

# 부록 3. 인텔리제이(IntelliJ) IDEA 설치 및 사용

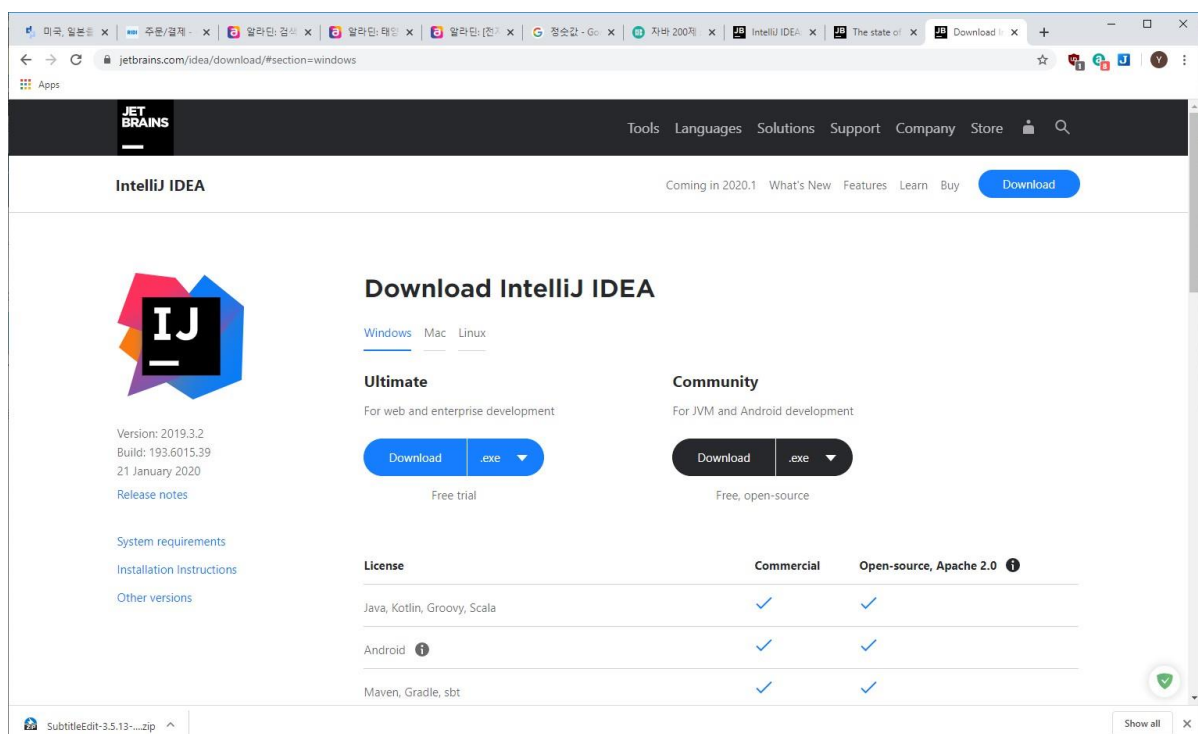
## 인텔리제이(IntelliJ) 설치

JDK를 설치한 후에 커맨드 윈도우에서 컴파일 하는 것을 보았다. 그렇게 컴파일하고 실행시키는 것도 가능하지만, 코드가 많아지면 컴파일하는 것이 불편해진다. 그래서 개발자들은 본인들에게 적합한 형태의 다양한 도구들을 사용하는데, 여기서 설치할 것은 인텔리제이라고 불리는 자바 통합 개발 환경(Integrated Development Environment 또는 IDE)이다. 통합 개발 환경이란 한 개 프로그램에서 코드 작성, 컴파일, 디버깅을 함께 할 수 있는 프로그래밍 도구로 초보자들뿐만 아니라 실무에서 일하는 개발자들도 많이 사용한다. 자바 IDE는 인텔리제이(IntelliJ) IDEA(이후 인텔리제이라고 부름), 이클립스(Eclipse), 넷빈즈(NetBeans) 등이 많이 알려져 있는데 여기서는 인텔리제이를 사용하기로 한다.

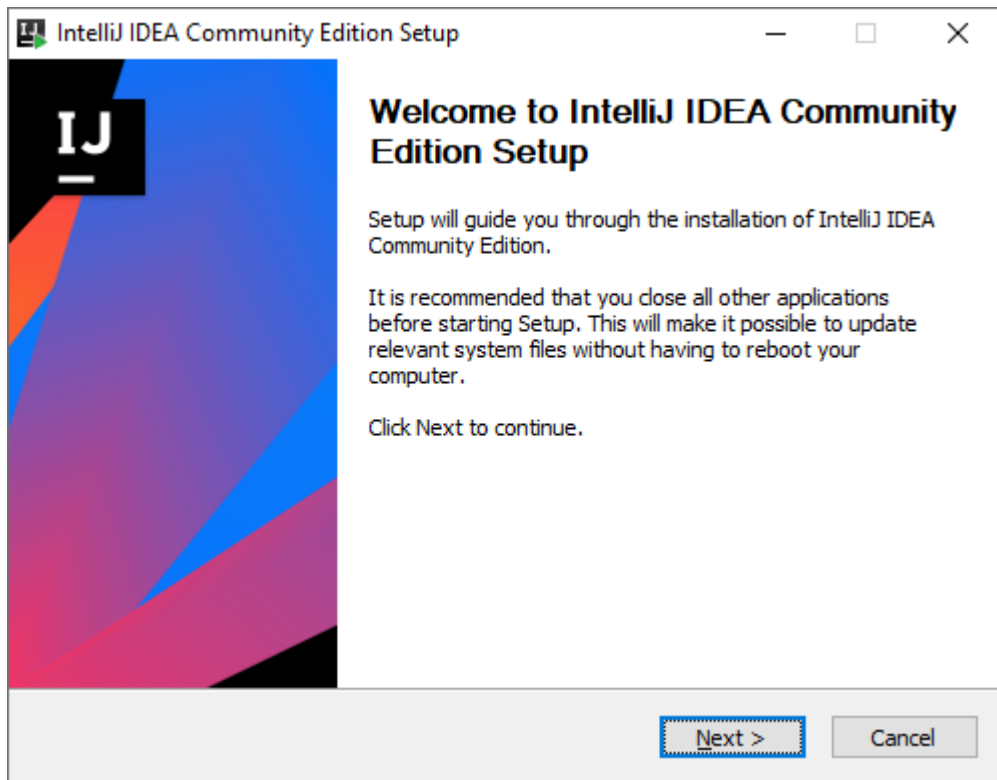
인텔리제이는 Jet Brains 라는 회사에서 개발한 프로그램이다. 많은 자바 책들은 전통적으로 무료로 제공되었던 이클립스를 주로 설명한다. 하지만 인텔리제이도 커뮤니티 버전(communitiy version)은 무료로 사용할 수 있고, 또 자바 프로그래머들이 관심을 많이 가지는 안드로이드 기기용 프로그램을 만들때 사용되는 안드로이드 스튜디오도 인텔리제이를 기반으로 만들어져 있으므로, 이 책에서는 인텔리제이를 사용하기로 한다.

## 인텔리제이 설치

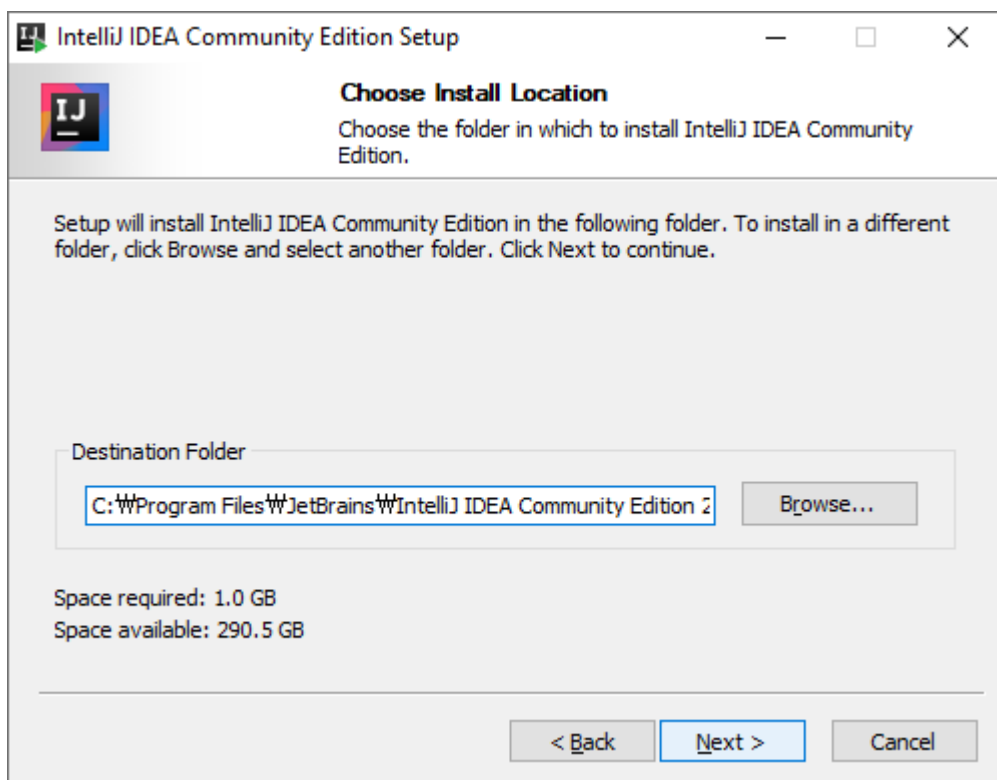
인텔리제이는 젯브레인즈(Jet Brains)사의 홈페이지(<https://jetbrains.com/idea/download>)에서 다운로드 받을 수 있다. 다음 그림은 다운로드 페이지를 보인다.



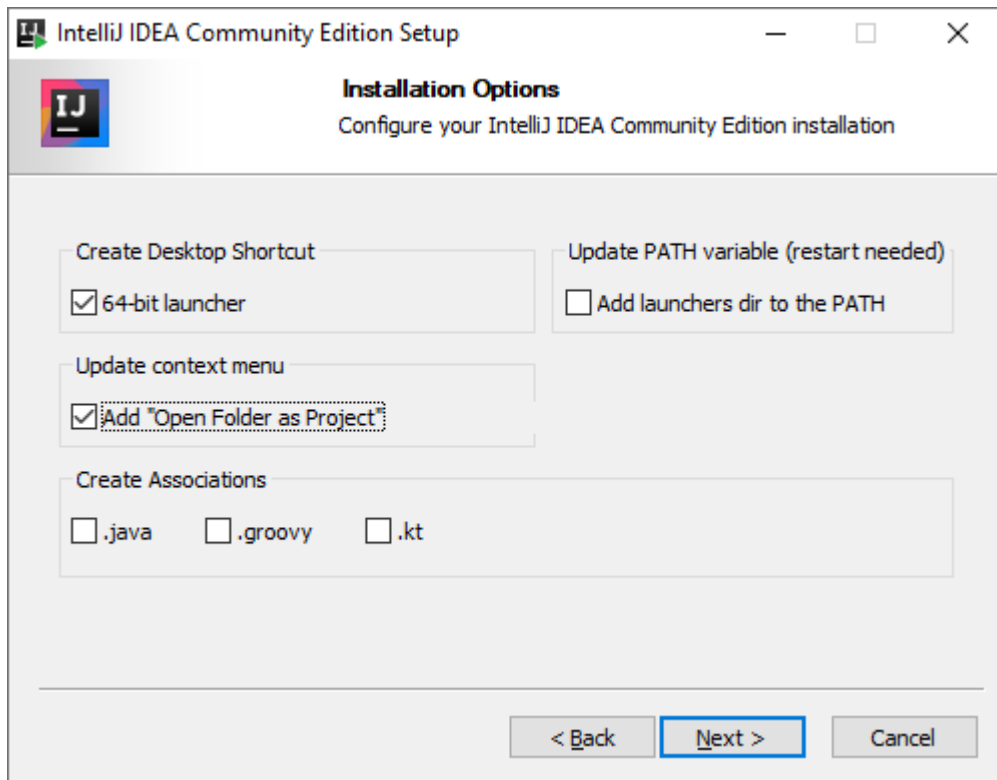
Community 버전을 다운로드 받은 후에 설치한다. 다음은 설치 시작 화면이다. Next 를 선택해서 다음으로 진행한다.



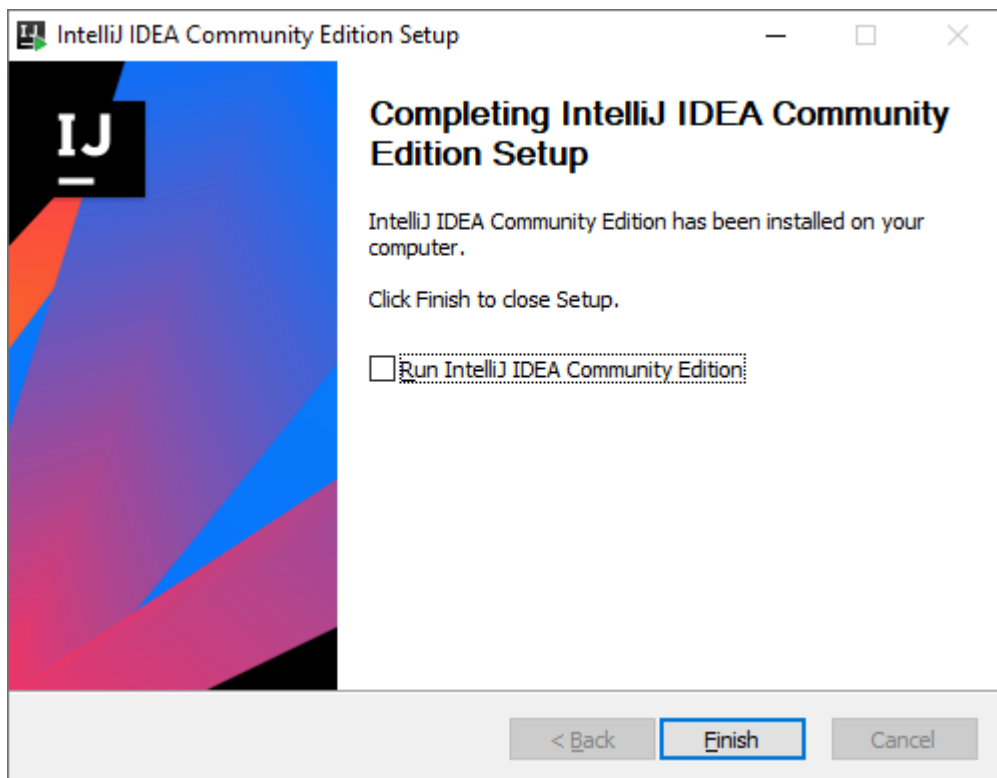
인텔리제이를 설치할 폴더(디렉토리)를 지정한다.



만약 바탕화면에 단축기를 설치하고 싶으면 "64-bit launcher"를 선택하고, 탐색기 메뉴에서 직접 프로젝트를 열고 싶다면 "Add Open Folder as Project"를 선택한다.

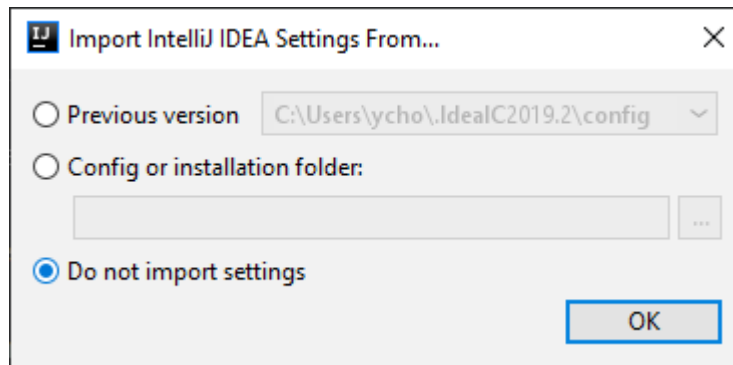


다음은 설치 종료 화면이다. Finish 를 선택해서 설치를 종료한다.

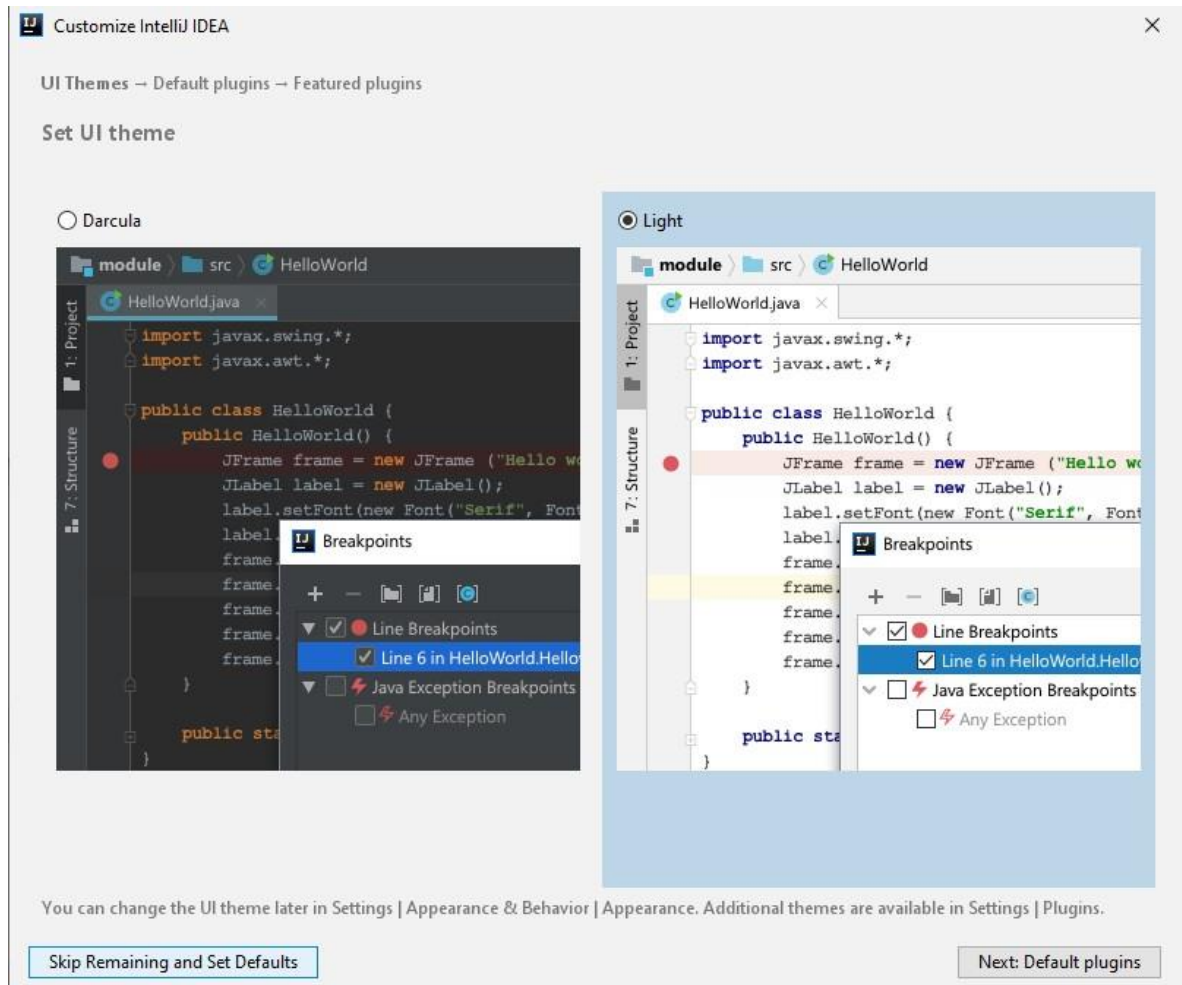


## 인텔리제이 설정하고 프로젝트 생성해서 실행하기

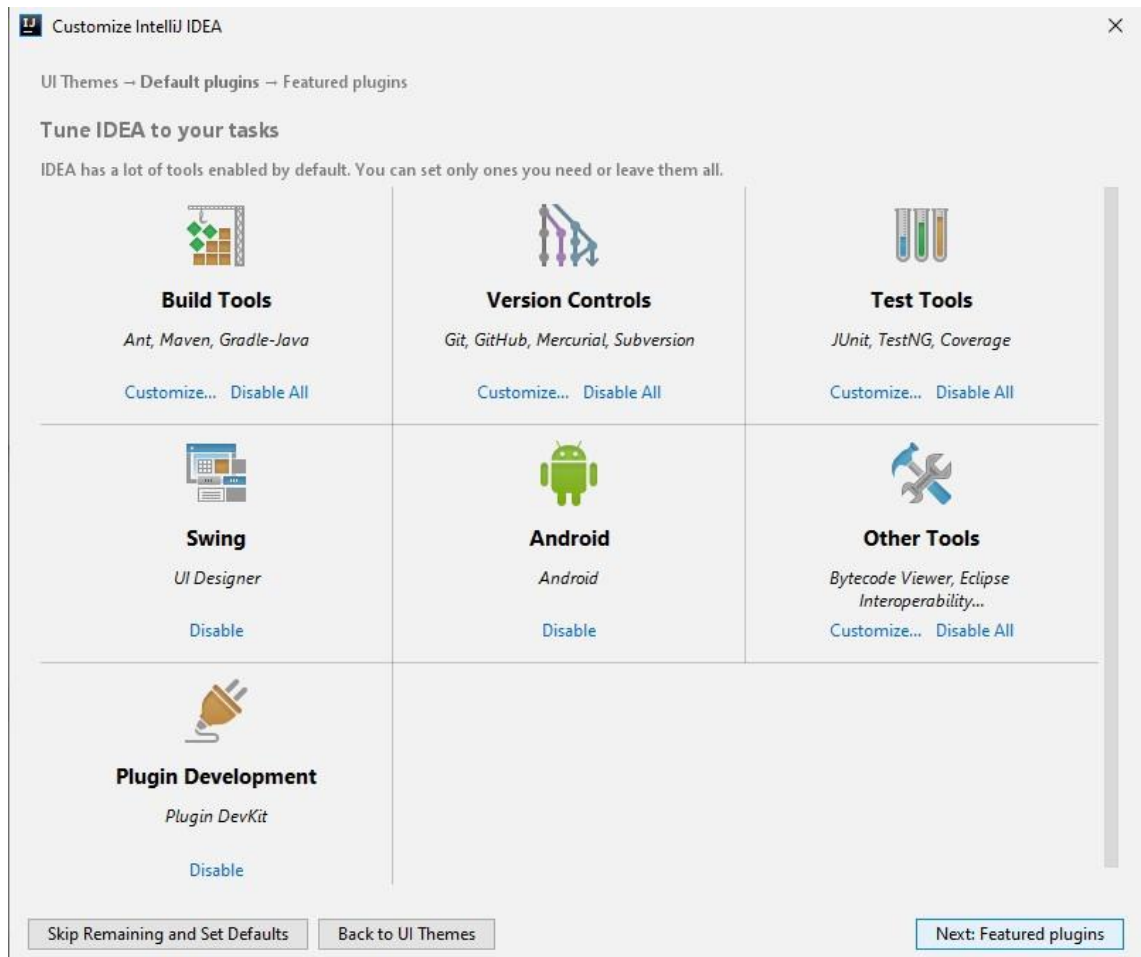
인텔리제이를 기존에 설치했었다면 다음과 같은 화면에 나타날 수 있다. 기존 설정 내용을 사용하고 싶은지 아니면 새로운 설정을 만들고 싶은지에 따라 선택하면 된다.



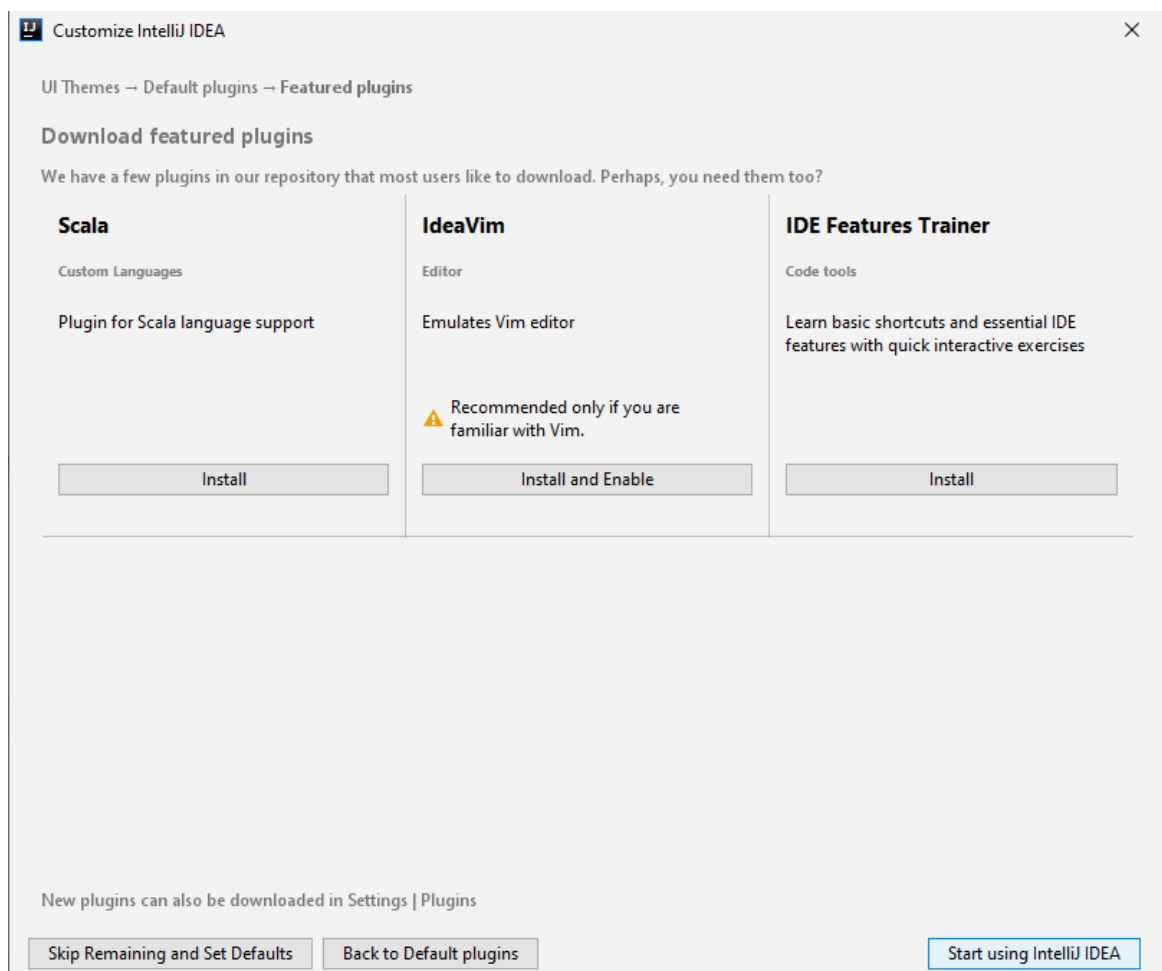
본인이 원하는 테마로 설정한다.



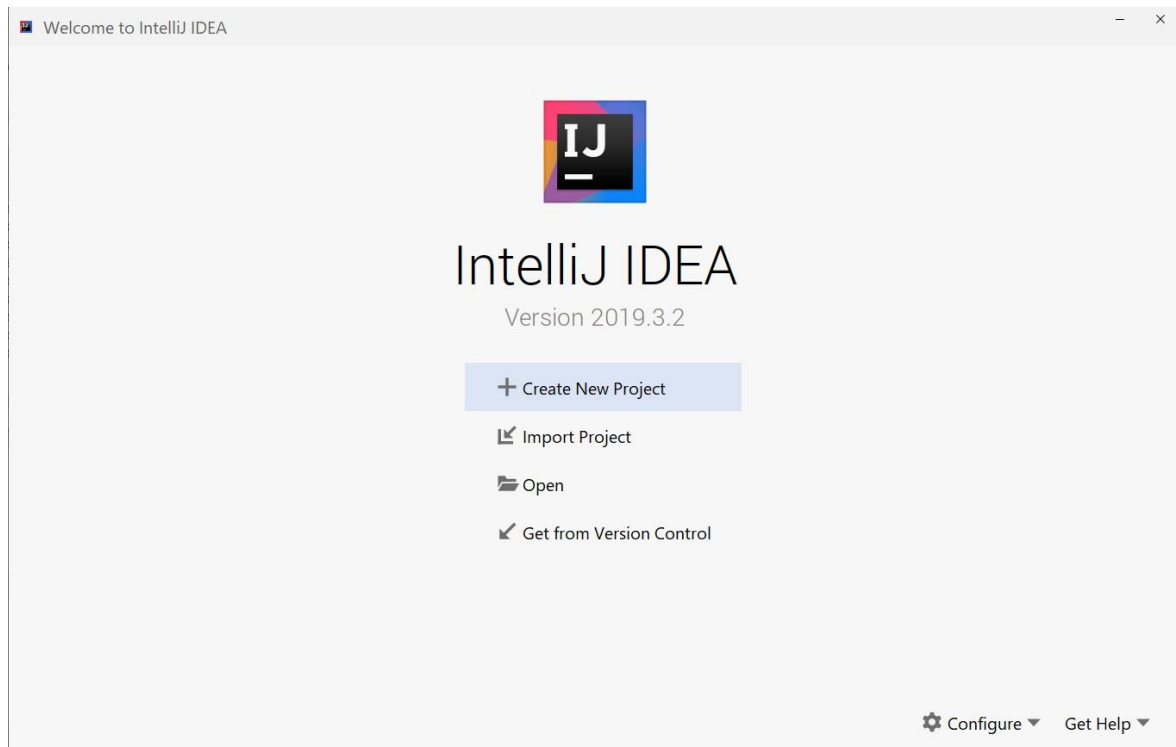
플러그인을 설정하는 화면이다. 내용을 안다면 본인이 원하는 것들만 설치하는 것이 가능하다. 잘 모른다면 그냥 다음으로 넘어간다.



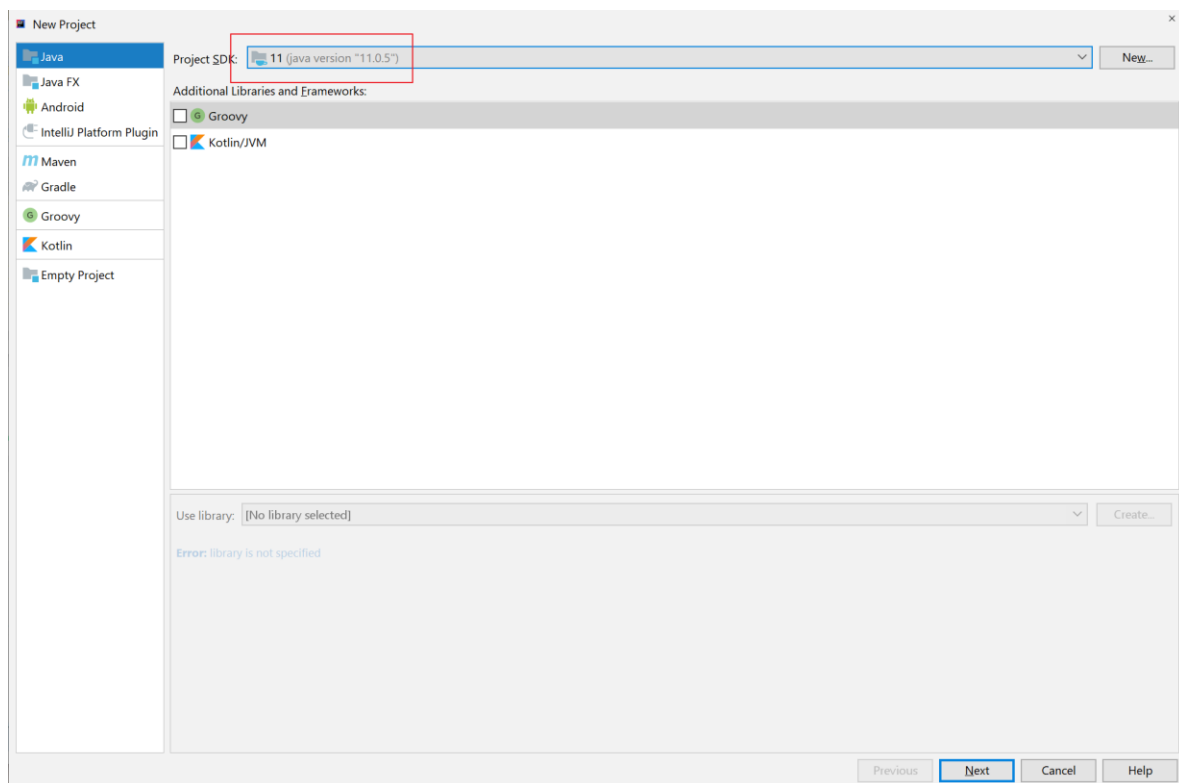
역시 마찬가지로 다음으로 넘어가도 된다. 스칼라 에디터나 vim 키를 사용하고 싶다면 혹은 인텔리제이 사용법을 익히고 싶다면 적절한 플러그인을 설치하도록 한다.



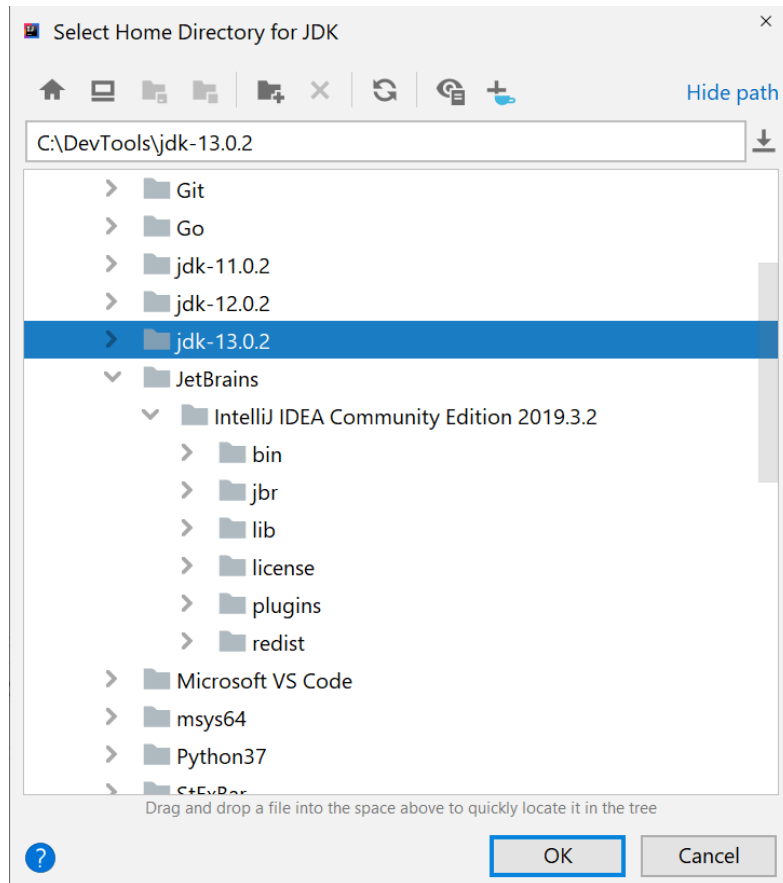
인텔리제이를 실행시킨 화면이다. 여기서는 프로젝트를 만들고 실행해보려 하니, "Create New Project"를 선택한다.



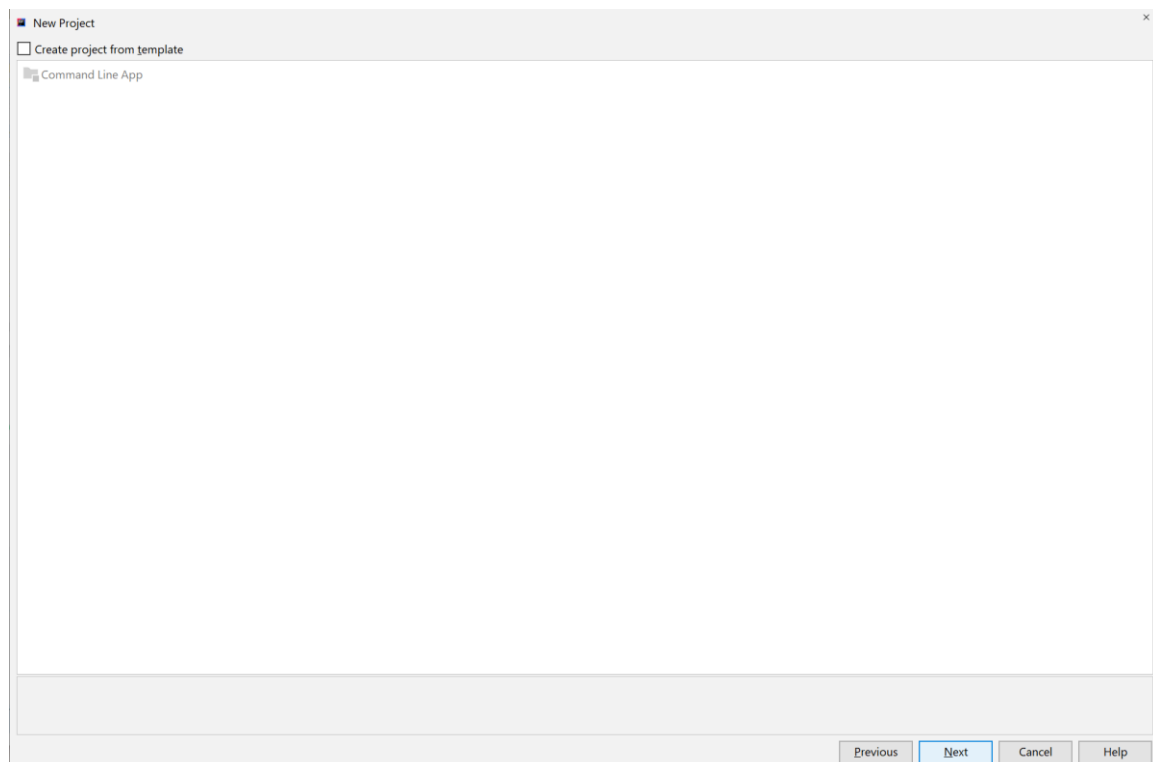
Project SDK 를 선택해야 할 수 있다. 만약 Project SDK 부분에 아무것도 없거나 혹은 원하는 JDK 버전 이 아니라면 "New..." 버튼을 눌러서 새로운 JDK 를 선택한다.



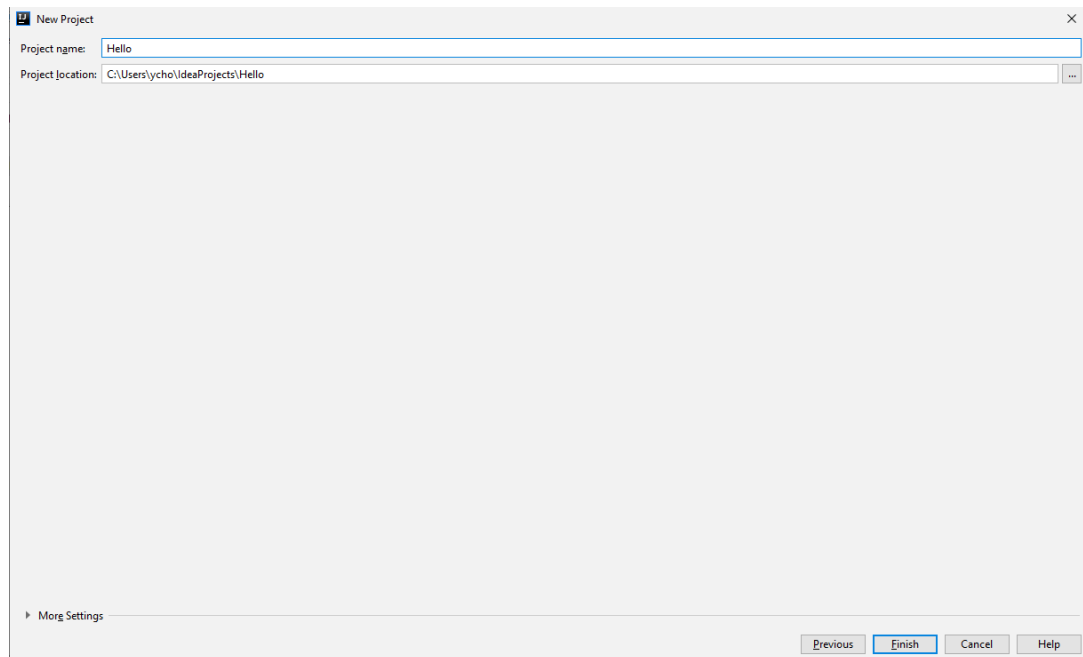
다음 그림은 "New..." 버튼을 누른 경우를 보인다. 여기서는 jdk 13.0.2 버전을 선택하였다. 사용자는 본인이 설치한 JDK 의 버전을 선택하면 된다.



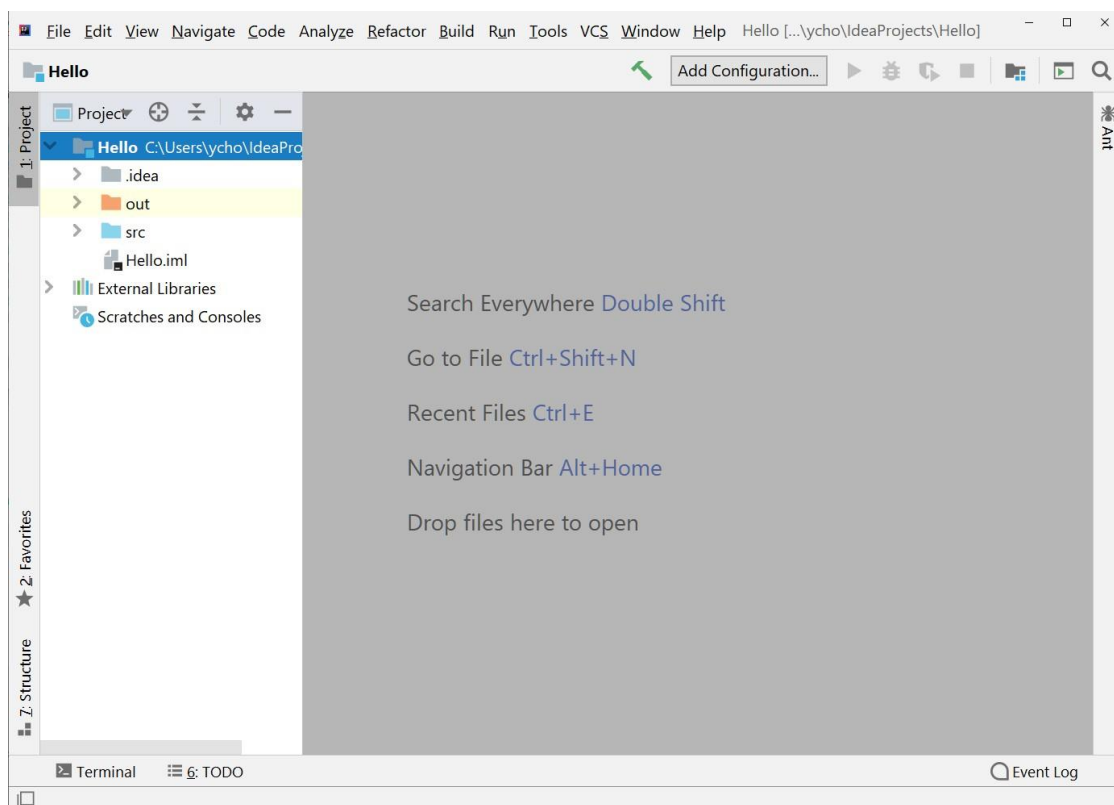
기본적인 뼈대가 되는 코드를 생성해주기를 원하면 빈 칸을 클릭해서 선택하고 아니면 바로 Next 를 선택한다.



프로젝트 이름을 지정한다.

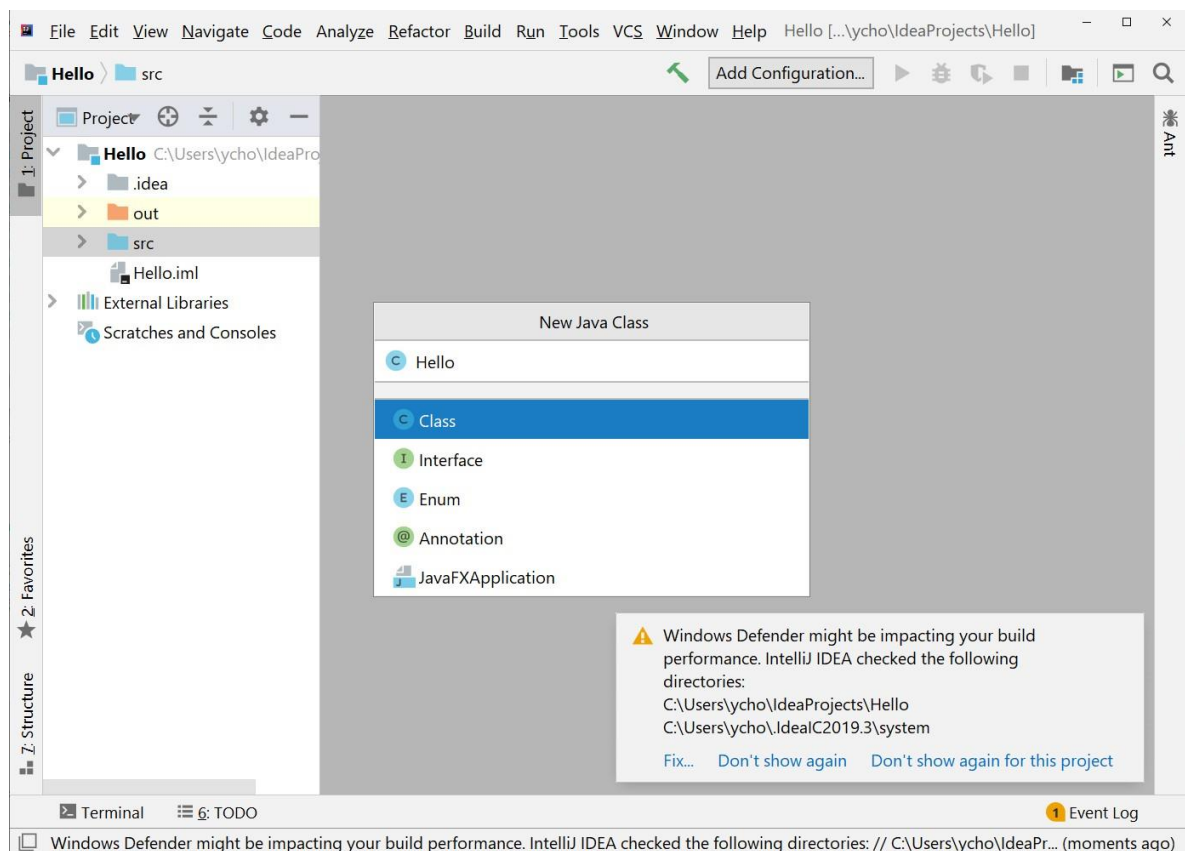
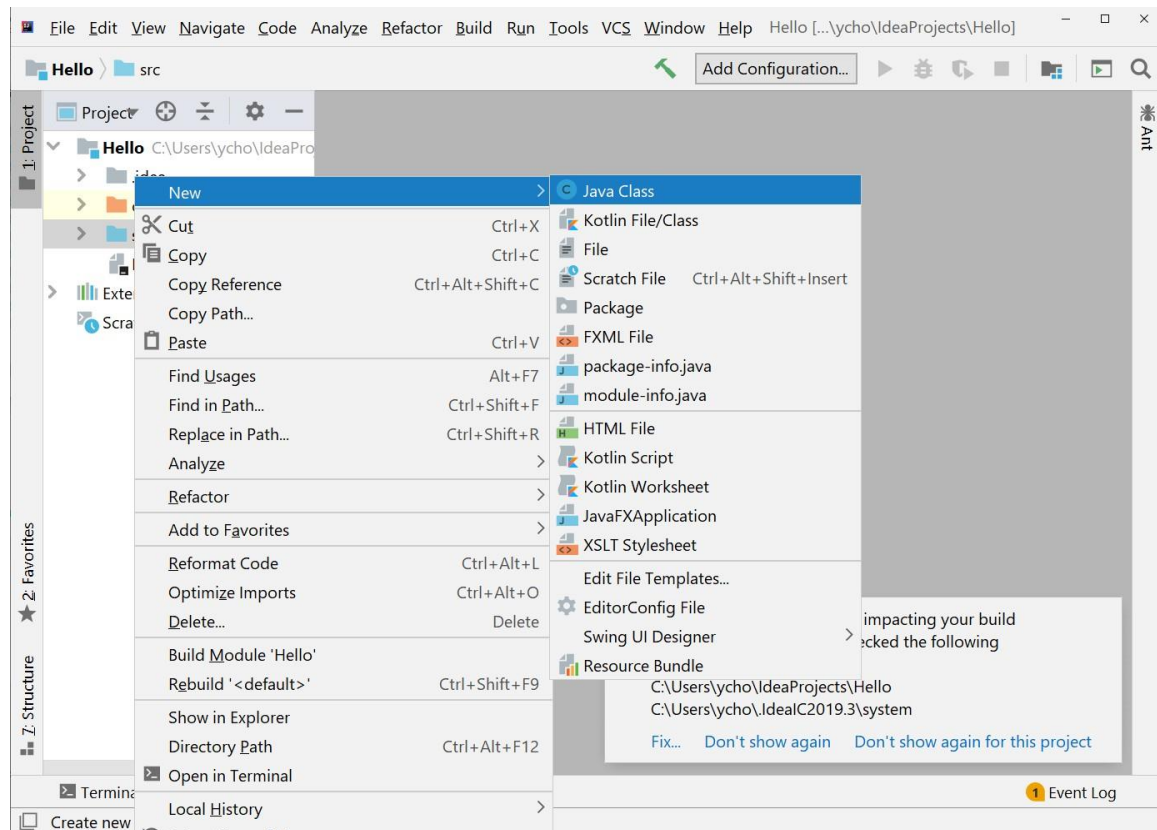


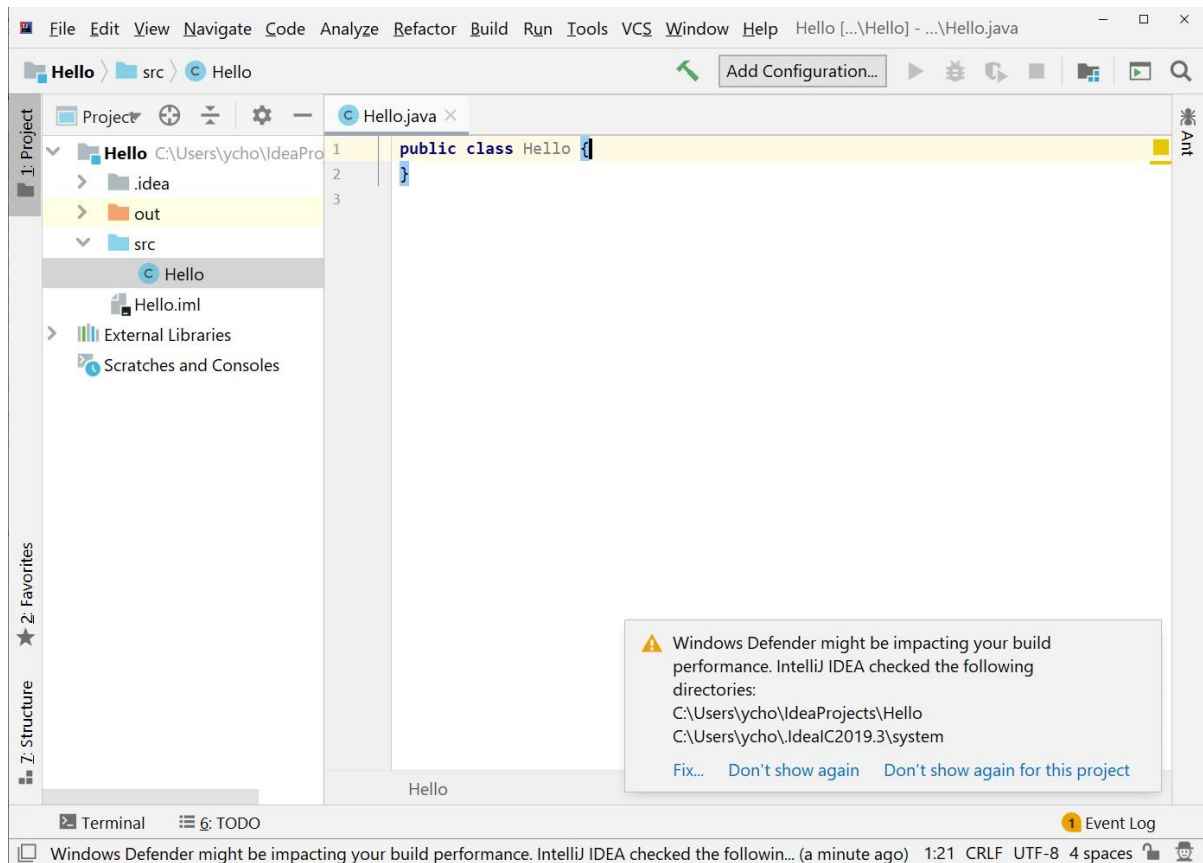
프로젝트를 생성한 후에 마우스를 클릭하면 다음과 같은 화면을 볼 수 있다. 이제 코드를 추가해본다.



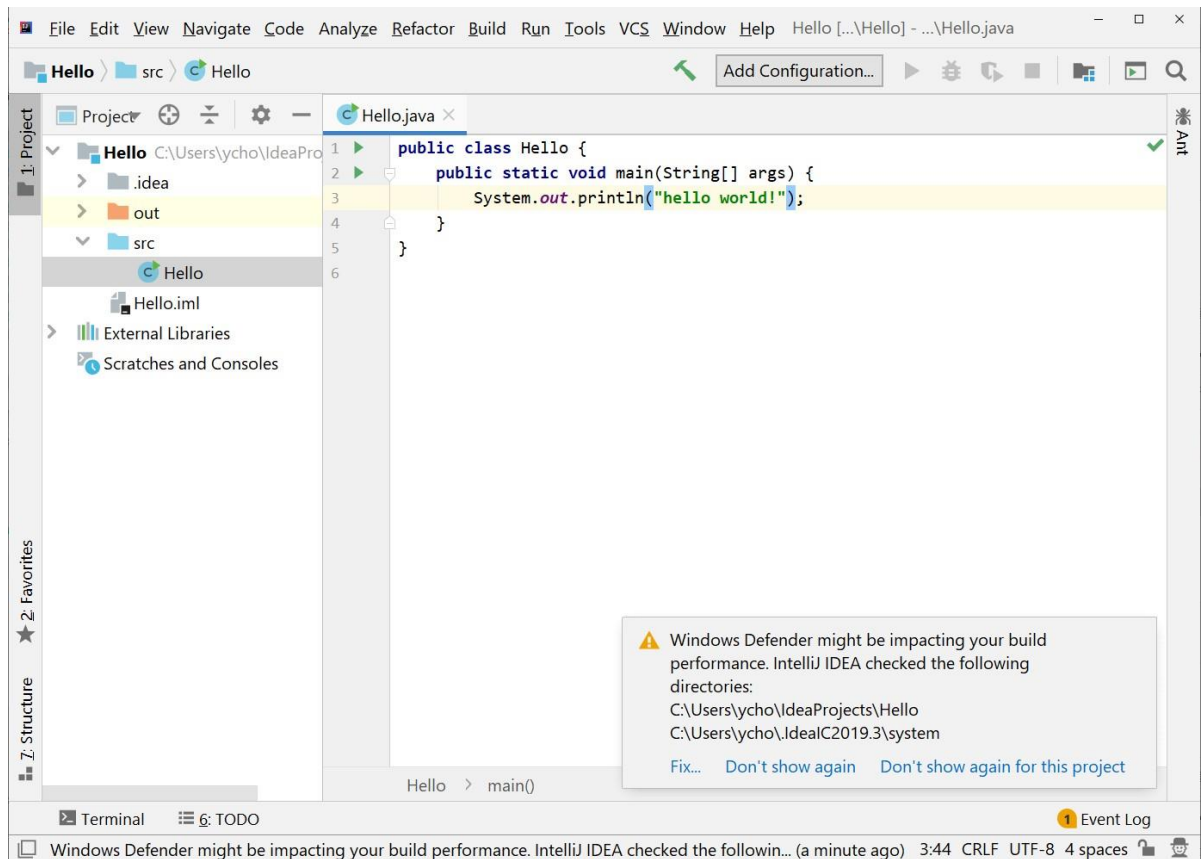


자바는 컴파일하려면 기본적으로 클래스가 한 개 이상 있어야 한다. 따라서 다음에서 보인 것처럼 클래스를 생성한다. 클래스 이름을 적고 엔터키를 누르면 클래스가 생성된다.

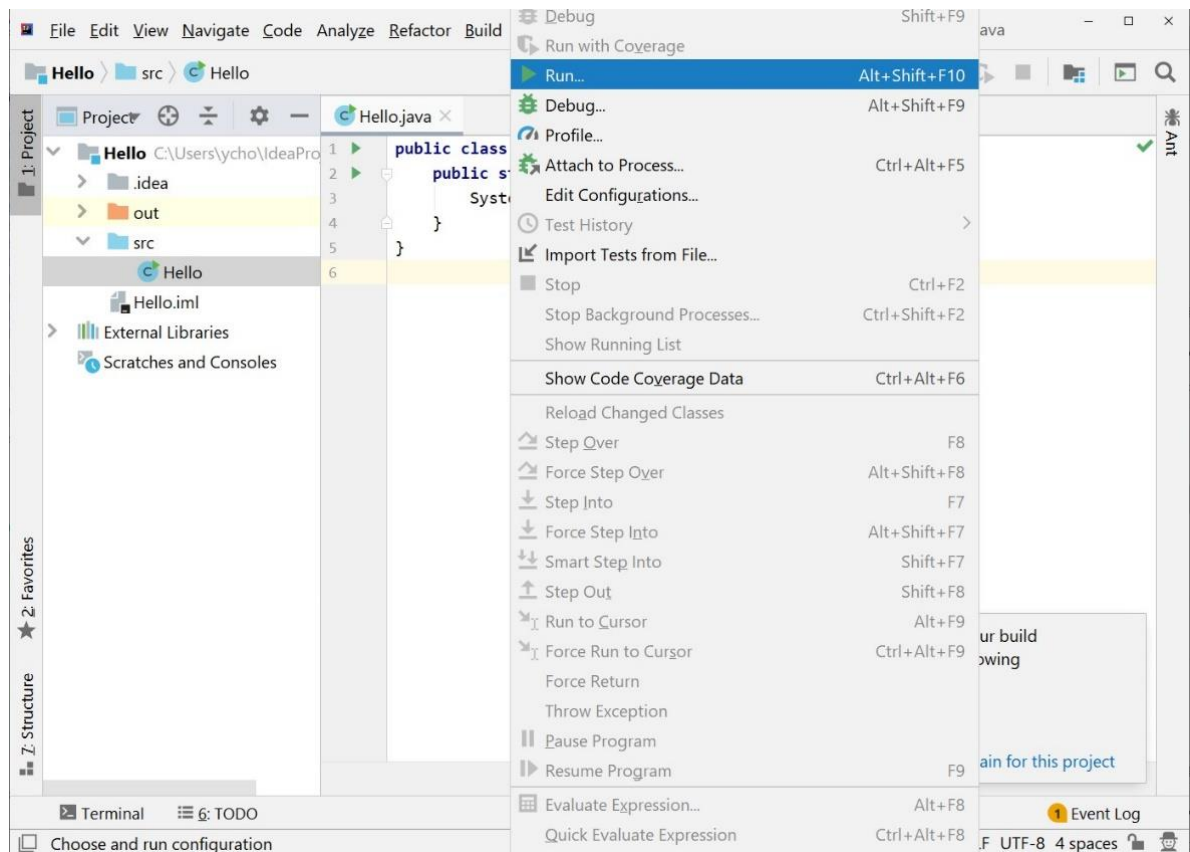




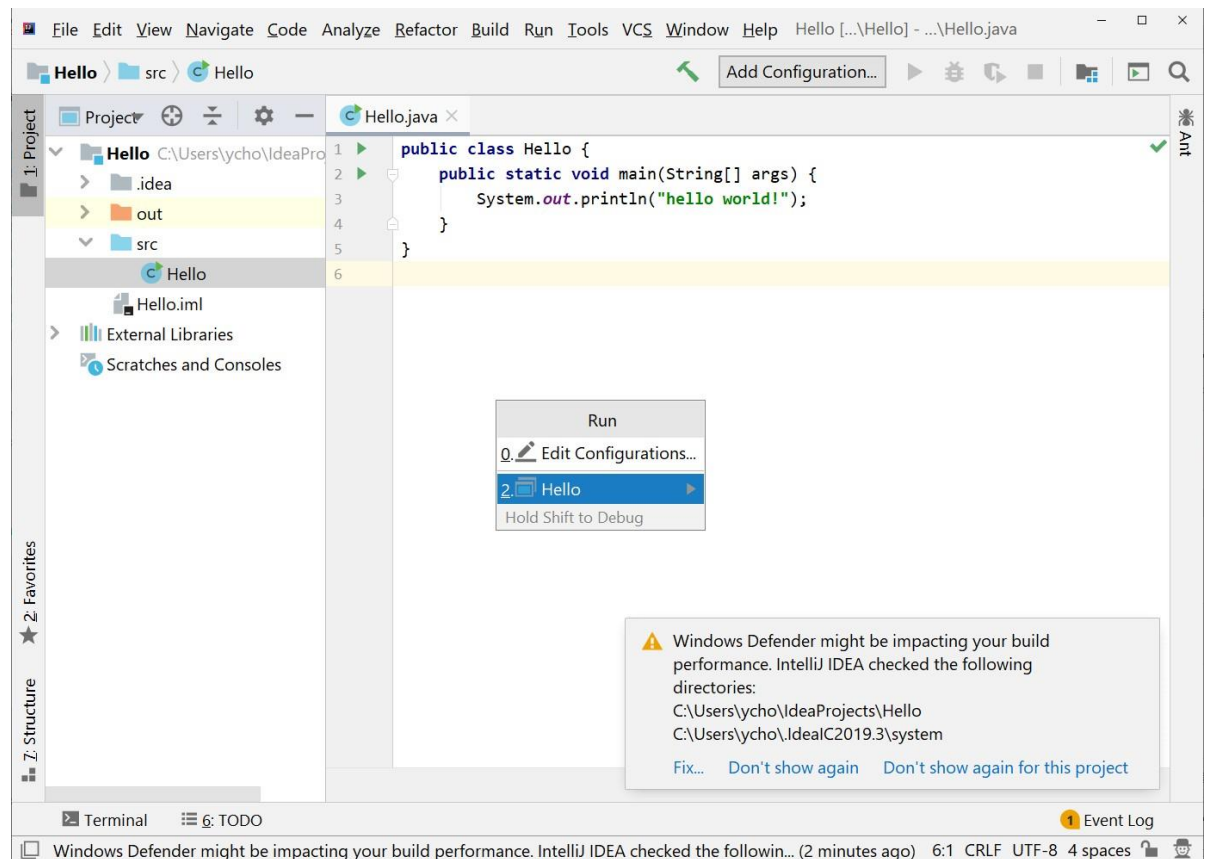
다음에서 보인 것처럼 입력해본다.



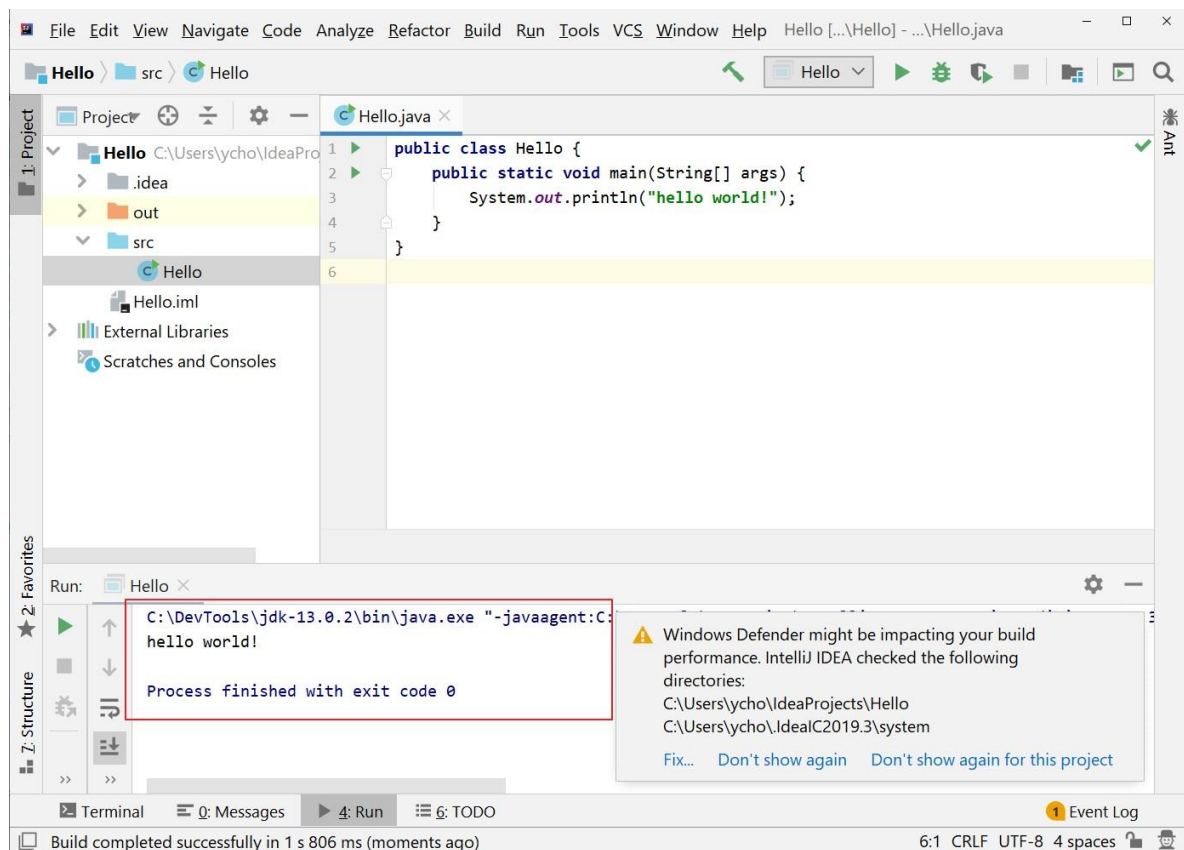
인텔리제이에서 처음으로 프로젝트를 실행시키려면 "Run..." 메뉴에서 실행할 클래스(main) 함수를 가지고 있는 클래스 중에서 한 개)를 지정해야 한다.



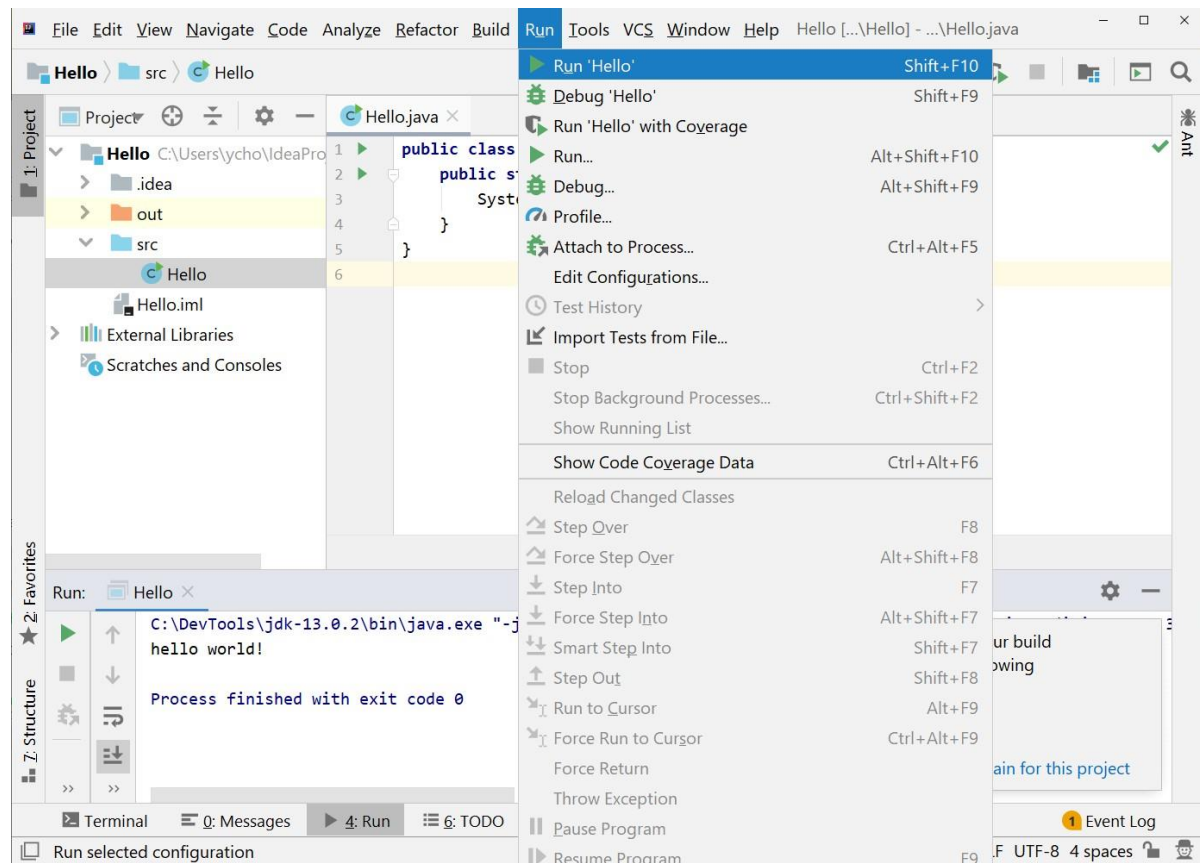
다음 화면에서는 "Hello" 클래스를 실행할 클래스로 선택하였다.



다음은 실행 화면을 보인다. 아래 빨간색 박스 영역에 실행 결과가 보인다.



앞에서 말했던 것처럼 "Run..." 메뉴는 프로젝트를 처음 실행시킬 때 사용한다. 일단 실행할 클래스가 지정되었다면 다음 그림처럼 Run 메뉴를 실행시키면 된다.

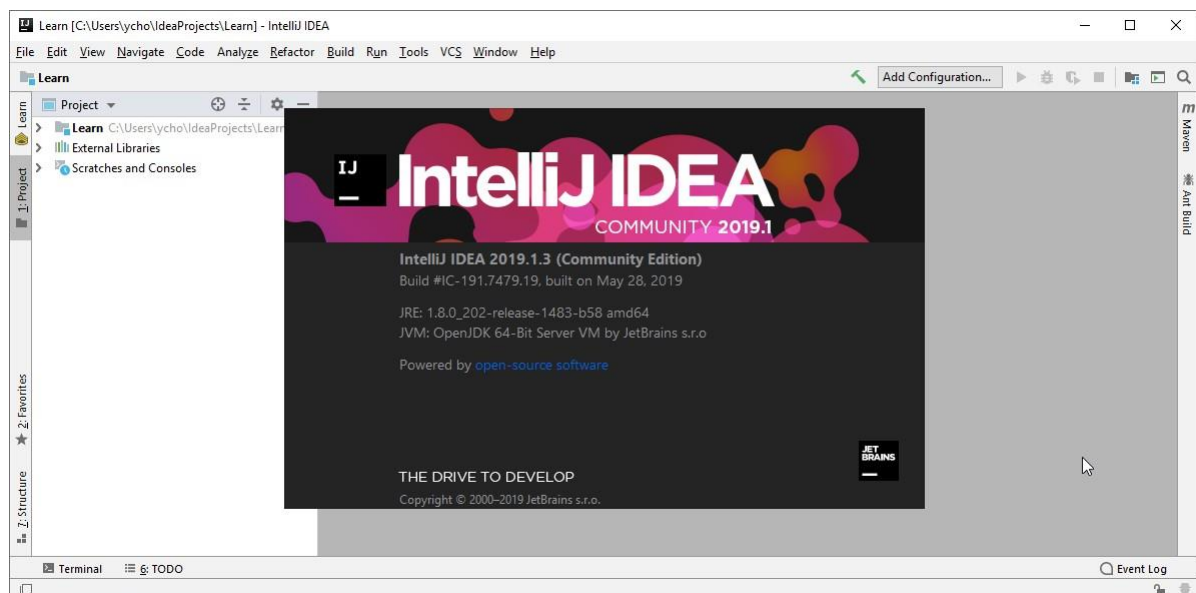
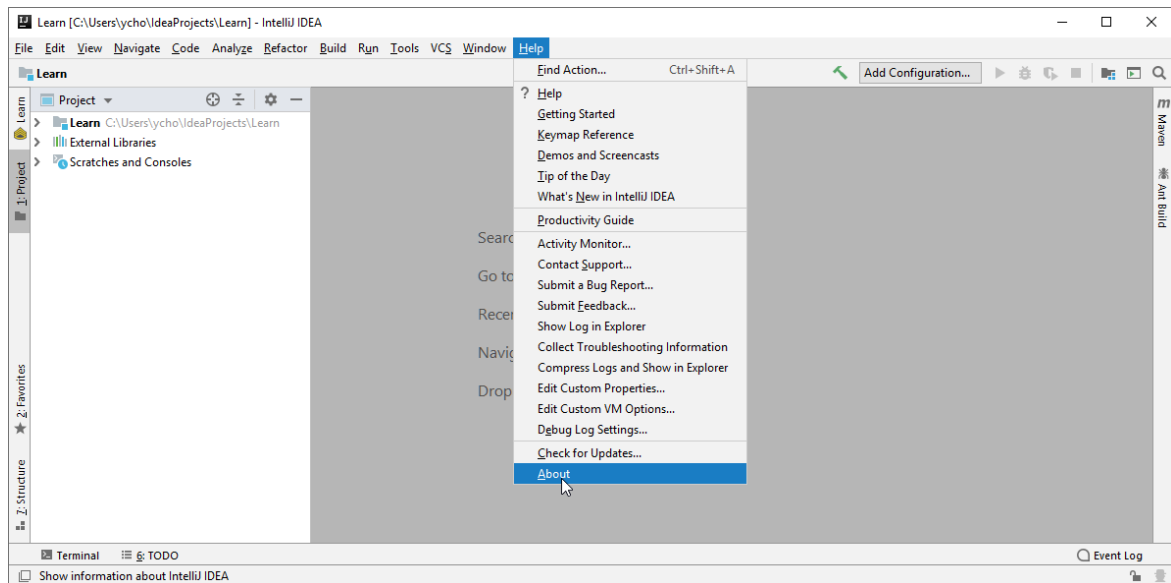




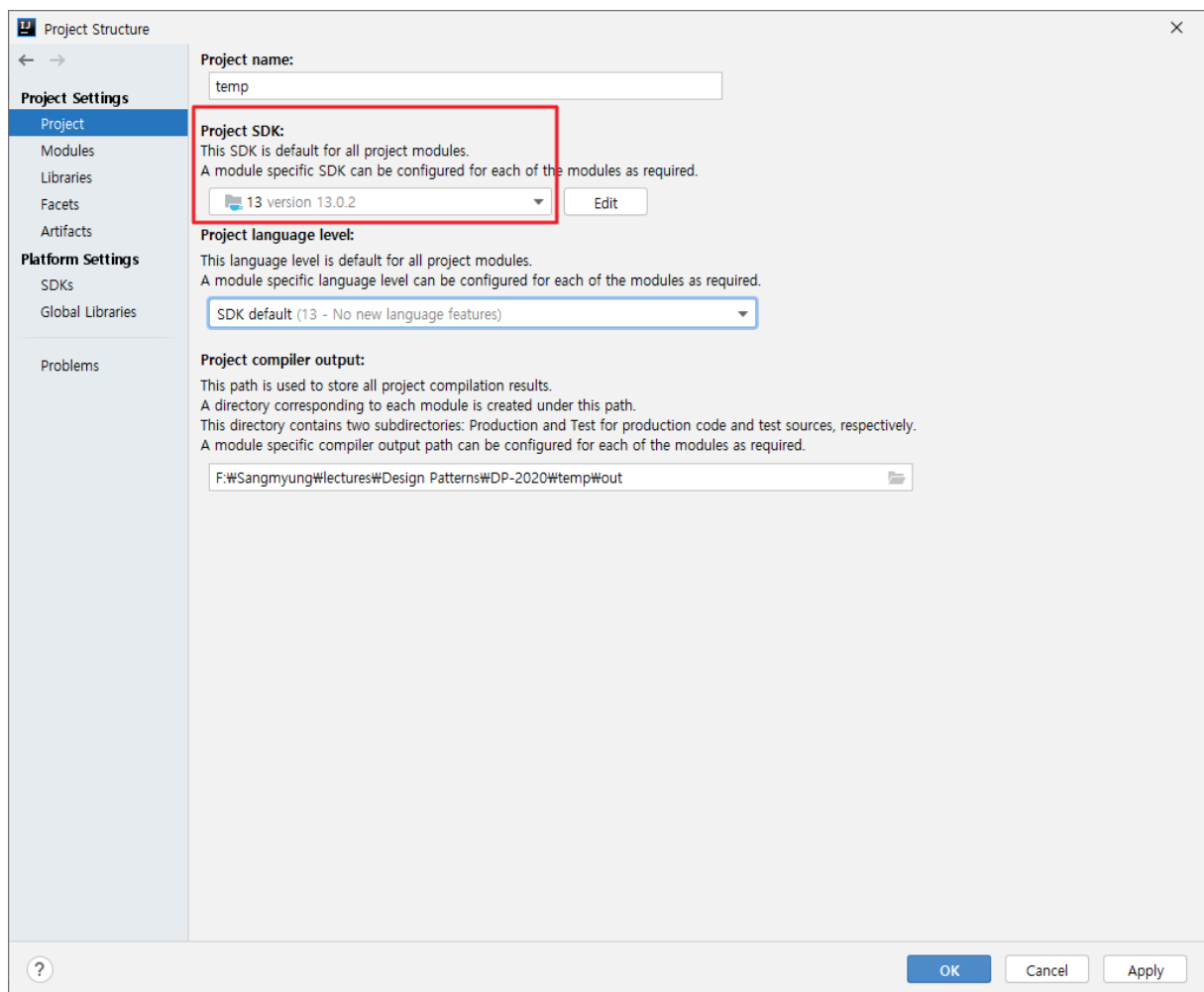
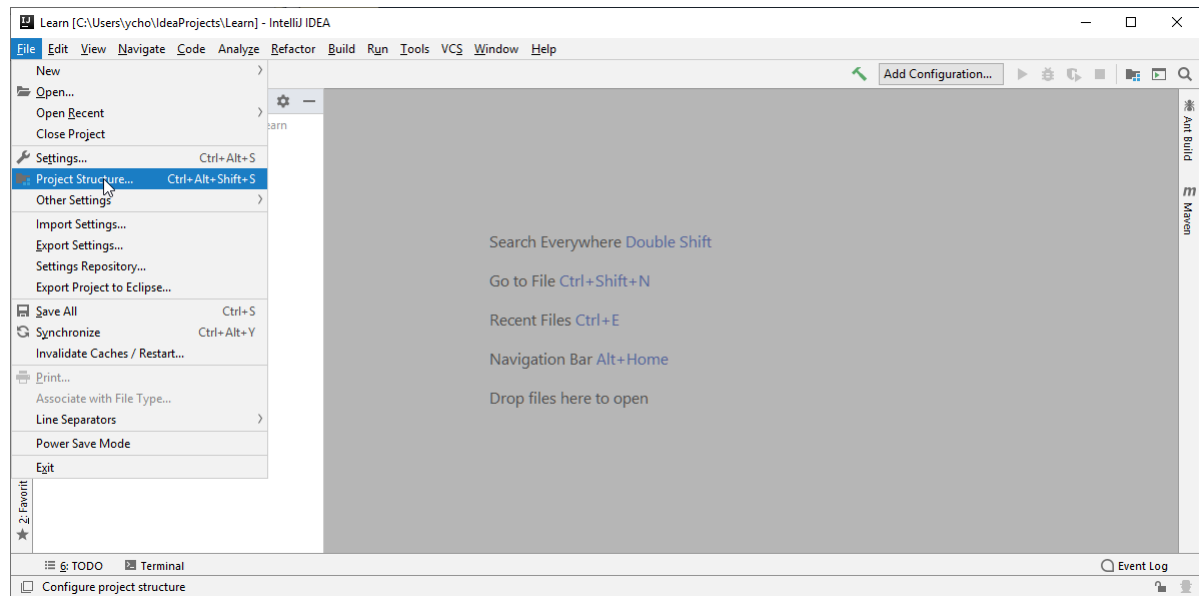
# 인텔리제이에서 JShell 사용하기

터미널에서 JShell 을 사용하면 기존에 자바 코드를 작성하고 컴파일하는 것 보다는 더 간단하고 빠르게 코드를 실행시켜보고 결과를 확인할 수 있다. 하지만 코드가 조금만 길어지면 JShell 에서 코드를 입력 하는 것이 불편해진다. 따라서 외부 에디터를 사용하게 되는데, 이것도 계속 에디터와 JShell 을 번갈아 가면서 코드를 편집하고 실행시키고 하기 때문에 불편할 수 있다. 다행히도 인텔리제이는 JShell 을 잘 지원하는 편이어서 여기서는 인텔리제이에서 JShell 을 사용하는 방법을 보기로 한다.

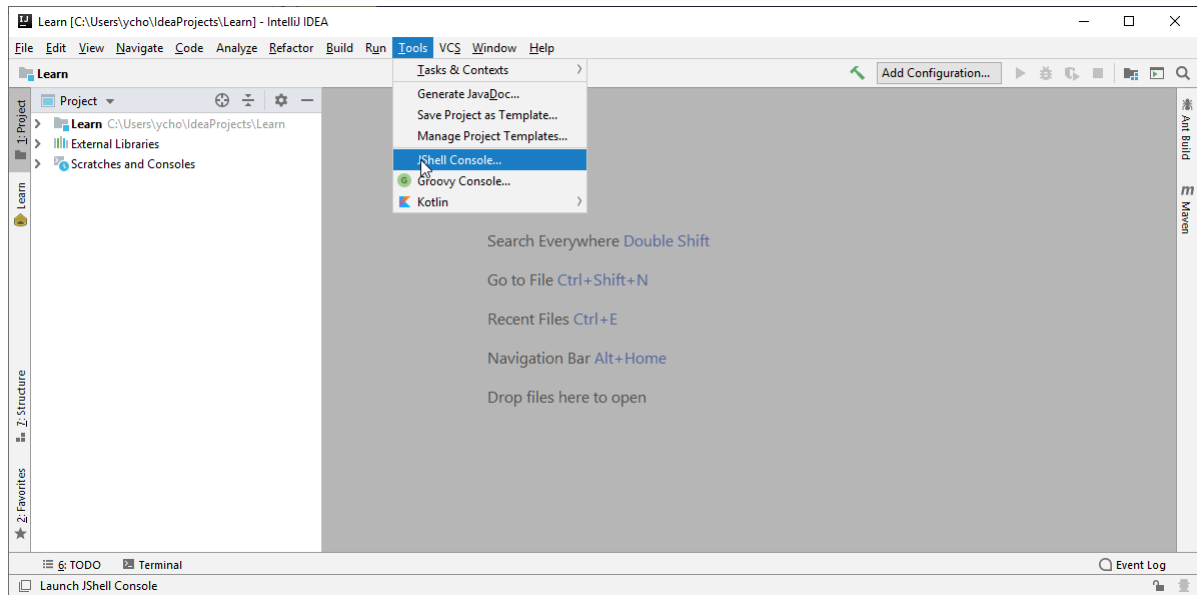
인텔리제이에서 JShell 을 사용하려면 IntelliJ IDEA 2017.3 또는 이후 버전이 필요하다 본인이 설치한 IDEA 의 버전 정보를 확인해보도록 한다. 버전 정보 확인은 Help 메뉴의 About 항목을 실행시키면 된다. 아래 그림에서는 Help|About 메뉴를 선택하는 화면과 버전 정보를 보인다.



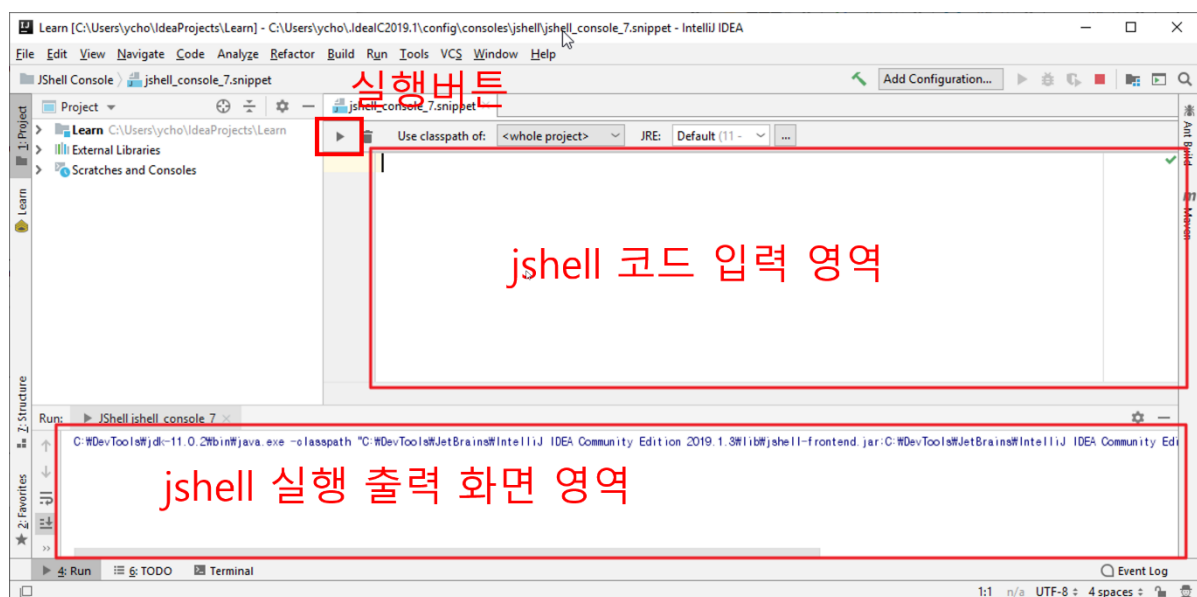
인텔리제이에서 JShell 을 사용하려면 먼저 프로젝트를 한 개 생성해야 한다. 원하는 이름으로 프로젝트를 생성하고, 다음에는 JDK 9 이상을 사용함을 확인하자. 아래 그림에서 보인 것처럼 File 메뉴의 Project Structure 항목을 선택하고, Project SDK 가 JDK 9 이상임을 확인한다.



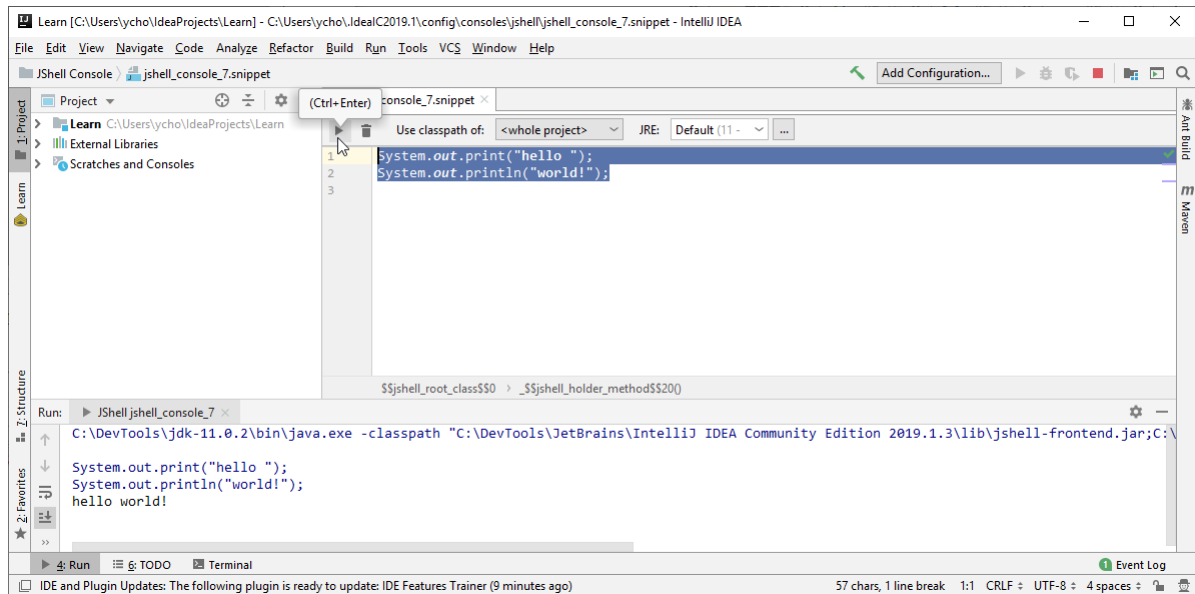
여기까지 확인되었으면 이제 JShell 을 사용할 수 있다. Tools 메뉴의 "JShell Console..." 항목을 선택해 서 JShell 을 실행시킨다.



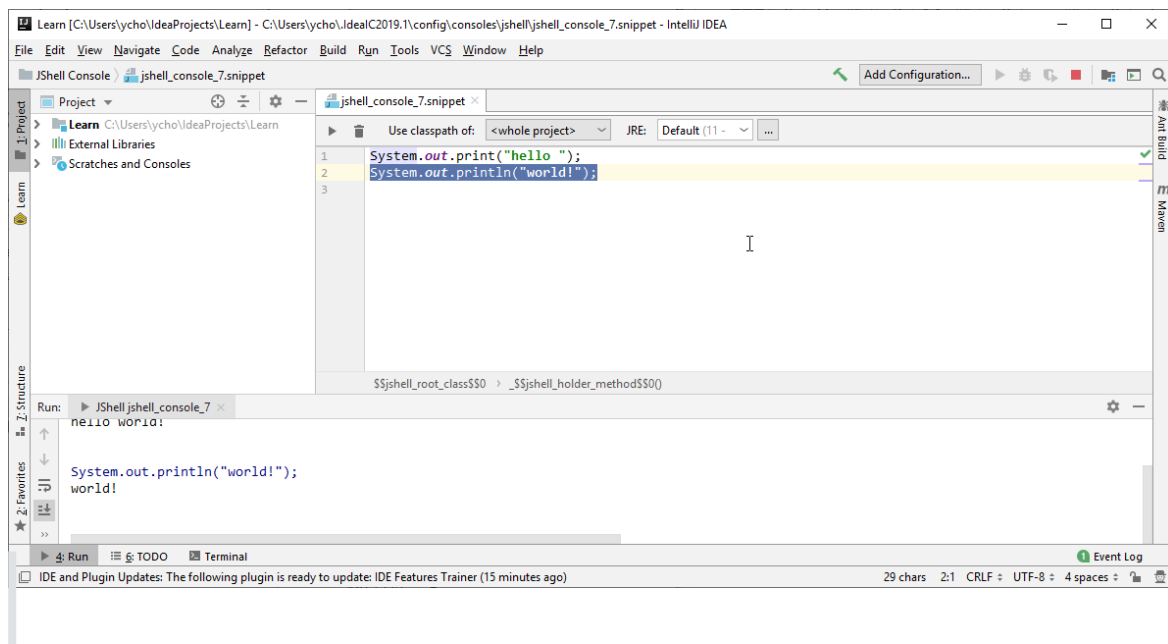
아래에서 보인 것 같은 화면이 나타난다. 코드 입력 영역에 코드를 입력하고 실행시키면 실행 출력 화면 영역에 그 결과가 나타난다.



다음에서 보인 코드를 입력하고 실행시키려고 하는 코드를 선택한 후에 후에 Ctrl+Enter(컨트롤 키와 엔터 키를 함께 누름) 또는 실행버튼을 눌러서 실행시킨다. 아래 그림은 두 줄의 자바 코드를 입력해서 실행시킨 것을 보인다.



입력한 코드 중에 일부만 선택해서 실행하는 것도 가능하다. 다음은 두 줄의 코드 중 두 번째 줄만 실행시킨 화면이다.



### 더 알아보기: 윈도우용 인텔리제이 JShell 의 한글 출력 문제

윈도우용 인텔리제이의 JShell 에서 한글을 제대로 출력 못하는 문제가 있다. 리눅스, 맥 등에서는 문제가 나타나지 않는 것으로 보아, 윈도우에서 사용하는 한글 코드 체계 때문인 것으로 보인다. 만약 한글 출력이 필요한 코드를 사용한다면, 인텔리제이의 JShell 이 아니라 명령행에서 직접 입력하는 형태로 실행시켜야 한다.

또 인텔리제이의 JShell 콘솔은 사용자로부터 입력을 받는 코드를 실행시키는 것은 불편하다. 그래서 이 책에서는 인텔리제이의 터미널 환경에서 JShell 을 사용하는 것을 설명한다. 두 가지 사용법을 모두 살펴보고 본인에게 적합한 방법을 선택해서 사용하는 것이 좋을 것이다. 이 책에서는 다음에 소개한 방법으로 사용하는 것을 기준으로 한다.



# 이 책에서 JShell 사용하기

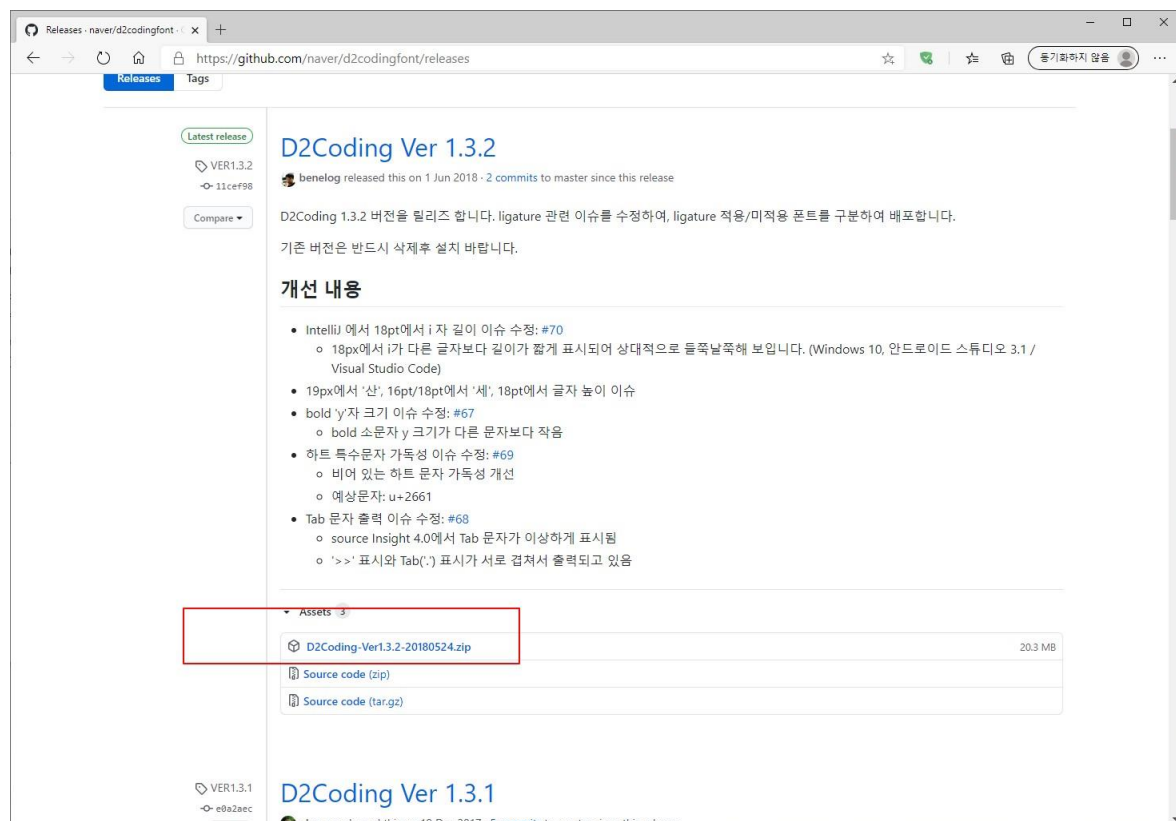
JShell에서는 바로 코드를 작성해서 실행시킬 수 있기 때문에 컴파일러를 사용하기 위해 클래스를 생성하거나 main() 함수를 작성해야 하는 불편함을 없애준다. 하지만 JShell은 코드가 좀 길어지면 입력하고 수정하는 것이 번거롭다는 점이 단점도 있다. 이러한 문제를 해결할 수 있도록 기본 에디터 또는 외부 에디터를 사용할 수 있도록 지원하지만 코드를 편집할 때마다 다른 윈도우로 이동해서 작업하는 것이 불편하다. 인텔리제이에서는 기존에 프로그래밍 에디터를 사용하는 것 같은 환경에서 프로그램을 작성하고, 이를 선택해서 실행시킬 수 있도록 해주고 있어 JShell의 활용성을 높여주는데 가장 많이 사용될 것으로 추정되는 윈도우에서는 한글 입력에 문제가 있다.

그래서 이 책에서 권하고 싶은 방법은 비주얼 스튜디오 코드나 인텔리제이의 터미널 창에서 JShell을 실행시키고, 간단한 코드는 JShell에서 입력하고, 좀 긴 코드는 에디터를 이용해서 작성한 후에 터미널 창에 복사해서 붙이는(copy and paste) 것이다. 불편하기는 하지만, 그래도 윈도우를 전환하지 않아도 되고, 한 프로그램에서 코드를 작성하고 실행시켜볼 수 있다는 점에서 좀 더 효율적이다. 여러 모로 아쉬움이 남지만 이 책에서는 다음 경우에 대해 먼저 JShell에서 실행시키고, 추후에 컴파일할 수 있는 전체 코드를 보이는 것으로 한다. 다음은 JShell에서 한글을 원활하게 사용할 수 있도록 네이버에서 만든 코딩용 한글 폰트를 설치하는 것을 먼저 설명하고, 인텔리제이의 터미널 화면에서 JShell을 사용할 수 있도록 설정을 지정하는 것에 대해서 설명한다.

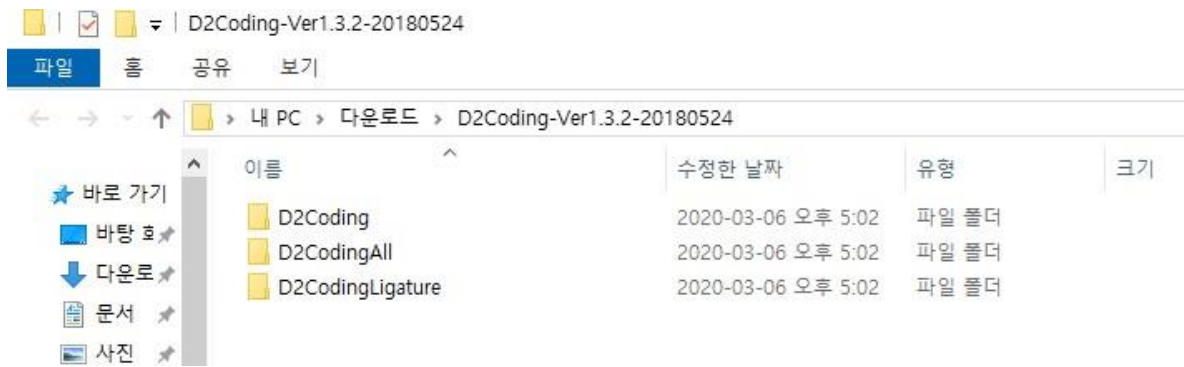
## 코딩용 한글 폰트 설치하기

여기서는 네이버에서 개발해서 오픈 폰트 라이선스로 공개한 코딩용 한글 폰트인 D2Coding을 설치하고 사용하는 것을 보인다. 여기에서는 윈도우에서 설치하는 것만을 설명함으로, 다른 플랫폼에서는 인터넷에서 설치 방법을 찾아보도록 한다.

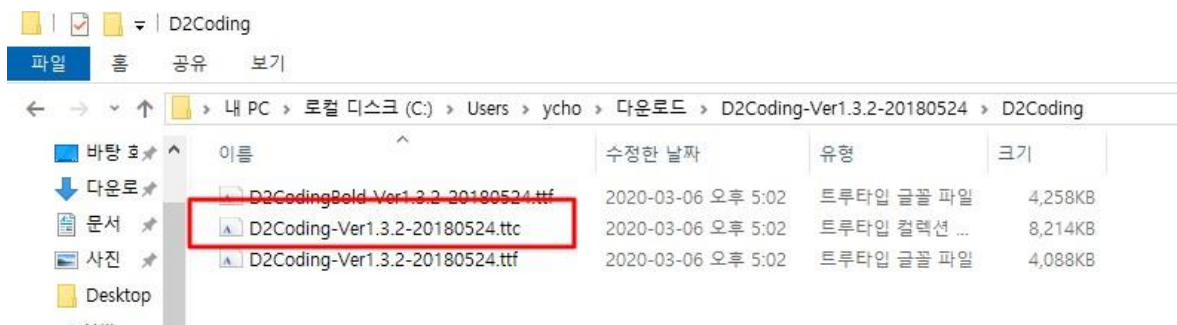
폰트를 설치하려면 D2Coding 사이트(<https://github.com/naver/d2codingfont/releases>)에서 폰트 파일 최신 버전을 다운로드 받는다. 2020년 3월 6일 현재 최신 버전은 D2Coding-Ver1.3.2-20180524.zip과 일이지만, 개선된 버전이 있으면 새로운 파일을 받는다.



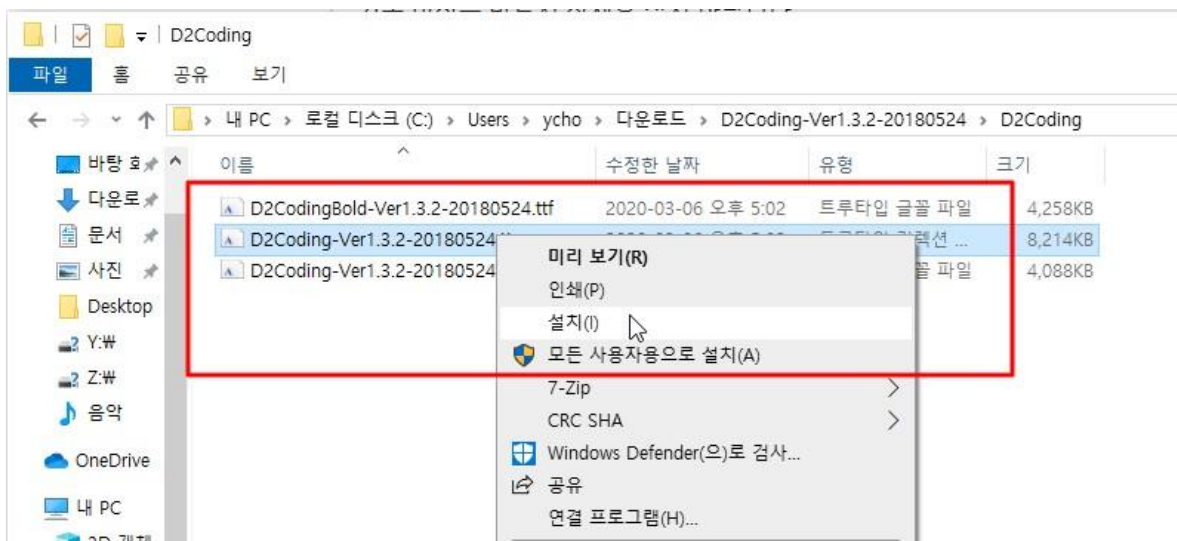
다운로드 받은 압축 파일을 풀면 3개 폴더가 나온다. D2Coding을 선택하자.



세 개 파일이 있을텐데, 이중에서 확장자가 ttc(혹은 유형이 트루타입 컬렉션)인 파일을 선택하거나 확장자가 ttf(유형이 트루타입 글꼴 파일)인 두 개 파일을 선택해서 설치해도 된다. ttc 파일은 두 개 ttf 를 한 파일에 모아놓은 것이다.



다음 그림은 확장자가 ttc 인 파일을 선택하고 마우스의 오른쪽 버튼을 눌러서 설치 메뉴를 선택하는 것을 보인다. 설치 된 후에는 비주얼 스튜디오 코드 또는 인텔리제이에서 사용할 수 있도록 해보자.

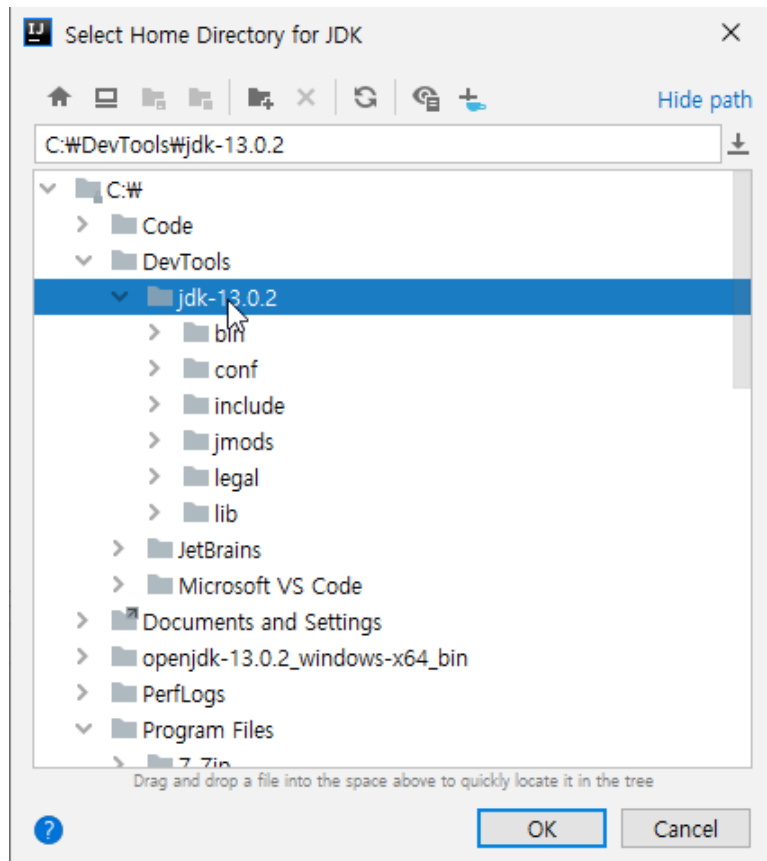


## 인텔리제이 IDEA 에서 JShell 사용하기

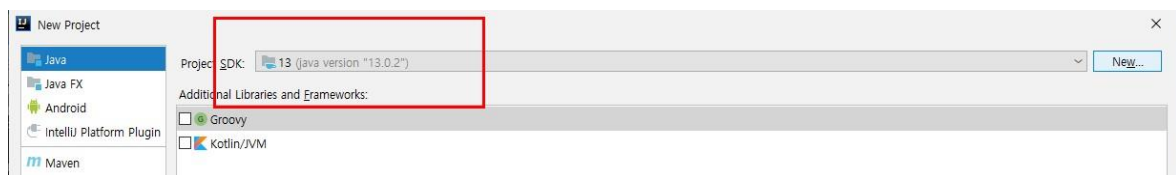
앞에서 보았던 것처럼 인텔리제이에는 JShell Console 이라는 것이 있다. 하지만, 한글 출력도 안되고, 또 사용자로부터 입력을 받아야 하는 경우들이 있는데, 그럴 때에는 제대로 동작 안 된다. 그래서 인텔리제이에서도 비주얼 스튜디오 코드와 비슷하게 터미널에서 JShell 을 실행시키고 임시 자바 파일을 이용하여 코드를 작성하는 형태로 사용한다.

인텔리제이 터미널 화면에서 앞에서 설치했던 D2Coding 폰트를 이용하도록 수정하자. File 메뉴의 Settings...를 선택하거나 Ctrl+Alt+S 키를 누른다. 다음 그림에서 보인 것처럼 왼쪽에서 Console Font, "Use console font instead of default"를 선택하고 Font 의 종류를 D2Coding 으로 바꾼다.

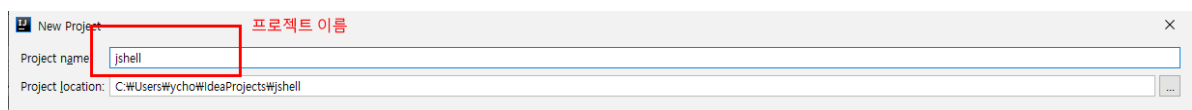




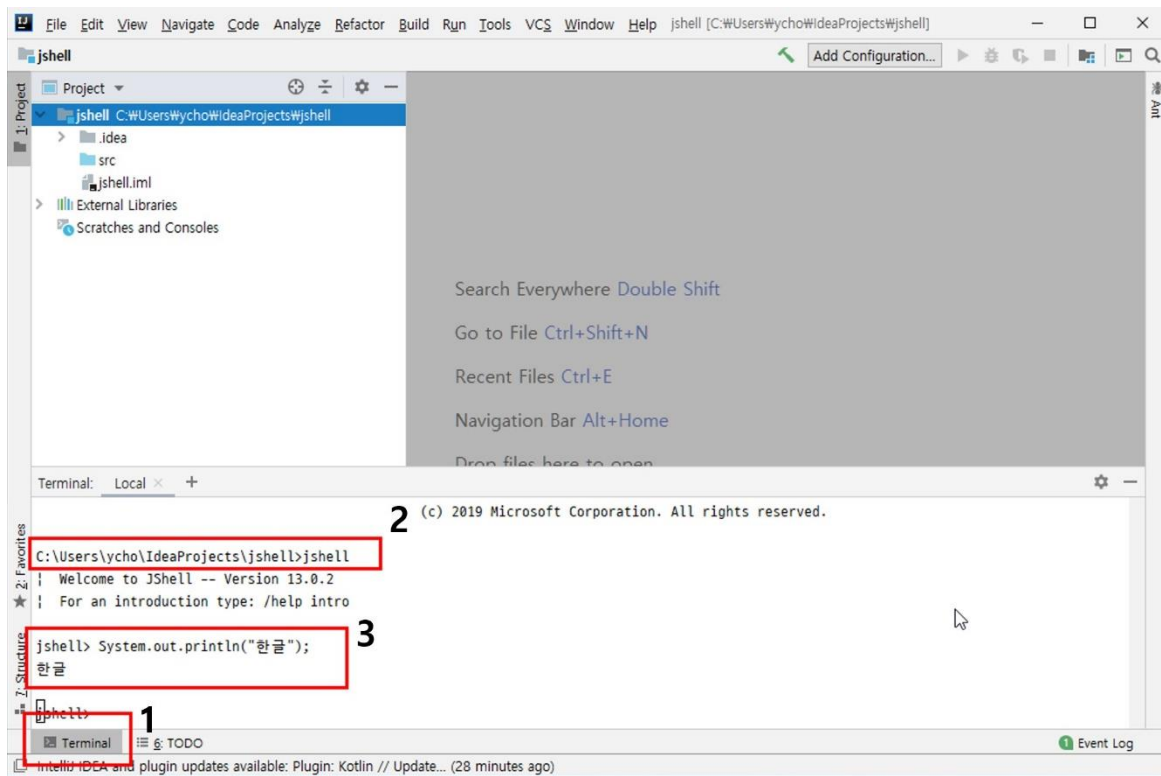
Project SDK 가 바뀐 것 확인한다.



다음 그림에서 보인 것처럼 JShell 같은 이름으로 생성한다.

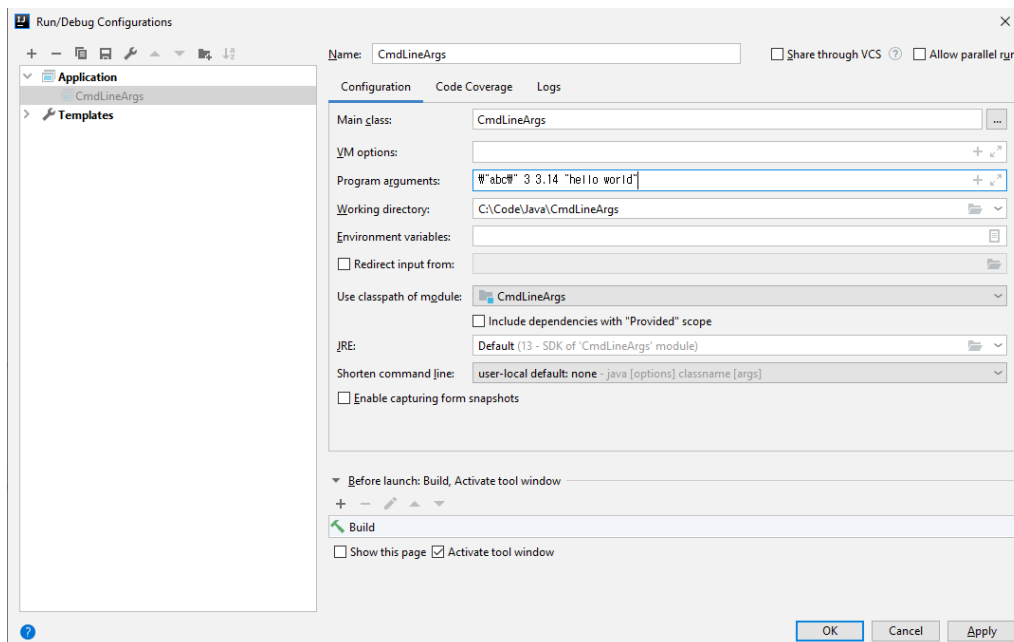


프로젝트를 생성한 후에는 IntelliJ의 화면 하단에 있는 Terminal 버튼을 클릭해서 터미널화면을 생성하고, JShell 을 실행시킨다. 마지막으로 한글을 출력하는 코드를 작성해서 실행시켜본다. 제대로 출력된다면 설치는 끝난 셈이다. 비주얼 스튜디오 코드와 마찬가지로 자바 코드 파일을 에디터에 열고 좀 긴 코드는 에디터에서 작성하고 터미널에서 실행되는 jshell 에 복사해서 붙여넣는 형태로 작성한다.



## 명령행 인자 지정

인텔리제이 IDEA 에서 명령행 인자를 입력하고 싶으면 메뉴에서 “Run|Edit configurations...”를 실행시킨 후에 다음 화면에서 보인 것처럼 “Program arguments” 부분에 내용을 입력하면 된다.



다음은 앞에서 작성했던 코드를 인텔리제이 IDEA 에서 실행시킨 화면의 일부를 보인다.

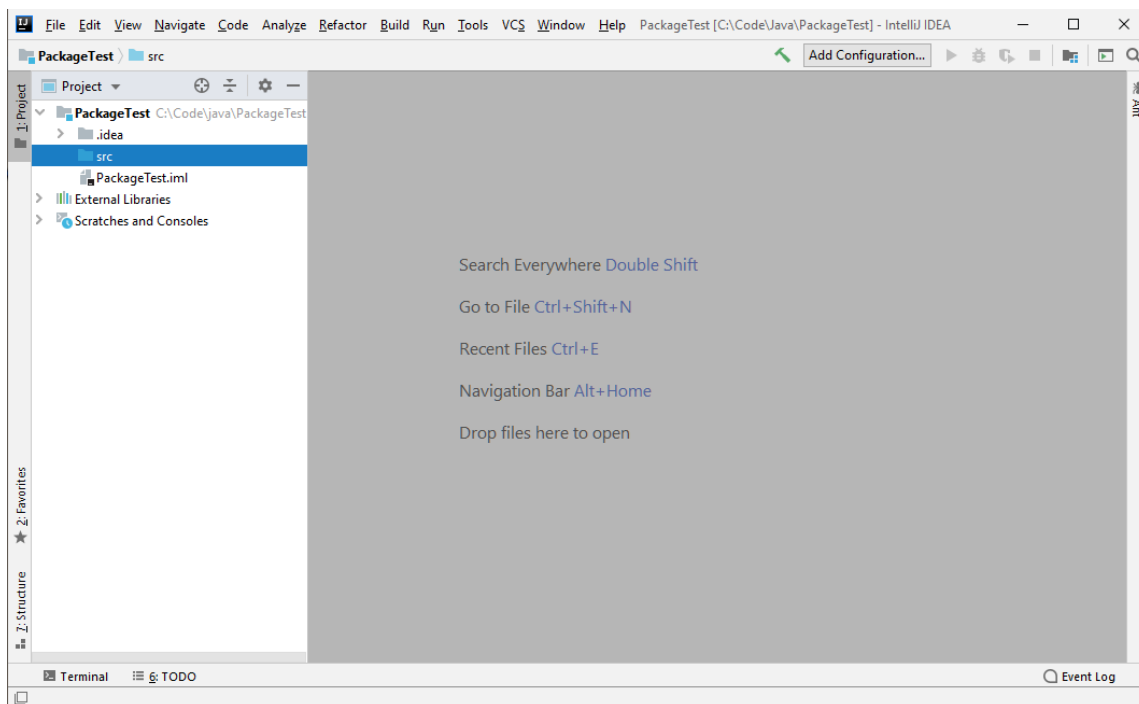


```
Run: CmdLineArgs x
C:\DevTools\jdk-13.0.2\bin\java.exe "-javaagent:C:\DevTools\JetBrains\IntelliJ IDEA Community Edition 2019.3.3\lib
args[0] = "abc"
args[1] = 3
args[2] = 3.14
args[3] = hello world

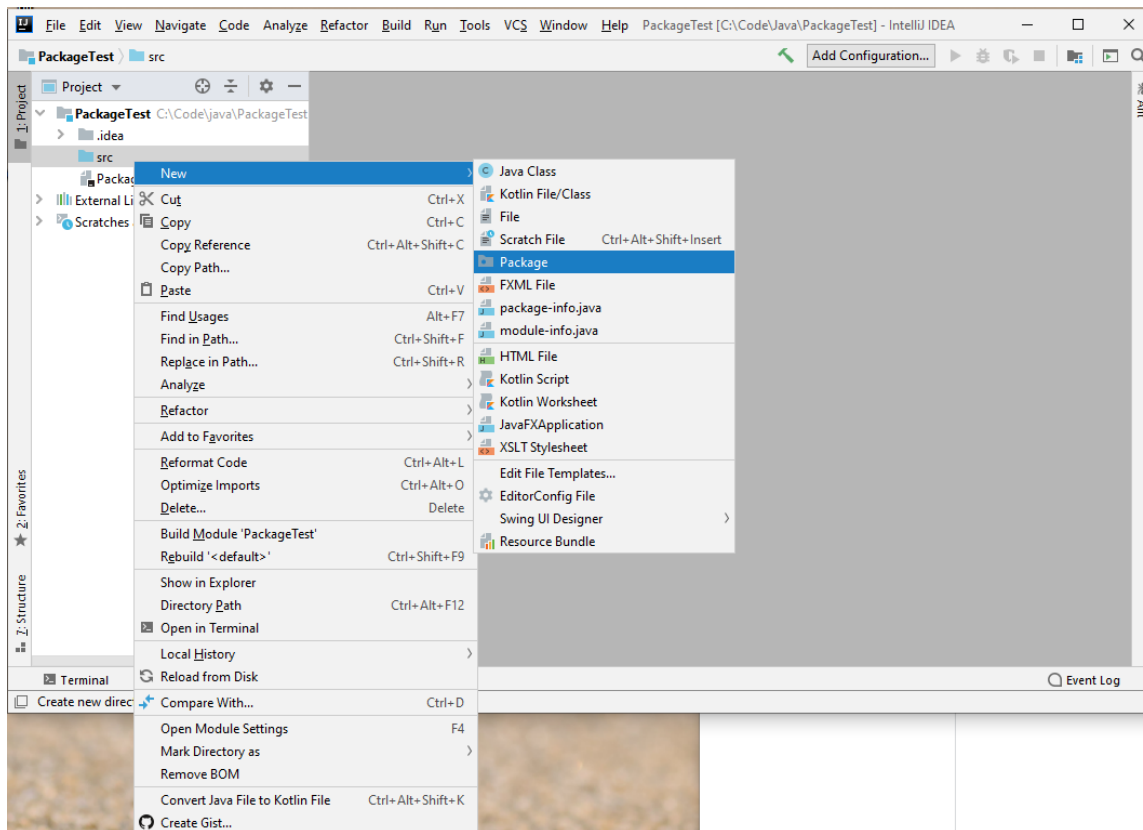
Process finished with exit code 0
```

## 인텔리제이 IDEA 에서 패키지 생성하고 컴파일하기

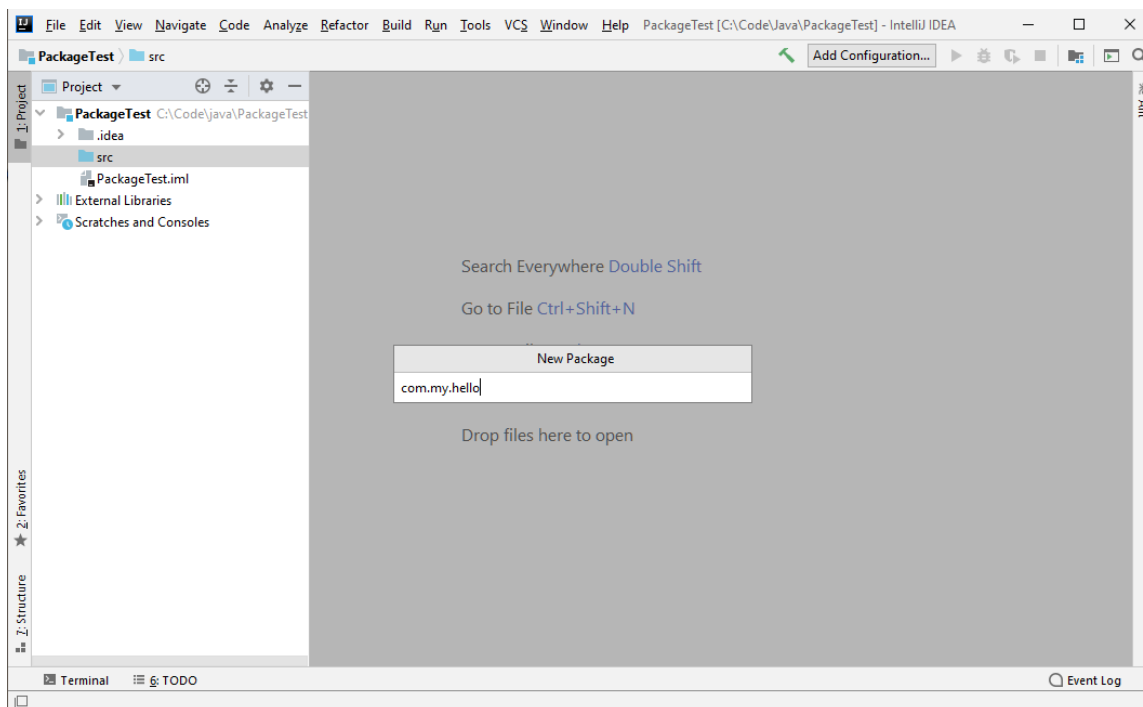
이번에는 인텔리제이 IDEA 에서 패키지를 생성하고 컴파일 해본다. 먼저 프로젝트를 먼저 생성한다. 여기에서는 PackageTest 라는 이름으로 프로젝트를 생성하였다. 다음 화면은 프로젝트가 만들어진 뒤의 IDEA 화면의 일부를 보인다.



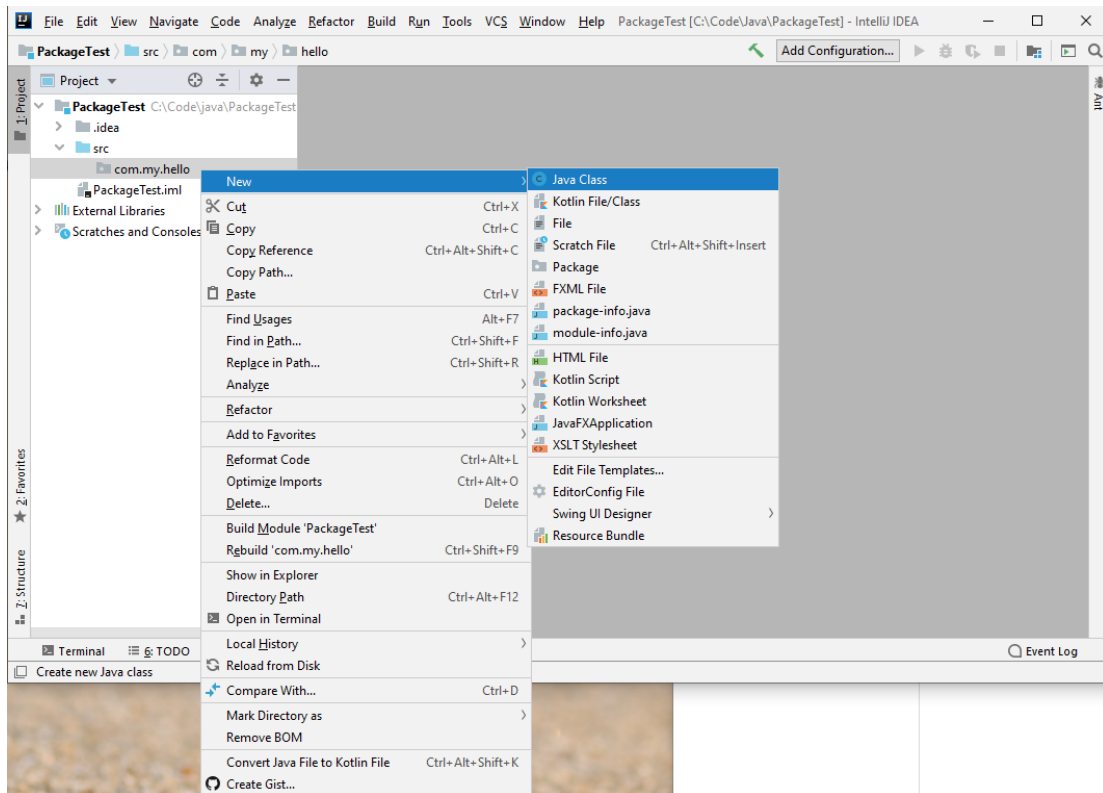
다음 그림에 보인 것처럼 화면 왼쪽에 있는 프로젝트 뷰에서 src 를 선택 -> 오른쪽 버튼 클릭 -> New -> Package 를 선택한다.



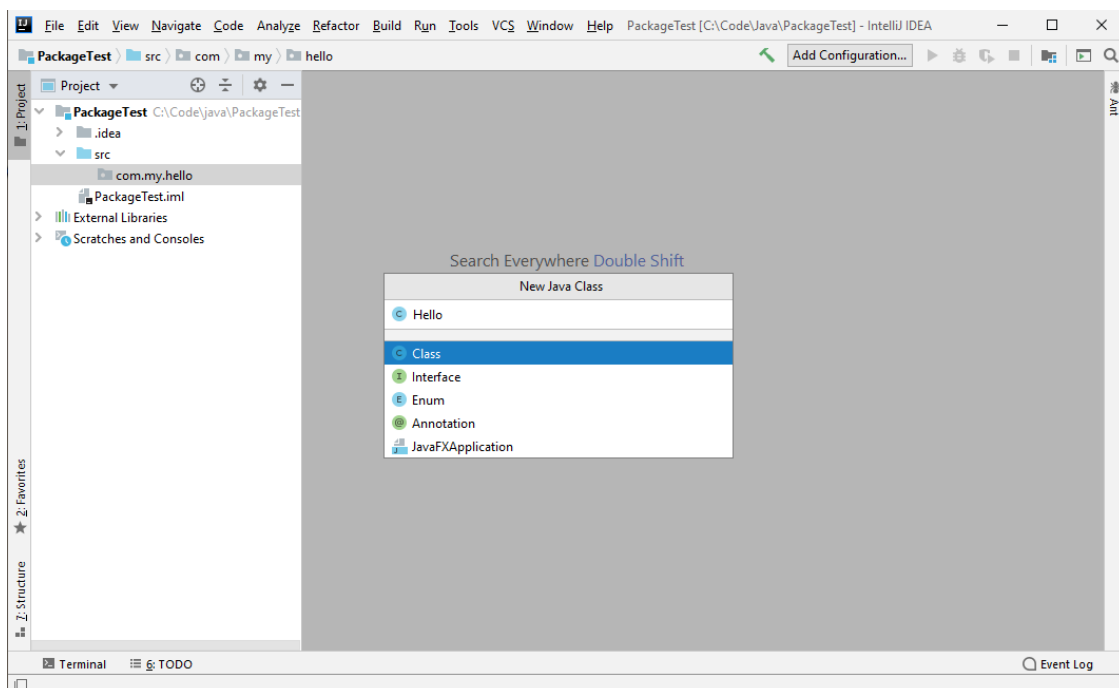
패키지 이름을 입력한다. 여기서는 먼저 com.my.hello 부터 만들기로 한다.



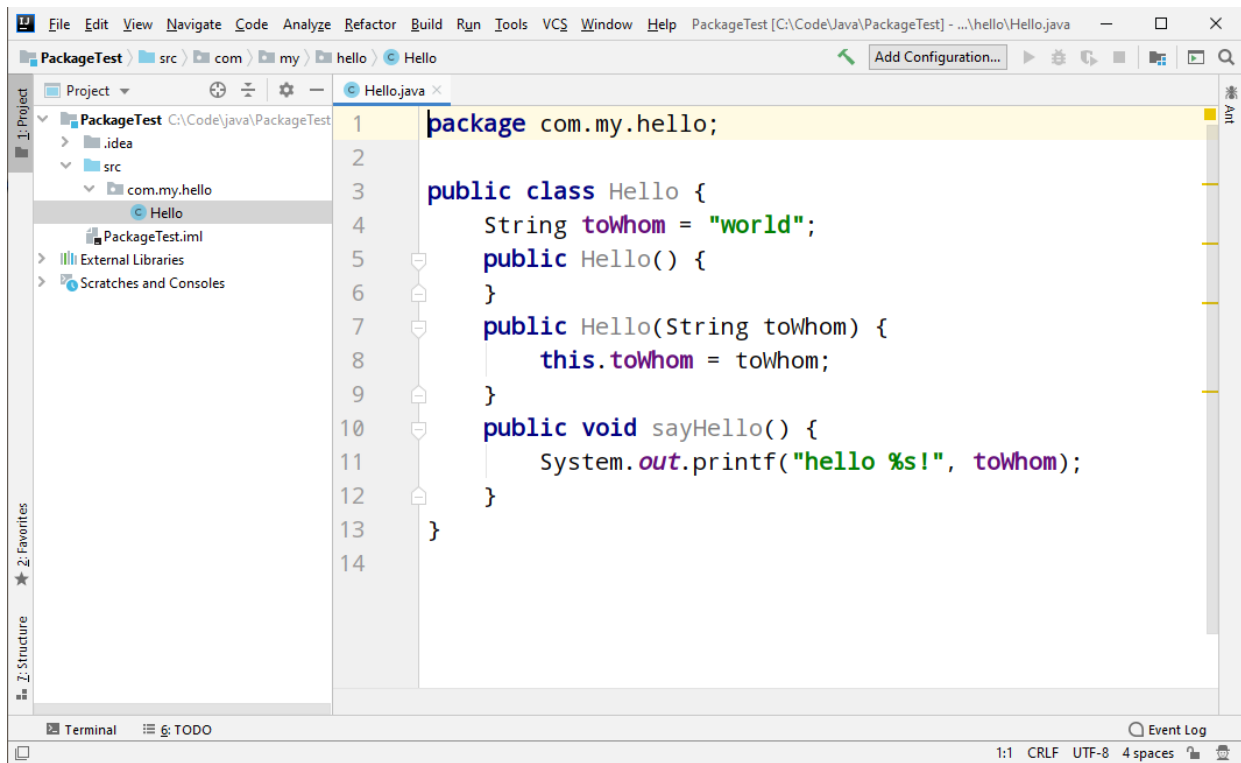
다음에는 클래스를 만든다. 앞에서 만들었던 패키지를 선택 -> 오른쪽 버튼 클릭 -> New -> Class 를 선택한다.



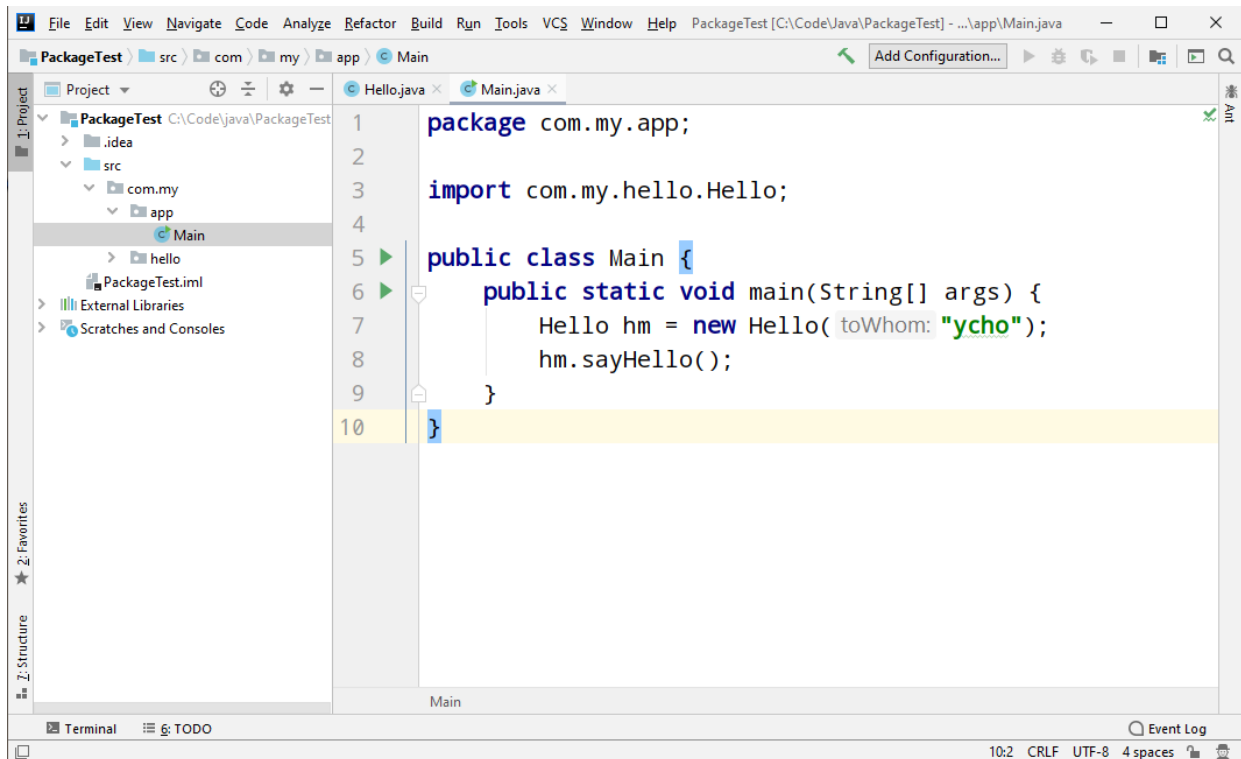
Hello 클래스를 만들고 입력한다.



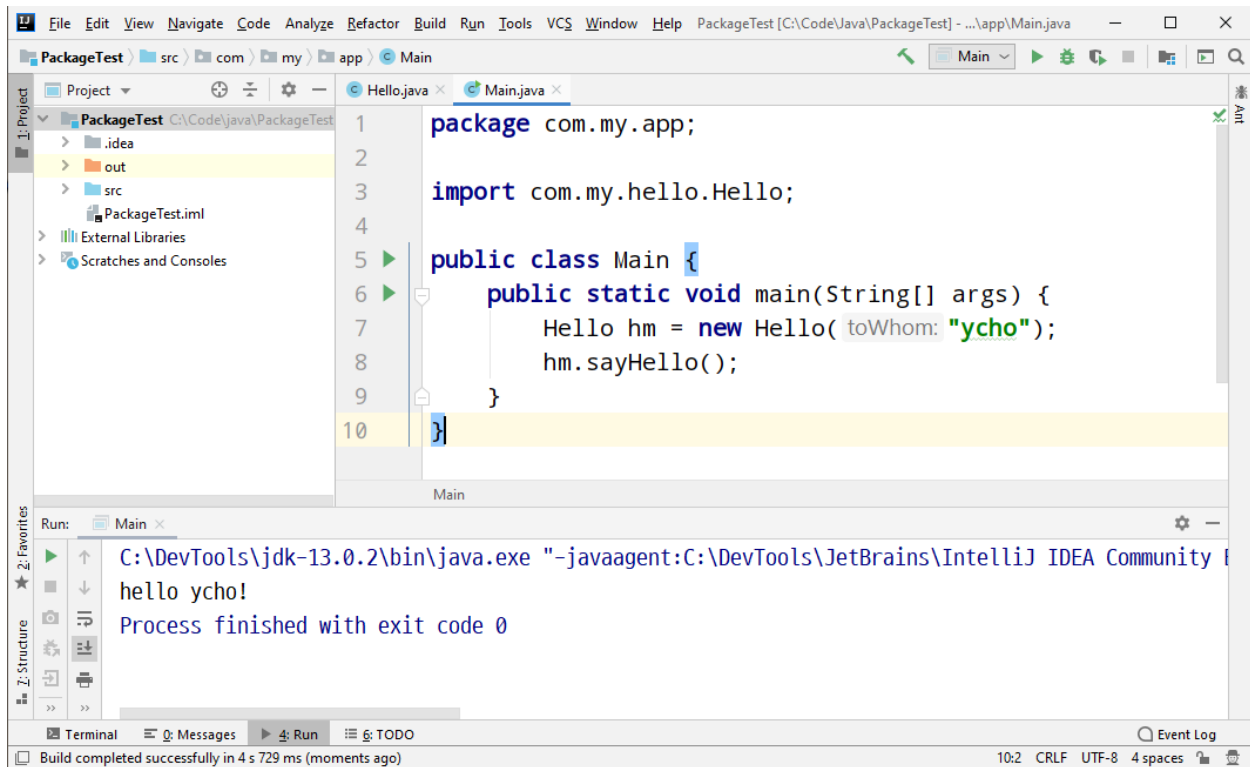




앞에서 했던 방법대로 com.my.app 패키지를 생성하고, Main 클래스를 구현한다.

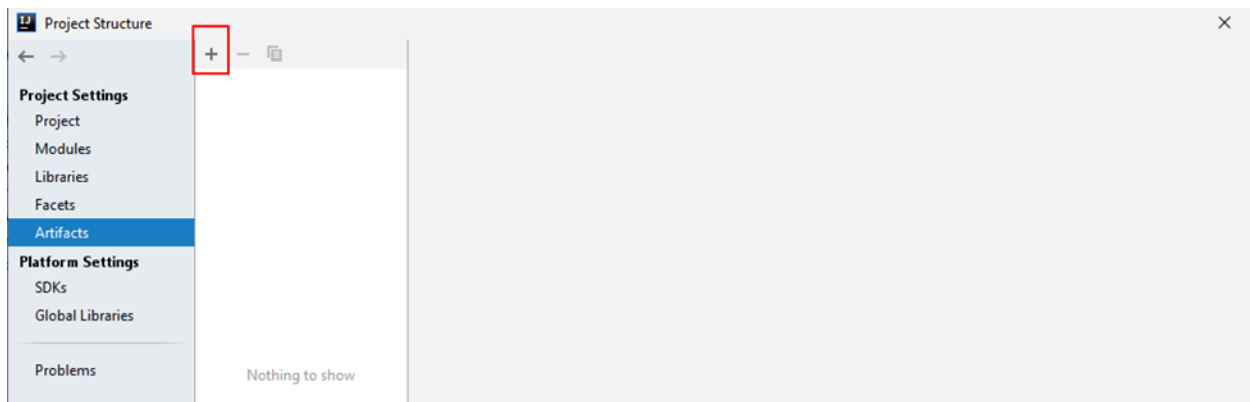


Main 클래스를 실행시킨다.

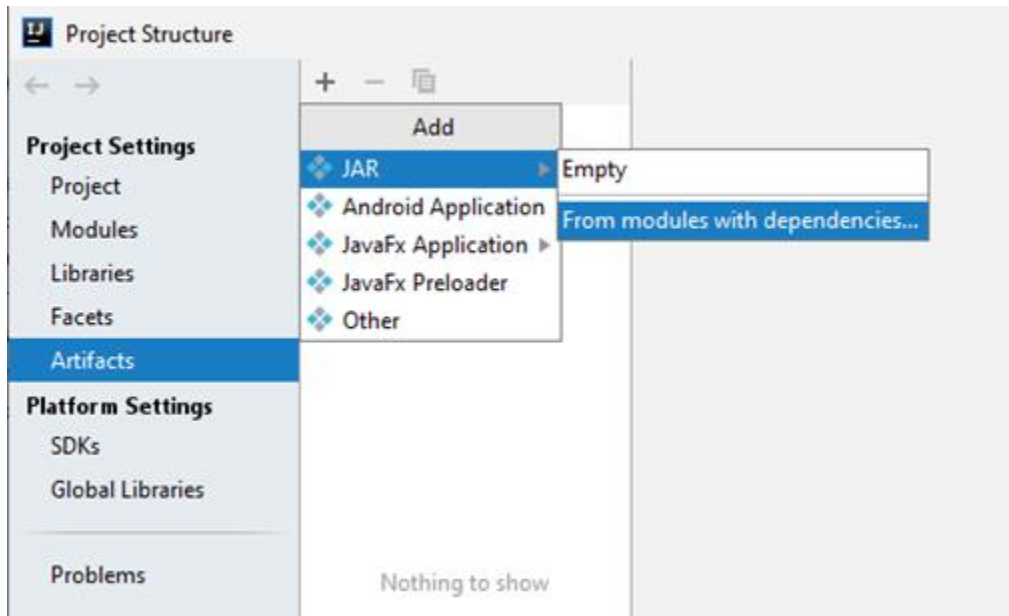


## 인텔리제이 IDEA 에서 jar 파일 생성하기

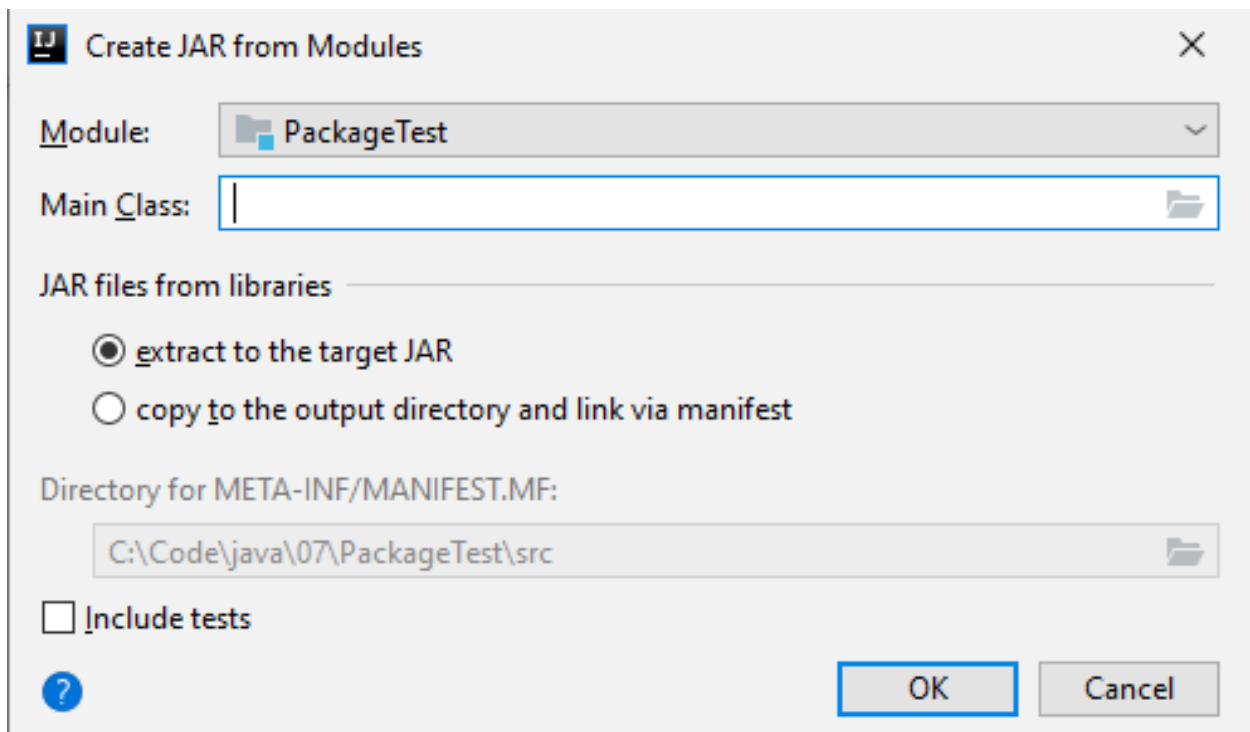
인텔리제이 IDEA 에서 프로그램을 구현한 후에 jar 파일로 만들어본다. 먼저 "File" 메뉴의 "Project Structure..."를 선택한다. 그림에서 보인 것처럼 "Project Structure" 대화 상자 왼쪽에서 "Artifacts"를 선택하고, 가운데 부분에서 '+'를 선택하고 OK 버튼을 클릭한다..

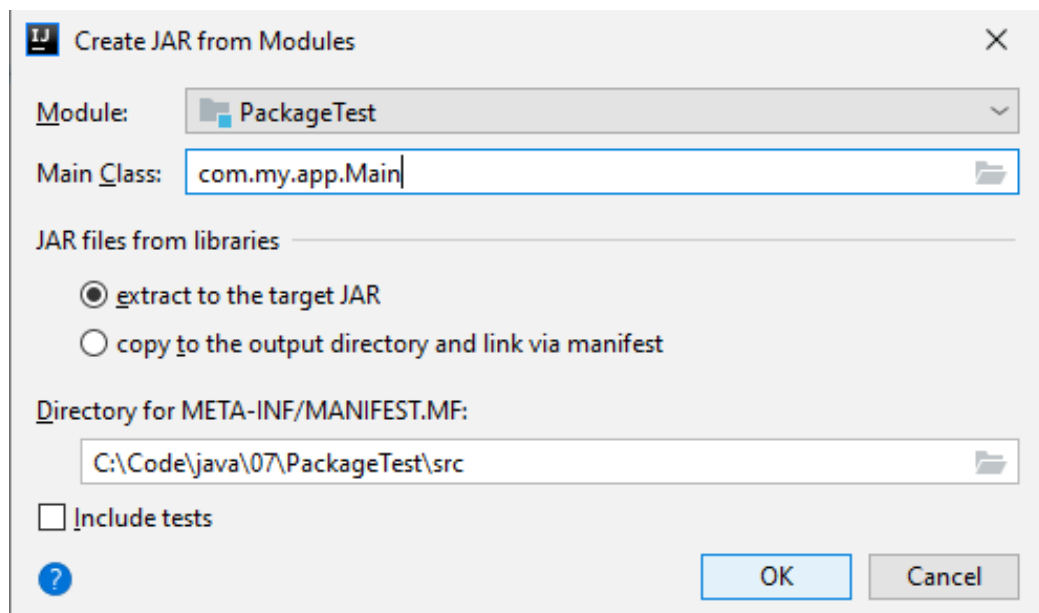
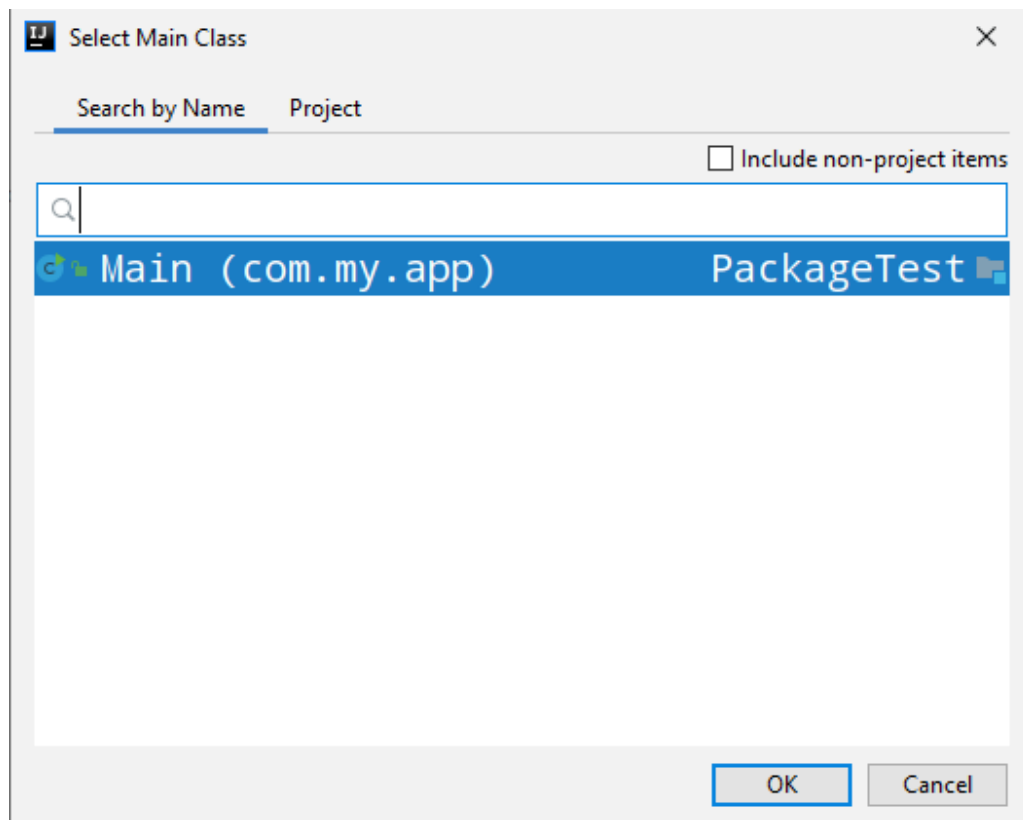


'+'를 누르면 나타나는 메뉴에서 JAR -> "From modules with dependencies..."를 선택한다.

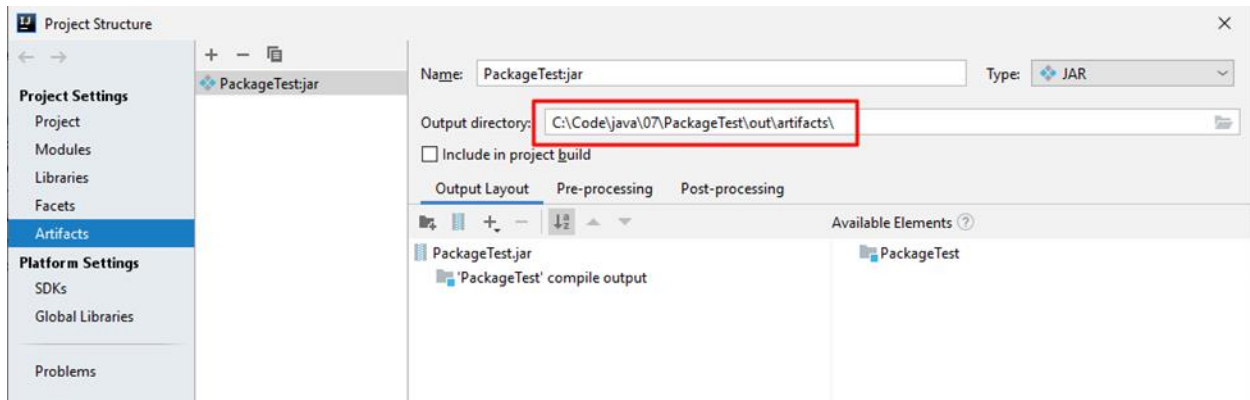


대화 상자에서 main() 함수가 있는 클래스를 선택한다. 여기서는 com.my.app 의 Main 클래스를 선택한다.



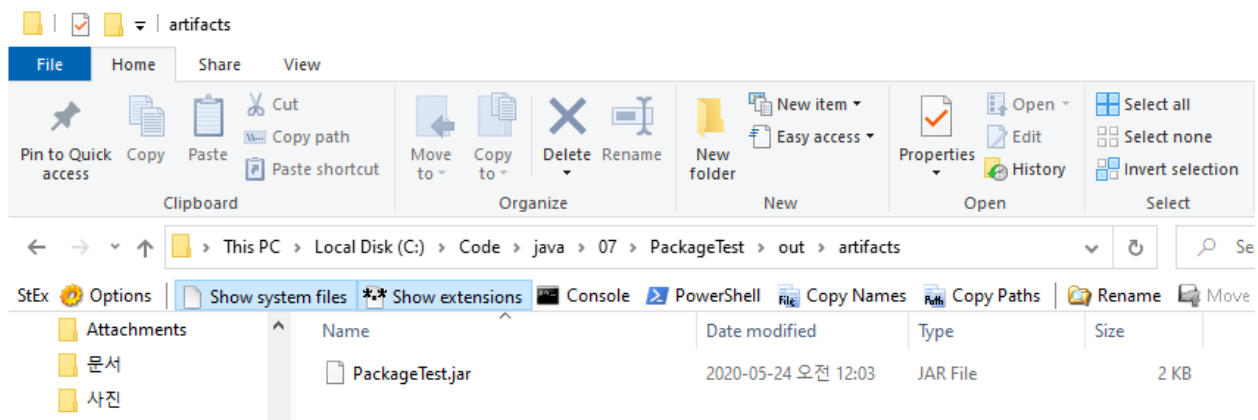


jar 파일을 생성할 경로를 확인하고 "OK" 버튼을 누른다.



메뉴에서 "Build" -> "Build Artifacts..."를 선택한다.

경로에 프로젝트 이름으로 jar 파일이 생성되었음을 확인할 수 있다.

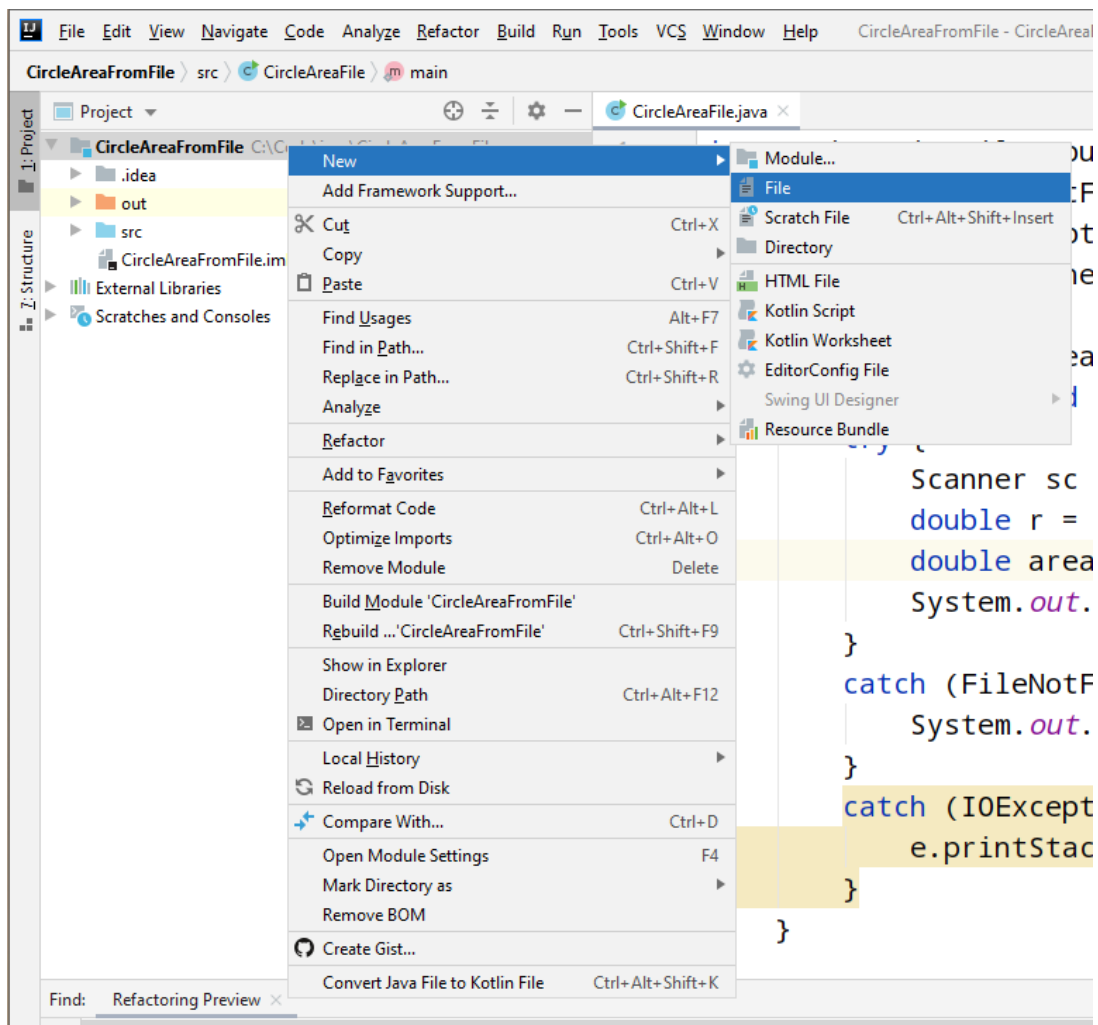


다음은 실행 결과를 보인다.

```
C:\Code\java\06\PackageTest\out\artifacts>java -jar PackageTest.jar  
hello ycho!
```

## 인텔리제이 IDEA 에서 파일 입출력 프로그램을 작성할 때 데이터 파일 위치

프로젝트 생성 후에 circle.txt 파일에서 입력을 받는다고 가정하자. 여기서는 인텔리제이 IDEA 에서 자바 프로그램을 실행시키면서 데이터 파일을 읽기 위해 데이터 파일을 생성하는 것을 보인다. 다음 그림에 보면 것처럼 프로젝트가 위치한 곳에 데이터 파일을 생성한다. 이를 위해 프로젝트를 선택한 후에 마우스의 오른쪽 버튼을 누르고 새로운 파일(이 경우에는 텍스트 파일)을 생성해서 반지름 정보를 넣는다. 만약 다른 종류의 데이터 파일이라면 프로젝트가 위치한 폴더에 파일을 복사한다.



다음 그림은 circle.txt 가 생성되었음을 보인다.

