

SPRING

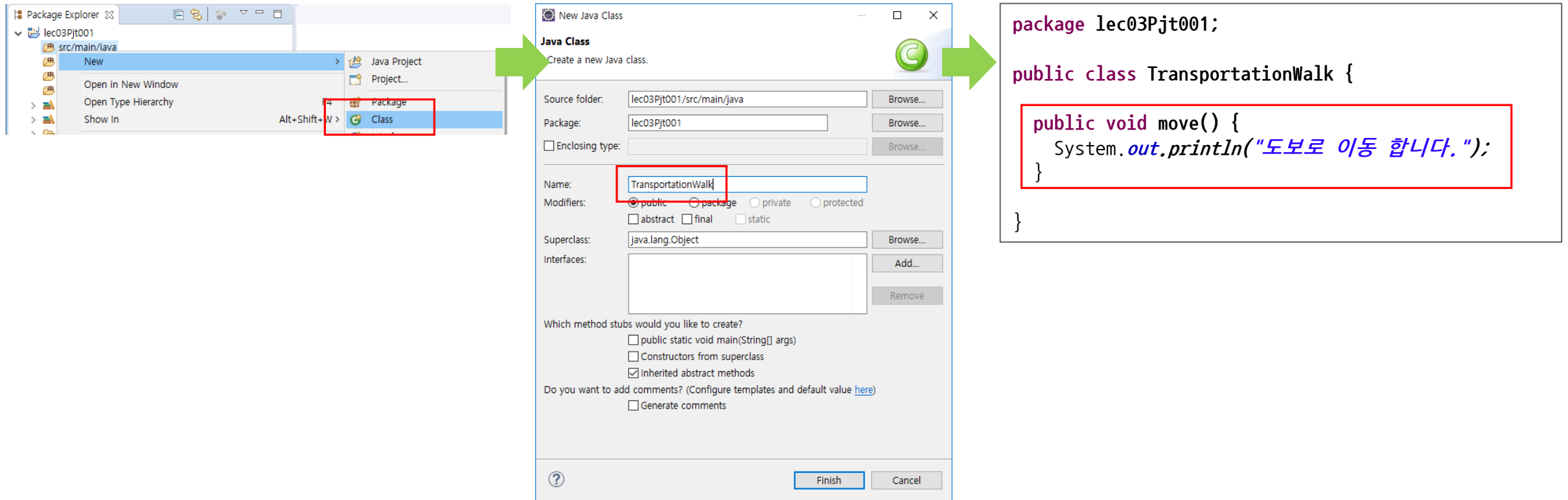
4강_처음해 보는 스프링 프로젝트

스프링을 이용해서 간단한 프로그램을 만들어 본다.

4-1 Java파일을 이용한 프로젝트 실행

4-2 우선 따라 해 보는 스프링 프로젝트

4-1 : Java파일을 이용한 프로젝트 실행



The image illustrates the process of creating a new Java class in an Eclipse IDE project. It shows the Package Explorer on the left, the 'New Java Class' dialog in the center, and the resulting Java code on the right.

Package Explorer: The project 'lec03Pjt001' is expanded, showing the 'src/main/java' package. The 'New' menu is open, and the 'Class' option is selected.

New Java Class Dialog: The dialog is titled 'New Java Class'. The 'Source folder' is 'lec03Pjt001/src/main/java', and the 'Package' is 'lec03Pjt001'. The 'Name' field is 'TransportationWalk'. The 'Modifiers' section has 'public' selected. The 'Superclass' is '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.

Java Code: The resulting code is as follows:

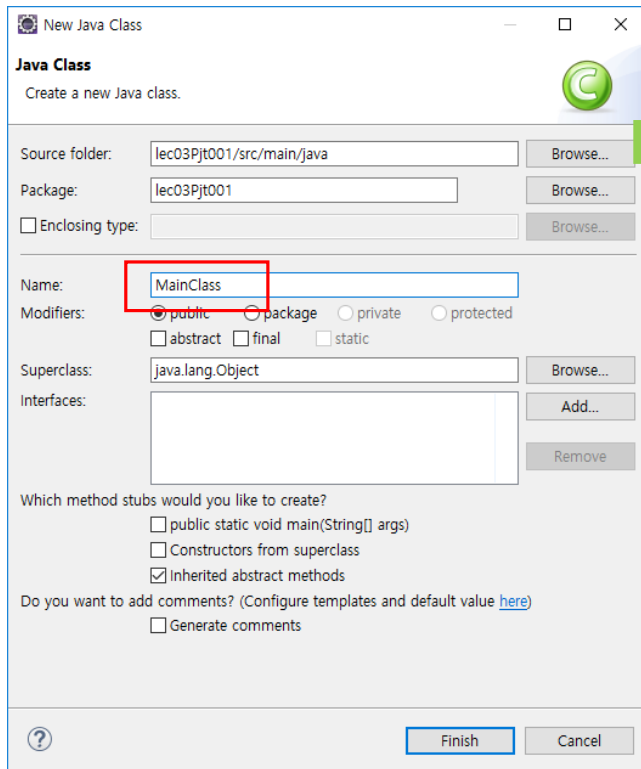
```
package lec03Pjt001;

public class TransportationWalk {

    public void move() {
        System.out.println("도보로 이동 합니다.");
    }

}
```

4-1 : Java파일을 이용한 프로젝트 실행



```
package lec03Pjt001;
```

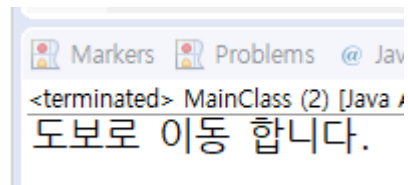
```
public class MainClass {
```

```
    public static void main(String[] args) {
```

```
        TransportationWalk transportationWalk = new TransportationWalk();  
        transportationWalk.move();
```

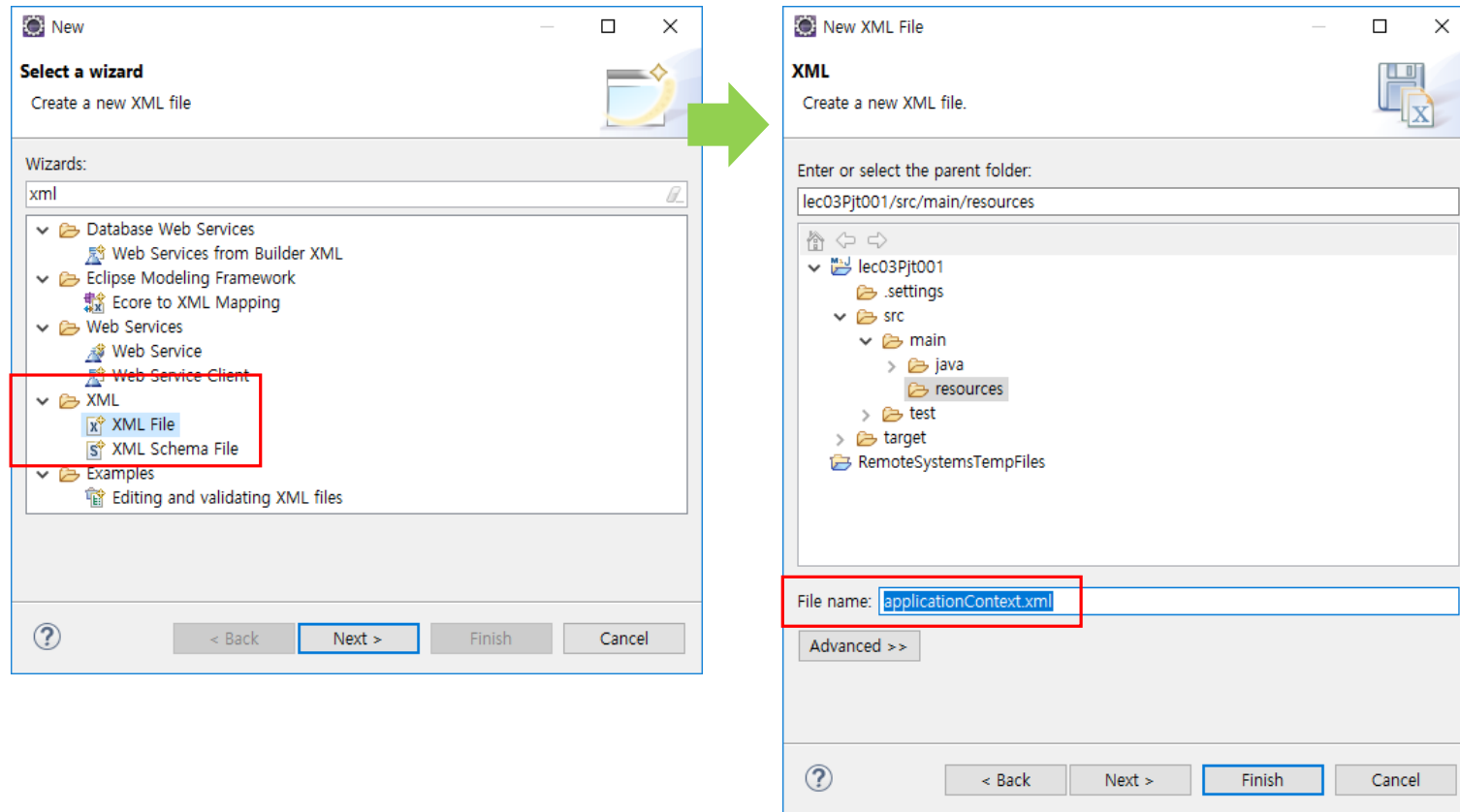
```
    }
```

```
}
```



4-2 : 우선 따라 해 보는 스프링 프로젝트

스프링 방식의 ‘의존’을 이용하기 위해서는 Main에서 TransportationWalk 객체를 직접 생성하지 않고, 스프링 설정 파일(XML)을 이용해 보기로 한다. 가장 큰 차이점은 Java 파일에서 이용한 new 연산자를 이용하지 않고 스프링 설정파일(XML)을 이용하는 것이다.



4-2 : 우선 따라 해 보는 스프링 프로젝트

applicationContext.xml

```
<?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="tWalk" class="lec03Pjt001.TransportationWalk" />

</beans>
```

개념부터 직접 경험해볼 것



MainClass.java

```
package lec03Pjt001;

import org.springframework.context.support.GenericXmlApplicationContext;

public class MainClass {

    public static void main(String[] args) {

        //TransportationWalk transportationWalk = new TransportationWalk();
        //transportationWalk.move();

        GenericXmlApplicationContext ctx =
            new GenericXmlApplicationContext("classpath:applicationContext.xml");

        TransportationWalk transportationWalk = ctx.getBean("tWalk",
            TransportationWalk.class);

        transportationWalk.move();

        ctx.close();

    }
}
```

applicationContext.xml의 리소스

경로 지정 (classpath:applicationContext.xml)

bean id

클래스명

시작할 외부 리소스 반환

```
<terminated> MainClass (2) [Java Application] C:\Program Files\Java\jre1.8.0_151\bin\javaw.exe (2018. 1. 29. 오후 4:41:18)
1월 29, 2018 4:41:19 오후 org.springframework.beans.factory.xml.XmlBeanDefinitionF
정보: Loading XML bean definitions from class path resource [applicationContext.xml]
1월 29, 2018 4:41:19 오후 org.springframework.context.support.GenericXmlApplicatio
정보: Refreshing org.springframework.context.support.GenericXmlApplicationContext@
도보로 이동 합니다.
1월 29, 2018 4:41:19 오후 org.springframework.context.support.GenericXmlApplicatio
정보: Closing org.springframework.context.support.GenericXmlApplicationContext@496
```