

프로그래밍 기초

▶ 프로그래밍

✓ 프로그램 (Program)

컴퓨터가 인식할 수 있는 명령어의 나열(집합)

✓ 프로그래밍 (Programming)

프로그램을 작성하는 과정 = 코딩

✓ 프로그래머 (Programmer)

프로그램을 작성하는 사람

▶ 프로그래밍 언어

프로그램을 작성하기 위한 언어체계, 사람이 컴퓨터와 소통하게 하는 요소

TIOBE Index

– 프로그래밍 언어를 이용하는 개발자 & 업체의 수

Apr 2020	Apr 2019	Change	Programming Language	Ratings	Change
1	1		Java	16.73%	+1.69%
2	2		C	16.72%	+2.64%
3	4	▲	Python	9.31%	+1.15%
4	3	▼	C++	6.78%	-2.06%
5	6	▲	C#	4.74%	+1.23%
6	5	▼	Visual Basic	4.72%	-1.07%
7	7		JavaScript	2.38%	-0.12%
8	9	▲	PHP	2.37%	+0.13%
9	8	▼	SQL	2.17%	-0.10%
10	16	▲▲	R	1.54%	+0.35%
11	19	▲▲	Swift	1.52%	+0.54%
12	18	▲▲	Go	1.36%	+0.35%
13	13		Ruby	1.25%	-0.02%
14	10	▼▼	Assembly language	1.16%	-0.55%
15	22	▲	PL/SQL	1.05%	+0.26%

PYPL

– 프로그래밍 언어 튜토리얼이 검색된 수

Worldwide, Apr 2020 compared to a year ago:				
Rank	Change	Language	Share	Trend
1		Python	30.61 %	+3.9 %
2		Java	18.45 %	-1.9 %
3		Javascript	7.91 %	-0.4 %
4		C#	7.27 %	-0.0 %
5		PHP	6.07 %	-1.1 %
6		C/C++	5.76 %	-0.2 %
7		R	3.8 %	-0.2 %
8		Objective-C	2.4 %	-0.4 %
9		Swift	2.23 %	-0.2 %
10	▲	TypeScript	1.85 %	+0.2 %
11	▼	Matlab	1.77 %	-0.2 %
12	▲▲	Kotlin	1.63 %	+0.4 %
13		VBA	1.33 %	+0.0 %
14	▲▲▲	Go	1.26 %	+0.2 %
15	▼▼▼	Ruby	1.23 %	-0.1 %

▶ 프로그래밍 언어

프로그램을 작성하기 위한 언어체계, 사람이 컴퓨터와 소통하게 하는 요소

* 프로그래밍 언어 순위 사이트

TIOBE Index (<https://www.tiobe.com/tiobe-index/>)

– 프로그래밍 언어의 인기를 나타내는 지표

Google, Yahoo, Youtube, Wikipedia 등의 검색엔진을 통해 검색되어지는 결과 수로 순위를 매김

PYPL(<http://pypl.github.io/PYPL.html>)

– Google trends 통계 데이터를 기반으로 특정 프로그래밍 언어 튜토리얼이 검색된 수로 순위를 매김

▶ Java 프로그래밍 언어 특징

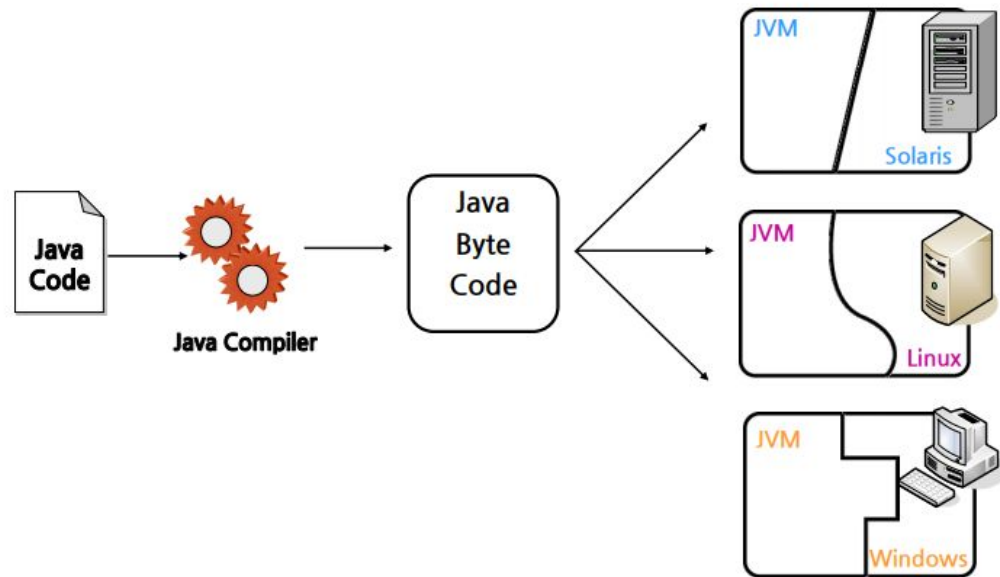


1. 운영체제(OS)에 독립적
-> (OS 관계없이 동일 코드로 동작. 이식성이 높다고도 표현함.)
2. 객체 지향 프로그래밍(OOP) 언어
3. 사용하기 쉬운 언어
 - 능률적이고 명확한 코드 작성 가능
 - 다른 언어의 단점 보완(포인터, 메모리 관리)
4. 자동 메모리 관리(Garbage Collection)
5. 동적 로딩 지원
6. 멀티쓰레드 지원
7. 네트워크와 분산환경 지원

▶ JVM(Java Virtual Machine)

Java를 실행하기 위한 가상 기계 로 OS(운영체제)에 관계없이 독립적으로 동작.

- C : OS가 코드를 직접 해석하기 때문에 C언어는 OS 따라 코드가 다른 부분이 존재.
- Java : OS에 맞는 JVM을 설치하여 OS 종류 관계없이 JVM이라는 가상머신이 Java 코드를 동일하게 해석 .



[JVM이 Java 코드를 해석하는 순서]

개발자가 작성한 Java 코드를

Compiler(컴파일러) 가 byte code(.class 파일)로 번역

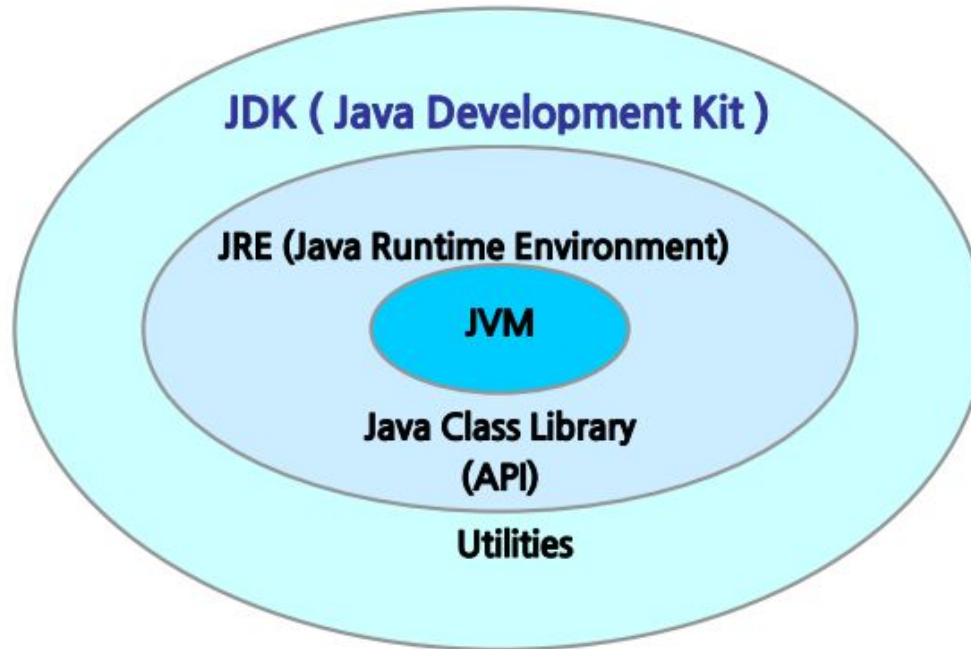
-> JVM에 전달

-> JVM이 **Interpreter(인터프리터)** 방식으로 한 줄 씩 해석함.

▶ Java 개발 환경

✓ 설치 범위

사용자/개발자 입장에 따라 설치하는 범위가 달라짐



Java SE : Java Standard Edition

Java EE : Java Enterprise Edition

Java ME : Java Micro Edition

▶ Open JDK 설치

OpenJDK 다운로드 페이지로 이동 (<https://jdk.java.net/archive/>)

[jdk.java.net](#)
 GA Releases
 JDK 22
 JavaFX 22
 JMC 9
 Early-Access
 Releases
 JDK 24
 JDK 23
 JavaFX 24
 JavaFX 23
 Jextract
 Leyden
 Loom
 Valhalla
 Reference
 Implementations
 Java SE 22
 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 SE 10
 Java SE 9
 Java SE 8
 Java SE 7
 Feedback
 Report a bug
 Archive

Archived OpenJDK General-Availability Releases

This page is an archive of previously released builds of the JDK licensed under the GNU General Public License, version 2, with Classpath Exception.

WARNING: These older versions of the JDK are provided to help developers debug issues in older systems. They are not updated with the latest security patches and are not recommended for use in production.

Releases

22.0.1 (build 22.0.1+8)

Windows	64-bit	zip (sha256) 192M
Mac/AArch64	64-bit	tar.gz (sha256) 187M
Mac/x64	64-bit	tar.gz (sha256) 189M
Linux/AArch64	64-bit	tar.gz (sha256) 191M
Linux/x64	64-bit	tar.gz (sha256) 193M
	Source	Tag jdk-22.0.1-ga

22 GA (build 22+36)

Windows	64-bit	zip (sha256) 192M
Mac/AArch64	64-bit	tar.gz (sha256) 187M
Mac/x64	64-bit	tar.gz (sha256) 189M
Linux/AArch64	64-bit	tar.gz (sha256) 191M
Linux/x64	64-bit	tar.gz (sha256) 193M
	Source	Tags are jdk-22+36, jdk-22-ga

21.0.2 (build 21.0.2+13)

Windows	64-bit	zip (sha256) 193M
Mac/AArch64	64-bit	tar.gz (sha256) 189M
Mac/x64	64-bit	tar.gz (sha256) 191M
Linux/AArch64	64-bit	tar.gz (sha256) 193M
Linux/x64	64-bit	tar.gz (sha256) 195M
	Source	Tag jdk-21.0.2-ga

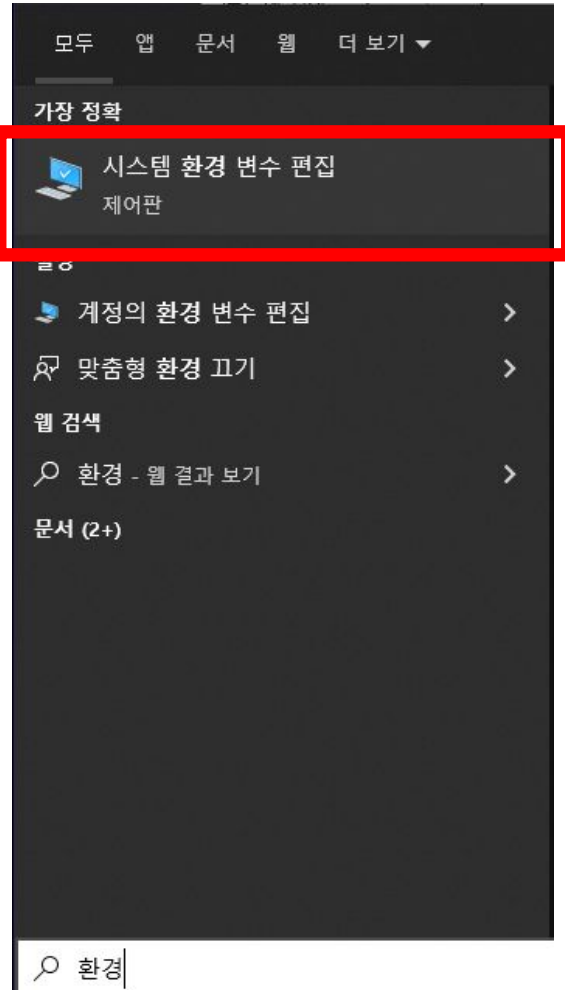
21.0.1 (build 21.0.1+12)

Windows	64-bit	zip (sha256) 188M
Mac/AArch64	64-bit	tar.gz (sha256) 184M
Mac/x64	64-bit	tar.gz (sha256) 186M
Linux/AArch64	64-bit	tar.gz (sha256) 187M
Linux/x64	64-bit	tar.gz (sha256) 189M
	Source	Tag jdk-21.0.1-ga

원하는 버전(17 or 21)을 찾아서 OS에 맞게 다운로드 진행

▶ 자바 설정

✓ 환경 변수 설정



1) 윈도우 키(시작 버튼) 클릭 후 “환경” 검색

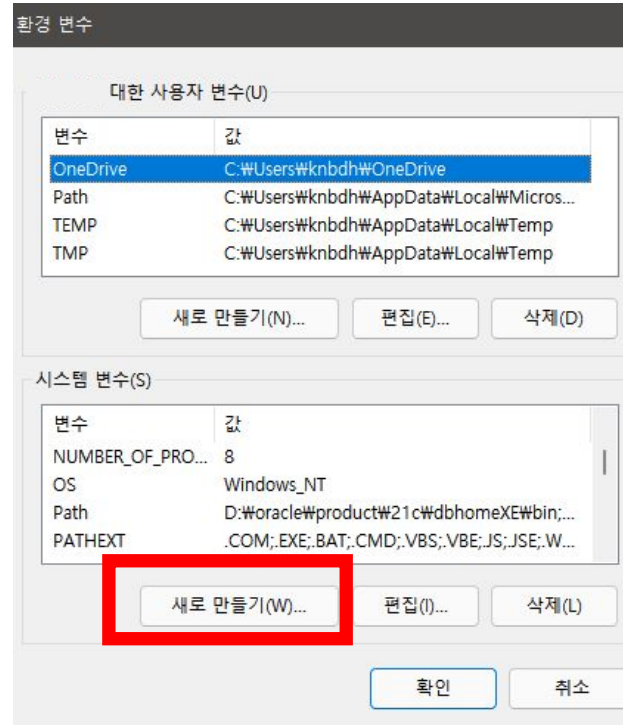
2) 시스템 환경 변수 편집 선택

▶ 자바 설정

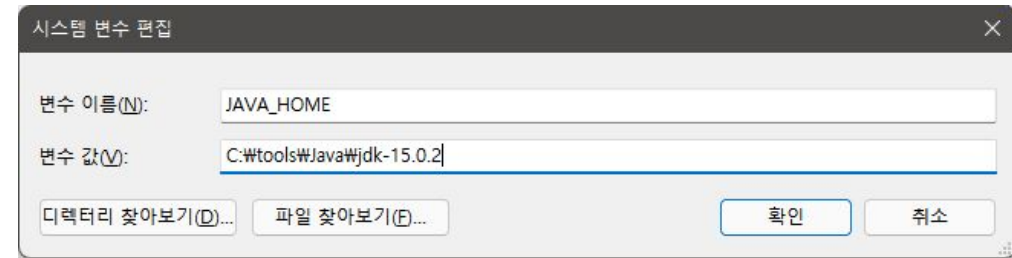
✓ 환경 변수 설정



3) 환경 변수 클릭



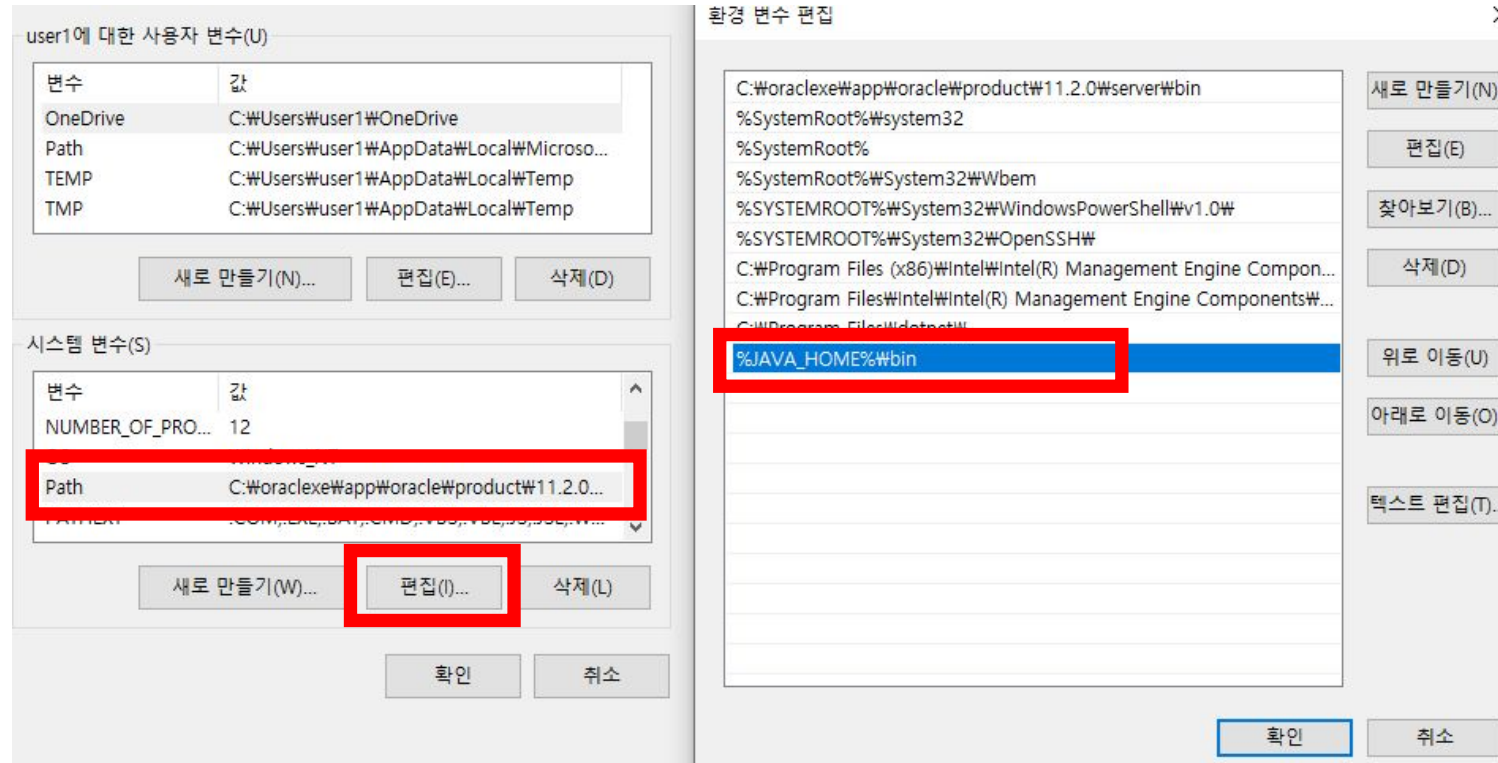
4) 시스템 변수 -> 새로 만들기 클릭



5) 내용 작성 후 확인
 변수 이름 : JAVA_HOME
 변수 값 : 다운로드 받은 JDK 압축 해제 폴더

▶ 자바 설정

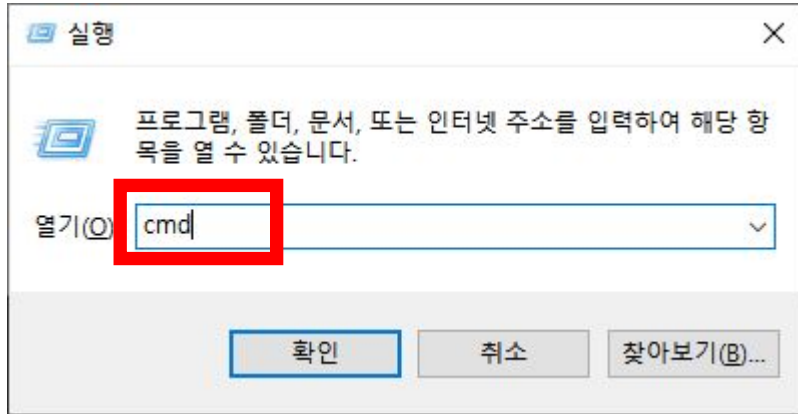
✓ 환경 변수 설정



6) 시스템 변수 -> Path 선택 -> 편집 -> %JAVA_HOME%\bin 추가

▶ 자바 설정

✓ 환경변수 테스트



윈도우키 + R 버튼 누르고 실행창에서 cmd 입력 후 확인

```
선택 C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Version 10.0.22000.493]
(c) Microsoft Corporation. All rights reserved.

c:\#>java -version
openjdk version "15.0.2" 2021-01-19
OpenJDK Runtime Environment (build 15.0.2+7-27)
OpenJDK 64-Bit Server VM (build 15.0.2+7-27, mixed mode, sharing)

c:\#>javac -version
javac 15.0.2

c:\#>_
```

java -version / javac -version 입력 시
이미지와 같은 버전 정보 나오면 설정완료

▶ 자바 개발 환경

✓ Compile Test

1. workspace 폴더 만들기

C:\workspace 폴더 생성

2. 메모장 실행 후 다음 내용 작성

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

▶ 자바 개발 환경

✓ Compile Test

3. 내용 작성 후 파일 저장

파일 저장 경로 : C:\workspace 폴더

파일명 : JavaTest.java (대소문자구분 유의)

파일 형식 : 모든 파일

4. 명령 프롬프트(cmd창) 열기

시작 - “명령 프롬프트” 또는 “cmd” 입력

5. 명령 프롬프트에서 workspace 경로 찾아가기

```
C:\W> cd c:\workspace
```

▶ 자바 개발 환경

✓ Compile Test

6. Compile 하기

```
C:\workspace> javac JavaTest.java
```

(컴파일 완료 시 workspace 폴더 내부에 JavaTest.class 파일이 생성됨.)

7. 실행하기 (HelloWord!! 메시지 출력 확인)

```
C:\workspace> java JavaTest
```

```
c:\workspace> javac JavaTest.java
```

```
c:\workspace> java JavaTest
```

```
Hello World!!!
```

```
c:\workspace> _
```

▶ Eclipse IDE(통합 개발 환경) 설치

* IDE(Integrated Development Environment, 통합 개발 환경)

- 애플리케이션 개발에 사용되는 공통된 개발자 도구를 하나의 GUI에 결합한 소프트웨어
- 소스 코드 편집기(작성 중 오류 검사, 자동완성, 구문 강조), 로컬 빌드 자동화(자동 컴파일, 패키징, 테스트), 디버거(버그 위치 표시) 를 내장하고 있음.
- Eclipse, Visual Studio Code, IntelliJ 등 여러 IDE가 존재함.

* Eclipse다운로드 페이지

(<https://www.eclipse.org/downloads/>)

▶ Eclipse설치

- Download Packages 클릭



Install your favorite desktop IDE packages

[Learn More](#)

[Download x86_64](#)

[Download Packages](#) | [Need Help](#)



The Eclipse Temurin™ project provides high-quality, TCK certified OpenJDK runtimes and associated technology for use across the Java™ ecosystem.

[Learn More](#)

[Download](#)

▶ Eclipse설치

Eclipse IDE for Enterprise Java And Web Developers – 해당되는 운영체제 클릭 – Download 클릭

Eclipse IDE 2024-06 R Packages

Eclipse IDE for Enterprise Java and Web Developers

531 MB 223,087 DOWNLOADS



Tools for developers working with Java and Web applications, including a Java IDE, tools for JavaScript, TypeScript, JavaServer Pages and Faces, Yaml, Markdown, Web Services, JPA and Data Tools, Maven and Gradle, Git, and more.



Windows x86_64

macOS x86_64 | AArch64

Linux x86_64 | AArch64

Click [here](#) to open a bug report with the Eclipse Web Tools Platform.
Click [here](#) to raise an issue with the Eclipse Platform.
Click [here](#) to raise an issue with Maven integration for web projects.
Click [here](#) to raise an issue with Eclipse Wild Web Developer (incubating).

Download

Download from: Japan - Yamagata University (<https>)

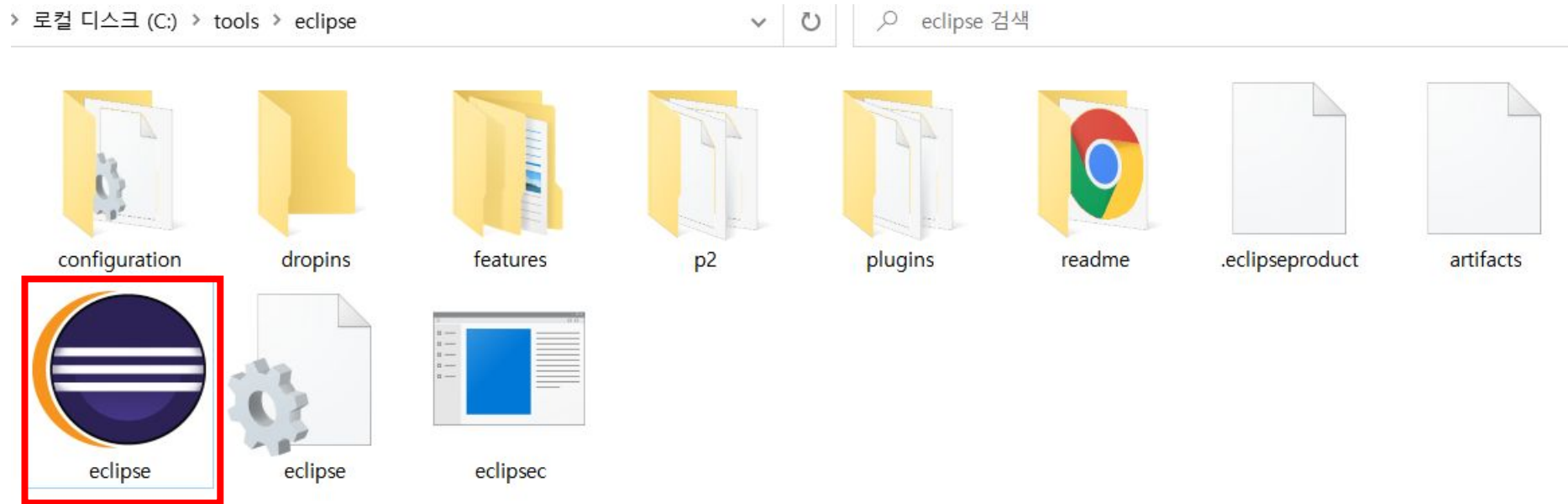
File: [eclipse-jee-2024-06-R-win32-x86_64.zip](#) SHA-512

>> [Select Another Mirror](#)

▶ Eclipse설치

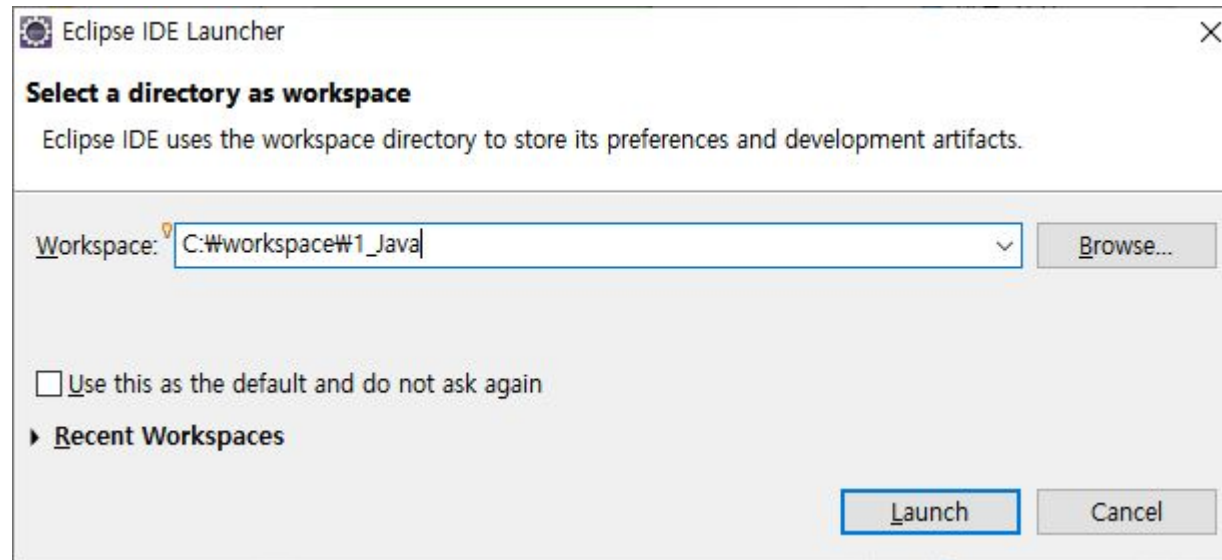
C드라이브에 tools 폴더를 생성하여 다운로드된 파일을 이동

- 압축 해제(여기에 풀기) - eclipse.exe 실행 (바탕화면에 바로가기 만들기 추천)



▶ 이클립스 설치

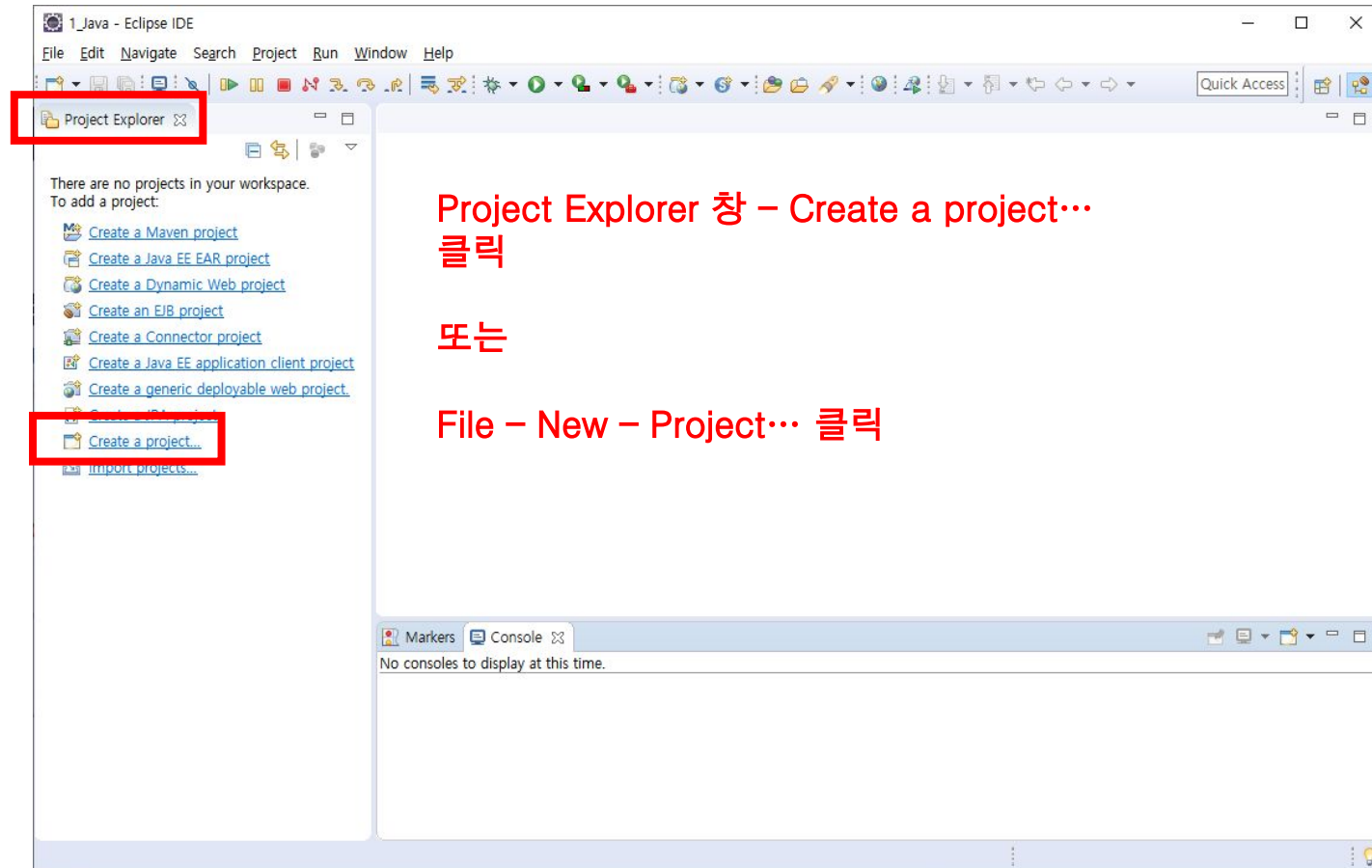
Workspace 입력란에 C:\workspace\1_Java 작성 후 Launch 클릭



▶ 자바 프로그래밍 순서

✓ Eclipse 환경

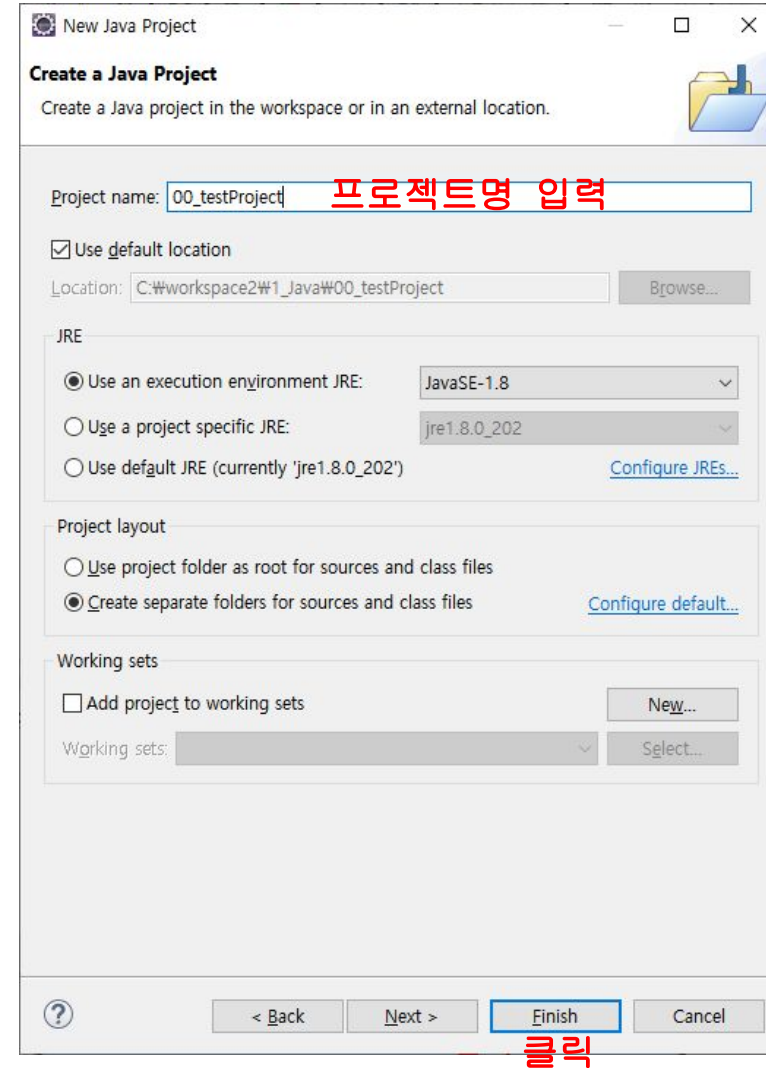
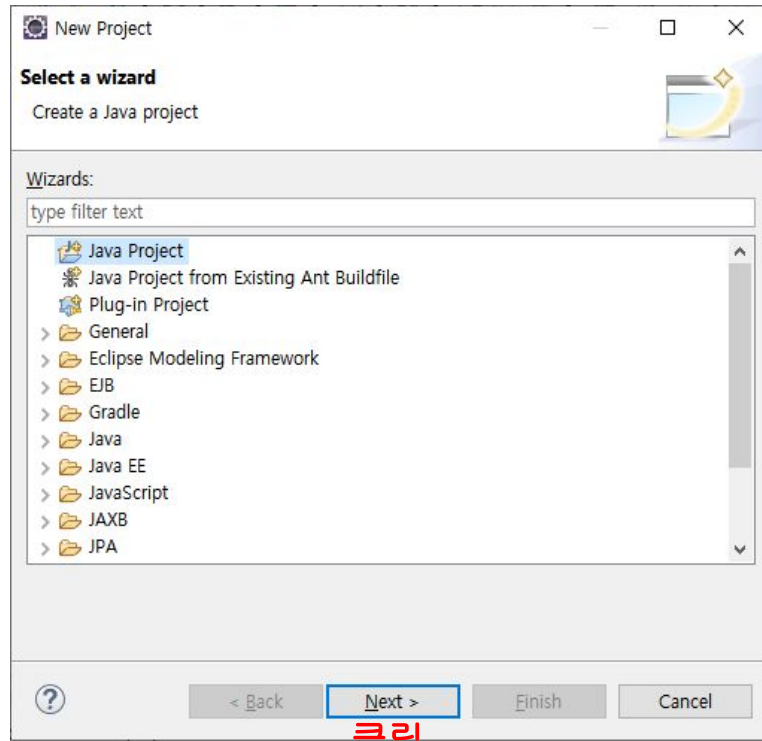
1. project 만들기



▶ 자바 프로그래밍 순서

✓ Eclipse 환경

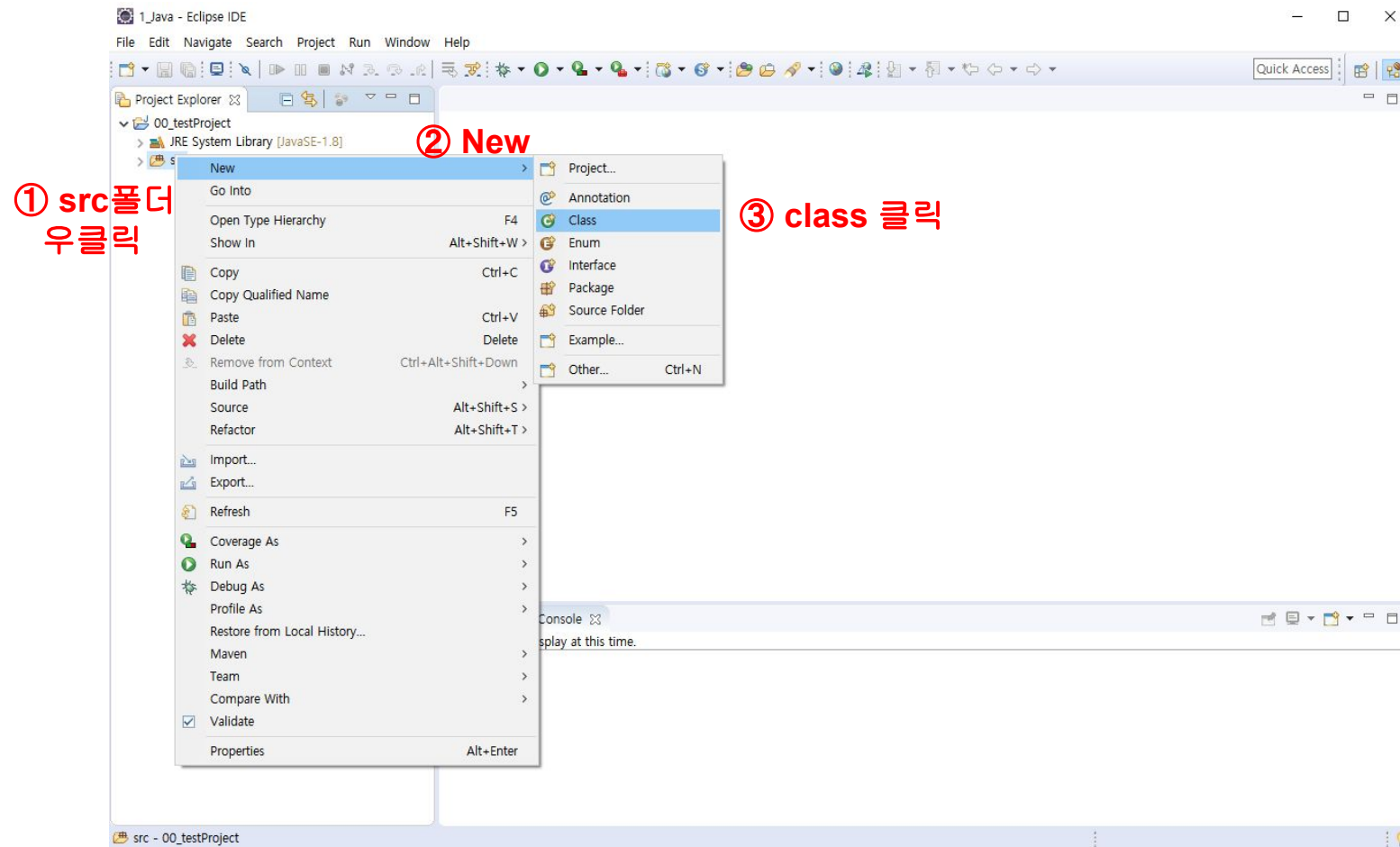
1. project 만들기



▶ 자바 프로그래밍 순서

✓ Eclipse 환경

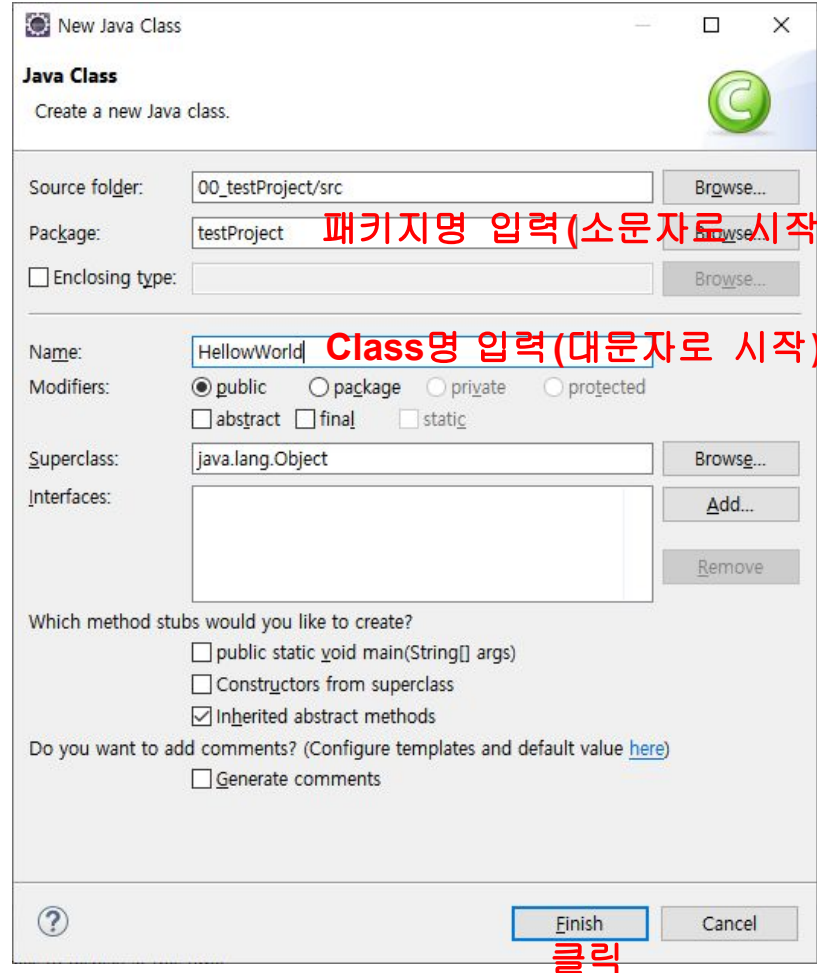
2. Class 만들기



▶ 자바 프로그래밍 순서

✓ Eclipse 환경


2. Class 만들기

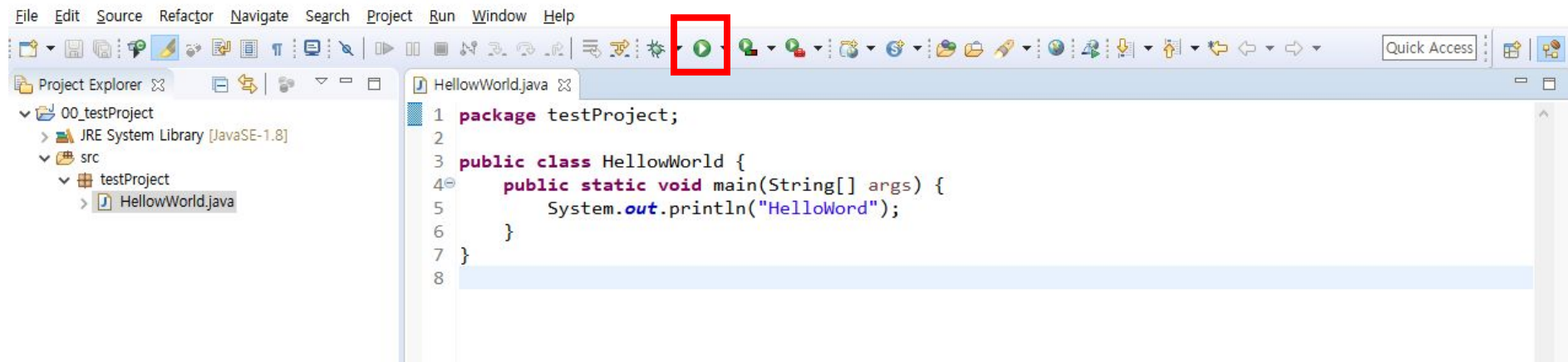


▶ 자바 프로그래밍 순서

✓ Eclipse 환경

3. Class 작성 후 실행

Class 작성 완료 후  클릭 또는 Ctrl + F11을 눌러 실행



▶ 자바 프로그래밍 기본

✓ class

객체를 만들기 위한 일종의 **설계도**.

자바에서 **모든 코드는 반드시 클래스 안에 존재** 해야 하며,

서로 관련된 코드들을 그룹으로 나누어 별도의 클래스를 구성

클래스들이 모여 하나의 Java 애플리케이션 구성

```
public class 클래스명 {  
  
    /*  
     * 주석을 제외한 모든 코드는 블록 클래스 { } 내에 작성  
     */  
  
}
```

▶ 자바 프로그래밍 기본

✓ 주석(comment)

코드에 대한 설명이나 그 외 다른 정보를 넣을 때 사용하는 것으로
컴파일 시 컴파일러가 주석 부분은 건너 뛴

`/* */` : 범위 주석, `/*` 와 `*/` 사이 내용은 주석으로 간주

`//` : 한 줄 주석, `//` 뒤의 내용은 주석으로 간주

▶ 자바 프로그래밍 기본

✓ main (main method)

`public static void main(String[] args)` 는 고정된 형태 의 메서드 선언부로
Java Application을 실행하는데 필요한 메서드 (프로그램 실행 시 java.exe에 의해 호출됨)
모든 클래스가 main메서드를 가지고 있어야 하는 것은 아니지만
하나의 Java애플리케이션에는 main메서드를 포함한 클래스가 반드시 하나 이상 존재해야 함.

```
public class 클래스명 {  
  
    public static void main(String[] args) { //메인 메서드의 선언부  
  
        // 실행될 코드를 작성  
  
    }  
}
```

▶ 자바 프로그래밍 순서

✓ Class 작성 예시

`package member.model.vo;` ① 패키지 (package) 선언

`import java.util.Date;` ② 임포트 (import) 선언

`public class Member {` ③ 클래스 (class) 작성부

`private String name;`
`private int age;`
`private Date enrollDate;`

} 필드 (또는 멤버 변수)

생성자

`public Member() {}`

`public Member(String name, int age, Date enrollDate) {`
`super();`
`this.name = name;`
`this.age = age;`
`this.enrollDate = enrollDate;`
`}`

`public String getName() {`
`return name;`
`}`

`public void setName(String name) {`
`this.name = name;`
`}`

`... 이하 생략...`

`}`

} (멤버) 메서드