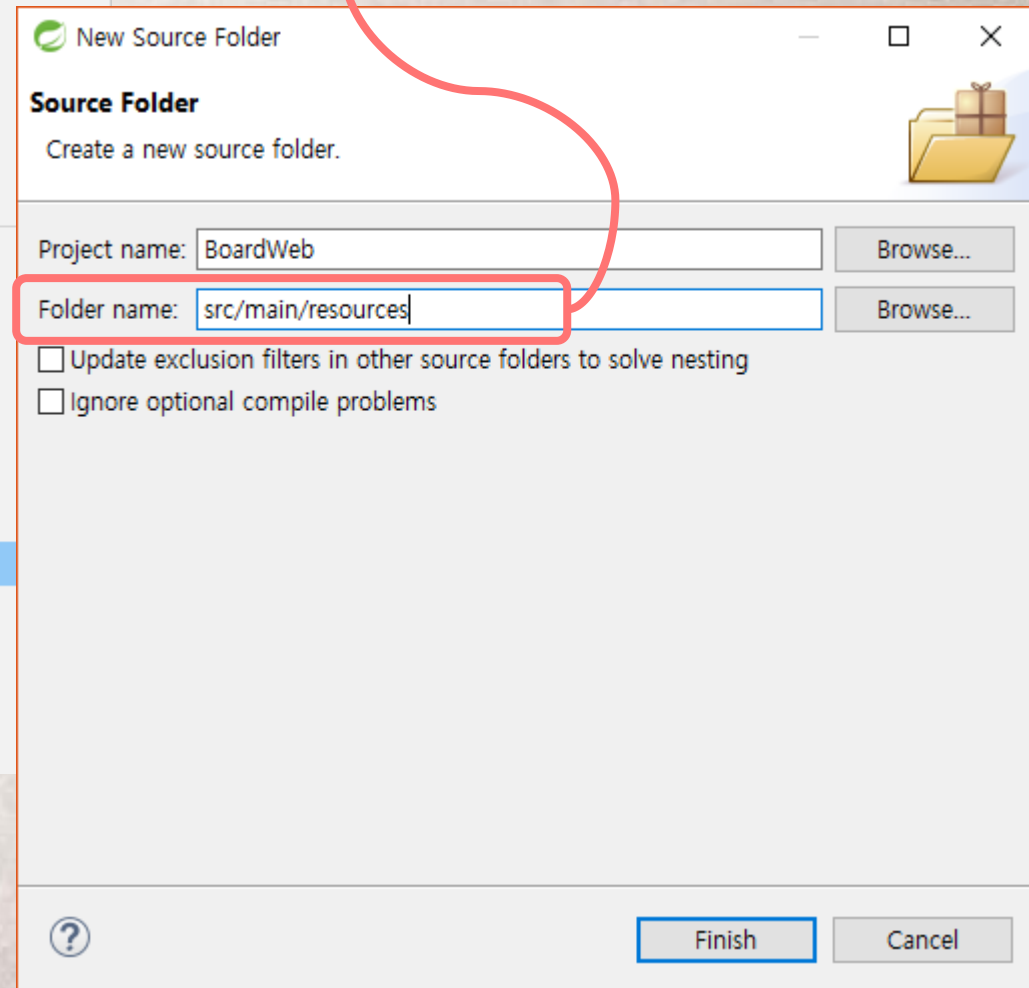
스프링 컨테이너와 설정 파일

- IoC 컨테이너는 각 컨테이너에서 관리할 객체들을 위한 별도의 설정 파일이 있다.
- Servlet 컨테이너는 `web.xml`, EJB 컨테이너는 `ejb-jar.xml` 파일에 해당 컨테이너가 생성하고 관리할 클래스 등록
- 스프링도 자신이 관리할 클래스들이 등록된 XML 설정 파일이 필요

4.1 스프링 IoC - 스프링 설정 파일 생성



Folder Name:
src/main/resources



스프링 설정 파일 생성

Package Explorer

- BoardWeb
 - src/main/java
 - src/test/java
 - src/main/resources
 - JRE System Li
 - Project and E
 - gradle
 - src
 - build.gradle
 - gradlew
 - gradlew.bat
 - settings.gradle
- Servers

New

- Open in New Window
- Open Type Hierarchy F4
- Show In Alt+Shift+W
- Copy Ctrl+C
- Copy Qualified Name
- Paste Ctrl+V
- Delete Delete
- Remove from Context Ctrl+Alt+Shift+Down
- Build Path
- Source Alt+Shift+S
- Refactor Alt+Shift+T
- Import...
- Export...
- Refresh F5
- Assign Working Sets...
- Run As
- Debug As
- Profile As
- Validate

TVUser.java BeanFactory.java

```
1 package polymorphism;  
2  
3 public class TVUser {  
4
```

Create a new Spring Bean Definition file

New Spring Bean Definition file

Select the location and give a name for the Spring Bean Definition file

Enter or select the parent folder:

BoardWeb/src/main/resources

BoardWeb

- .gradle
- .settings
- bin
- gradle
- src
 - main
 - java
 - resources
 - test
- RemoteSystemsTempFiles
- Servers

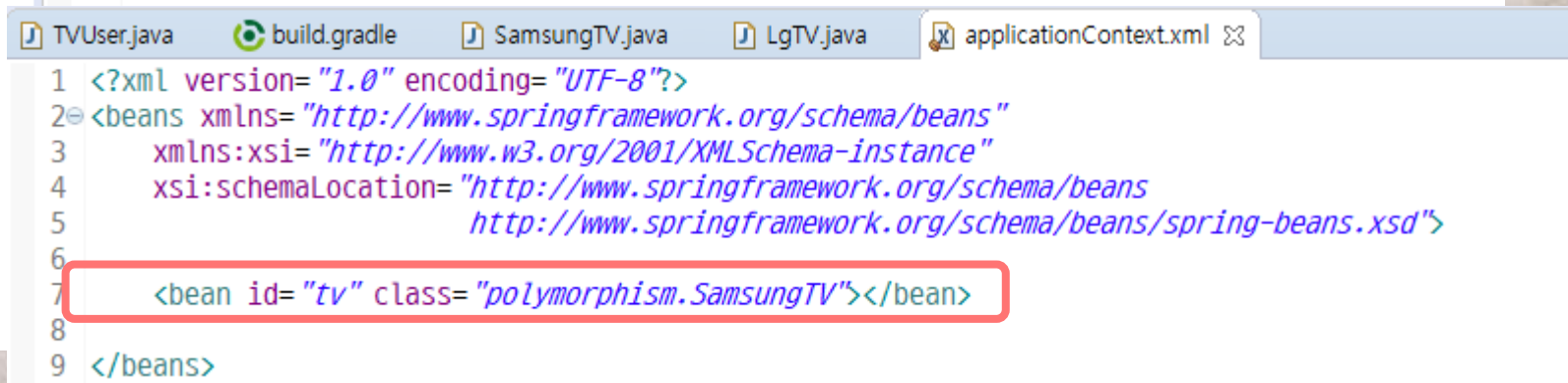
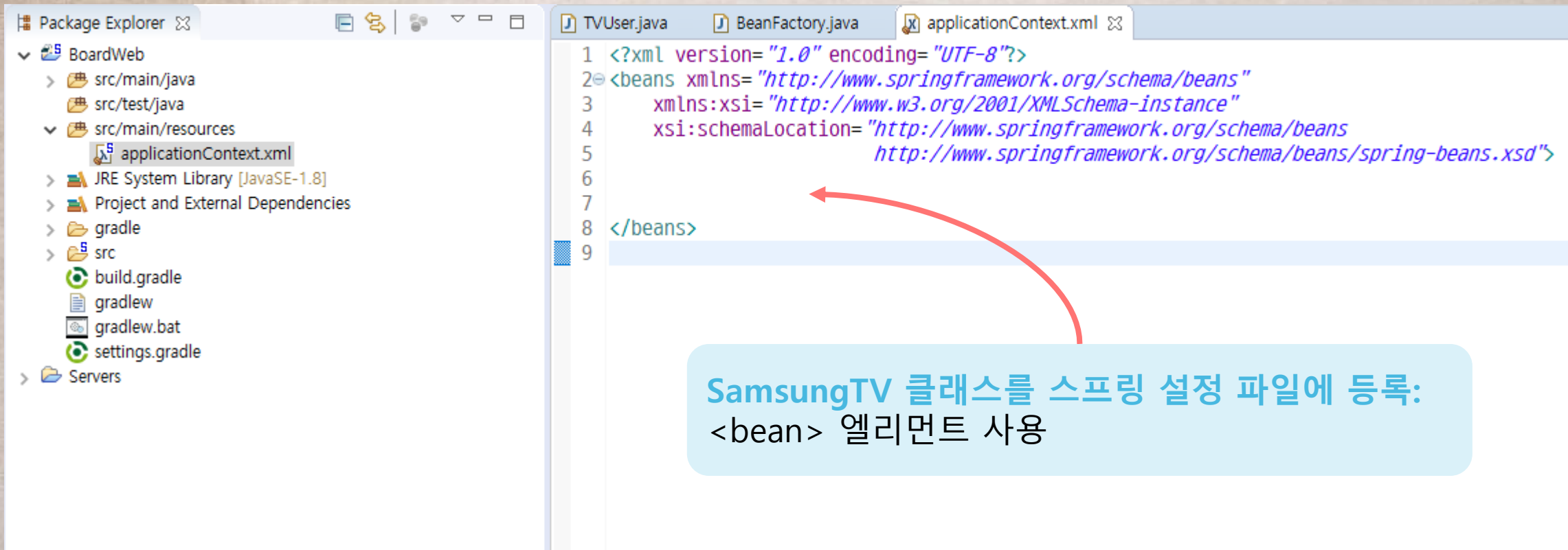
File name: applicationContext

Advanced >>

☒ Add Spring project nature if required

Finish

스프링 설정 파일 생성-applicationContext.xml



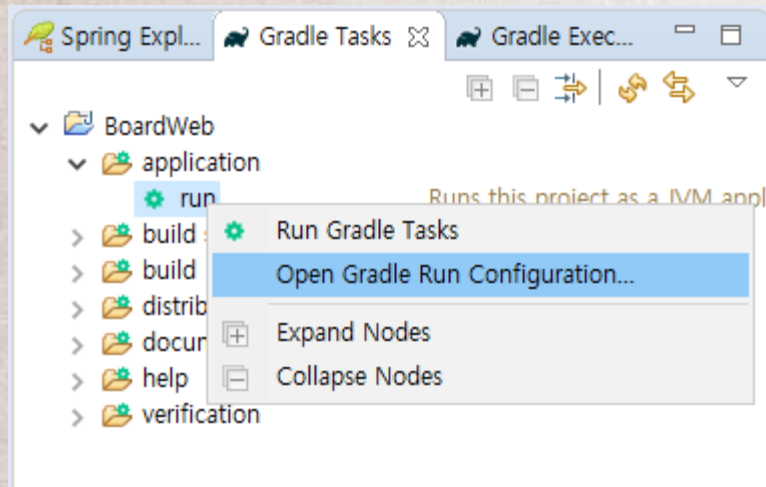
TVUser.Java 클래스수 수정

```
TVUser.java build.gradle SamsungTV.java LgTV.java applicationContext.xml
1 package polymorphism;
2
3 import org.springframework.context.support.AbstractApplicationContext;
4 import org.springframework.context.support.GenericXmlApplicationContext;
5
6 public class TVUser {
7
8     public static void main(String[] args){
9
10         // Spring 컨테이너를 구동
11         AbstractApplicationContext factory = new GenericXmlApplicationContext("applicationContext.xml");
12
13         //BeanFactory factory = new BeanFactory();
14         TV tv = (TV) factory.getBean("tv");
15
16         tv.powerOn();
17         tv.volumeUp();
18         tv.volumeDown();
19         tv.powerOff();
20
21         // Spring 컨테이너를 종료
22         factory.close();
23     }
24 }
25
26
```

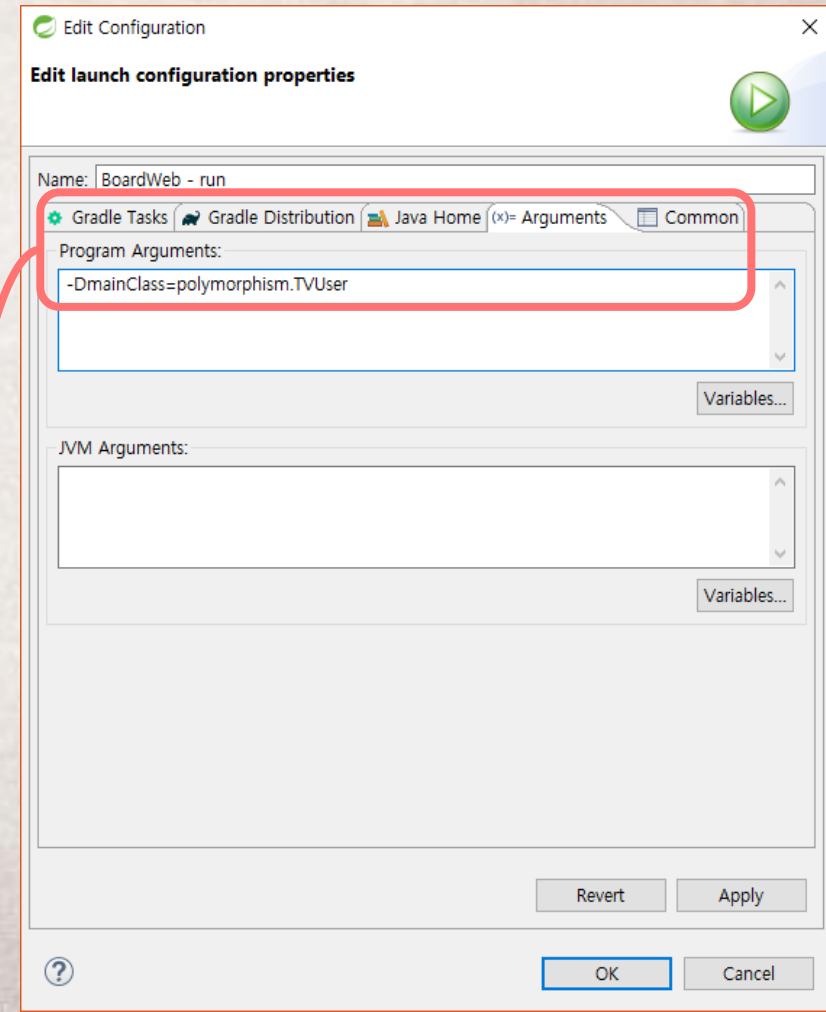

<bean> 엘리먼트

```
<bean id="tv" class="polymorphism.SamsungTV"></bean>
```

- <bean> 엘리먼트에서 class 속성이 가장 중요
- Package 경로가 포함된 전체 클래스 경로를 지정



-DmainClass=kr.ac.inje.comsi.pkt.Application



build.gradle

```
TVUser.java *build.gradle SamsungTV.java
1 /*
2  * This build file was generated by the Gradle 'init' task.
3  *
4  * This generated file contains a sample Java project to get you started.
5  * For more details take a look at the Java Quickstart chapter in the Gradle
6  * user guide available at https://docs.gradle.org/3.3/userguide/tutorial_java_projects.html
7  */
8
9 // Apply the java plugin to add support for Java
10 apply plugin: 'application'
11
12 mainClassName = System.getProperty("mainClass");
13
14 compileJava.options.encoding = 'UTF-8'
15
16 // In this section you declare where to find the dependencies of your project
17 repositories {
18     mavenCentral()
19 }
20
21 dependencies {
22     compile 'org.springframework:spring-context:4.3.8.RELEASE'
23 }
24
25
26
```

gradle 의존 Library를 자동 추가

The screenshot shows an IDE interface with the `build.gradle` file open. The file content is as follows:

```
1 // Apply the java plugin to add support for Java
2 apply plugin: 'application'
3
4 mainClassName = System.getProperty("mainClass");
5
6 // In this section you declare where to find the dependencies of your project
7 repositories {
8     mavenCentral()
9 }
10
11 dependencies {
12     compile 'org.springframework:spring-context:4.3.8.RELEASE'
13 }
14
```

A context menu is open over the `build.gradle` file, showing various actions. The `Gradle` option is expanded, and `Refresh Gradle Project` is highlighted. A red arrow points from the Korean text "gradle 의존 Library를 자동 추가" to the `Refresh Gradle Project` option.

Below the menu, the console output shows the following messages:

```
gTV -- Power ON
gTV -- Volume UP
gTV -- Volume DOWN
```

Gradle 실행

The screenshot shows an IDE window with a console on the left and a Spring Explorer on the right. The console displays the output of a Gradle 'run' task for a project named 'BoardWeb'. The output includes system information, task execution steps, and Spring Framework logs. The Spring Explorer on the right shows the project structure with the 'run' task highlighted under the 'application' folder.

BoardWeb - run [Gradle Project] run in D:\workspace\java\2017_1\BoardWeb (2017. 5. 8 오후 11:37:42)

Working Directory: D:\workspace\java\2017_1\BoardWeb
Gradle User Home: C:\Users\Kyungtae\.gradle
Gradle Distribution: Local installation at C:\gradle-3.3
Gradle Version: 3.3
Java Home: C:\Program Files\Java\jdk1.8.0_121
JVM Arguments: None
Program Arguments: -DmainClass=polymorphism.TVUser
Gradle Tasks: run

:compileJava
:processResources UP-TO-DATE
:classes
:run5월 08, 2017 11:37:43 오후 org.springframework.beans.factory.xml.XmlBeanDefinitionReader loadBeanDefinitions
정보: Loading XML bean definitions from class path resource [applicationContext.xml]
5월 08, 2017 11:37:43 오후 org.springframework.context.support.GenericXmlApplicationContext prepareRefresh
정보: Refreshing org.springframework.context.support.GenericXmlApplicationContext@378fd1ac: startup date [Mon May 08 23:37:43 KST 2017]
SamsungTV -- 전원 켜기
SamsungTV -- 소리 올리기
SamsungTV -- 소리 내리기
SamsungTV -- 전원 끄기
5월 08, 2017 11:37:43 오후 org.springframework.context.support.GenericXmlApplicationContext doClose
정보: Closing org.springframework.context.support.GenericXmlApplicationContext@378fd1ac: startup date [Mon May 08 23:37:43 KST 2017]
BUILD SUCCESSFUL
Total time: 0.931 secs

Spring Explorer

- BoardWeb
 - application
 - run
 - build setup
 - build
 - distribution
 - documentation
 - help
 - verification

어떤 과정일까?

applicationContext.xml

```
<beans>  
<bean id="tv" class=polymorphism.SamsungTV/>  
</beans>
```



③ `getBean(" tv ")`

① `Container(ApplicationContext)`

②

SamsungTV

④ `return`

- ① TVUser 클라이언트가 스프링 설정 파일을 로딩하여 컨테이너 구동
- ② 스프링 설정 파일에 `<bean>`으로 등록된 SamsungTV 객체 생성
- ③ `getBean()` 메소드로 이름이 ' tv ' 인 객체를 요청(Lookup)
- ④ SamsungTV 객체 반환

스프링 컨테이너의 종류

■ BeanFactory

- ① 스프링 설정 파일에 등록된 <bean> 객체 생성하고 관리하는 기본 기능
- ② 클라이언트의 요청(Lookup)에 의해서만 <bean> 객체가 생성되는 지연 로딩(Lazy Loading) 방식
 - ➔ 컨테이너가 구동될 때 <bean> 객체를 생성하는 것이 아님.

■ ApplicationContext

- ① BeanFactory의 기본 기능 외에 트랜잭션 관리, 메시지 기반의 다국어 처리
- ② 컨테이너 구동시점에 <bean>으로 등록된 객체를 즉시 로딩(pre-loading)하는 방식
- ③ 웹 어플리케이션 개발도 지원하므로 대부분의 스프링 프로젝트에서 사용

ApplicationContext의 구현 클래스(일부)

구현 클래스	기능
GenericXmlApplicationContext	파일시스템이나 클래스 경로에 있는 XML 설정 파일을 로딩하여 구동하는 컨테이너(TVUser 클라이언트에서 구동한 컨테이너)
XmlWebApplicationContext	웹 기반의 스프링 어플리케이션을 개발할 때 사용하는 컨테이너 (유저가 직접 생성하지 않으며 SpringMVC에서 클라이언트의 요청을 처리하는 DispatcherServlet에서 호출)

구동된 컨테이너로부터 객체 생성확인하기

```
TVUser.java build.gradle SamsungTV.java applicationContext.xml
1 package polymorphism;
2
3 public class SamsungTV implements TV {
4
5     public SamsungTV(){
6         System.out.println("==> SamsungTV 객체 생성");
7     }
8
9     @Override
10    public void powerOn() {
11        System.out.println("SamungTV -- 전원을 켜다.");
12    }
13
14
15    @Override
16    public void powerOff() {
17        System.out.println("SamungTV -- 전원을 끈다.");
18    }
19
20
21    @Override
22    public void volumeUp() {
23        System.out.println("SamungTV -- 소리를 올린다.");
24    }
25
26
27    @Override
28    public void volumeDown() {
29        System.out.println("SamungTV -- 소리를 내린다.");
30    }
31
32
33 }
34
```

기본 생성자 추가

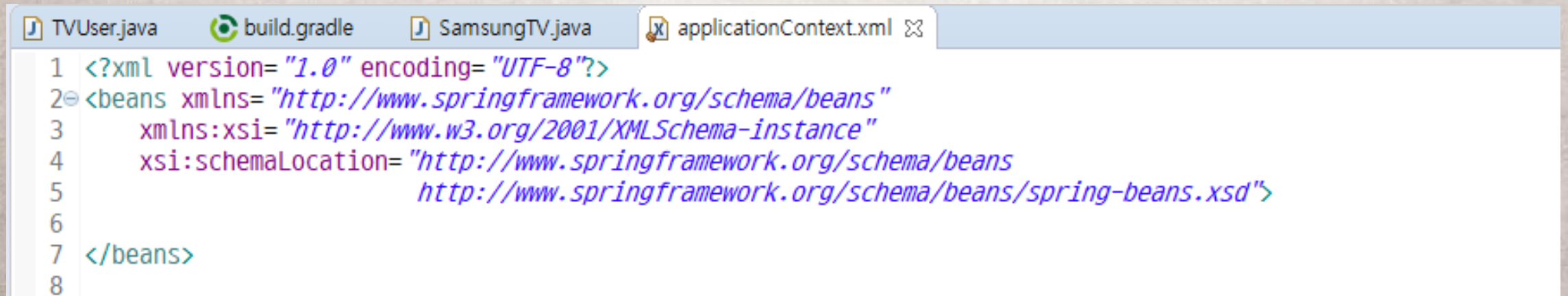
GenericXmlApplicationContext가 생성된 후 객체가 생성됨

```
Console Progress Problems
BoardWeb - run [Gradle Project] run in D:\workspace\java\2017_1\BoardWeb (2017. 5. 9 오전 12:20:45)
:processResources UP-TO-DATE
:classes
:run5월 09, 2017 12:20:46 오전 org.springframework.beans.factory.xml.XmlBeanDefinitionReader
정보: Loading XML bean definitions from class path resource [applicationContext.xml]
5월 09, 2017 12:20:46 오전 org.springframework.context.support.GenericXmlApplicationContext p
정보: Refreshing org.springframework.context.support.GenericXmlApplicationContext@378fd1ac: s
==> SamsungTV 객체 생성
SamungTV -- 전원을 켜다.
SamungTV -- 소리를 올린다.
SamungTV -- 소리를 내린다.
SamungTV -- 전원을 끈다.
5월 09, 2017 12:20:46 오전 org.springframework.context.support.GenericXmlApplicationContext o
정보: Closing org.springframework.context.support.GenericXmlApplicationContext@378fd1ac: star
BUILD SUCCESSFUL
Total time: 0.682 secs
```

스프링 XML 설정

4.1 <beans> 루트 엘리먼트

- 스프링 컨테이너는 <bean> 저장소에 해당하는 XML 설정 파일을 참조하여 <bean>의 생명 주기 관리 및 다양한 서비스 제공
- 설정 파일을 정확하게 작성 및 관리가 중요
- 스프링 설정 파일 이름은 무엇이든 상관없으나 <beans>를 루트 엘리먼트로 사용해야 한다.
- beans 네임스페이스가 기본 네임스페이스로 선언되어야 함
- spring-beans.xsd 스키마 문서가 schemaLocation에 등록
- <bean>, <description>, <alias>, <import> 등의 자식 엘리먼트



```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <beans xmlns="http://www.springframework.org/schema/beans"
3       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4       xsi:schemaLocation="http://www.springframework.org/schema/beans
5                           http://www.springframework.org/schema/beans/spring-beans.xsd">
6
7 </beans>
8
```


<import>, <bean> 엘리먼트

- <import> - 기능별로 여러 XML 파일 나누어서 설정할 때

context-datasource.xml	context-transaction.xml
<pre><beans> DataSource 관련 설정 </beans></pre>	<pre><beans> Transaction 관련 설정 </beans></pre>
applicationContext.xml	
<pre><beans> <import resource="context-datasource.xml"/> <import resource="context-transaction.xml"/> </beans></pre>	

- <bean> - 스프링 설정 파일에 클래스 등록할 때
 - id와 class 속성 사용
 - id는 생략 가능하지만 class는 정확한 패키지 경로와 클래스 이름을 지정

```
<bean id="tv" class="polymorphism.SamsungTV">
```

- id 속성값에 해당하는 문자열은 자바 식별자 규칙을 따르며 일반적으로 나탁표기법 (CamelCase)을 사용
- id 대신 name 속성을 사용할 수 있으며 식별자 작성 규칙을 따르지 않아도 된다.

```
name = "http://www.inje.ac.kr" 또는 name = "http://www.daum.net"
```

<bean> 엘리먼트 속성

- init-method 속성: 객체 생성후 멤버변수 초기화 작업

```
<bean id="tv" class="polymorphism.SamsungTV" init-method="init"/>
```

- destroy-method 속성: 컨테이너가 객체를 삭제하기 전에 호출

```
<bean id="tv" class="polymorphism.SamsungTV" destroy-method="destroyMethod"/>
```

- lazy-init 속성: 컨테이너가 생성되는 시점이 아닌 <bean>사용 시점에 객체를 생성하도록 설정(효율적 메모리 관리)

```
<bean id="tv" class="polymorphism.SamsungTV" lazy-init="true"/>
```

- scope 속성: 컨테이너가 생성한 <bean>을 어느 범위에서 사용할 지를 지정. “ singleton ” 이 기본값

```
<bean id="tv" class="polymorphism.SamsungTV" scope="singleton"/>
```

- “ prototype ” 도 있는데 컨테이너가 해당 <bean>이 요청될 때마다 매번 새로운 객체를 생성하여 반환


```

1 package polymorphism;
2
3 import org.springframework.context.support.AbstractApplicationContext;
4
5
6 public class TVUser {
7
8     public static void main(String[] args){
9
10         // Spring 컨테이너를 구동
11         AbstractApplicationContext factory = new GenericXmlApplicationContext("applicationContext.xml");
12
13         //BeanFactory factory = new BeanFactory();
14         TV tv1 = (TV) factory.getBean("tv");
15         TV tv2 = (TV) factory.getBean("tv");
16         TV tv3 = (TV) factory.getBean("tv");
17
18         // Spring 컨테이너를 종료
19         factory.close();
20     }
21 }

```

```

:compileJava UP-TO-DATE
:processResources
:classes
:run5월 09, 2017 12:15:13 오후 org.springframework.beans.factory.xml.XmlBean
정보: Loading XML bean definitions from class path resource [applicationCor
5월 09, 2017 12:15:14 오후 org.springframework.context.support.GenericXmlAp
정보: Refreshing org.springframework.context.support.GenericXmlApplicationC

```

```

===> SamsungTV 객체 생성
5월 09, 2017 12:15:14 오후 org.springframework.context.support.GenericXmlAp
정보: Closing org.springframework.context.support.GenericXmlApplicationCont

```

BUILD SUCCESSFUL

Total time: 0.544 secs

scope= “ singleton ”

```

:classes
:run5월 09, 2017 12:16:27 오후 org.springframework.beans.factory.xml.XmlBean
정보: Loading XML bean definitions from class path resource [applicationCor
5월 09, 2017 12:16:27 오후 org.springframework.context.support.GenericXmlAp
정보: Refreshing org.springframework.context.support.GenericXmlApplicationC

```

```

===> SamsungTV 객체 생성
===> SamsungTV 객체 생성
===> SamsungTV 객체 생성
5월 09, 2017 12:16:27 오후 org.springframework.context.support.GenericXmlAp
정보: Closing org.springframework.context.support.GenericXmlApplicationCont

```

BUILD SUCCESSFUL

Total time: 0.537 secs

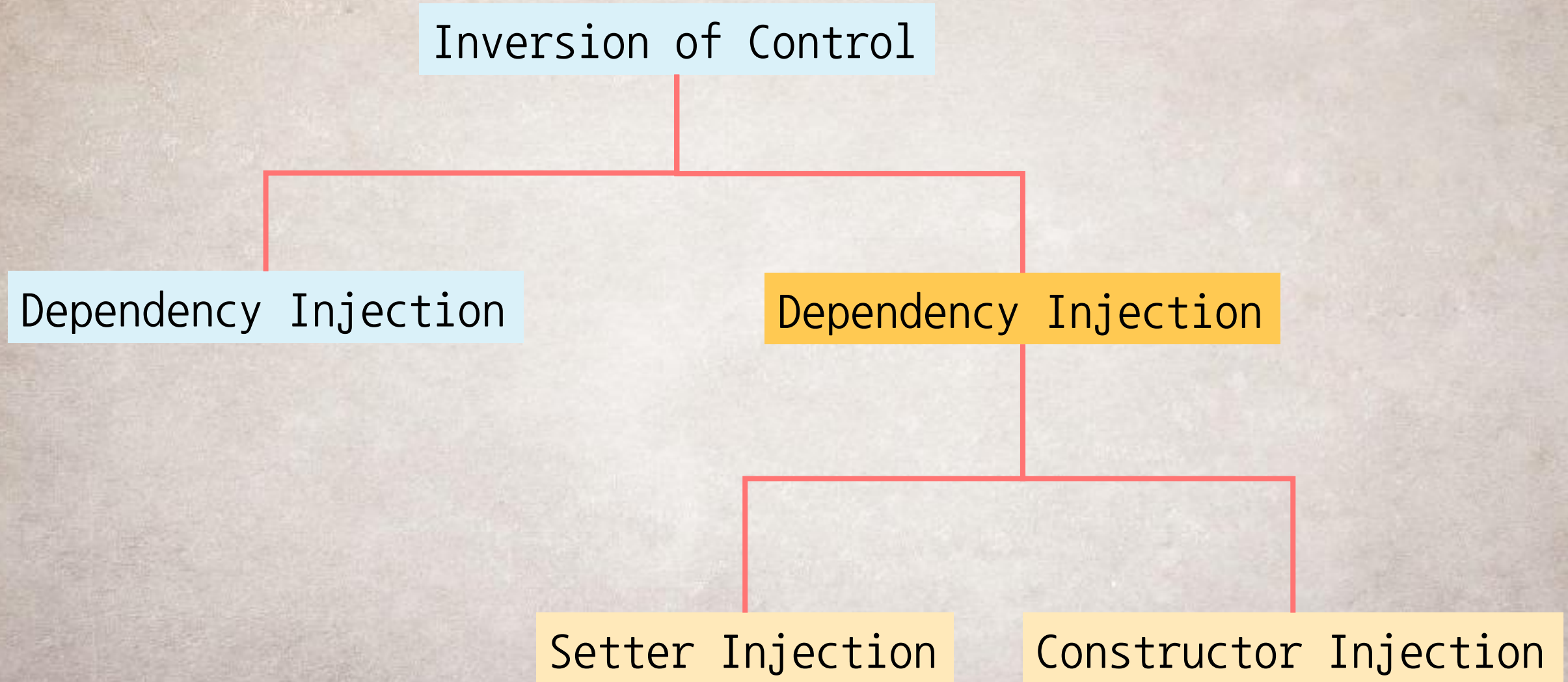
scope= “ prototype ”

5. 의존성 주입

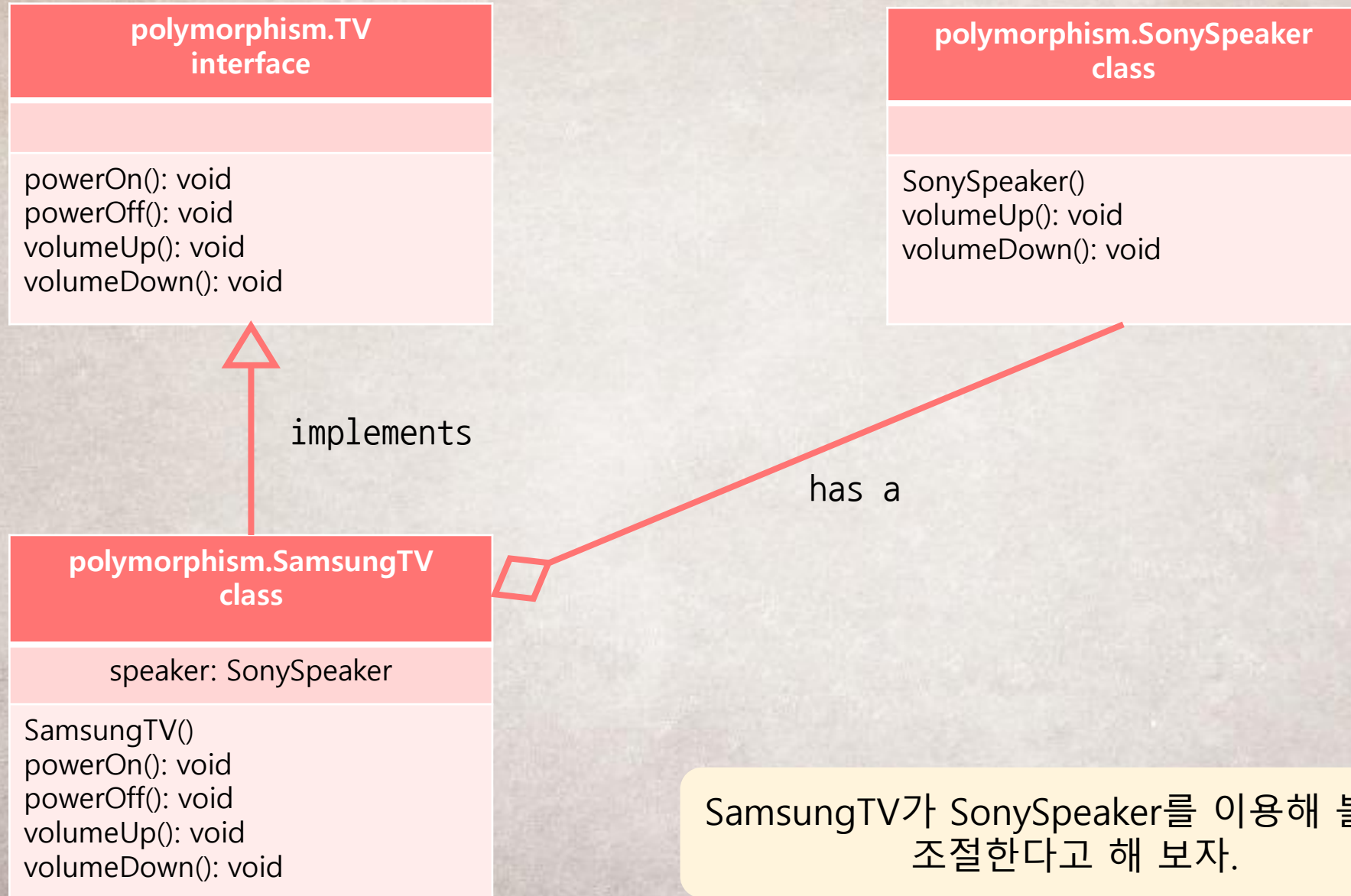
5.1 의존성 관리

- 스프링은 객체의 생성과 의존관계를 컨테이너가 자동 관리
- IoC(제어의 역행)의 핵심 원리이기도 하며, 두 가지 형태를 지원
 - Dependency Lookup
 - 컨테이너가 객체를 생성하고 클라이언트가 생성한 객체를 검색(lookup)하여 사용
 - 지금까지 컨테이너가 사용해왔던 방법
 - 실제 어플리케이션 과정에서는 사용하지 않음
 - Dependency Injection
 - 객체 사이의 의존관계를 스프링 설정 파일에 등록 정보를 바탕으로 컨테이너가 처리
 - 의존성 설정을 변경하고자 한다면 프로그램 코드가 아닌 스프링 설정 파일을 수정
 - ➔ 유지보수가 향상
 - 컨테이너가 직접 객체들 사이의 의존관계를 처리하는 것을 의미
 - Setter 메소드 기반의 세터 인젝션(Setter Injection)과 생성자를 기반으로 하는 생성자 인젝션(Constructor Injection)으로 나뉨

스프링의 IoC



의존성 관계



SamsungTV가 SonySpeaker를 이용해 볼륨을 조절한다고 해 보자.

SonySpeaker.Java 클래스

```
TVUser.java SamsungTV.java SonySpeaker.java ✖
1 package polymorphism;
2
3 public class SonySpeaker {
4
5     public SonySpeaker(){
6         System.out.println("==> SonySpeaker 객체 생성");
7     }
8
9     public void VolumeUp(){
10        System.out.println("SonySpeaker -- 소리를 올린다.");
11    }
12
13    public void VolumeDown(){
14        System.out.println("SonySpeaker -- 소리를 내린다.");
15    }
16 }
17
18
```

SamsungTV.Java 클래스

```
TVUser.java SamsungTV.java SonySpeaker.java
1 package polymorphism;
2
3 public class SamsungTV implements TV {
4
5     private SonySpeaker speaker;
6
7     public SamsungTV(){
8         System.out.println("====> SamsungTV 객체 생성");
9     }
10
11     @Override
12     public void powerOn() {
13         System.out.println("SamungTV -- 전원을 켜다.");
14     }
15
16
17     @Override
18     public void powerOff() {
19         System.out.println("SamungTV -- 전원을 끈다.");
20     }
21
22
23     @Override
24     public void volumeUp() {
25         speaker = new SonySpeaker();
26         speaker.VolumeUp();
27     }
28
29
30     @Override
31     public void volumeDown() {
32         speaker = new SonySpeaker();
33         speaker.VolumeDown();
34     }
35
36 }
37
```

SonySpeaker의 볼륨 조절 기능을 수행하기 위해서 SonySpeaker 타입의 speaker 멤버 변수 선언

SonySpeaker 타입의 객체 생성과 VolumeUp 메소드 실행

SonySpeaker 타입의 객체 생성과 VolumeDown 메소드 실행

➔ volumeUp(), volumeDown() 메소드 실행 순서를 모르기 때문에 각 메소드 실행 전에 Speaker 객체 생성

실행 결과

```
BoardWeb - run [Gradle Project] run in D:\workspace\java\2017_1\BoardWeb (2017. 5. 9 오후 6:31:19)
:classes
:run5월 09, 2017 6:31:20 오후 org.springframework.beans.factory.xml.XmlBeanDefinitionReader loadBeanDefinitions
정보: Loading XML bean definitions from class path resource [applicationContext.xml]
5월 09, 2017 6:31:20 오후 org.springframework.context.support.GenericXmlApplicationContext prepareRefresh
정보: Refreshing org.springframework.context.support.GenericXmlApplicationContext@378fd1ac: startup date [Tue May
===> SamsungTV 객체 생성
SamungTV -- 전원을 켜다.
===> SonySpeaker 객체 생성
SonySpeaker -- 소리를 올린다.
===> SonySpeaker 객체 생성
SonySpeaker -- 소리를 내린다.
SamungTV -- 전원을 끈다.
5월 09, 2017 6:31:20 오후 org.springframework.context.support.GenericXmlApplicationContext doClose
정보: Closing org.springframework.context.support.GenericXmlApplicationContext@378fd1ac: startup date [Tue May 09
BUILD SUCCESSFUL
```

- 1) SonySpeaker 객체를 쓸데 없이 두 개를 생성
- 2) 운영 과정에서 SonySpeaker의 성능이 떨어져서 AppleSpeaker와 같은 다른 Speaker로 변경하고자 할 때, volumeUp(), volumeDown() 두 개의 메소드를 모두 수정

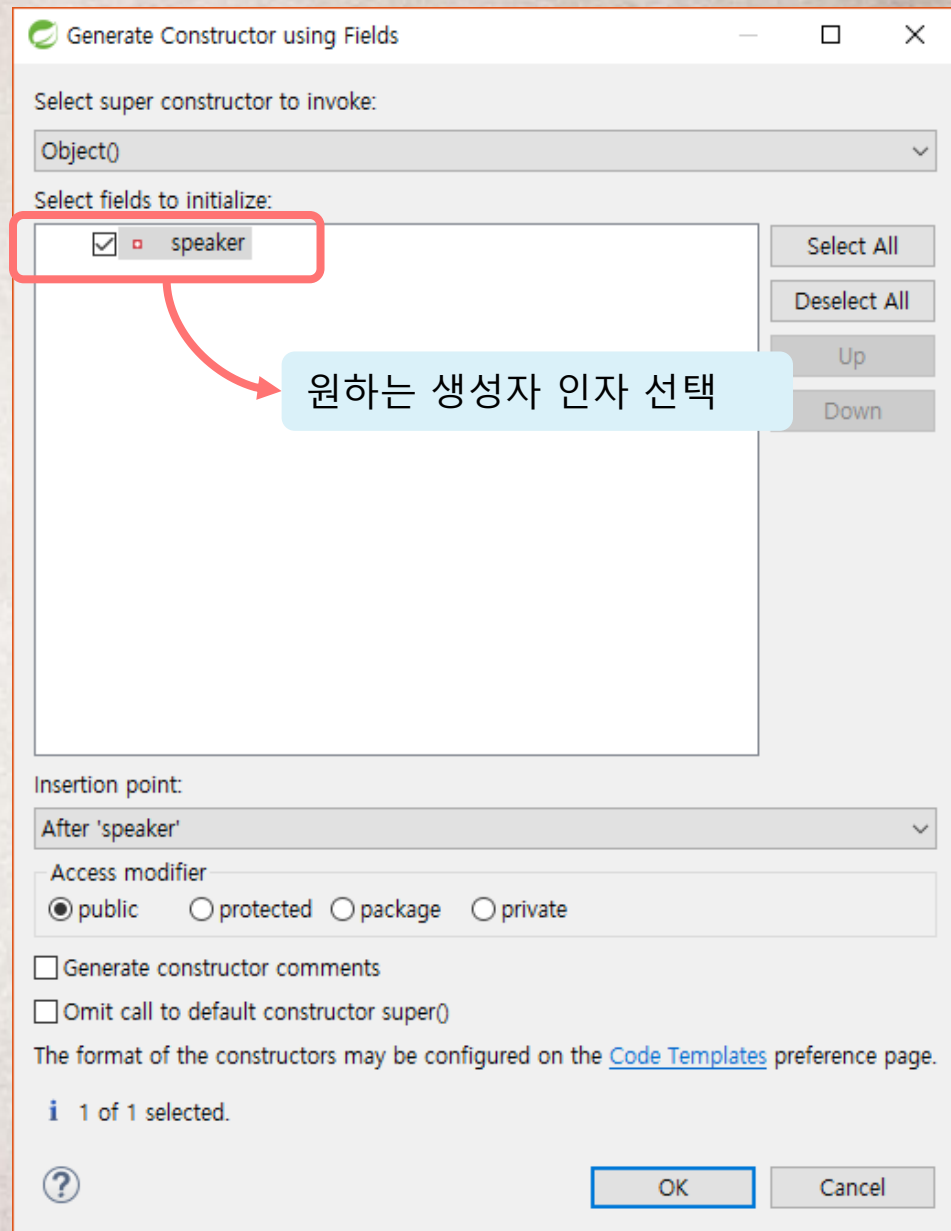
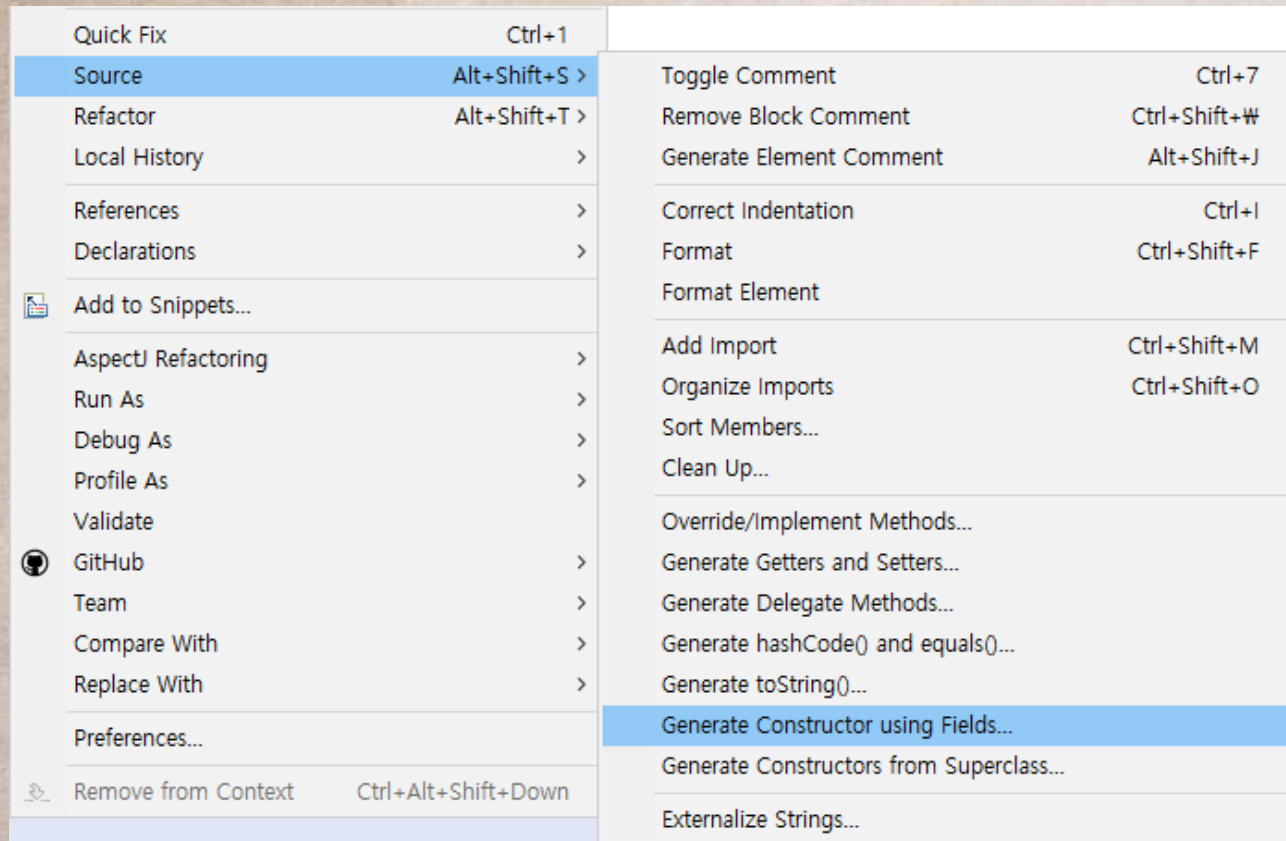
문제 발생 이유? 의존관계에 있는 Speaker 객체에 대한 객체 생성 코드를 직접 SamsungTV 소스에 명시했기 때문

➔ **의존성 주입으로 해결**

생성자 인젝션 이용하여 문제점 해결하기

- 생성자 인젝션을 사용하면 생성자의 매개변수로 의존관계에 있는 객체의 주소 정보를 전달
- SamsungTV 클래스에 생성자를 추가

생성자 만들기



SamsungTV.Java 클래스 수정

```
4
5  private SonySpeaker speaker;
6
7
8  public SamsungTV(SonySpeaker speaker) {
9      super();
10     this.speaker = speaker;
11 }
12
13 public SamsungTV(){
14     System.out.println("==> SamsungTV 객체 생성");
15 }
16
```



```
4
5  private SonySpeaker speaker;
6
7
8  public SamsungTV(SonySpeaker speaker) {
9      System.out.println("==> SamsungTV 객체 생성 (2)");
10     this.speaker = speaker;
11 }
12
13 public SamsungTV(){
14     System.out.println("==> SamsungTV 객체 생성 (1)");
15 }
16
```

applicationContext.xml 수정

```
TVUser.java SamsungTV.java SonySpeaker.java applicationContext.xml
1 <?xml version="1.0" encoding="UTF-8"?>
2 <beans xmlns="http://www.springframework.org/schema/beans"
3     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4     xsi:schemaLocation="http://www.springframework.org/schema/beans
5         http://www.springframework.org/schema/beans/spring-beans.xsd">
6
7     <bean id="tv" class="polymorphism.SamsungTV">
8         <constructor-arg ref="sony"></constructor-arg>
9     </bean>
10
11     <bean id="sony" class="polymorphism.SonySpeaker"></bean>
12 </beans>
13
14
```

- 생성자 인젝션을 위해 <bean> 태그 내부에 <constructor-arg> 엘리먼트를 추가
- 생성자 인자로 전달할 객체의 id를 <constructor-arg> 엘리먼트의 ref 속성으로 참조

실행 결과

```
Console Progress Problems
BoardWeb - run [Gradle Project] run in D:\workspace\java\2017_1\BoardWeb (2017. 5. 9 오후 7:14:36)
정보: Refreshing org.springframework.context.support.GenericXmlApplicationContext@378fd1ac: startup date [Tue
====> SonySpeaker 객체 생성
====> SamsungTV 객체 생성 (2)
SamungTV -- 전원을 켜다.
====> SonySpeaker 객체 생성
SonySpeaker -- 소리를 올린다.
====> SonySpeaker 객체 생성
SonySpeaker -- 소리를 내린다.
SamungTV -- 전원을 끈다.
5월 09, 2017 7:14:36 오후 org.springframework.context.support.GenericXmlApplicationContext doClose
정보: Closing org.springframework.context.support.GenericXmlApplicationContext@378fd1ac: startup date [Tue Ma
BUILD SUCCESSFUL
Total time: 0.648 secs
```

- 1) SamsungTV 클래스 객체의 생성자가 기본 생성자가 아닌 두번째 생성자가 사용됨
- 2) 스프링 설정 파일에 SonySpeaker가 SamsungTV 클래스 밑에 등록되었는데도 먼저 생성되고 있다는 점

다중 변수 매핑

```
TVUser.java SamsungTV.java SonySpeaker.java applicationContext.xml
1 package polymorphism;
2
3 public class SamsungTV implements TV {
4
5     private SonySpeaker speaker;
6     private int price;
7
8     public SamsungTV(SonySpeaker speaker, int price) {
9         System.out.println("==> SamsungTV 객체 생성 (3)");
10        this.speaker = speaker;
11        this.price = price;
12    }
13
14    public SamsungTV(SonySpeaker speaker) {
15        System.out.println("==> SamsungTV 객체 생성 (2)");
16        this.speaker = speaker;
17    }
18
19    public SamsungTV(){
20        System.out.println("==> SamsungTV 객체 생성 (1)");
21    }
22
23    @Override
24    public void powerOn() {
25        System.out.println("SamungTV -- 전원을 켜다.(가격: " + this.price + ")");
26    }
27
28 }
```

applicationContext.xml 수정

```
TVUser.java SamsungTV.java SonySpeaker.java applicationContext.xml
1 <?xml version="1.0" encoding="UTF-8"?>
2 <beans xmlns="http://www.springframework.org/schema/beans"
3     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4     xsi:schemaLocation="http://www.springframework.org/schema/beans
5         http://www.springframework.org/schema/beans/spring-beans.xsd">
6
7     <bean id="tv" class="polymorphism.SamsungTV">
8         <constructor-arg ref="sony"></constructor-arg>
9         <constructor-arg value="2700000"></constructor-arg>
10    </bean>
11
12    <bean id="sony" class="polymorphism.SonySpeaker"></bean>
13 </beans>
14
15
```

- 1) 등록된 다른 객체일 때는 ref 속성을 이용
- 2) 고정된 문자열이나 정수 같은 기본형은 value 속성을 사용

실행 결과

```
BoardWeb - run [Gradle Project] run in D:\workspace\java\2017_1\BoardWeb (2017. 5. 9 오후 7:38:18)
5월 09, 2017 7:38:19 오후 org.springframework.context.support.GenericXmlApplicationContext prepareRefresh
정보: Refreshing org.springframework.context.support.GenericXmlApplicationContext@378fd1ac: startup date [Tue
May 9, 2017 7:38:19 PM KST]; root of org.springframework.context.support.GenericXmlApplicationContext
===> SonySpeaker 객체 생성
===> SamsungTV 객체 생성 (3)
SamungTV -- 전원을 켜다.(가격: 2700000)
===> SonySpeaker 객체 생성
SonySpeaker -- 소리를 올린다.
===> SonySpeaker 객체 생성
SonySpeaker -- 소리를 내린다.
SamungTV -- 전원을 끈다.
5월 09, 2017 7:38:19 오후 org.springframework.context.support.GenericXmlApplicationContext doClose
정보: Closing org.springframework.context.support.GenericXmlApplicationContext@378fd1ac: startup date [Tue Ma
y 9, 2017 7:38:19 PM KST]; root of org.springframework.context.support.GenericXmlApplicationContext

BUILD SUCCESSFUL

Total time: 0.682 secs
```


생성자 호출이 불분명 - 생성자가 오버로딩 되어있을 경우

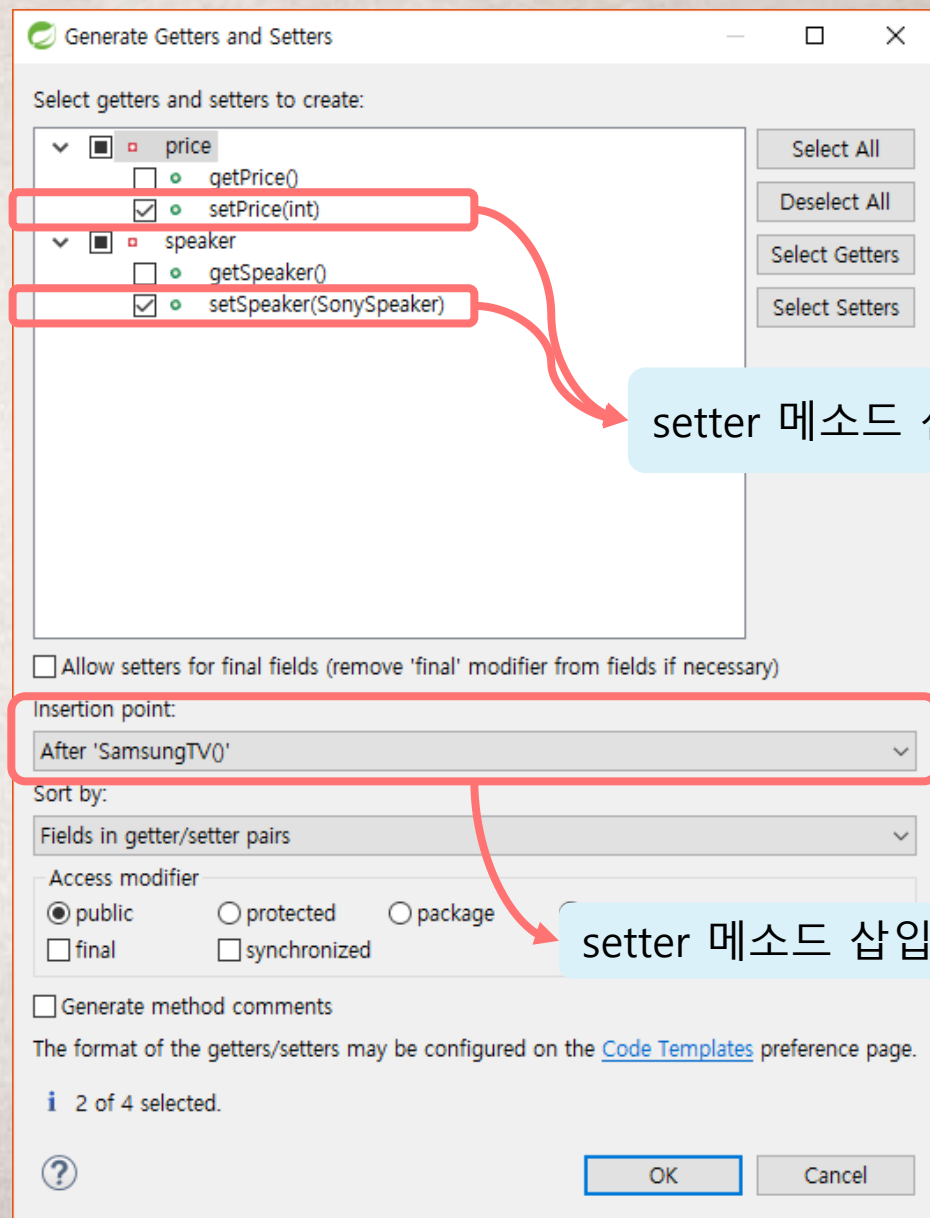
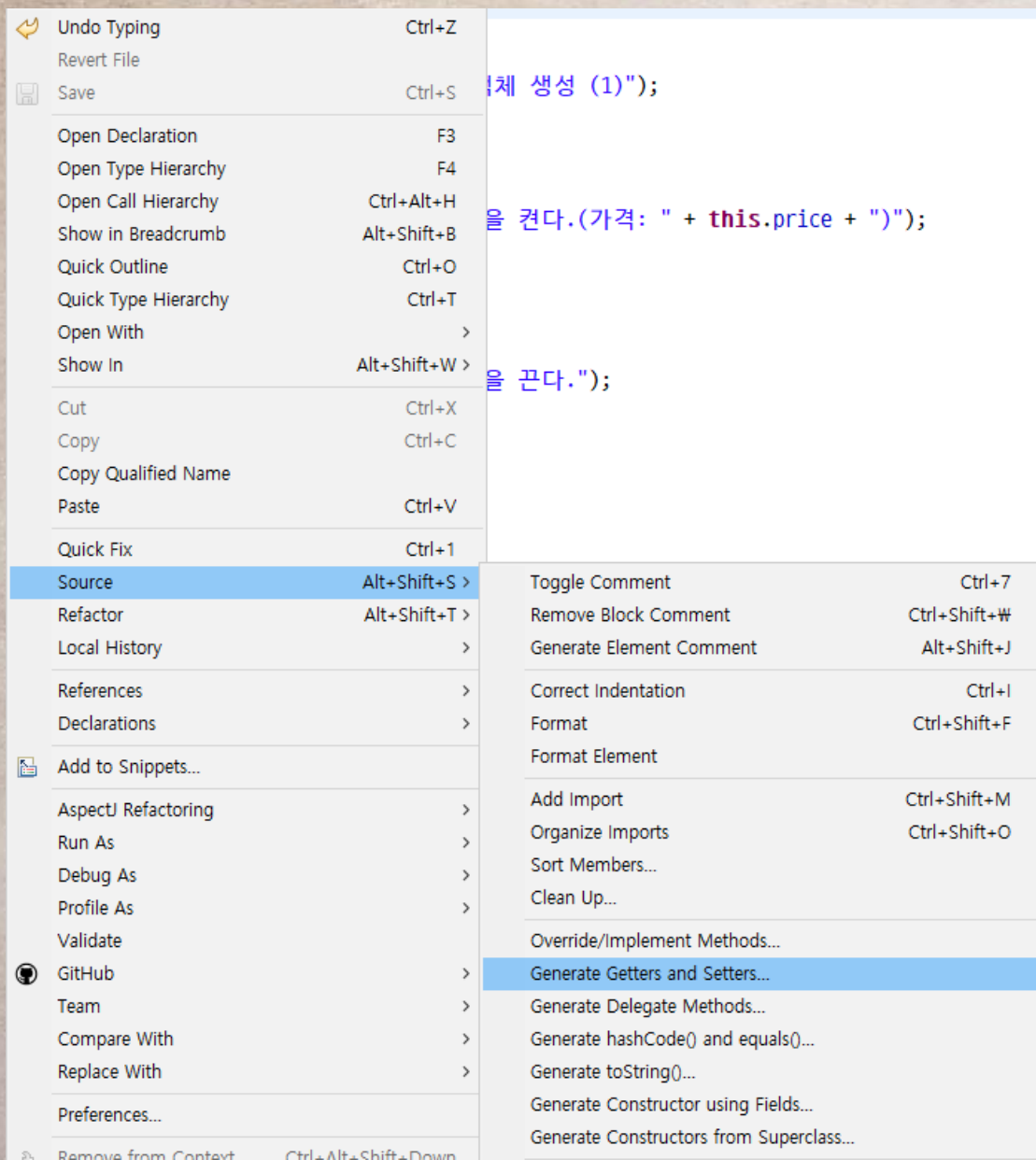
```
TVUser.java SamsungTV.java SonySpeaker.java applicationContext.xml
1 <?xml version="1.0" encoding="UTF-8"?>
2 <beans xmlns="http://www.springframework.org/schema/beans"
3     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4     xsi:schemaLocation="http://www.springframework.org/schema/beans
5         http://www.springframework.org/schema/beans/spring-beans.xsd">
6
7     <bean id="tv" class="polymorphism.SamsungTV">
8         <constructor-arg index="0" ref="sony"></constructor-arg>
9         <constructor-arg index="1" value="2700000"></constructor-arg>
10    </bean>
11
12    <bean id="sony" class="polymorphism.SonySpeaker"></bean>
13 </beans>
14
15
16
```

index 속성을 사용

Setter 인젝션 이용하기

- Setter 메소드를 호출하여 의존성을 주입하는 방법
- 두 가지 방법 모두 멤버변수에 원하는 값을 설정하는 것이 목적
- 코딩 컨벤션에 따라 한가지를 선택해서 사용하면 됨
 - 주로 Setter 인젝션을 사용
 - Setter 메소드를 제공되지 않는 클래스에 대해서는 생성자 인젝션 사용

SamsungTV 클래스에 Setter 메소드를 추가하기



SamsungTV.Java 클래스

```
TVUser.java SamsungTV.java SonySpeaker.java applicationContext.xml
1 package polymorphism;
2
3 public class SamsungTV implements TV {
4
5     private SonySpeaker speaker;
6     private int price;
7
8     public SamsungTV(){
9         System.out.println("==> SamsungTV 객체 생성 (1)");
10    }
11
12    public void setSpeaker(SonySpeaker speaker) {
13        System.out.println("==> setSpeaker() 호출");
14        this.speaker = speaker;
15    }
16
17    public void setPrice(int price) {
18        System.out.println("==> setPrice() 호출");
19        this.price = price;
20    }
21
22    @Override
23    public void powerOn() {
24        System.out.println("SamungTV -- 전원을 켜다.(가격: " + this.price + ")");
25    }
26
27 }
```

setSpeaker 메소드

setPrice 메소드

applicationContext.xml

```
TVUser.java SamsungTV.java SonySpeaker.java applicationContext.xml
1 <?xml version="1.0" encoding="UTF-8"?>
2 <beans xmlns="http://www.springframework.org/schema/beans"
3     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4     xsi:schemaLocation="http://www.springframework.org/schema/beans
5         http://www.springframework.org/schema/beans/spring-beans.xsd">
6
7     <bean id="tv" class="polymorphism.SamsungTV">
8         <property name="speaker" ref="sony"></property>
9         <property name="price" value="2700000"></property>
10    </bean>
11
12    <bean id="sony" class="polymorphism.SonySpeaker"></bean>
13 </beans>
14
15
16
```

- Setter 인젝션을 사용할 경우 <property> 엘리먼트를 사용하고 name 속성값이 호출하고자 하는 메소드 이름
:예) 메소드 이름: setSpeaker()
name 속성값: name= " speaker "

실행 결과

```
BoardWeb - run [Gradle Project] run in D:\workspace\java\2017_1\BoardWeb (2017. 5. 9 오후 8:12:04)
:run5월 09, 2017 8:12:05 오후 org.springframework.beans.factory.xml.XmlBeanDefinitionReader loadBeanDefinitions
정보: Loading XML bean definitions from class path resource [applicationContext.xml]
5월 09, 2017 8:12:05 오후 org.springframework.context.support.GenericXmlApplicationContext prepareRefresh
정보: Refreshing org.springframework.context.support.GenericXmlApplicationContext@378fd1ac: startup date [Tue May 09 20:12:05 KST 2017]
===> SamsungTV 객체 생성 (1)
===> SonySpeaker 객체 생성
===> setSpeaker() 호출
===> setPrice() 호출
SamungTV -- 전원을 켜다.(가격: 2700000)
===> SonySpeaker 객체 생성
SonySpeaker -- 소리를 올린다.
===> SonySpeaker 객체 생성
SonySpeaker -- 소리를 내린다.
SamungTV -- 전원을 끈다.
5월 09, 2017 8:12:05 오후 org.springframework.context.support.GenericXmlApplicationContext doClose
정보: Closing org.springframework.context.support.GenericXmlApplicationContext@378fd1ac: startup date [Tue May 09 20:12:05 KST 2017]

BUILD SUCCESSFUL

Total time: 0.574 secs
```

1. SamsungTV 객체 생성
2. SonySpeaker 객체 생성
3. Setter 메소드 호출

p 네임스페이스 사용하기

➔ 좀 더 효율적인 의존성 주입 처리

p 네임스페이스 사용하기

```
SamsungTV.java SonySpeaker.java applicationContext.xml
1 <?xml version="1.0" encoding="UTF-8"?>
2 <beans xmlns="http://www.springframework.org/schema/beans"
3       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4       xmlns:p="http://www.springframework.org/schema/p"
5       xsi:schemaLocation="http://www.springframework.org/schema/beans
6                           http://www.springframework.org/schema/beans/spring-beans.xsd">
7
8   <bean id="tv" class="polymorphism.SamsungTV">
9       <property name="speaker" ref="sony"></property>
10      <property name="price" value="2700000"></property>
11  </bean>
12
13  <bean id="sony" class="polymorphism.SonySpeaker"></bean>
14 </beans>
15
16
```

p 네임스페이스 사용하기

SamsungTV.java SonySpeaker.java applicationContext.xml

Namespaces

Configure Namespaces

Select XSD namespaces to use in the configuration file

- ☐ aop - http://www.springframework.org/schema/aop
- ☒ beans - http://www.springframework.org/schema/beans
- ☐ c - http://www.springframework.org/schema/c
- ☐ cache - http://www.springframework.org/schema/cache
- ☐ context - http://www.springframework.org/schema/context
- ☐ jee - http://www.springframework.org/schema/jee
- ☐ lang - http://www.springframework.org/schema/lang
- ☒ p - http://www.springframework.org/schema/p
- ☐ task - http://www.springframework.org/schema/task
- ☐ util - http://www.springframework.org/schema/util

Source Namespaces Overview beans

- p네임스페이스 선언 후 참조형 변수에 참조할 객체 할당

p:변수명-ref= “ 참조할 객체의 이름이나 아이디 ”

- 기본형이나 문자형 변수에 직접 값을 설정하는 방법

p:변수명= “ 설정할 값 ”

- p네임스페이스를 이용한 SamsungTV와 SonySpeaker의 의존성 주입 설정

```
SamsungTV.java SonySpeaker.java applicationContext.xml
1 <?xml version="1.0" encoding="UTF-8"?>
2 <beans xmlns="http://www.springframework.org/schema/beans"
3     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4     xmlns:p="http://www.springframework.org/schema/p"
5     xsi:schemaLocation="http://www.springframework.org/schema/beans
6         http://www.springframework.org/schema/beans/spring-beans.xsd">
7
8     <bean id="tv" class="polymorphism.SamsungTV" p:speaker-ref="sony" p:price="2700000">
9     </bean>
10
11     <bean id="sony" class="polymorphism.SonySpeaker"></bean>
12 </beans>
13
14
15
```

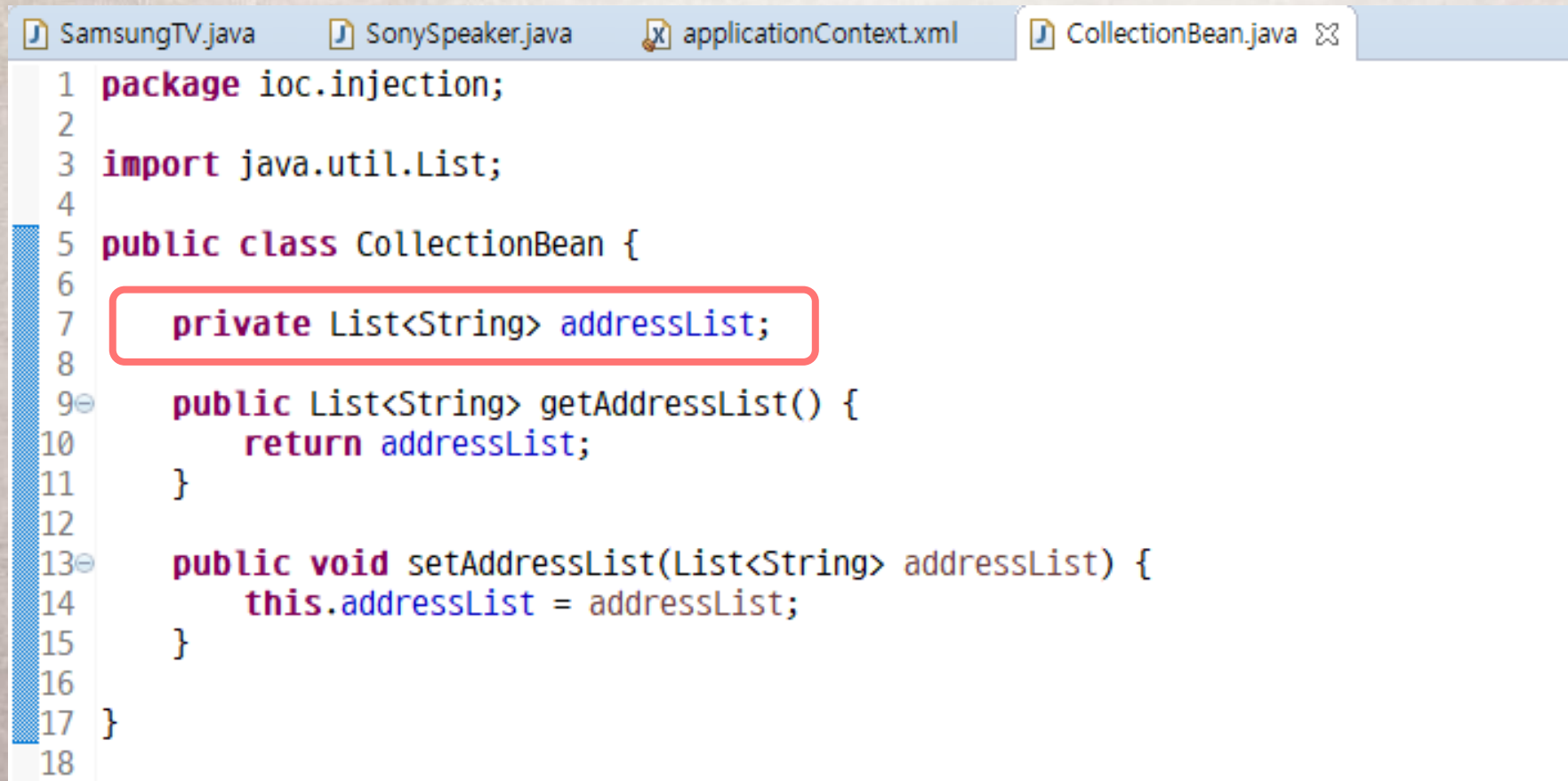

컬렉션(Collection) 객체 설정

- 배열이나 List와 같은 컬렉션(Collection)객체를 이용할 경우
- 스프링에서는 컬렉션 매핑과 관련된 엘리먼트 지원

컬렉션 유형	엘리먼트
java.util.List, 배열	<list>
java.util.Set	<set>
java.util.Map	<map>
java.util.Properties	<props>

List 타입 매핑

- 배열 객체나 java.util.List타입의 컬렉션 객체는 <list> 태그 사용
- CollectionBean.Java 클래스



```
1 package ioc.injection;
2
3 import java.util.List;
4
5 public class CollectionBean {
6
7     private List<String> addressList;
8
9     public List<String> getAddressList() {
10         return addressList;
11     }
12
13     public void setAddressList(List<String> addressList) {
14         this.addressList = addressList;
15     }
16
17 }
18
```


applicationContext.xml

```
SamsungTV.java  SonySpeaker.java  applicationContext.xml  CollectionBean.java
1  <?xml version="1.0" encoding="UTF-8"?>
2  <beans xmlns="http://www.springframework.org/schema/beans"
3      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4      xmlns:p="http://www.springframework.org/schema/p"
5      xsi:schemaLocation="http://www.springframework.org/schema/beans
6                          http://www.springframework.org/schema/beans/spring-beans.xsd">
7
8      <bean id="collectionBean" class="ioc.injection.CollectionBean">
9          <property name="addressList">
10             <list>
11                 <value>김해시 어방동</value>
12                 <value>부산시 동대신동</value>
13             </list>
14          </property>
15      </bean>
16 </beans>
17
18
```

setAddressList() 메소드의 인자
List의 값

- 주소가 저장된 List객체를 CollectionBean 객체의 setAddressList() 메소드를 호출할 때, 인자로 전달하여 addressList 멤버 변수를 초기화

CollectionBeanClient.Java 클래스

```
applicationContext.xml  CollectionBean.java  CollectionBeanClient.java ✕
1 package ioc.injection;
2
3 import java.util.List;
4
5 import org.springframework.context.support.AbstractApplicationContext;
6 import org.springframework.context.support.GenericXmlApplicationContext;
7
8 public class CollectionBeanClient {
9     public static void main(String[] args) {
10
11         // Spring 컨테이너를 구동
12         AbstractApplicationContext factory = new GenericXmlApplicationContext("applicationContext.xml");
13
14         CollectionBean bean = (CollectionBean) factory.getBean("collectionBean");
15         List<String> addressList = bean.getAddressList();
16         for (String address : addressList){
17             System.out.println(address.toString());
18         }
19         // Spring 컨테이너를 종료
20         factory.close();
21     }
22 }
23
```

실행 결과

```
BoardWeb - run [Gradle Project] run in D:\workspace\java\2017_1\BoardWeb (2017. 5. 9 오후 10:05:55)

:compileJava
:processResources
:classes
:run5월 09, 2017 10:05:56 오후 org.springframework.beans.factory.xml.XmlBeanDefinitionReader loadBeanDefinitions
정보: Loading XML bean definitions from class path resource [applicationContext.xml]
5월 09, 2017 10:05:56 오후 org.springframework.context.support.GenericXmlApplicationContext prepareRefresh
정보: Refreshing org.springframework.context.support.GenericXmlApplicationContext@378fd1ac: startup date [Tue May 09
김해시 어방동
부산시 동대신동
5월 09, 2017 10:05:56 오후 org.springframework.context.support.GenericXmlApplicationContext doClose
정보: Closing org.springframework.context.support.GenericXmlApplicationContext@378fd1ac: startup date [Tue May 09

BUILD SUCCESSFUL

Total time: 0.656 secs
```

setAddressList() 메소드에 의해 출력된 addressList List값

Set 타입 매핑-CollectionBean.Java 클래스

```
applicationContext.xml  CollectionBean.java  CollectionBeanClient.java
1 package ioc.injection;
2
3 import java.util.Set;
4
5 public class CollectionBean {
6
7     private Set<String> addressList;
8
9     public void setAddressList(Set<String> addressList) {
10         this.addressList = addressList;
11     }
12
13     public Set<String> getAddressList() {
14         return addressList;
15     }
16
17 }
18
```


applicationContext.xml

```
applicationContext.xml  CollectionBean.java  CollectionBeanClient.java
1  <?xml version="1.0" encoding="UTF-8"?>
2  <beans xmlns="http://www.springframework.org/schema/beans"
3        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4        xmlns:p="http://www.springframework.org/schema/p"
5        xsi:schemaLocation="http://www.springframework.org/schema/beans
6                             http://www.springframework.org/schema/beans/spring-beans.xsd">
7
8      <bean id="collectionBean" class="ioc.injection.CollectionBean">
9          <property name="addressList">
10             <set value-type="java.lang.String">
11                 <value>김해시 어방동</value>
12                 <value>부산시 동대신동</value>
13                 <value>부산시 동대신동</value>
14             </set>
15          </property>
16      </bean>
17 </beans>
18
19
```

value가 중복??
set은 중복처리해서 저장함

CollectionBeanClient.Java 클래스

```
applicationContext.xml  CollectionBean.java  CollectionBeanClient.java  ✖
1  package ioc.injection;
2
3  import java.util.Set;
4
5  import org.springframework.context.support.AbstractApplicationContext;
6  import org.springframework.context.support.GenericXmlApplicationContext;
7
8  public class CollectionBeanClient {
9      public static void main(String[] args) {
10
11         // Spring 컨테이너를 구동
12         AbstractApplicationContext factory = new GenericXmlApplicationContext("applicationContext.xml");
13
14         CollectionBean bean = (CollectionBean) factory.getBean("collectionBean");
15         Set<String> addressList = bean.getAddressList();
16         for (String address : addressList){
17             System.out.println(address.toString());
18         }
19         // Spring 컨테이너를 종료
20         factory.close();
21     }
22 }
23
```

Map 타입 매핑-CollectionBean.Java 클래스

```
applicationContext.xml  CollectionBean.java  CollectionBeanClient.java
1 package ioc.injection;
2
3 import java.util.Map;
4
5 public class CollectionBean {
6
7     private Map<String, String> addressList;
8
9     public Map<String, String> getAddressList() {
10         return addressList;
11     }
12
13     public void setAddressList(Map<String, String> addressList) {
14         this.addressList = addressList;
15     }
16
17 }
18
```


applicationContext.xml

```
applicationContext.xml  CollectionBean.java  CollectionBeanClient.java
1  <?xml version="1.0" encoding="UTF-8"?>
2  <beans xmlns="http://www.springframework.org/schema/beans"
3      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4      xmlns:p="http://www.springframework.org/schema/p"
5      xsi:schemaLocation="http://www.springframework.org/schema/beans
6                          http://www.springframework.org/schema/beans/spring-beans.xsd">
7
8      <bean id="collectionBean" class="ioc.injection.CollectionBean">
9          <property name="addressList">
10             <map>
11                 <entry>
12                     <key><value>인제대</value></key>
13                     <value>김해시 어방동</value>
14                 </entry>
15                 <entry>
16                     <key><value>박경태</value></key>
17                     <value>부산시 동대신동</value>
18                 </entry>
19             </map>
20          </property>
21      </bean>
22 </beans>
23
24
```

CollectionBeanClient.Java 클래스

```
applicationContext.xml  CollectionBean.java  CollectionBeanClient.java  ✕
1 package ioc.injection;
2
3 import java.util.Map;
4
5 import org.springframework.context.support.AbstractApplicationContext;
6 import org.springframework.context.support.GenericXmlApplicationContext;
7
8 public class CollectionBeanClient {
9     public static void main(String[] args) {
10
11         // Spring 컨테이너를 구동
12         AbstractApplicationContext factory = new GenericXmlApplicationContext("applicationContext.xml");
13
14         CollectionBean bean = (CollectionBean) factory.getBean("collectionBean");
15         Map<String, String> addressList = bean.getAddressList();
16         for (Map.Entry<String, String> address: addressList.entrySet()){
17             System.out.println("Key: " + address.getKey() + ", Value: " + address.getValue());
18         }
19         // Spring 컨테이너를 종료
20         factory.close();
21     }
22 }
23
```


실행 결과

```
BoardWeb - run [Gradle Project] run in D:\workspace\java\2017_1\BoardWeb (2017. 5. 9 오후 10:27:15)

:compileJava
:processResources
:classes
:run5월 09, 2017 10:27:15 오후 org.springframework.beans.factory.xml.XmlBeanDefinitionReader loadBeanDefinitions
정보: Loading XML bean definitions from class path resource [applicationContext.xml]
5월 09, 2017 10:27:15 오후 org.springframework.context.support.GenericXmlApplicationContext prepareRefresh
정보: Refreshing org.springframework.context.support.GenericXmlApplicationContext@378fd1ac: startup date [Tue May
Key: 인제대, Value: 김해시 어방동
Key: 박경태, Value: 부산시 동대신동
5월 09, 2017 10:27:15 오후 org.springframework.context.support.GenericXmlApplicationContext doClose
정보: Closing org.springframework.context.support.GenericXmlApplicationContext@378fd1ac: startup date [Tue May 09

BUILD SUCCESSFUL

Total time: 0.675 secs
```