

2024년 상반기 K-디지털 트레이닝

자바 시작하기

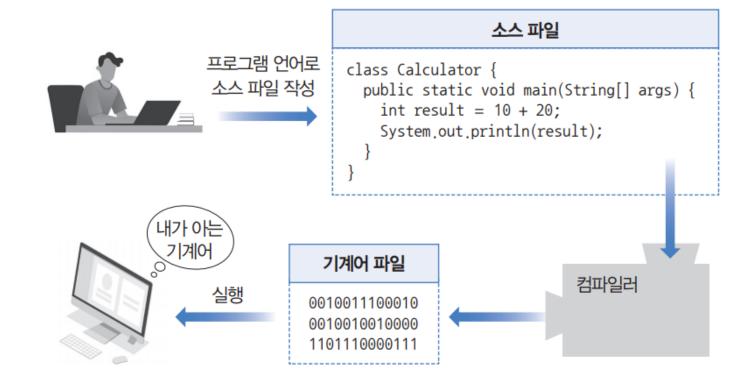
[KB] IT's Your Life



1

☑ 프로그래밍 언어의 역할

- ㅇ 사람과 컴퓨터의 대화 도움
- o 사람의 언어와 기계어 사이에서 다리 역할 → 컴파일
- ㅇ 고급 언어와 저급 언어로 구분



💟 자바 소개

- ㅇ 안드로이드 및 데스크톱 애플리케이션이나 웹사이트를 개발하는 핵심 언어
- o 1995년 썬마이크로시스템즈(Sun Microsystems)에서 처음 발표
- o 2010년 오라클에서 썬을 인수, 자바 개발 도구(JDK) 배포해 기술 지원



💟 자바 특징

- o 윈도우, 맥OS, 리눅스 등 모든 운영체제에서 실행 가능
- 먼저 객체(부품)를 만들고, 객체들을 서로 연결해서 더 큰 프로그램을 완성시키는 객체 지향 프로그래밍(OOP)에 최적화된 언어
- o 메모리(RAM)를 자동 정리해 메모리 관리에 용이
- o 무료로 다운로드해서 사용할 수 있는 오픈 소스 라이브러리(Open Source Library)가 풍부

자바 개발 도구(JDK) 설치

o JDK에는 Open JDK와 Oracle JDK 두 가지 있음

구분	Open JDK	Oracle JDK
라이선스	GNU GPL version 2 Oracle Technology Network License	
사용료	무료	개발 및 학습용: 무료, 상업용: 유료
개발 소스 공개 의무	없음	없음

- Open JDK 다운로드: https://jdk.java.net 및 https://adoptium.net
- Oracle JDK 다운로드: https://www.oracle.com/java/technologies/downloads
- JDK LTS 버전 다운로드: https://adoptium.net

OpenJDK 17

- o https://jdk.java.net/java-se-ri/17
 - openjdk-17+35_windows-x64_bin.zip 다운로드

jdk.java.net

GA Releases JDK 21 JavaFX 21 JMC 8

Early-Access Releases JDK 23 JDK 22 JavaFX 22 Jextract

Loom

Valhalla Reference

Implementations Java SE 21 Java SE 20 Java SE 19 Java SE 18 Java SE 17 Java SE 16

Java SE 15 Java SE 14

Java SE 13

Java SE 12 Java SE 11

Java Platform, Standard Edition 17 Reference Implementations

The official Reference Implementation for Java SE 17 (JSR 392) is based solely upon open-source code available from the JDK 17 Project in the OpenJDK Community.

The binaries are available under the GNU General Public License version 2, with the Classpath Exception.

These binaries are for reference use only!

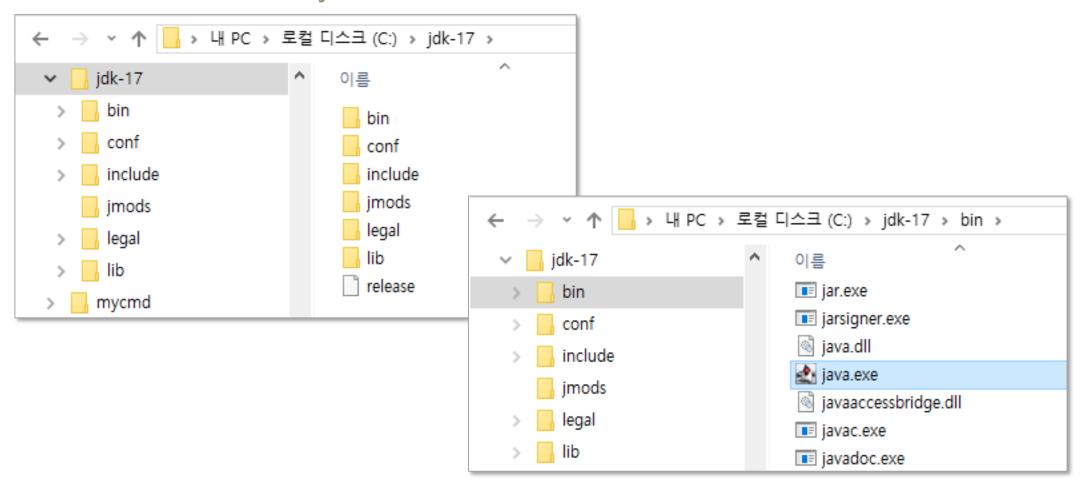
These binaries are provided for use by implementers of the Java SE 17 Platform Specification and are for reference purposes only. This Reference Implementation has been approved through the Java Community Process. Production-ready binaries under the GPL are available from Oracle; and will be in most popular Linux distributions.

RI Binary (build 17+35) under the GNU General Public License version 2

- Oracle Linux 8.3 x64 Java Development Kit (sha256) 179 MB
- Windows 10 x64 Java Development Kit (sha256) 178 MB

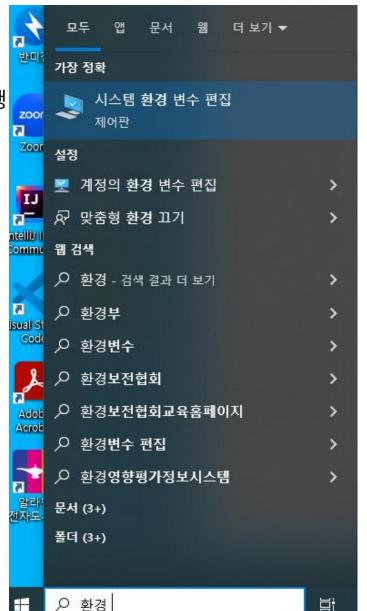
OpenJDK 17

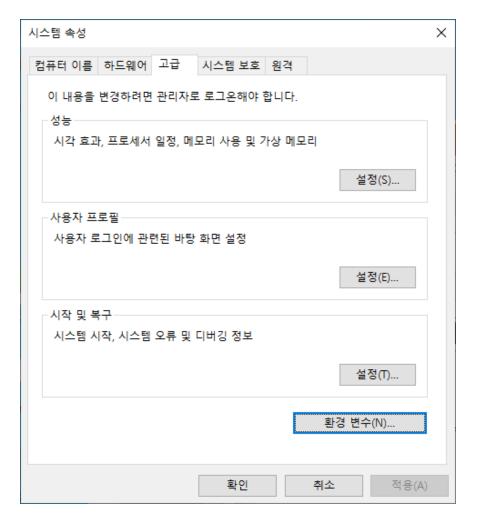
o 압축 해제 후 다음과 같이 jdk-17 폴더 이동



OpenJDK 17

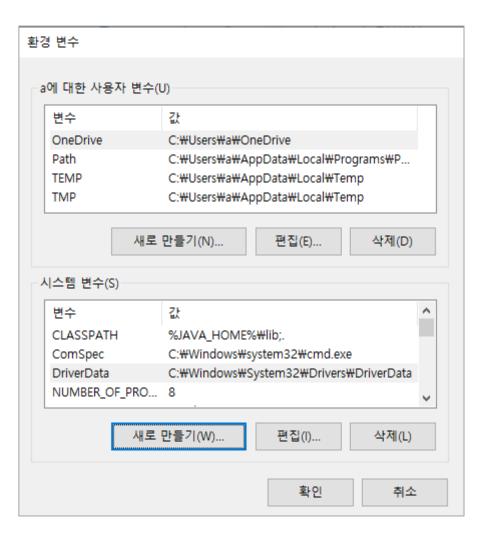
- o 환경 변수 설정
 - 시스템 환경 변수 편집 실행





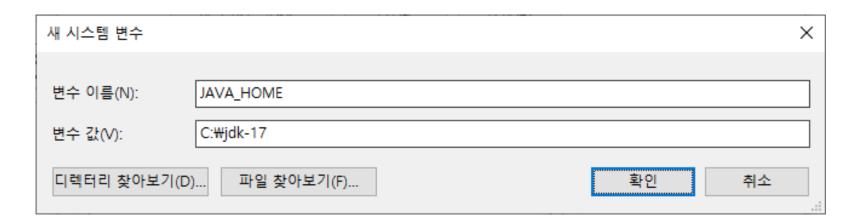
OpenJDK 17

o 환경 변수 설정



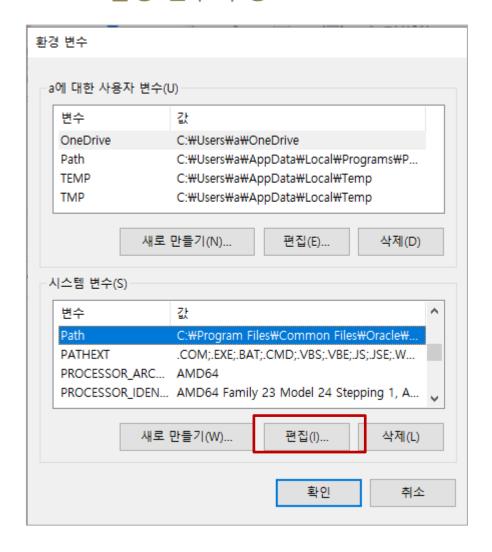
OpenJDK 17

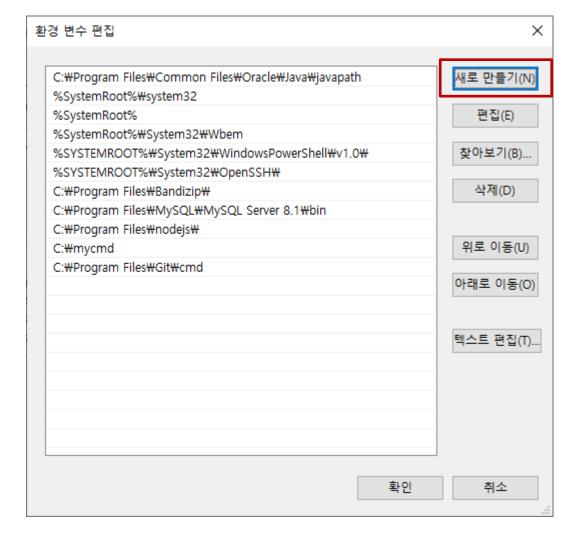
- o JAVA_HOME 환경 변수 추가
 - 변수 이름: JAVA_HOME
 - 변수 값: C:₩jdk-17



OpenJDK 17

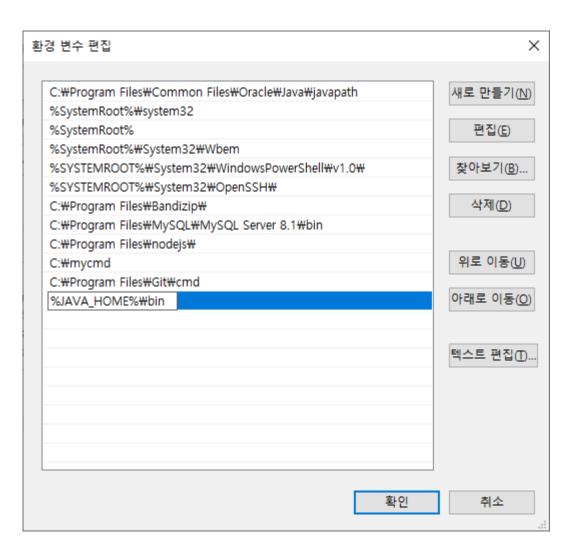
o PATH 환경 변수 수정





OpenJDK 17

- o PATH 환경 변수 수정
 - %JAVA_HOME%\bin 추가



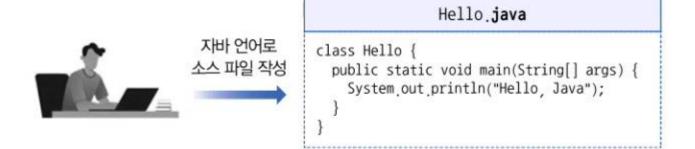
OpenJDK 17

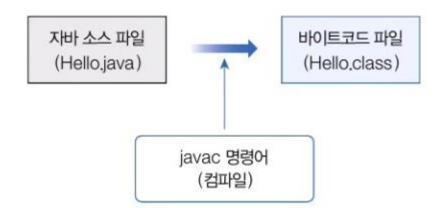
- o 설치 확인
 - 명령창(cmd)

```
C:\Users\a>java -version
openjdk version "17" 2021-09-14
OpenJDK Runtime Environment (build 17+35-2724)
OpenJDK 64-Bit Server VM (build 17+35-2724, mixed mode, sharing)
```

바이트코드 파일

- o 컴파일 compile
 - 소스 파일(.java)을 작성한 후 자바 가상머신이 실행 수 있는 바이트코드 파일(.class)로 번역하는 작업

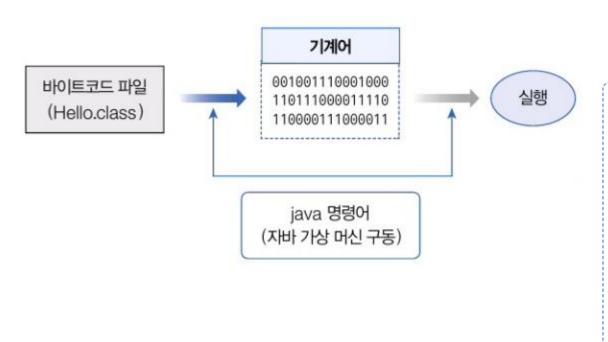


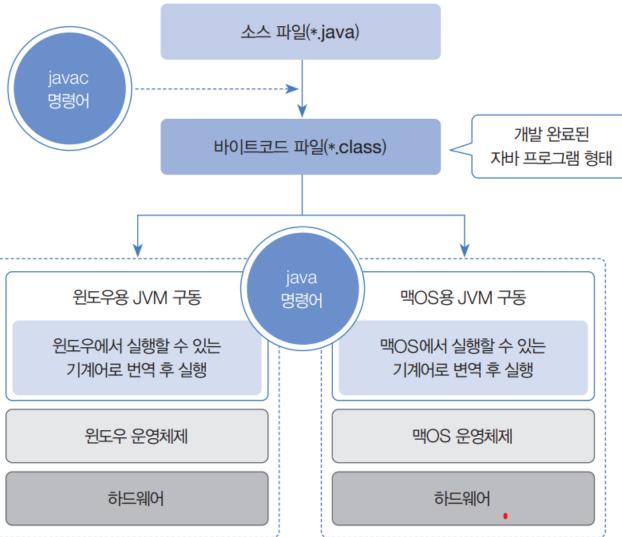


2 바이트코드 파일과 자바 가상 머신

💟 자바 가상 머신

- o java 명령어로 자바 가상 머신(JVM)을 구동
- o 바이트코드 파일(.class)을 실행
- o 자바가상머신은 운영체제별로 다르게 설치됨





소스 작성부터 실행까지

🤼 소스 파일 생성

- o c:₩temp 디렉토리에 Hello.java 작성
 - 메모장 이용

Hello.java

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

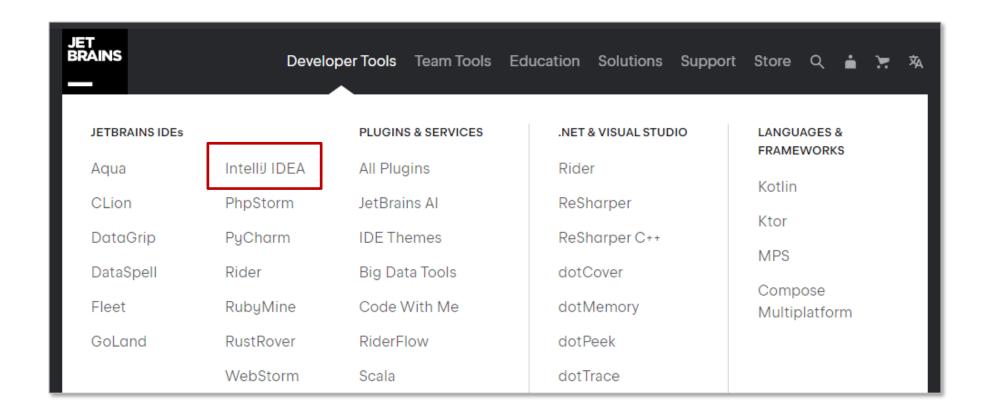
o 명령창

- 컴파일: javac Hello.java → Hello.class 생성
- 실행: java Hello

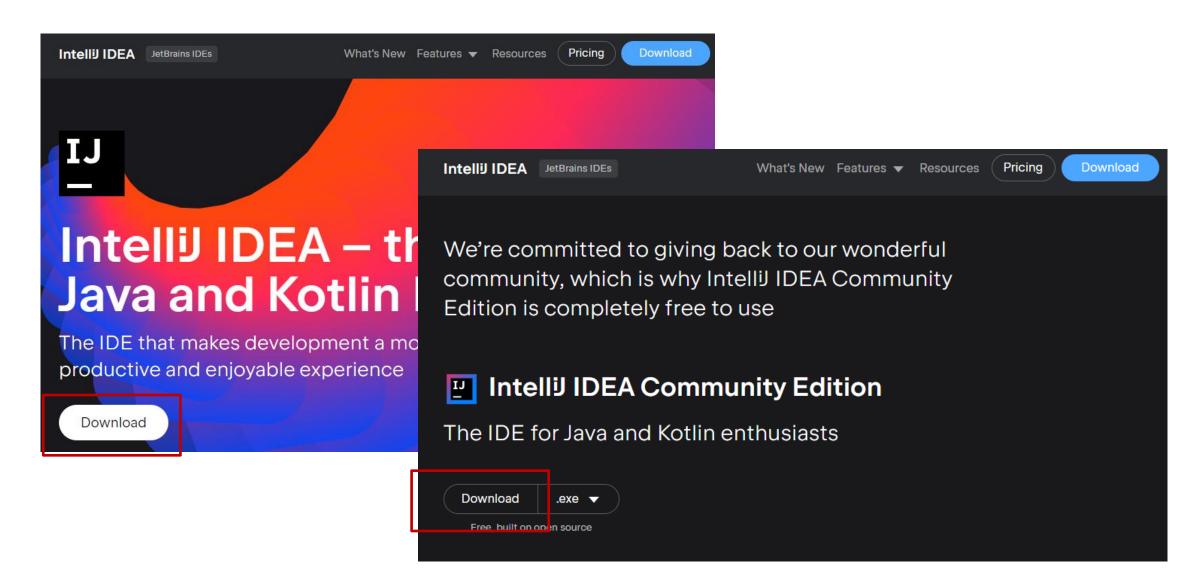
```
C:\Temp>javac Hello.java
C:\Temp>java Hello
Hello, Java
```

개발툴

- eclipse, IntelliJ가 가장 대표적인 툴
- IntelliJ
 - 상용버전(Ultimate), 무료버전(community)
 - https://www.jetbrains.com/

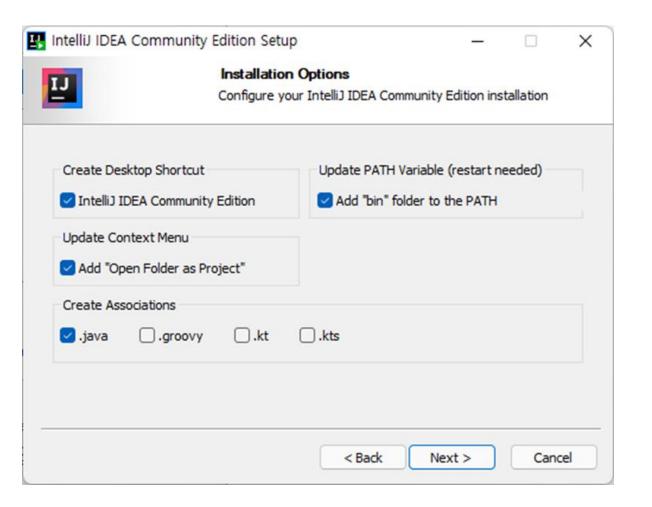


IntelliJ IDEA Community

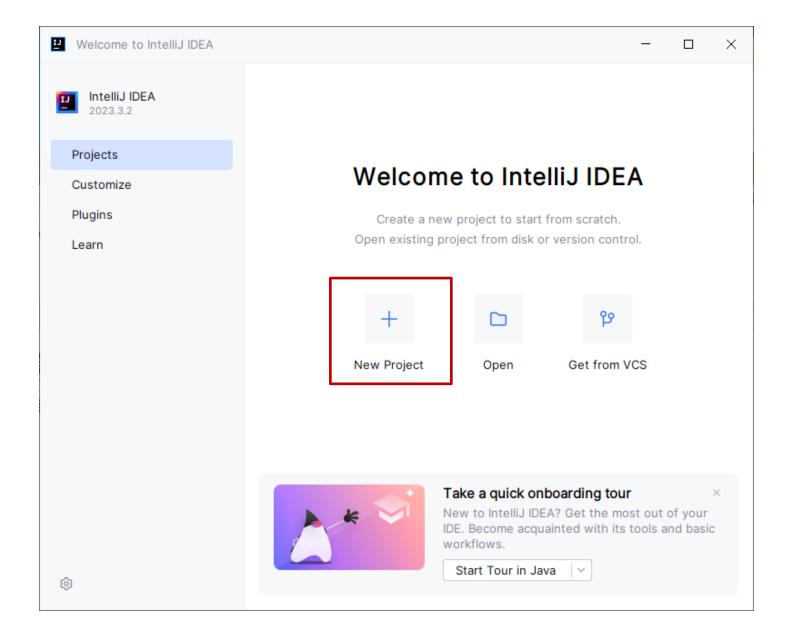


설치

- o 다운받은 idealC-2023.3.2.exe 실행
- o 설치 옵션을 다음과 같이 체크
- ㅇ 디폴트로 모두 설치

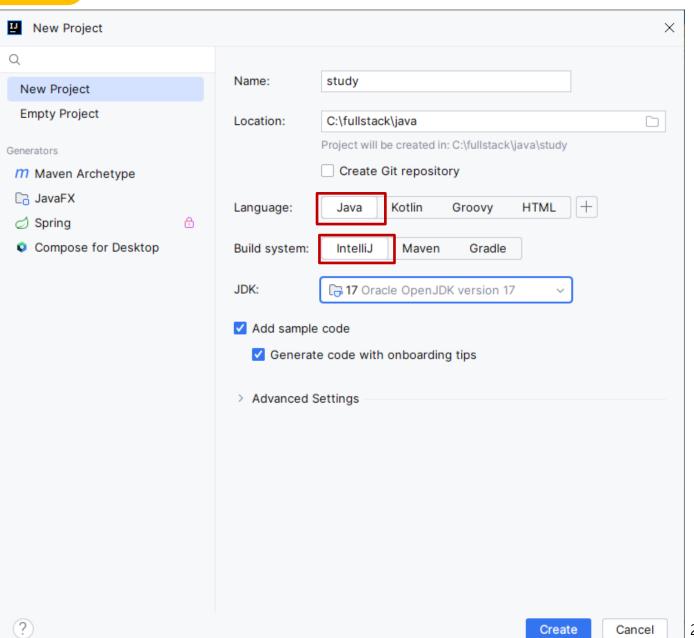


프로젝트 생성



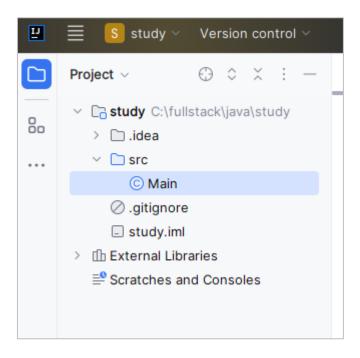
프로젝트 생성

- 작업 디렉토리: C:₩fullstack₩java
- 프로젝트명: study
- Language: java
- Build system: IntelliJ
- o JDK: 설치한 17버전



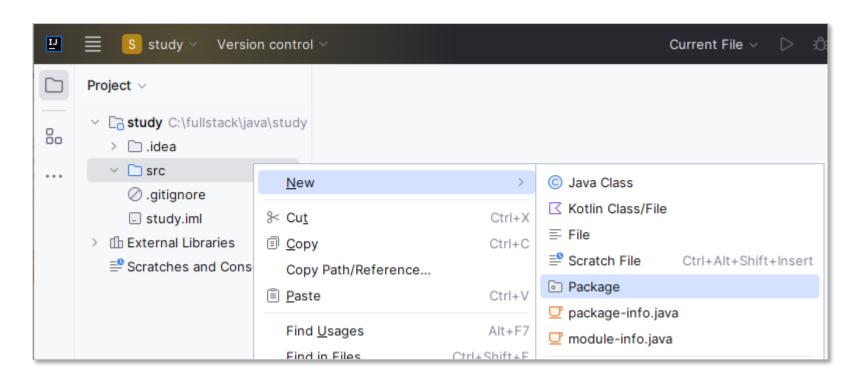
디폴트 프로젝트 구조

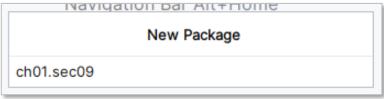
- o src/Main.java 샘플 파일 존재
- o 학습
 - 챕터 폴더₩섹션 폴더에 자바 파일 배치
 - 폴더를 package라고 부름



◎ 클래스 파일 만들기

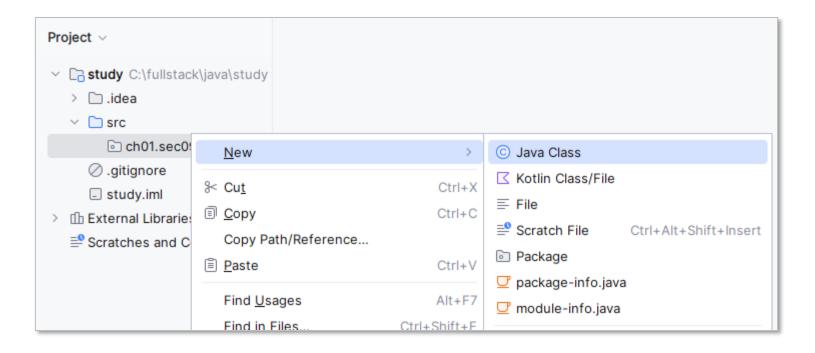
- o ch01.sec09 패키지 만들기
 - src >> New > Package





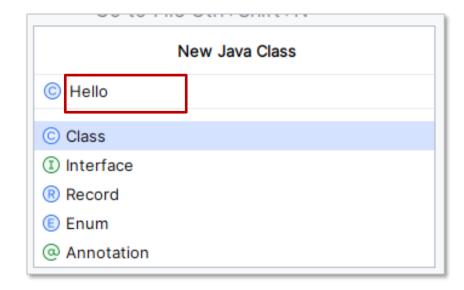
◎ 클래스 파일 만들기

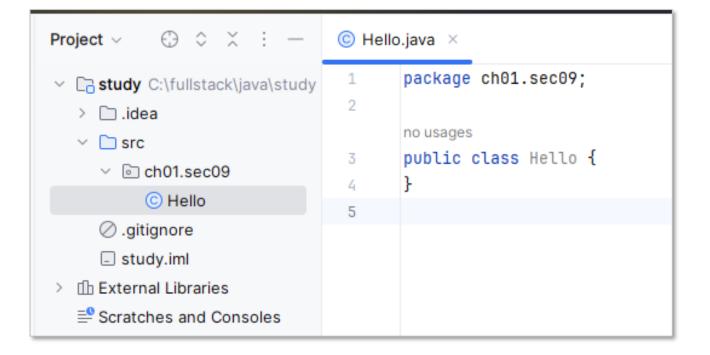
o ch01.sec09 >> New > Java Class



◎ 클래스 파일 만들기

- o 클래스명 Hello 입력
- o 유형은 Class

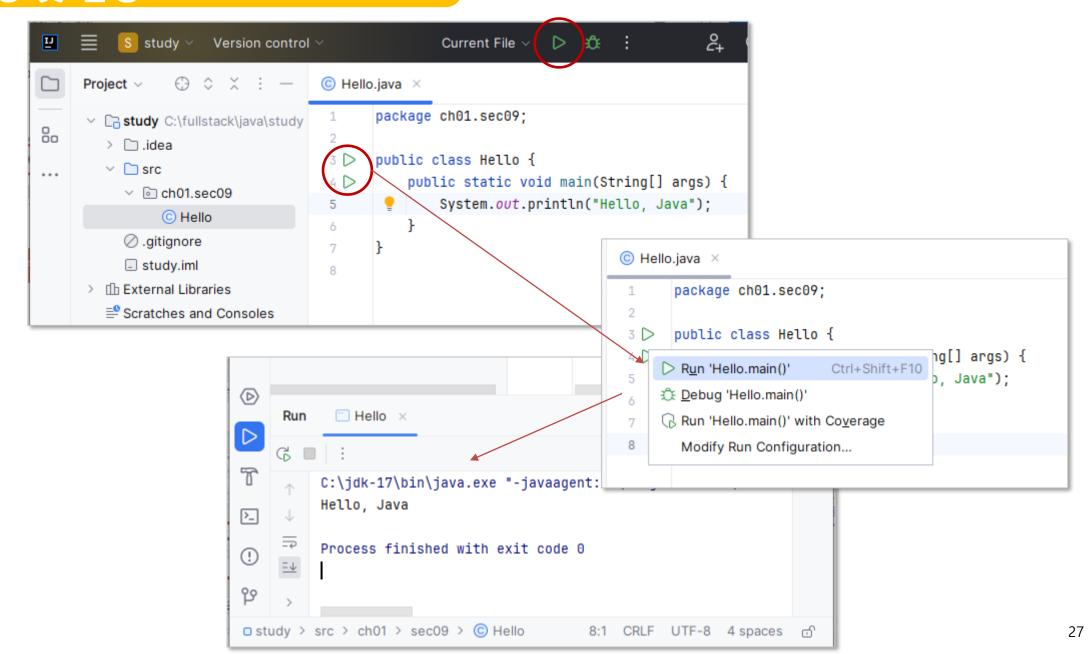




ch01.sec09.Hello.java

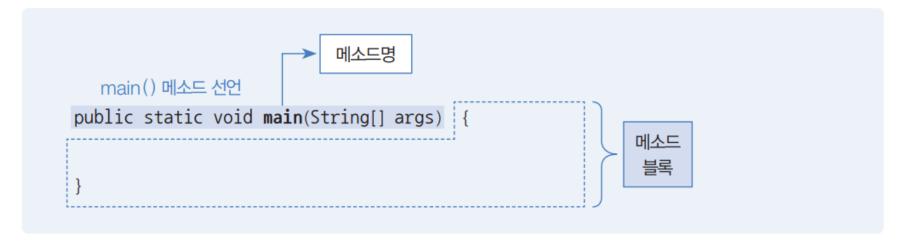
```
package ch01.sec09;
public class Hello {
   public static void main(String[] args) {
        System.out.println("Hello, Java");
```

☑ 실행



☑ 클래스 선언

☑ 메소드



♥ 주석 기호와 설명

ㅇ 프로그램 실행과 상관없이 코드에 설명 붙임

구분	주석 기호	설명
행 주석	//	//부터 행 끝까지 주석으로 처리한다.
범위 주석	/*···*/	/*와 */ 사이에 있는 내용은 모두 주석으로 처리한다.
도큐먼트 주석	/**···*/	/**와 */ 사이에 있는 내용은 모두 주석으로 처리한다. javadoc 명령어로 API 도큐먼트를 생성하는 데 사용한다.

o 주석 기호는 문자열("") 내부에 작성하면 안 됨

System.out.println("Hello, /*주석이 될 수 없음*/ welcome to the java world!");

6 코드 주석 달기

ch01.sec11.Hello.java

```
package ch01.sec11;
/**
* @author 신용권
*/
/*
장제목: 1장 자바 시작하기
작성일: 2022.08.24
*/
public class Hello {
 //프로그램 실행 진입점
 public static void main(String[] args) {
   //콘솔에 출력하는 실행문
   System.out.println("Hello, Java");
```

☑ 실행문

- o 실행문은 변수 선언, 변수값 저장, 메소드 호출에 해당하는 코드
- o 실행문 끝에는 반드시 세미콜론(;)을 붙여 실행문의 끝 표시

ch01.sec11. Calculator.java

3