

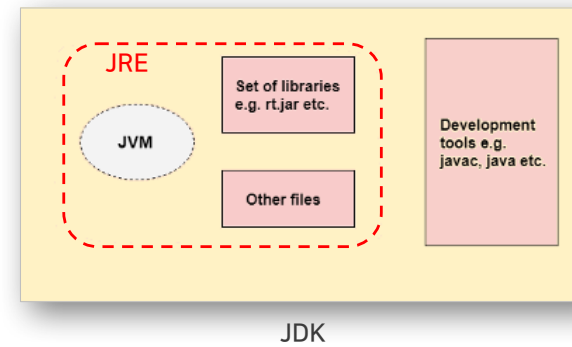
01_3 Environment Setup

Object-Oriented Programming

이번 강의에서는 자바 프로그램을 개발하는데 필요한 소프트웨어 환경을 셋업 합니다.
어렵지 않게 설치하고 사용할 수 있으니, 잘 따라와 주시기 바랍니다.

JDK and JRE

- JDK (Java Development Kit)
 - Including tools for developers
 - JVM, Java Compiler, ...
- JRE (Java Runtime Environment)
 - Execution environment for program users
 - Including JVM but no development tools
 - $JRE \subset JDK$
- Distributing JDK and JRE
 - Oracle's Java Technology Site
 - <https://www.oracle.com/java/technologies/>



Java Development Kit (JDK) 는 JVM 뿐아니라 자바 프로그램 개발자를 위한 여러가지 tool들을 포함하고 있는데, 예를 들면, javac, 즉, Java compiler, java, 즉, byte code interpreter 등입니다.

JDK 중에서 개발자가 아닌 프로그램 사용자만을 위한 Subset을 추린 것을 Java Runtime Environment (JRE) 라고 합니다.

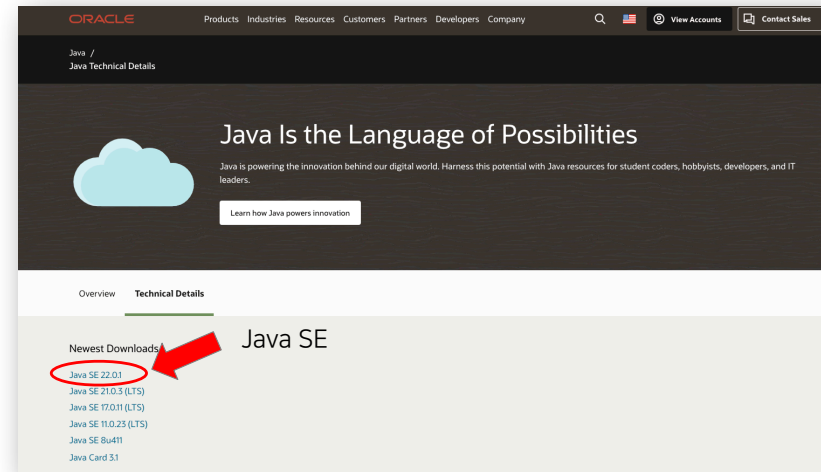
즉, 사용자는 프로그램을 개발하는 사람은 아니기 때문에 개발에만 사용되는 tool들을 JDK로 부터 제외한 것이 JRE라고 볼 수 있습니다.

간혹 Java로 개발된 프로그램을 Install하여 실행시키려 할 때, JRE가 없으면 다운로드를 받으라는 안내가 나오는데, Java로 만들어진 프로그램의 실행에 있어서, JRE는 필수이기 때문입니다.

JDK와 JRE는 Java의 소유사인 Oracle의 Java Technology Site에 가면 다운로드 받을 수 있습니다.

Installing JDK - 1

- <https://www.oracle.com/java/technologies/>



JDK를 다운 받기 위해, Oracle Technology 사이트에 가서 JavaSE의 최신 버전을 click 합니다.

Installing JDK - 2

- Download appropriate installation file for your platform:
- Windows
 - x64 installer
 - https://download.oracle.com/java/22/latest/jdk-22_windows-x64_bin.exe
- macOS
 - For M1, M2, ... (ARM) Chips
 - https://download.oracle.com/java/22/latest/jdk-22_macos-aarch64_bin.dmg
 - For Intel Chips:
 - https://download.oracle.com/java/22/latest/jdk-22_macos-x64_bin.dmg

사용자의 OS가 Windows, macOS M Chips, macOS intel chips인지에 따라 적절한 JDK를 다운받아 설치합니다.

Checking JDK Installation

- Executing Console (Command prompt, Terminal)
 - Windows
 - Windows Key + R > type "cmd" > confirm
 - Start menu > Windows system > command prompt
 - Mac: terminal.app
- Type `java -version` and `javac -version` to confirm the installation

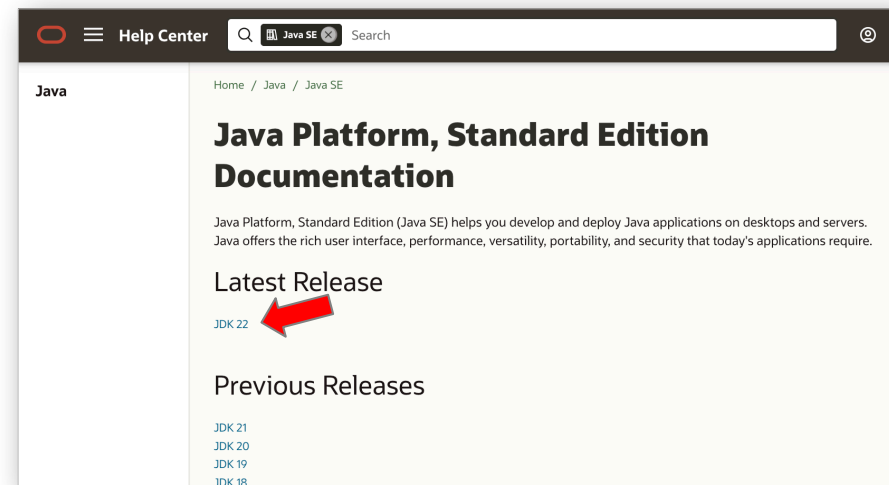
```
C:\> java -version
java version "17.0.1" 2021-10-19 LTS

C:\> javac -version
javac 17.0.1
```

JDK가 제대로 설치되었는지 check하기 위해 console window를 띄우고 거기에서 java -version 이라는 명령어를 실행합니다.
Java version이 표시되면 JDK가 정확히 설치된 것입니다.

JavaSE Documentation

<https://docs.oracle.com/en/java/javase>

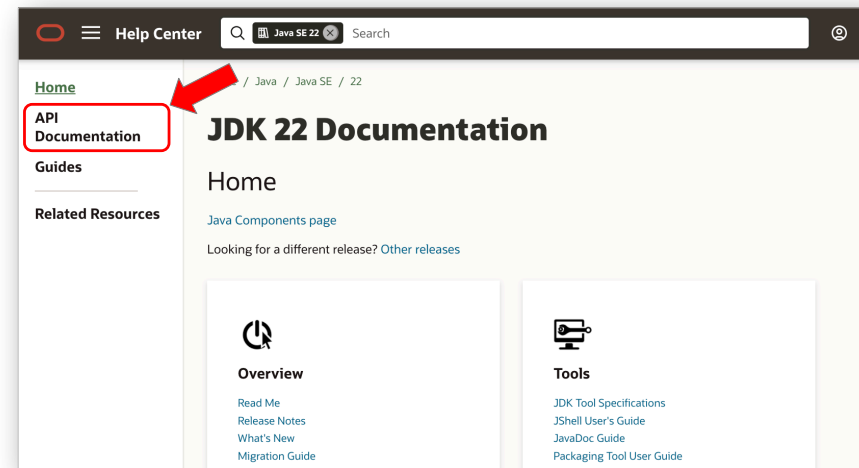


JDK에 포함된 API들의 documentation은 Oracle에서 online으로 제공하고 있습니다.
docs.oracle.com/en/java/javase 를 방문하여 설치한 JDK version을 click 합니다.

JDK Documentation

- Tools, Tutorials, API Documentation, ...

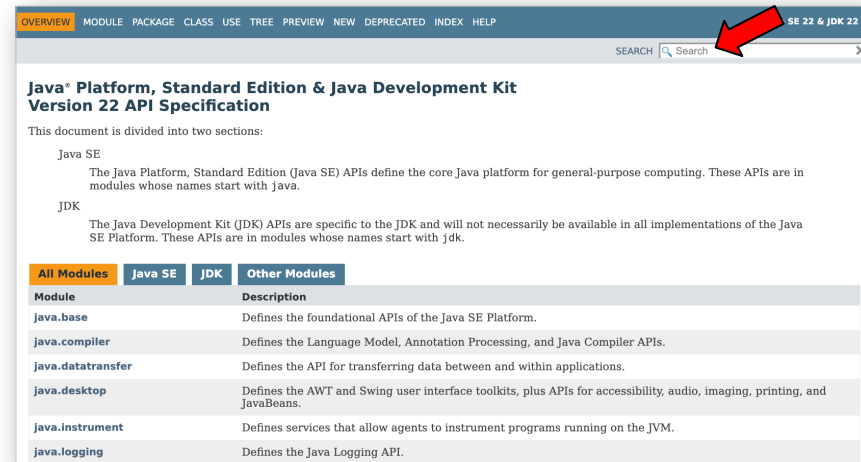
ex) <https://docs.oracle.com/en/java/javase/22>



왼쪽 메뉴 영역에서 API Documentation을 선택합니다.

API Documentation (1/3)

- ex) <https://docs.oracle.com/en/java/javase/22/docs/api/index.html>
- Try Search 'System' for the reference of the class the class 'System'



API Documentation page에서 search 창에 원하는 class의 이름을 typing하여 search 합니다.
예를 들면 'System' 이라는 class를 search 해 볼 수 있습니다.

API Documentation (2/3)

The screenshot shows the Java API documentation for the `System` class. The navigation bar at the top includes links for OVERVIEW, MODULE, PACKAGE, CLASS (highlighted), USE, TREE, PREVIEW, NEW, DEPRECATED, INDEX, and HELP. Below this, there are tabs for SUMMARY, NESTED, FIELD, CONSTR, and METHOD. The main content area shows the package hierarchy: `Module java.base`, `Package java.lang` (highlighted with a red box and an arrow pointing to it with the text "package including the class"), and `Class System` (highlighted with a red box). Below the class name, it shows the inheritance hierarchy: `java.lang.Object` and `java.lang.System`. The class declaration is shown as `public final class System extends Object`. A description follows: "The System class contains several useful class fields and methods. It cannot be instantiated. Among the facilities it provides are: a means of loading files and libraries; a means of accessing the environment; and a means of accessing system resources." The "Since" section indicates version 1.0. The "Field Summary" section is highlighted with a red box and an arrow pointing to it with the text "Fields: variables". It contains a table with the following data:

Fields	Modifier and Type	Field	Description
	static final <code>PrintStream</code>	<code>err</code>	The "standard" error output stream.
	static final <code>InputStream</code>	<code>in</code>	The "standard" input stream.
	static final <code>PrintStream</code>	<code>out</code>	The "standard" output stream.

'System' class에 대한 documentation이 표시됩니다.
Class 이름 위에는 class가 속한 package의 이름이 표시 됩니다.
예를 들면 System class는 java.lang package 안에 있습니다.
Fields에는 class에 속하는 instance variable이나 constant들에 대한 설명이 있습니다.

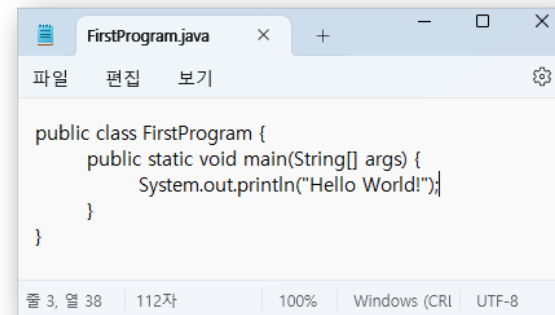
API Documentation (3/3)

Method Summary

All Methods	Static Methods	Concrete Methods	Deprecated Methods
Modifier and Type	Method	Description	
static void	arraycopy(Object src, int srcPos, Object dest, int destPos, int length)	Copies an array from the specified source to the specified destination.	
static String	clearProperty(String key)	Removes the property with the specified key.	
static Console	console()	Returns the system console, or null if none.	
static long	currentTimeMillis()	Returns the current time in milliseconds.	
static void	exit(int status)	Initiates the shutdown of the virtual machine.	
static void	gc()	Runs the garbage collection.	
static Map<String,String>	getenv()	Returns an unmodifiable map of environment variables.	
static String	getenv(String name)	Gets the value of the environment variable with the specified name.	
static System.Logger	getLogger(String name)	Returns an instance of the logger with the specified name.	
static System.Logger	getLogger(String name, ResourceBundle bundle)	Returns a localized instance of the logger with the specified name.	

그 아래에는 class에 속하는 method들이 나열되어 사용법을 볼 수 있습니다.
method의 return type, parameter들과 어떤 작업을 하는지를 볼 수 있는데
각 method 이름을 click하면 더 자세한 설명을 볼 수 있습니다.

Java Program Development Using JDK - 1



The screenshot shows a text editor window titled "FirstProgram.java". The menu bar includes "파일" (File), "편집" (Edit), and "보기" (View). The code content is as follows:

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

The status bar at the bottom indicates "줄 3, 열 38" (Line 3, Column 38), "112자" (112 characters), "100%", "Windows (CRLF)", and "UTF-8".

- Type the program using Notepad (Windows) or Text editor (macOS)
- Save the program as "FirstProgram.java" (the same name as the class)

뒤에 설치할 IDE, 즉, Integrated Development Environment 를 사용해 보기 전에 Console window에서 Java source program을 compile하고 실행하는 과정을 살펴 보겠습니다. 적당한 text editor로 java 프로그램의 source code를 작성합니다. 이 예제에서는 FirstProgram.java 라는 class source 파일을 작성하였습니다. class의 이름과 source file의 이름이 일치해야 합니다.

Java Program Development Using JDK - 2

```
PS C:\Users\iklee\java> javac FirstProgram.java
PS C:\Users\iklee\java> dir

디렉터리: C:\Users\iklee\java

Mode                LastWriteTime         Length Name
----                -
-a-----         2024-06-23 오후 11:26            430 FirstProgram.class
-a-----         2024-06-23 오후 11:22            116 FirstProgram.java

PS C:\Users\iklee\java> java FirstProgram
Hello World!
PS C:\Users\iklee\java>
```

"ls" command instead of "dir" in macOS

- Executing cmd (terminal in macOS)
- "javac FirstProgram.java" (compilation). Output is 'FirstProgram.class' (byte-code program)
- "java FirstProgram" (interpretation: executing the byte-code program on JVM)

12

Console 에서 source code를 byte code로 compile하기 위해

"Javac FirstProgram.java" 명령을 typing하여 실행합니다.

Compile에 성공하면 FirstProgram.class 라는 bytecode 프로그램이 생성된 것을 확인할 수 있습니다.

현재 폴더 안의 파일 리스트를 확인하기 위해 Windows에서는 "dir", macOS에서는 "ls" 명령을 사용할 수 있습니다.

이제 Byte code 프로그램을 실행하기 위해

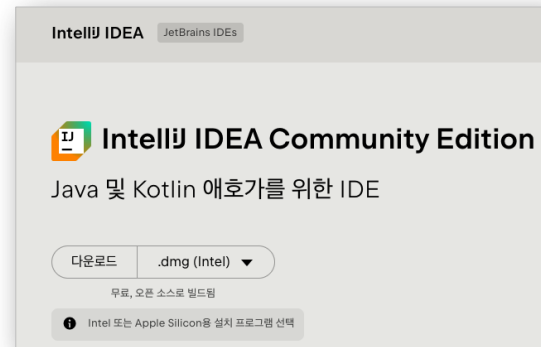
JVM interpreter "java FirstProgram" 명령을 실행합니다.

Byte code 프로그램 파일 이름인 FirstProgram.class 에서 extension인 .class는 제외하고 명령을 typing해야 합니다.

JVM 이 실행되면서 "Hello World!" 라는 output이 console에 출력되는 것을 볼 수 있습니다.

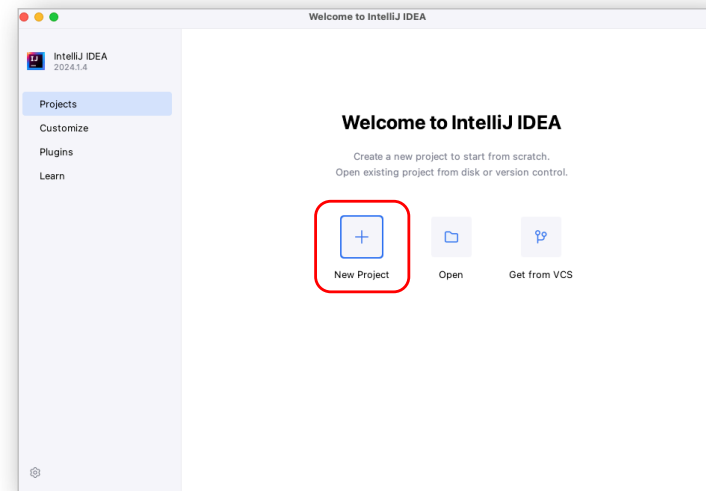
Installing IntelliJ IDEA

- <https://www.jetbrains.com/ko-kr/idea/download>
- Download “Community Edition” and Install it.
 - Windows, Intel mac, Apple Silicon mac



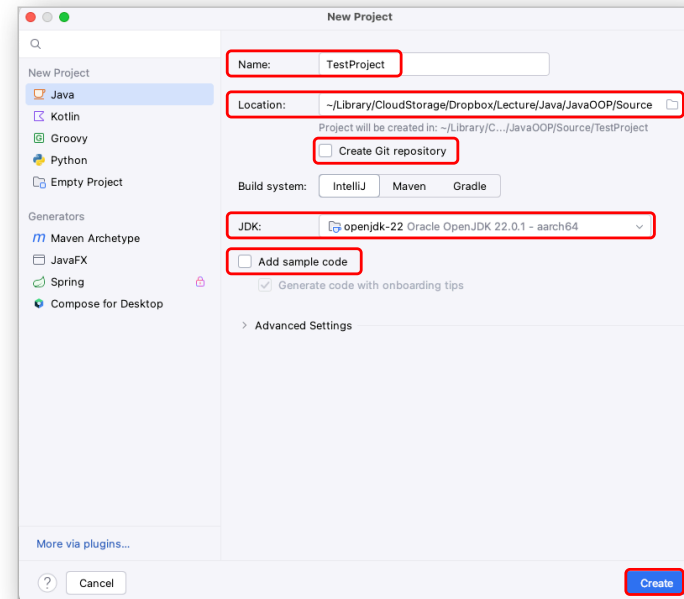
이제 IDE, 즉, Integrated Development Environment 인 IntelliJ IDEA를 설치해 보겠습니다.
jetbrains사의 홈페이지에서 IntelliJ IDEA의 "community edition"을 다운 받아 설치하십시오.
Community version은 무료 버전으로 본 코스 정도의 학습에는 충분한 기능을 갖추고 있습니다.

Creating New Project (1/3)



IntelliJ IDEA를 설치한 후 실행하여 New Project 버튼을 click합니다.

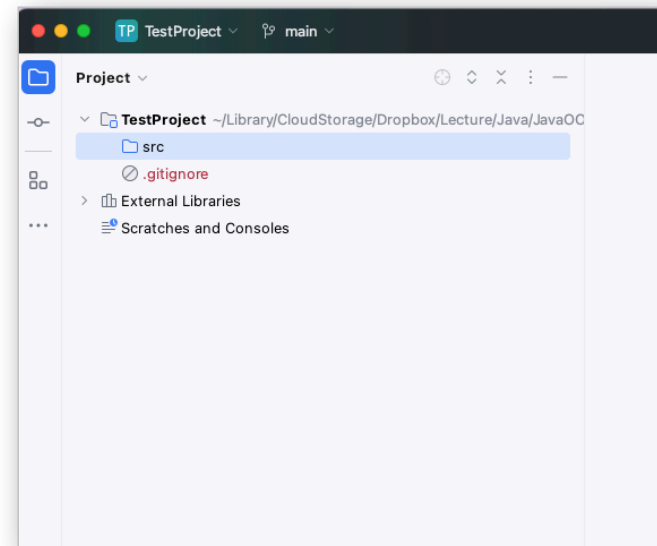
Creating New Project (2/3)



15

프로젝트의 이름을 입력하고
프로젝트 파일들을 저장할 폴더 위치를 선택합니다.
Git을 사용한다면 "Create Git repository"를 check하여 완성된 source code들을 github에 자동으로 upload할 수 있습니다.
여기서는 이 옵션을 선택하지는 않도록 하겠습니다.
여러 개의 JDK version들이 설치되어 있다면 그 중에 원하는 JDK를 선택할 수 있습니다.
보통 가장 최신에 설치한 최신 버전의 JDK를 선택하면 됩니다.
"Add sample code" option은 IntelliJ IDEA가 자동으로 sample code를 생성해 주는 기능입니다만
여기에서는 이 옵션을 선택하지 않도록 하겠습니다.
모든 입력이 끝나면 왼쪽 맨 아래 "Create" button을 눌러 project를 생성합니다.

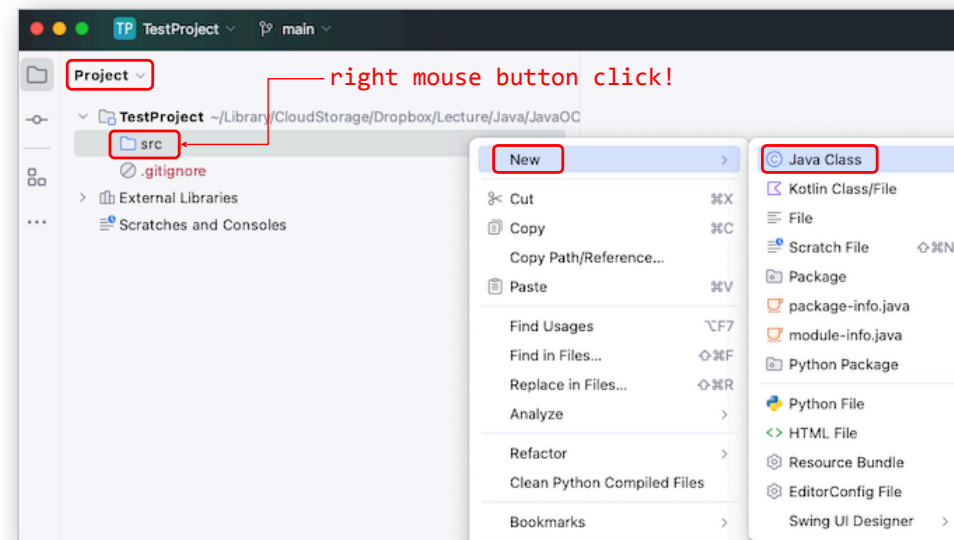
Creating New Project (3/3)



16

이제 방금 생성한 프로젝트 화면으로 전환됩니다.
아직 아무런 source code를 생성하지 않았기 때문에 src 폴더는 비어있습니다.

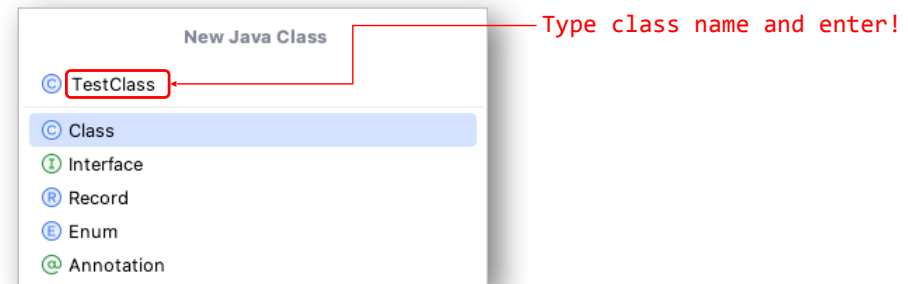
Creating New Class (1/3)



17

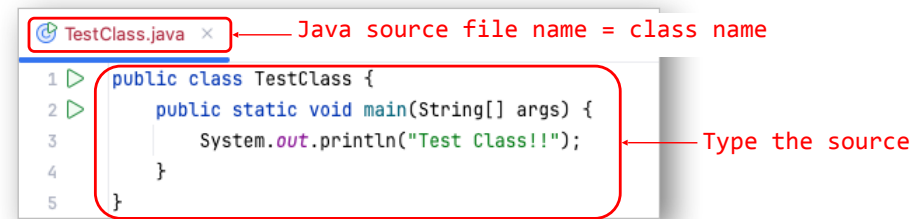
Src 폴더 위에 mouse pointer를 위치시키고 right mouse button을 눌러 popup 메뉴가 나오게 합니다.
메뉴에서 New 를 선택한 후
보여지는 sub menu에서 Java Class를 선택합니다.

Creating New Class (2/3)



New Java Class 창이 나오면 맨 위에 생성할 class 이름을 typing하고 enter를 누릅니다.
여기서는 "TestClass" 라는 class 이름을 입력하였습니다.

Creating New Class (3/3)



이제 "TestClass.java" source file이 생성됩니다.
Class name과 source file의 이름이 반드시 같아야 한다는 점을 다시한번 강조합니다.
이제 source 입력 창에 프로그램 source를 typing하면 됩니다.

Build Project (Compilation)

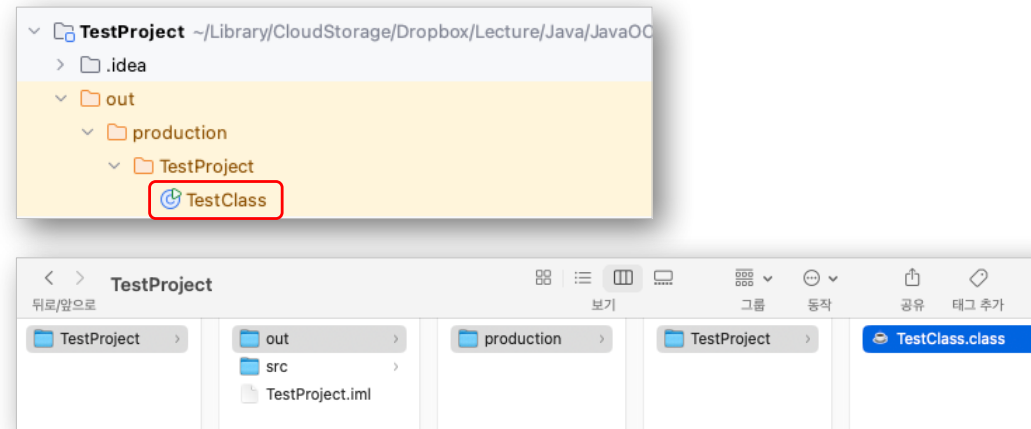
- Java Source to Bytecode
- Build Project
 - Build menu > Build Project
 - Windows
 - Ctrl + F9
 - MacOS
 - ⌘ fn F9 (Command + fn + F9)

20

이제 source file을 bytecode로 compile하기 위해
Build menu에서 Build Project를 선택합니다.
단축키는 Windows에서 Ctrl + F9, macOS에서는 Command + fn + F9 입니다.

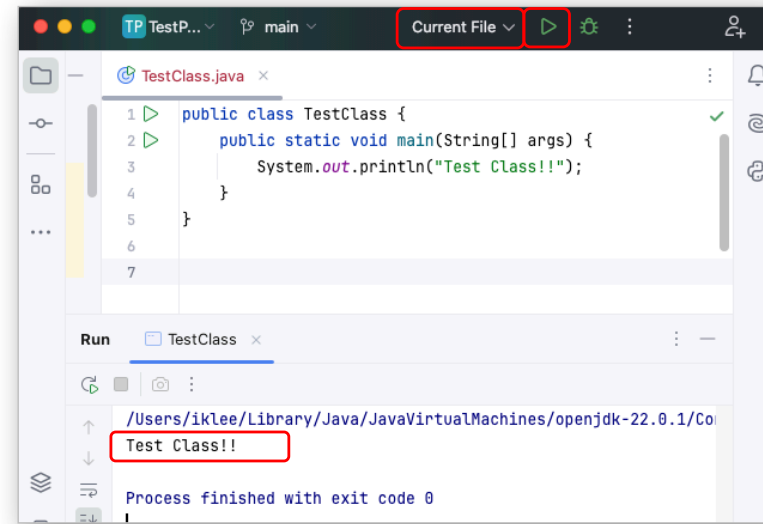
Bytecode

- Output of Compilation



이제 왼쪽의 file navigator에 "out" folder가 생성된 것을 확인할 수 있습니다.
Out folder 아래 "TestClass" 가 생성되었는데, 이것이 bytecode file인 TestClass.class 파일을 나타냅니다.
실제로 bytecode파일이 생성되었는지 확인하기 위해 explorer (Windows) 또는 finder (macOS) 에서
TestProject 폴더 아래 out 폴더 아래 TestClass.class 파일이 생성된 것을 확인할 수 있습니다.

Run



Ctrl + r for run
(Interpreted by JVM)

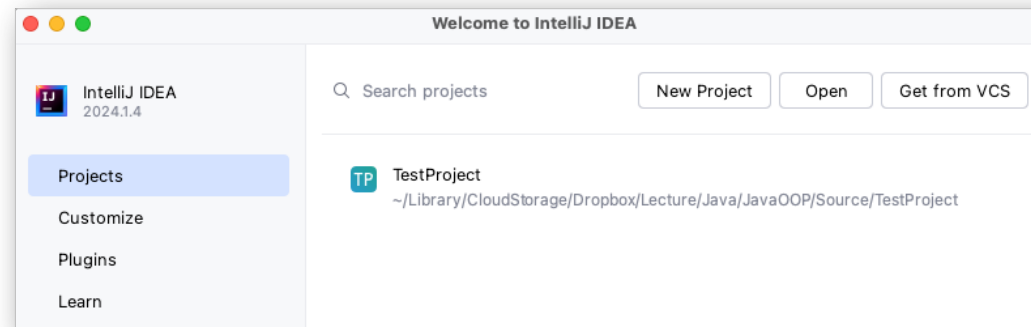
or run button

Output of the program

JVM interpreter를 실행하여 bytecode 프로그램을 실행하려면
Ctrl + r 을 선택하던지
Run button (play 삼각형 모양) 을 click 합니다.
program의 output이 IntelliJ 하단의 terminal window에 출력되는 것을 확인할 수 있습니다.

Close Project

- Main menu "File > Close Project"
- After closing the project:

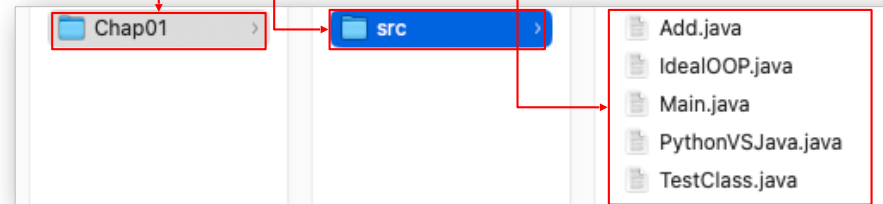


작업을 마친 후에는 "File > Close Project" menu를 선택하여 project를 닫을 수 있습니다.
Project를 close한 후에는 작성한 모든 project들의 list를 볼 수 있으며 각 프로젝트를 마우스로 선택하여 project로 진입할 수 있습니다.

New Project from Existing Source Files (1/3)

- Prepare the source

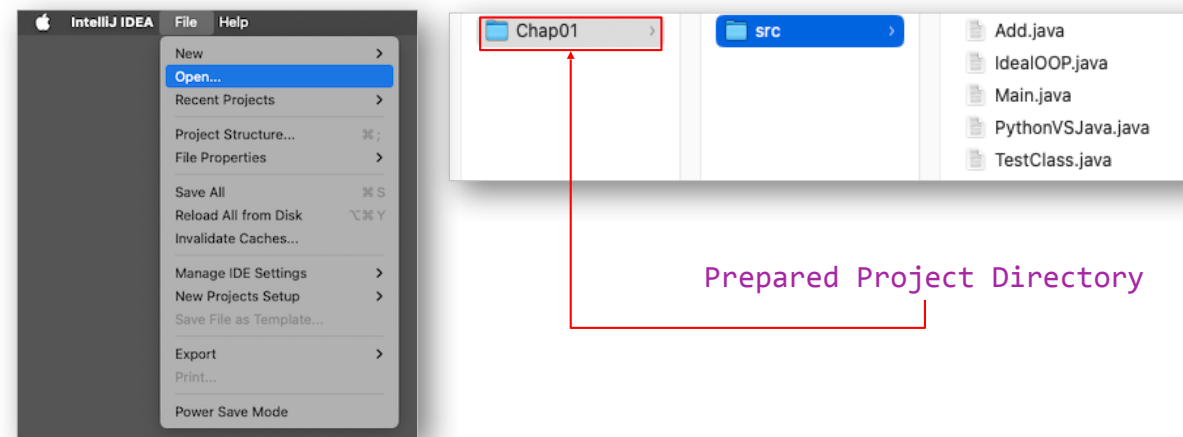
- Project name > src > Java source files



이번에는 이미 작성된 source file들을 IntelliJ에 불러들이는 방법에 대해 알아보겠습니다.
Source file들은 프로젝트 이름을 가진 폴더 아래 src 폴더 아래 java source file들을 가지고 있는 형식이어야 합니다.

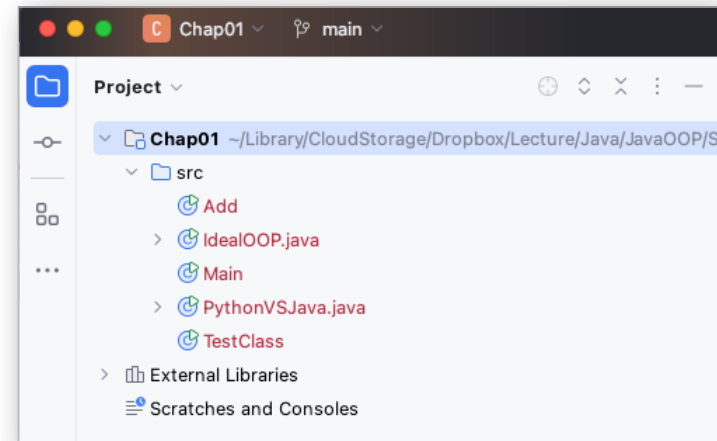
New Project from Existing Source Files (2/3)

- Main menu "File > Open"
- Select the project directory



Main menu에서 File > Open 메뉴를 선택하고
미리 준비된 project 폴더를 선택합니다.

New Project from Existing Source Files (3/3)



프로젝트가 IntelliJ로 읽어져 만들어지면서
Src 폴더 아래 source file들이 자리잡게 됩니다.