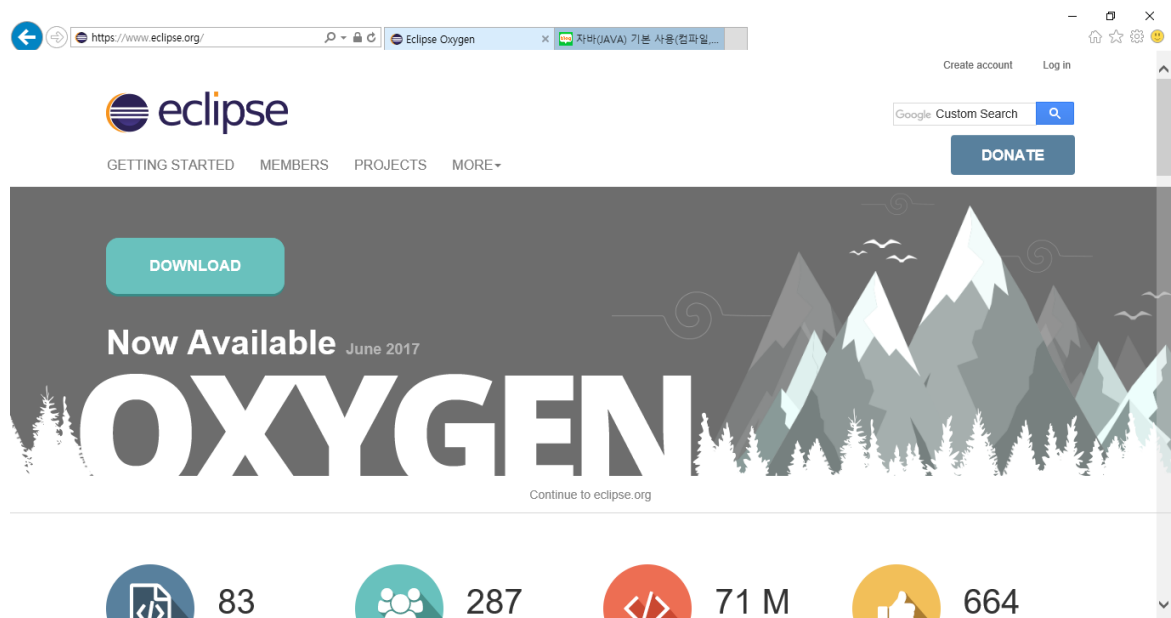


## 0. 설치 전에

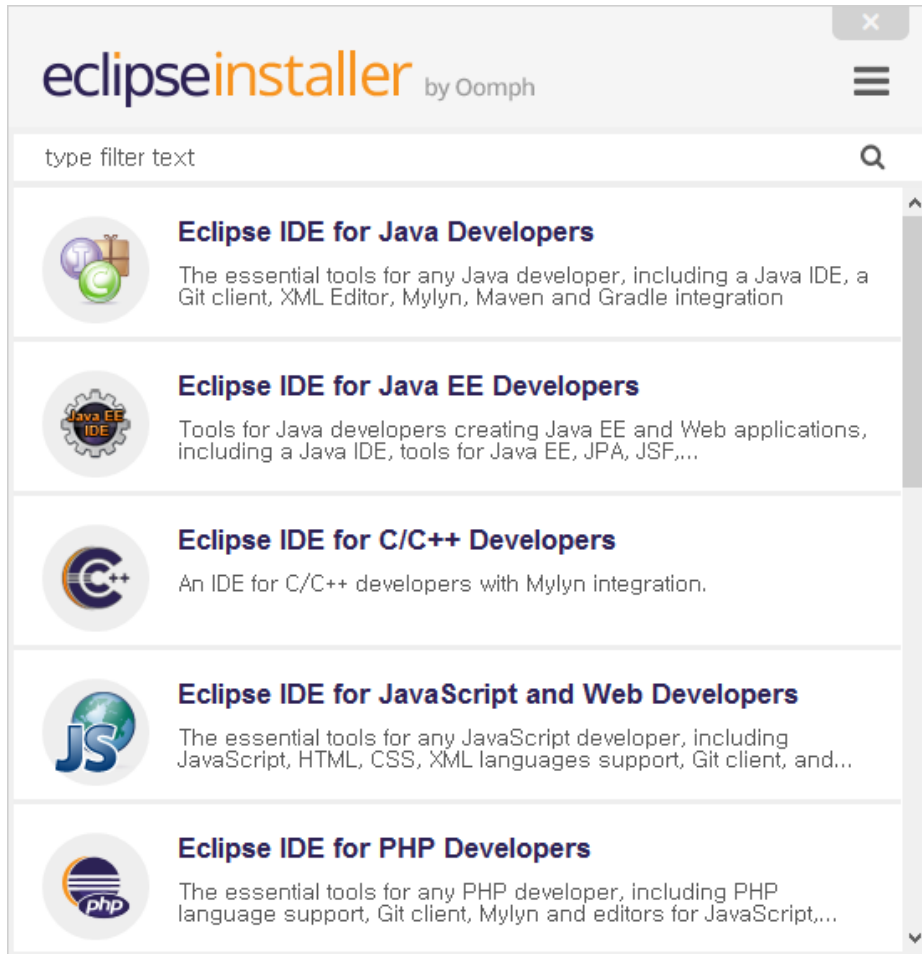
JDK 설치가 완료된 상황에서 아래의 과정을 진행하시기 바랍니다.

## 1. 설치

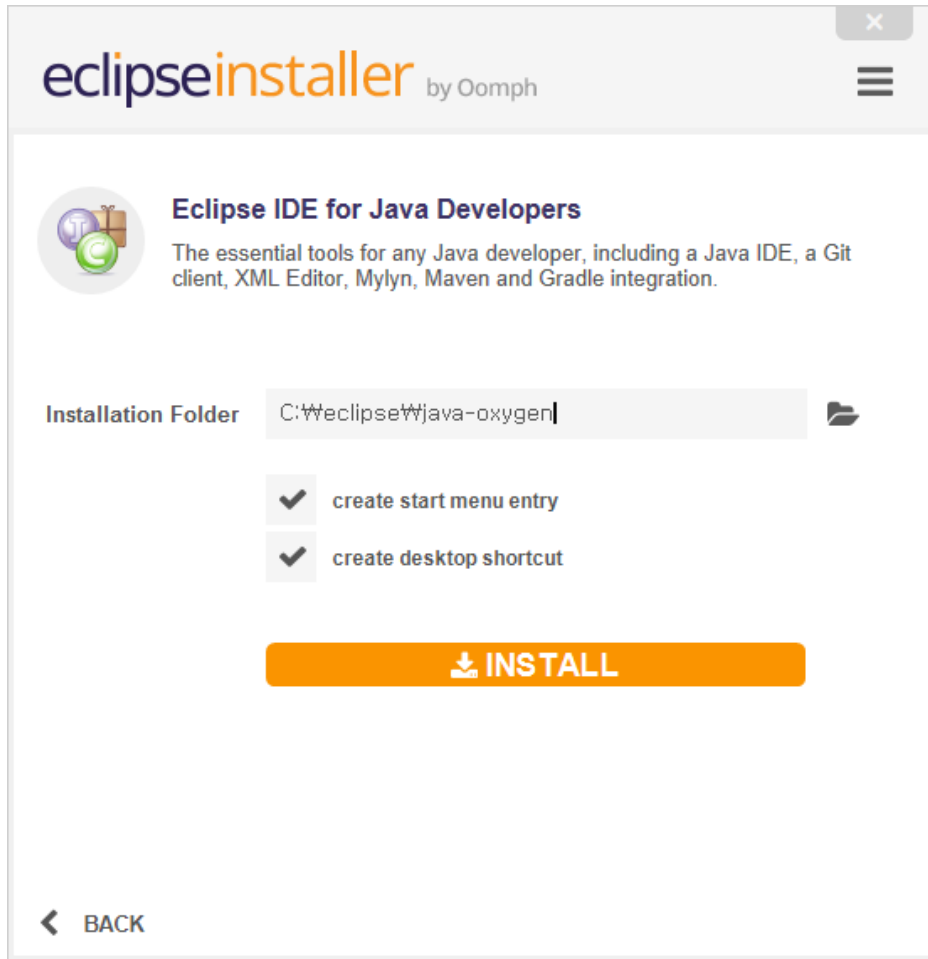
[www.eclipse.org](http://www.eclipse.org)에 접속을 합니다. 그리고 페이지에 바로 보이는 DOWNLOAD를 눌러서 다운로드를 합니다.



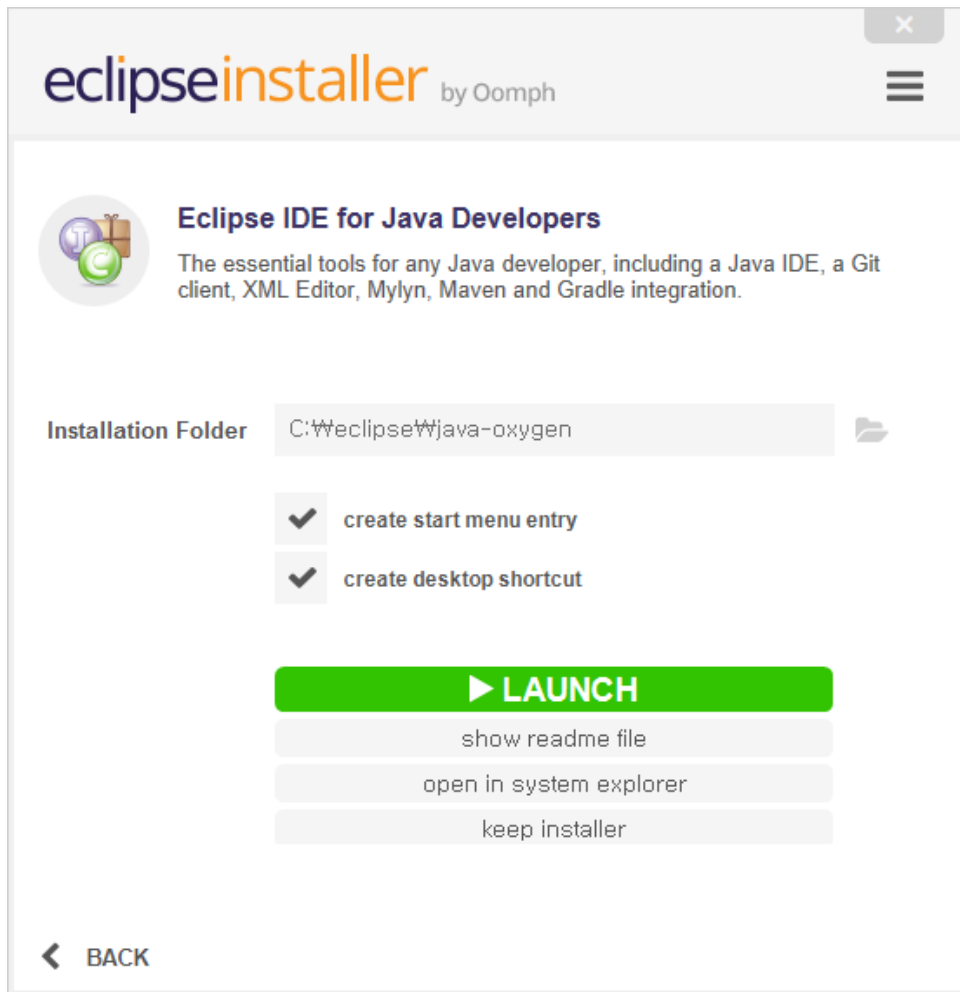
다운로드의 방식 및 경로는 조금씩 차이를 보이게 됩니다. 그러나 실제 다운로드를 하는 데는 전혀 어려움이 없습니다. 이어서 다운로드 한 설치파일을 실행하면 다음 내용을 보게 됩니다.



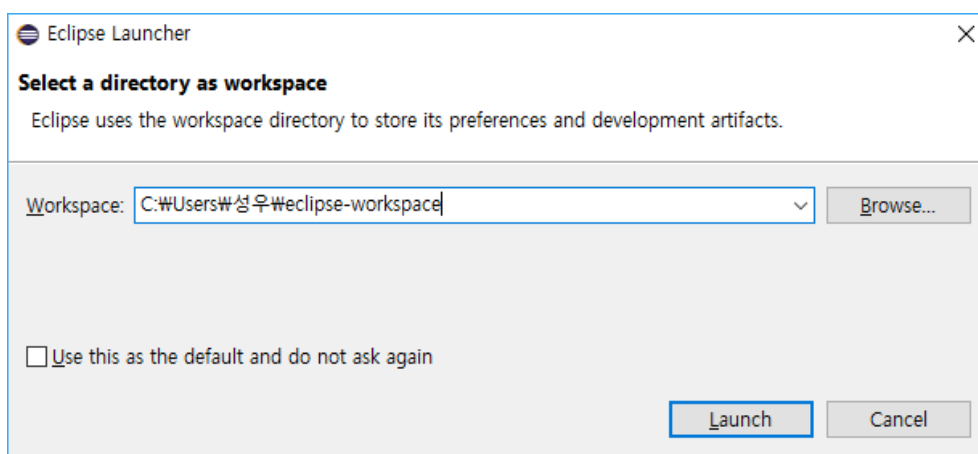
여러가지 버전 및 형태의 이클립스가 소개가 되는데요. 자바 기본을 공부하는 수준에서는 첫 번째로 소개되는 것을 선택하고 다운로드 하면 됩니다. 그럼 Eclipse IDE for Java Developers를 선택합니다.



위에서는 설치 경로를 묻고 있습니다. 그리고 스타트 메뉴에 추가를 할지 바로가기 만들지를 묻고 있습니다. 이렇게 해서 설치 경로를 설정했다면 이어서 INSTALL 버튼을 눌러서 설치를 진행합니다. 이왕이면 설치 경로 정도는 기억해 두는 것이 좋습니다.



설치가 끝이 나면 위의 페이지를 볼 수 있습니다. 그리고 이어서 LAUNCH를 선택하면 다음과 같이 무엇인가를 물어봅니다.



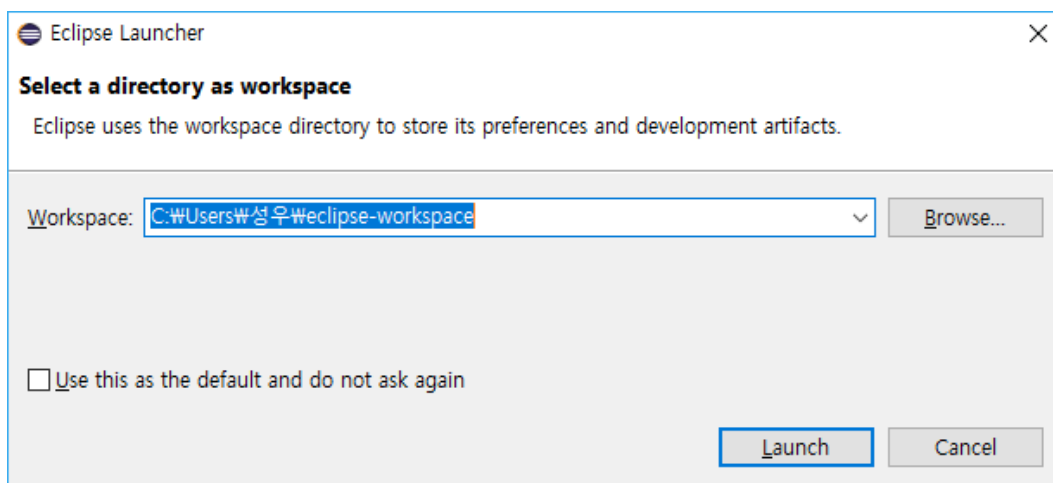
이는 당장 코딩을 하기 위한 정보를 묻는 것인데요 일단 설치가 끝이 났으니 여기서는 Cancel을 선택하여 종료를 하고 설치를 마무리 하기로 합니다.

## 2. 새로운 파일을 생성해서 Hello world 출력하기

이클립스를 설치할 때 바로가기도 만들었으니 이를 통해서 이클립스를 실행할 수도 있습니다. 그러나 실행파일을 직접 선택해서 실행을 한 번 해보도록 하겠습니다. 이클립스가 설치된 경로로 가보세요. 그러면 다음 파일 구성을 볼 수 있습니다.

configuration	2017-07-03 오후...	파일 폴더
dropins	2017-07-03 오후...	파일 폴더
plugins	2017-07-03 오후...	파일 폴더
readme	2017-07-03 오후...	파일 폴더
.eclipseproduct	2016-06-29 오전...	ECLIPSEPRODUCT...
eclipse.exe	2017-06-12 오후...	응용 프로그램
eclipse.ini	2017-07-03 오후...	구성 설정
eclipsesec.exe	2017-06-12 오후...	응용 프로그램

이중에서 eclipse.exe를 실행하면 이클립스가 실행이 됩니다. 한번 선택을 해보죠. 그러면 다음과 같이 설치 마지막 단계에서 보던 페이지를 다시 볼 수 있습니다.



영문 내용을 조금 읽어보면 '워크스페이스(Workspace)'라는 공간으로 활용할 경로를 묻는 것임을 알 수 있습니다. 워크스페이스에 대해서 간단히 설명하면 이렇습니다. 코딩을 편히 하기 위해서는 일종의 '관리'가 필요한데, 이 관리를 이클립스가 상당부분 대신해주게 됩니다. 물론 관리를 위해

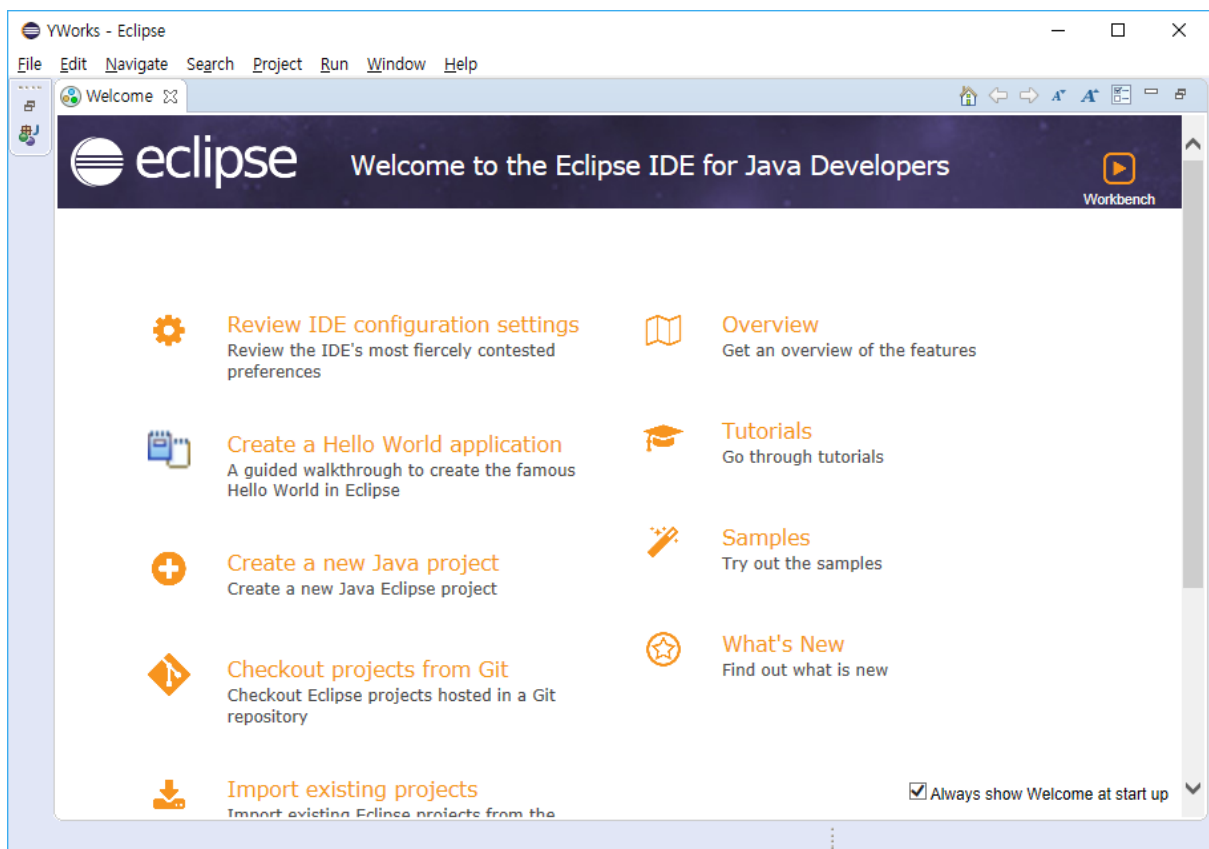
서는 기록해 두어야 할 다양한 정보들이 있겠지요. 바로 그 정보들을 묶어 두기 위한 공간이 워크스페이스(Workspace)입니다. 그러니까 위에서는 다음과 같은 질문을 하는 것으로 볼 수 있지요?

지금 진행할 작업들의 정보를 어느 디렉토리에 두고 관리를 하겠냐?

지금은 이 정도만 이해해도 충분합니다. 이쯤 이제 Launch를 누를 차례인데요. 그 전에 위의 워크스페이스 경로도 가까운 곳에 두기를 바랍니다. 위의 화면에는 경로에 한글이 들어가 있는데요. 사실 경로에 한글이 들어가는 것은 여러 이유로 좋지 않습니다. 경로 인식에 대한 문제가 발생할 수도 있고요. 따라서 다음과 같이 저도 경로를 수정하였습니다.

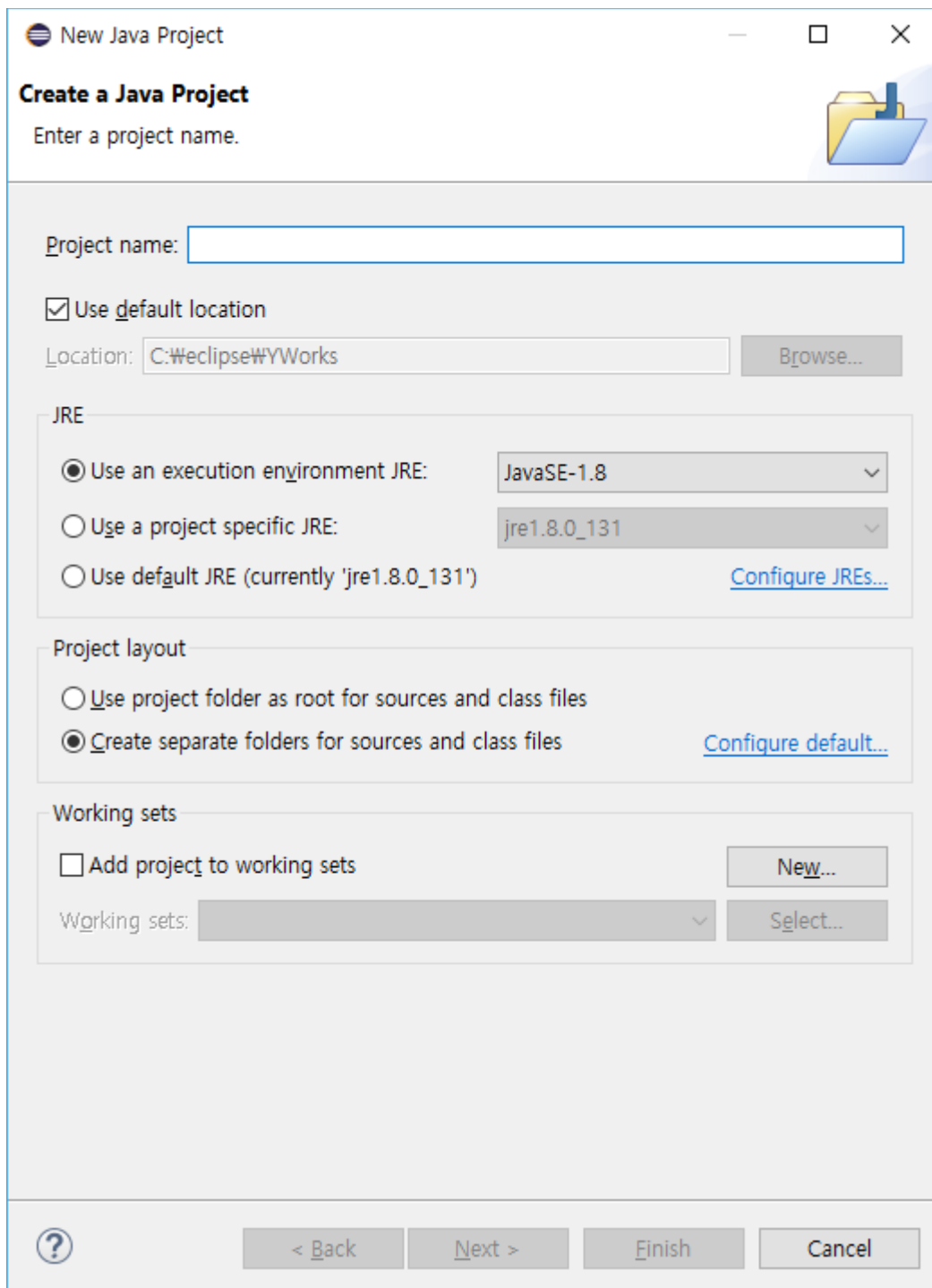
C:\eclipse\YWorks

이렇게 경로 설정까지 마무리 하고 Launch를 누르면 다음 페이지를 보게 됩니다. 그리고 이것이 실제 이클립스 실행 페이지라 할 수 있지요.



앞서 우리는 '워크스페이스'라는 것을 만들었지요? 사실 이 워크스페이스 안에서 다수의 프로젝트

를 생성해서 관리할 수 있습니다. 그런데 지금 우리가 다수의 프로젝트를 진행할 일이 없으니 이러한 관리 방식이 조금 부담스러울 수 있습니다. 그러나 이클립스는 실제 프로그램 개발용 소프트웨어이니 이러한 부분은 감수해야하겠지요. 어쨌든 예제 하나를 실행하기 위해서라도 우리는 앞서 만든 워크 스페이스 안에서 다시 프로젝트라는 것을 생성해야 합니다. 그럼 위의 페이지에서 무엇을 선택해야 할 지 조금 보이시죠? 그렇습니다. Create a new Java Project를 선택하는 겁니다. 그럼 다음 페이지가 뜹니다.



The screenshot shows the 'New Java Project' dialog box in the Eclipse IDE. The window title is 'New Java Project'. Below the title bar, there's a section 'Create a Java Project' with a folder icon and the instruction 'Enter a project name.'.

The main area contains several sections:

- Project name:** A text input field.
- Use default location:** A checked checkbox. Below it, the 'Location' is set to 'C:\eclipse\YWorks' with a 'Browse...' button.
- JRE:** A section with three radio buttons:
  - ☒ Use an execution environment JRE: A dropdown menu showing 'JavaSE-1.8'.
  - ☐ Use a project specific JRE: A dropdown menu showing 'jre1.8.0\_131'.
  - ☐ Use default JRE (currently 'jre1.8.0\_131') with a link to 'Configure JREs...'.
- Project layout:** A section with two radio buttons:
  - ☐ Use project folder as root for sources and class files.
  - ☒ Create separate folders for sources and class files with a link to 'Configure default...'.
- Working sets:** A section with a checkbox 'Add project to working sets' and a 'New...' button. Below it, a 'Working sets:' dropdown menu and a 'Select...' button.

At the bottom, there's a navigation bar with a help icon (?), '< Back', 'Next >', 'Finish', and 'Cancel' buttons.

뭐 많은 것을 묻는 것 같지만 워크스페이스 안에 생성할 프로젝트의 이름과 그 프로젝트를 관리할 디렉토리를 묻는 겁니다. 따라서 다음과 같이 프로젝트에 대한 경로 정보를 입력하고 JDK를 설치할 때 설치된 JRE를 기반으로 실행 결과를 확인하기 위해서 다음과 같이 입력하고 선택합니다. (프로젝트 이름을 입력하면 경로 정보에도 이 내용이 함께 반영이 됩니다.)

**New Java Project**

**Create a Java Project**

Create a Java project in the workspace or in an external location.

**Project name:** FirstPrj

☒ Use default location

**Location:** C:\eclipse\workspace\FirstPrj [Browse...](#)

**JRE**

☐ Use an execution environment JRE: JavaSE-1.8

☐ Use a project specific JRE: jre1.8.0\_131

☒ Use default JRE (currently 'jre1.8.0\_131') [Configure JREs...](#)

**Project layout**

☐ Use project folder as root for sources and class files

☒ Create separate folders for sources and class files [Configure default...](#)

**Working sets**

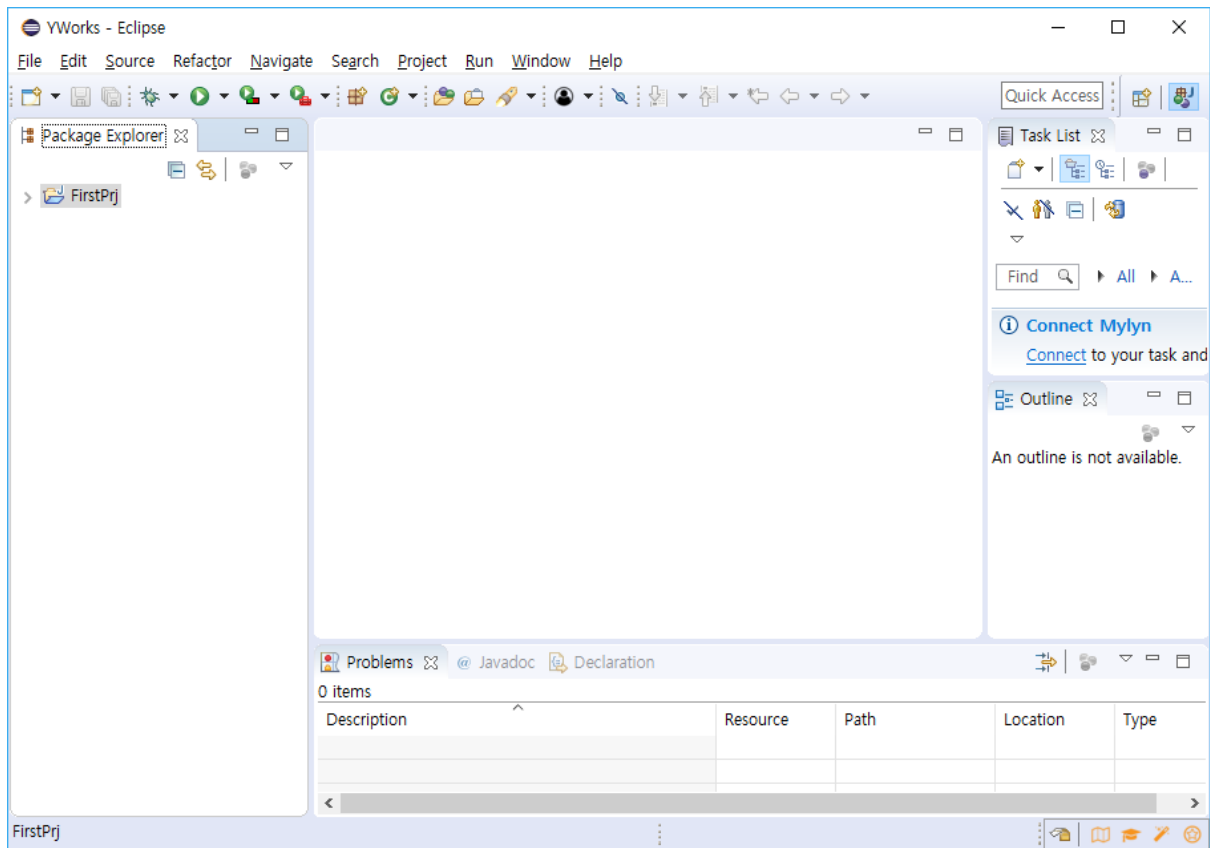
☐ Add project to working sets [New...](#)

**Working sets:** [Select...](#)

[?](#) [< Back](#) [Next >](#) **Finish** [Cancel](#)



이제 마지막으로 Finish를 누릅니다. 그러면 다음과 같이 실제 코딩을 할 수 있는 바탕이 만들어 집니다.



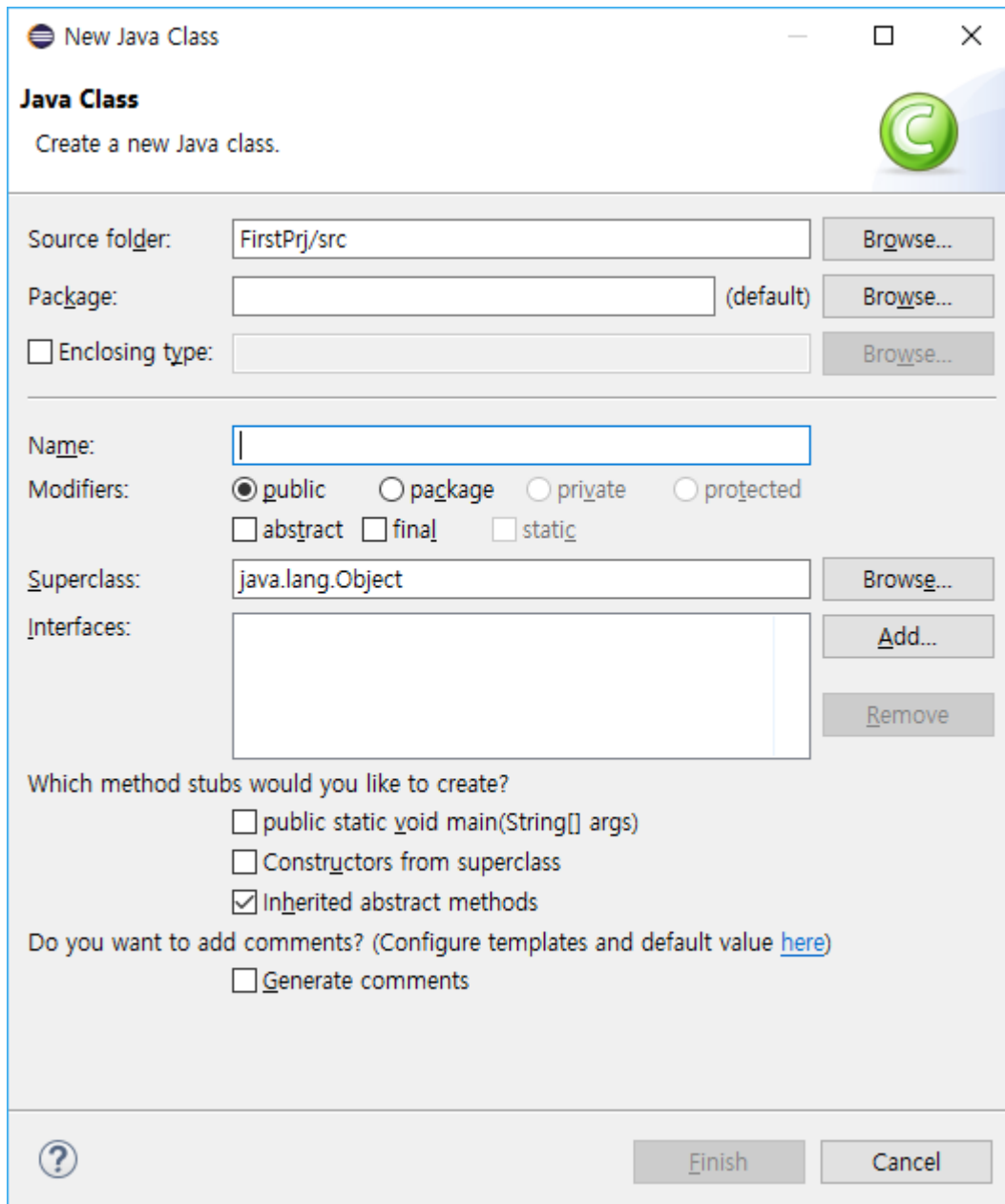
자! 그럼 Hello world를 출력하기 위한 다음 코드의 입력을 진행해보겠습니다.

```
class HelloWorld {  
  
    public static void main(String[] args) {  
  
        System.out.println("Hello world");  
  
    }  
  
}
```

위의 코드에서 보이듯이 클래스를 정의하는 것이 목적입니다. 그래서 다음과 같이 메뉴에서 시작해서 class를 선택합니다.

File ► new ► class

그러면 다음과 같이 생성하고자 하는 클래스의 정보를 입력하는 페이지가 뜹니다.



The image shows a 'New Java Class' dialog box. At the top, it says 'Java Class' and 'Create a new Java class.' with a green 'C' icon. The 'Source folder' is 'FirstPrj/src' with a 'Browse...' button. The 'Package' is '(default)' with a 'Browse...' button. There is an unchecked checkbox for 'Enclosing type' with a 'Browse...' button. The 'Name' field is empty. Under 'Modifiers', 'public' is selected with a radio button, and 'abstract', 'final', and 'static' are unchecked with checkboxes. The 'Superclass' is 'java.lang.Object' with a 'Browse...' button. The 'Interfaces' section is empty with an 'Add...' button and a 'Remove' button. Below this, it asks 'Which method stubs would you like to create?' with three options: 'public static void main(String[] args)' (unchecked), 'Constructors from superclass' (unchecked), and 'Inherited abstract methods' (checked). It then asks 'Do you want to add comments? (Configure templates and default value [here](#))' with a 'Generate comments' checkbox (unchecked). At the bottom, there is a help icon, 'Finish', and 'Cancel' buttons.

위의 내용은 공부한만큼 보입니다. 책의 중반 이후를 달리고 있다면 대부분의 내용이 이해가 되기 때문에 원하는 옵션을 직접 선택할 수 있습니다. 그러나 Hello world 출력이 목적이므로 다음 두 가지 사항에만 주목을 합니다.

“main 메소드가 있어야 한다.”

“클래스 이름이 HelloWorld이다.”

그래서 이 둘을 반영하여 다음과 같이 입력하고 선택을 합니다. (이클립스가 친절하게도, 디폴트 패키지 사용하는 건 좀 그렇다고 경고 메시지를 살짝 던져 주지만 지금은 그냥 넘어갑니다. 패키지를 배우지 않은 상태일 수도 있으니까요. 그리고 공부할 때는 디폴트 패키지도 괜찮습니다. ^^)

**New Java Class**

**Java Class**

⚠ The use of the default package is discouraged.

Source folder: FirstPrj/src Browse...

Package: (default) Browse...

☐ Enclosing type: Browse...

Name: HelloWorld

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

Superclass: java.lang.Object Browse...

Interfaces: Add...  
Remove

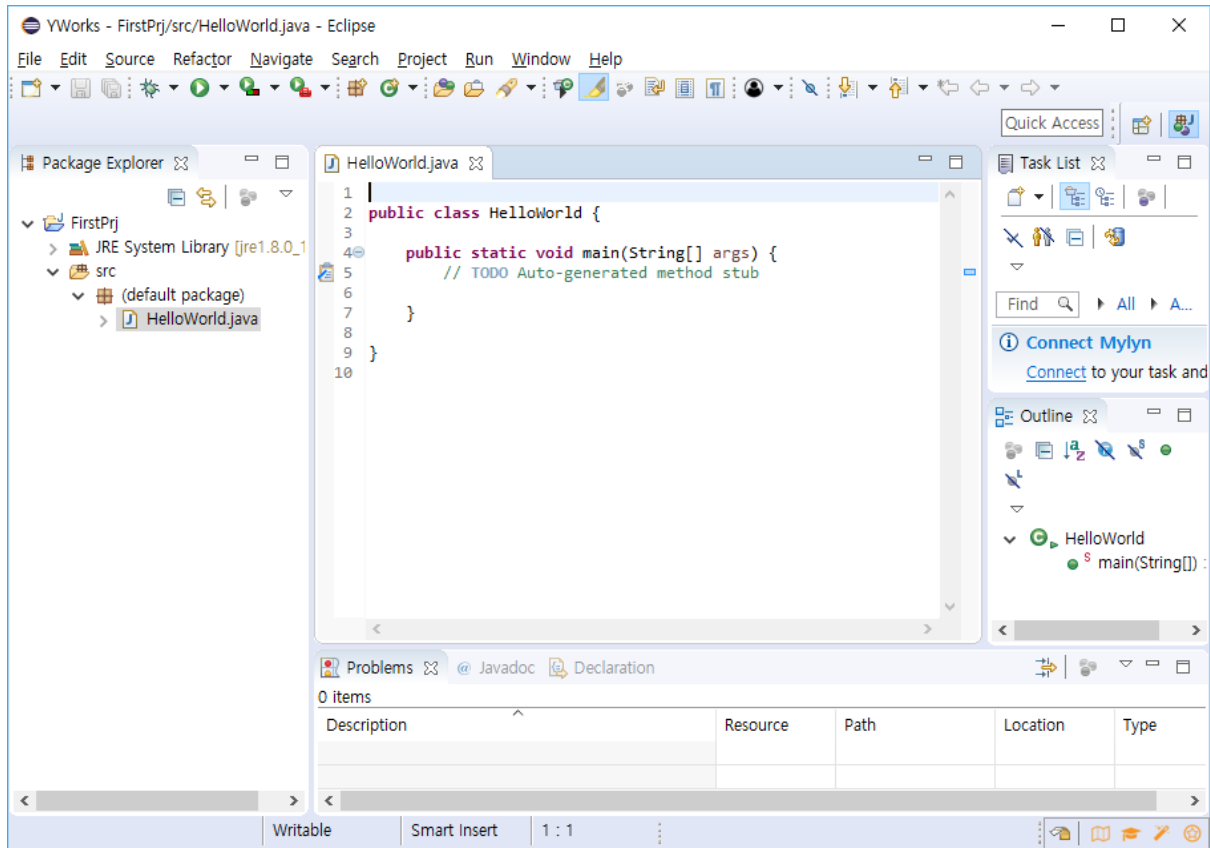
Which method stubs would you like to create?

☒ 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

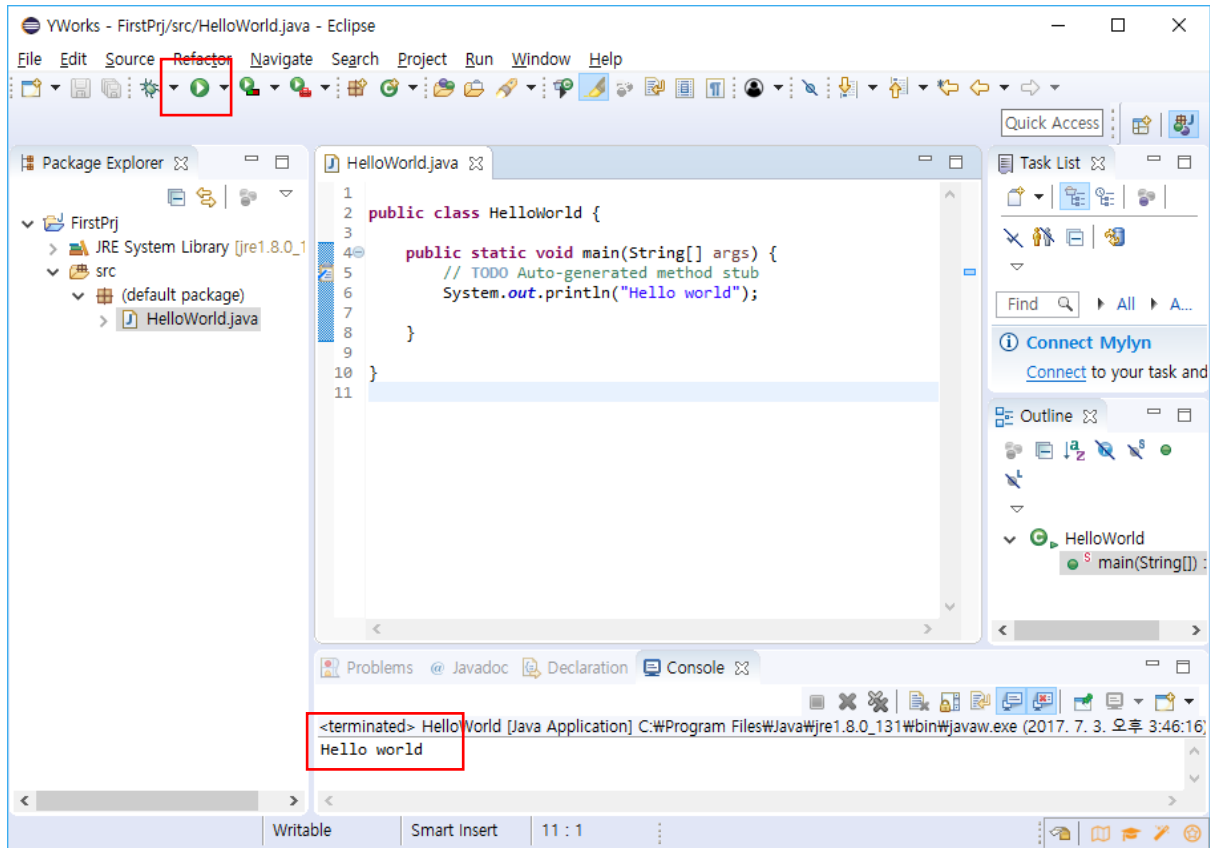
이제 Finish를 선택합니다. 그러면 다음과 같이 생각보다 많은 것을 대신 입력해줍니다. 클래스 이름만 입력했을 뿐인데 소스파일도 생성해줍니다. 어디에 생성되었는지 꼭 찾아보세요. 예상이 되겠지만 그래도 직접 찾아보세요. 앞서 워크스페이스 안에 프로젝트 디렉토리를 생성했지요? 그 안에 src라는 디렉토리가 있는데 그 안을 보시면 됩니다.



이제 main 메소드 안에 다음 문장을 직접 넣어 볼까요? 자동으로 들여쓰기가 되는 것도 느껴 보면서 말이죠.

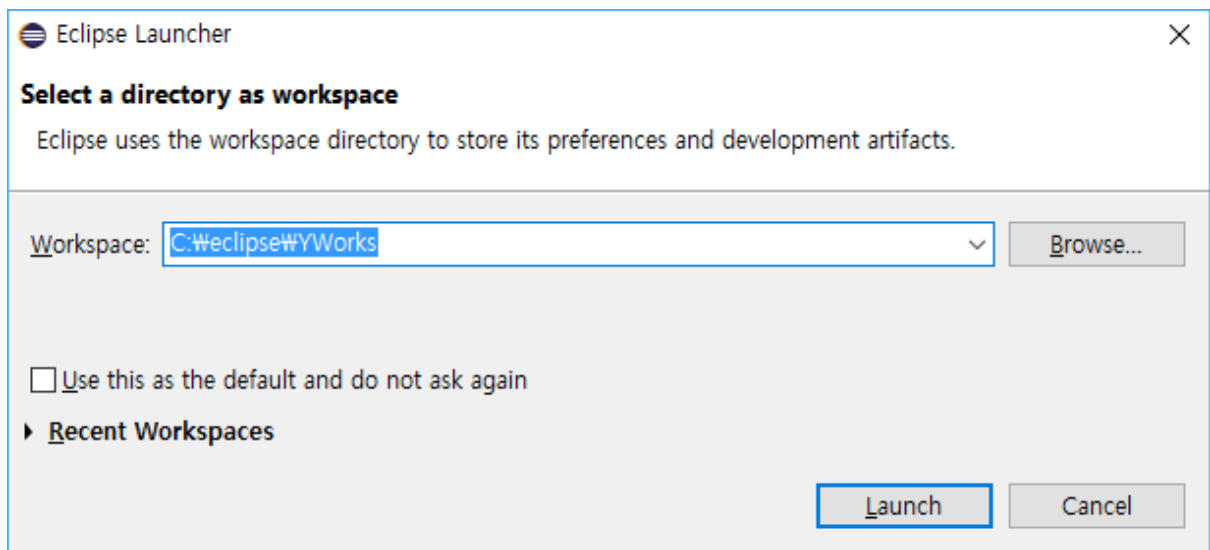
```
System.out.println("Hello world");
```

끝으로 입력이 끝났으면 ▶ 기호가 새겨진 것을 누릅니다. 그러면 다음과 같이 실행 결과를 확인할 수 있습니다.

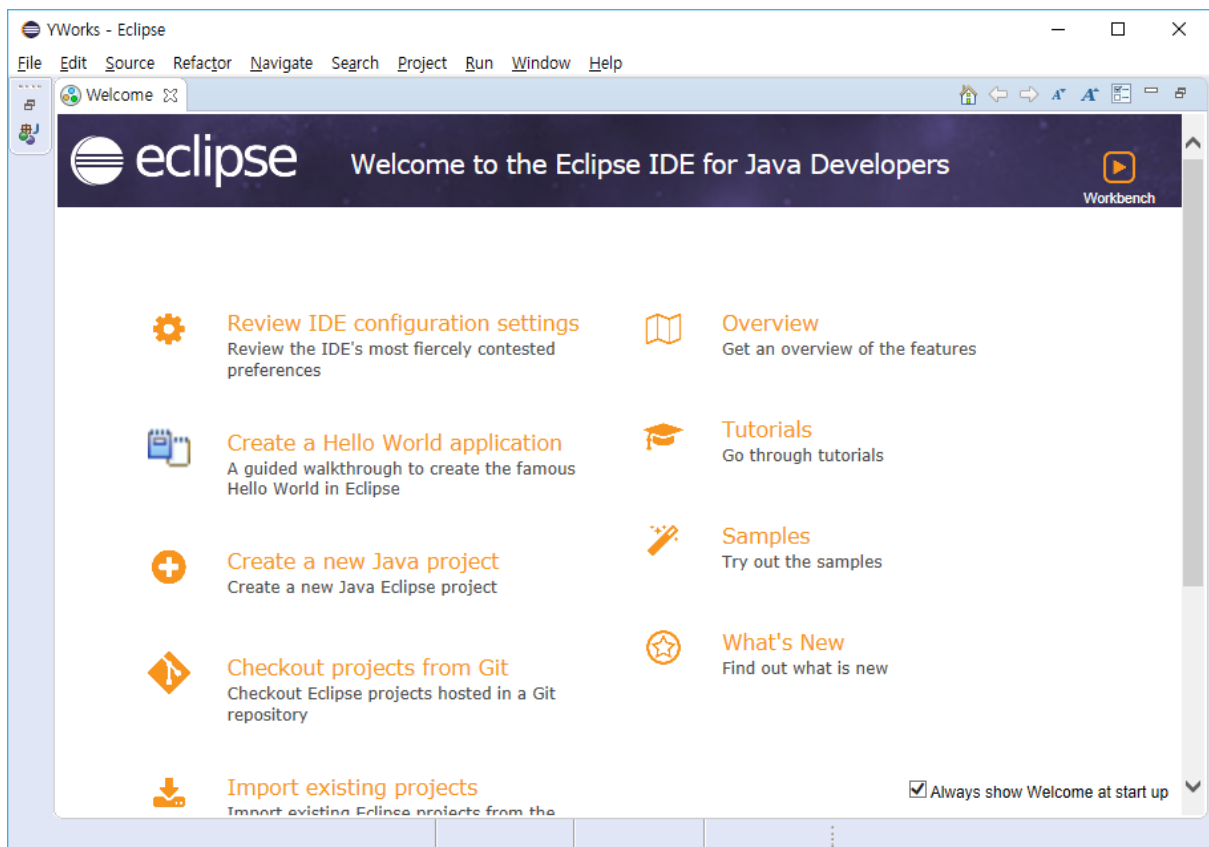


### 3. 책에서 제공하는 파일을 프로젝트에 포함해서 Hello World 출력하기

이번에는 책에서 제공하는 소스파일을 실행해보려고 합니다. 일단 이클립스를 종료 후에 다시 한번 실행해 볼까요?



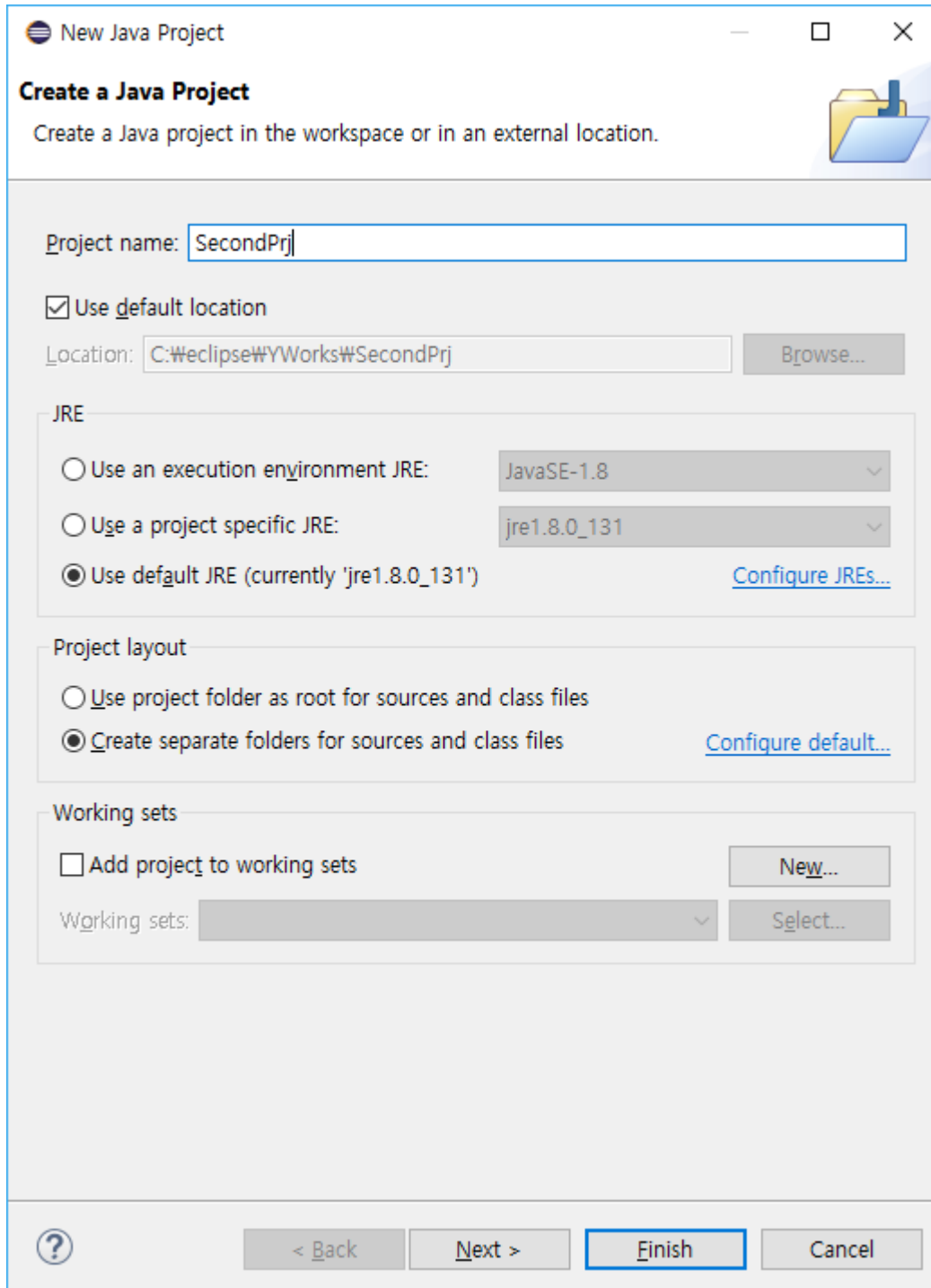
네 워크스페이스 어떻게 하겠냐고 또 물어보죠? 근데 워크스페이스를 계속해서 만들 필요는 없습니다. 그 안에 새로운 프로젝트를 다수 추가할 수 있으니까요. 따라서 그냥 Launch를 누릅니다. 그러면 앞서 한 차례 만들었던 워크스페이스 내에서 새로운 프로젝트를 생성할 수 있게 됩니다. 그리고 이를 위해 다음 페이지를 보게 되지요.



물론 앞서 만든 프로젝트에서 기존에 생성한 소스파일을 지우고 새 소스파일을 추가하는 것도 가능합니다. 그런데 여기서는 새 프로젝트를 하나 더 만들어 보도록 하겠습니다. 하나의 워크스페이스 안에 다수의 프로젝트를 생성할 수 있음을 보이기 위해서 말이죠. 그럼 위의 페이지에서 다음을 선택합니다.

Create a new Java project

그리고 다음과 같이 입력을 합니다. 물론 프로젝트 이름은 원하는 대로 붙여주면 됩니다.



**New Java Project**

**Create a Java Project**  
Create a Java project in the workspace or in an external location.

Project name:

☒ Use default location

Location:

**JRE**

☐ Use an execution environment JRE:

☐ Use a project specific JRE:

☒ Use default JRE (currently 'jre1.8.0\_131') [Configure JREs...](#)

**Project layout**

☐ Use project folder as root for sources and class files

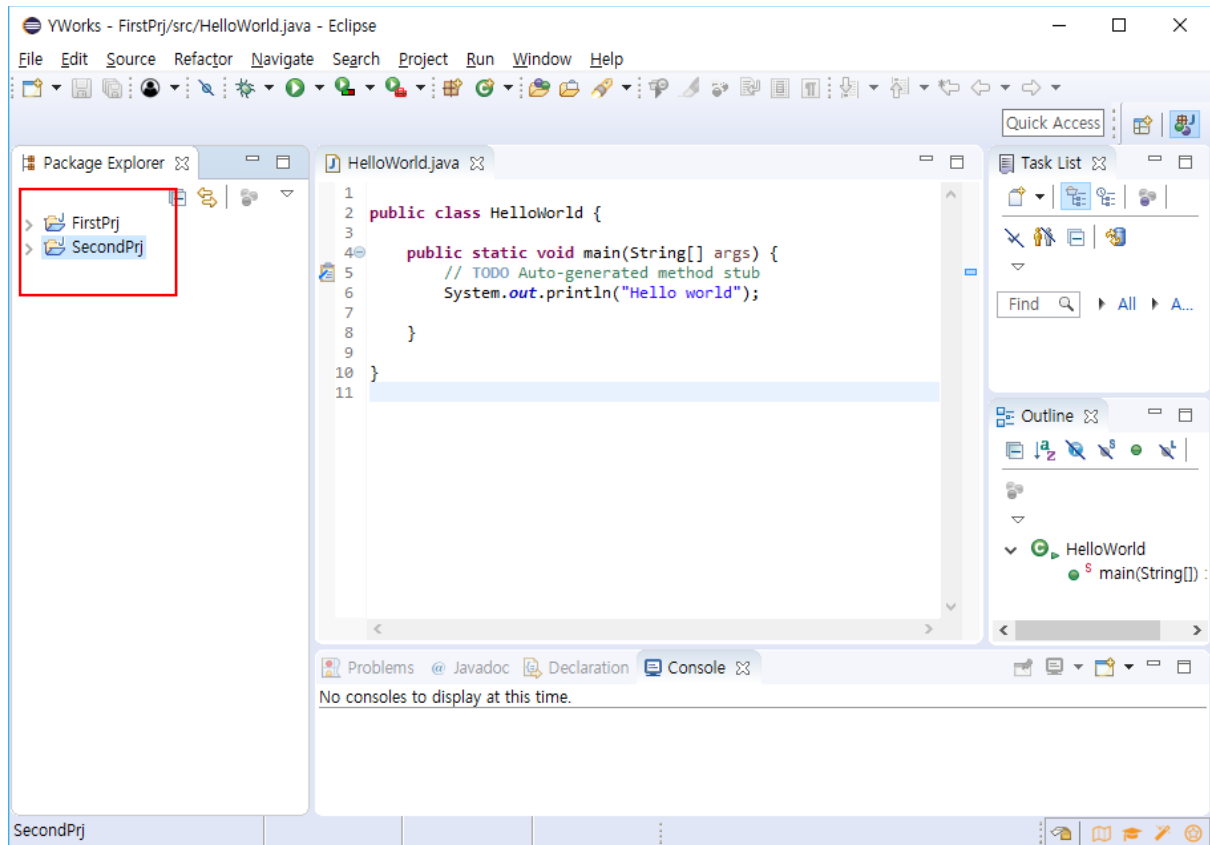
☒ Create separate folders for sources and class files [Configure default...](#)

**Working sets**

☐ Add project to working sets

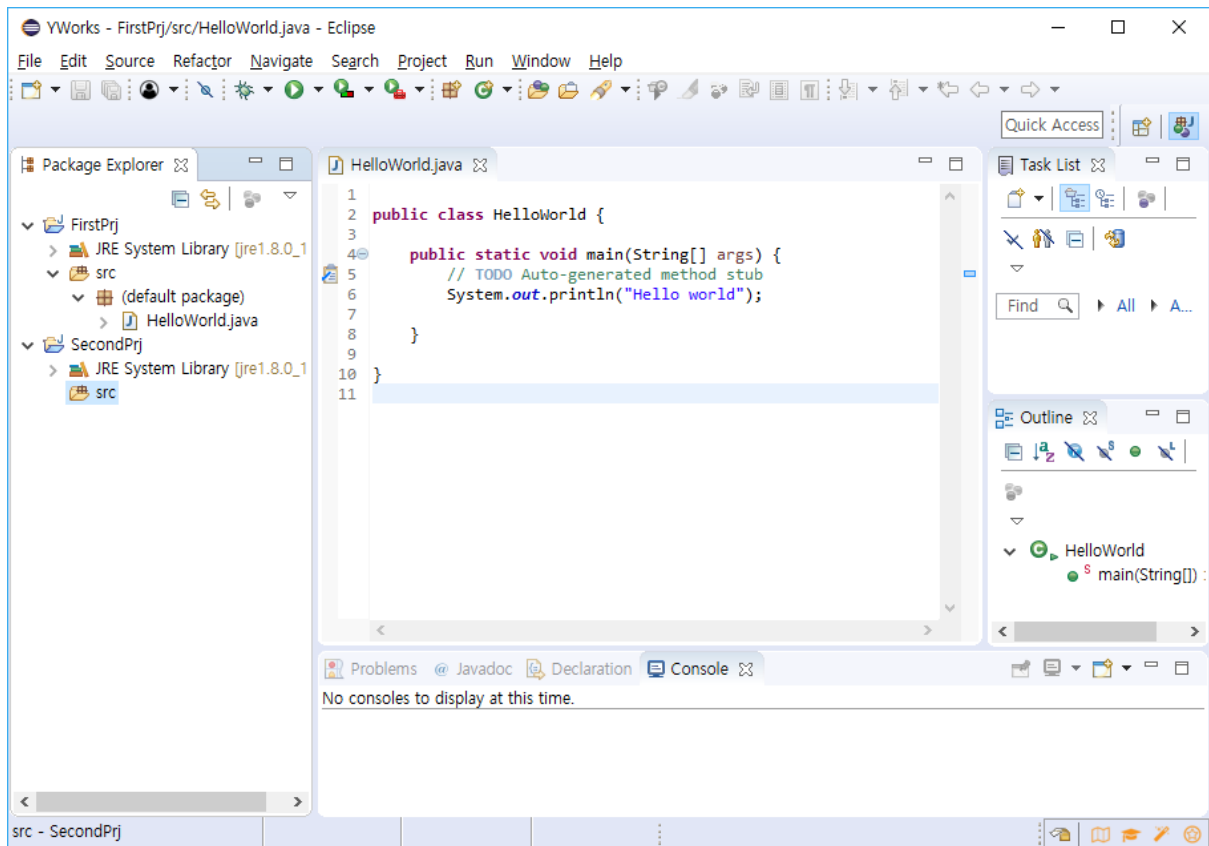
Working sets:

그리고 Finish를 눌러 보죠. 그러면 하나의 워크스페이스 안에 두 개의 프로젝트로 존재하는 것을 눈으로 확인할 수 있습니다. 다음과 같이 말이죠.

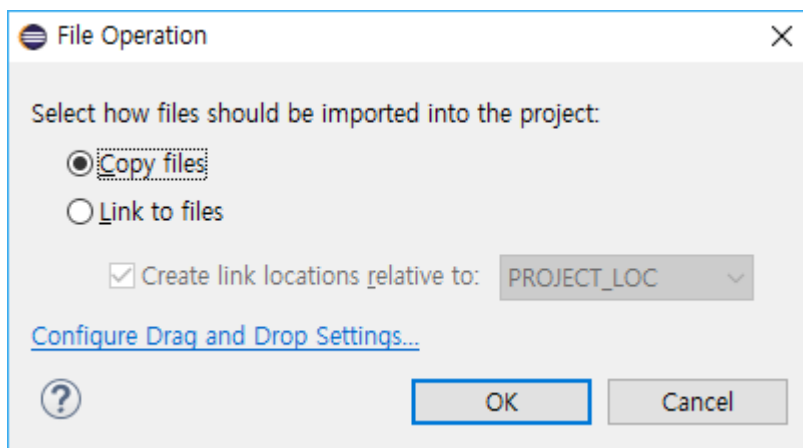




왼편의 프로젝트 정보를 클릭해 보면 다음과 같이 해당 프로젝트별 파일 구성도 볼 수 있습니다. 물론 SecondPrj는 비어 있겠죠.

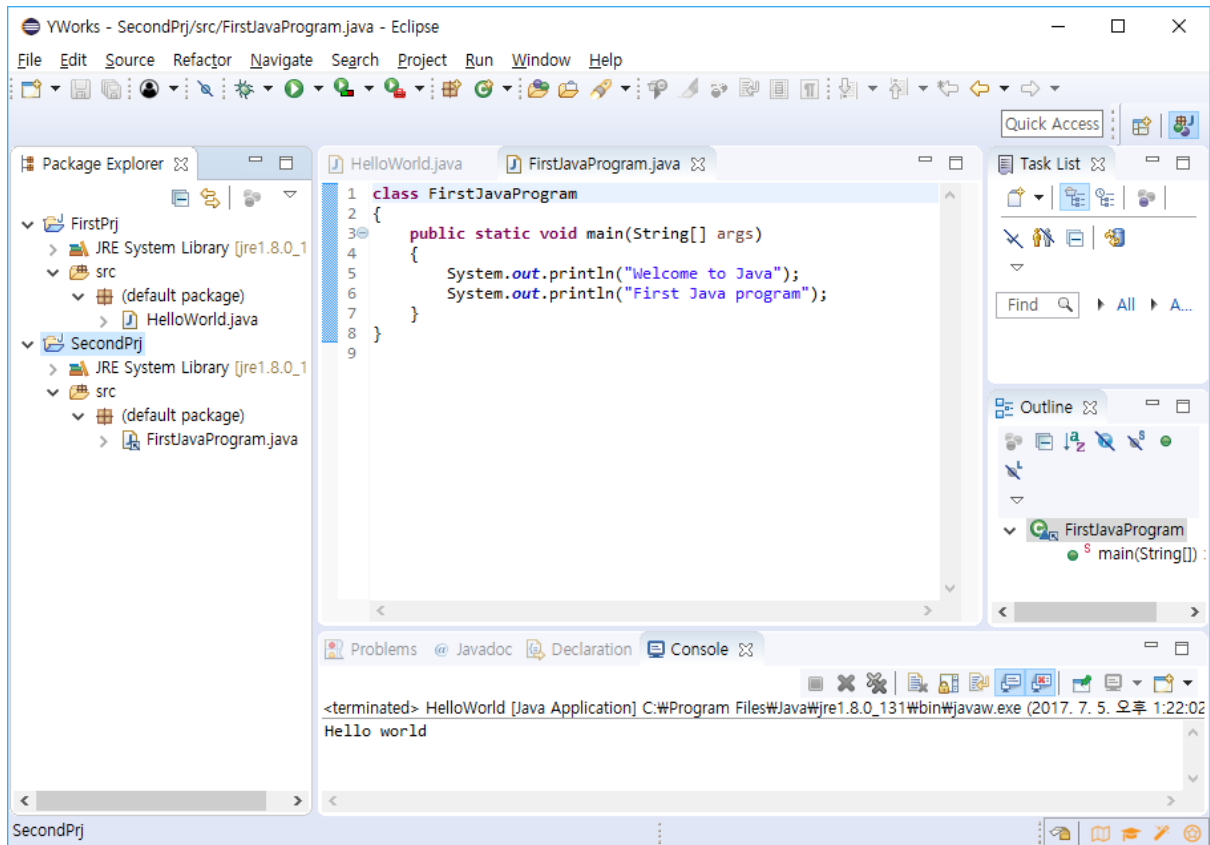


그럼 이제 제가 쓴 책의 예제 중 하나인 FirstJavaProgram.java을 프로젝트 SecondPrj에 추가해보도록 하지요. 추가하는 방법은 몇 가지고 있는데요. 우선 가장 쉬운 방법! 우리가 잘 아는 방법을 이용해서 파일을 추가해 보겠습니다. 간단해요. 위 페이지의 왼편에 있는 SecondPrj 정보 아래에 보면 src 폴더가 보이지요? 윈도우 탐색기 같은 것을 열어서 추가하고자 하는 Java 파일을 이곳으로 Drag & Drop 하면 됩니다. 그러면 다음 페이지가 뜹니다.



이때 추가하는 방법이 두 가지가 있습니다. Copy files를 선택하면 SecondPrj의 src 폴더에 소스파일이 복사가 되면서 프로젝트에 포함이 됩니다. 반면 Link to files를 선택하면 복사는 이뤄지지 않고 그냥 프로젝트에 포함이 되기만 하지요. 둘 다 실행 결과를 확인하는데 지장은 없습니다. 그래도 지금은 Copy files를 선택해서 파일이 복사가 된 것을 확인해 보시기 바랍니다.

이렇게 해서 자바 파일을 프로젝트에 포함을 시켰으면 다음과 같이 왼편에서 확인이 됩니다.



그럼 이제 실행을 해 봅시다. 그런데 왼편에 보면 프로젝트가 둘 입니다. 상관 없습니다. 프로젝트를 마우스로 선택하고 실행 버튼을 누르면 해당 프로젝트가 실행이 되니까요. 그러니까 위의 상황에서는 FirstPrj 선택하고 실행 한번 하고 SecondPrj 선택하고 실행 한번 하는 이런 일이 가능합니다.

이렇게 해서 이클립스의 기본적인 사용방법을 설명해 드렸습니다. 참고로 조금 더 말씀을 드리자면요 이클립스는 분명 좋은 아니 훌륭한 소프트웨어 개발 툴임이 분명합니다. 그런데 절대적이지는 않아요. 실력 있는 개발자들 중에서는 편집기 수준의 단순 툴을 더 선호하는 경우도 적지 않게 있거든요. (물론 메모장 수준의 편집기를 말하는 건 아닙니다.) 이클립스는 목적에 따라 그 기능이 매우 많아요. 따라서 지금은 공부에 필요한 만큼만 알고 사용하면 충분하다는 개인적인 생각입니다.