

# 프로그래밍 기초

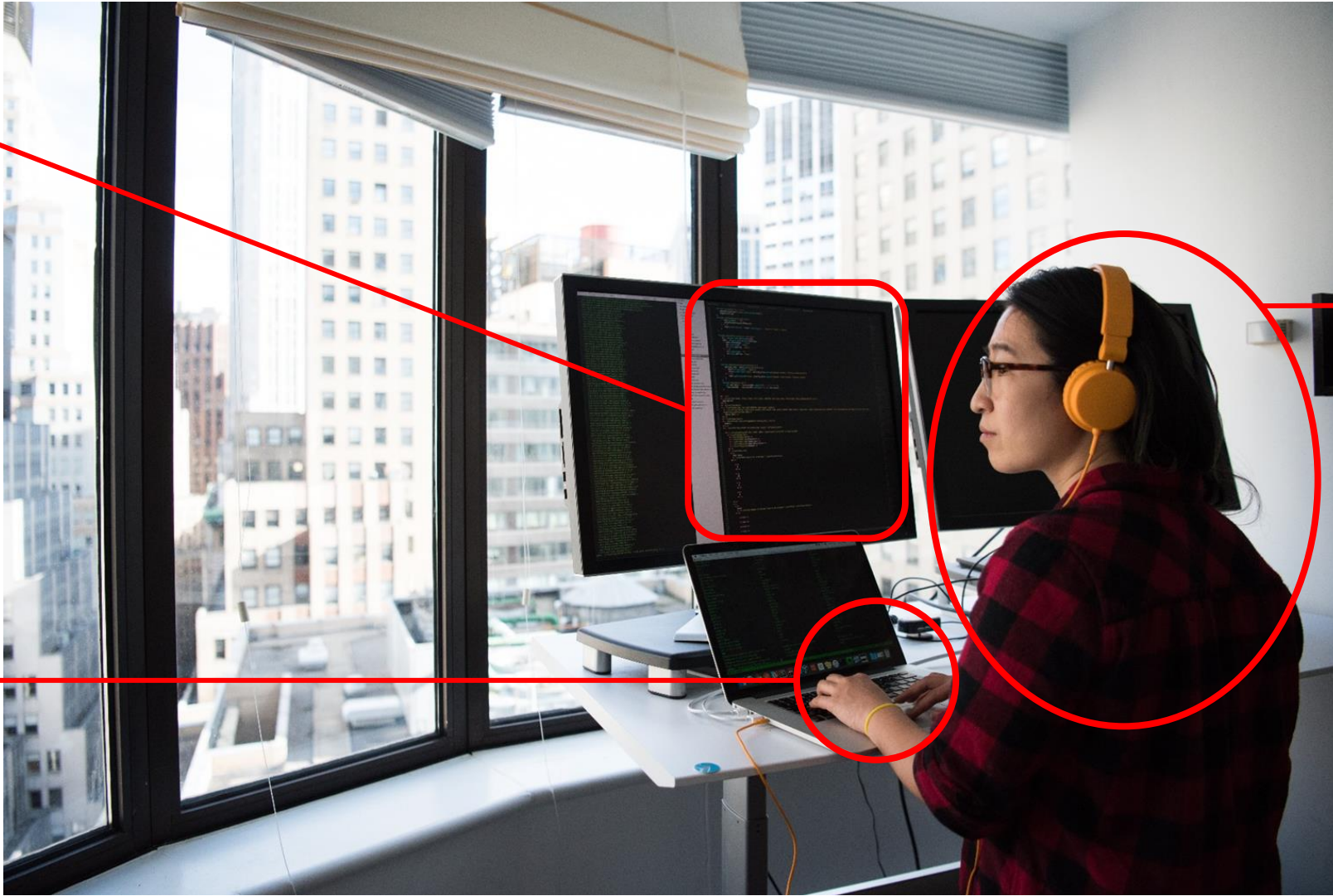
# ▶ 프로그래밍

## 프로그램

컴퓨터가  
인식할 수 있는  
명령어의 나열  
(집합)

## 프로그래밍

프로그램을 작성  
하는 과정 = 코딩



## 프로그래머



프로그램을  
작성하는 사람

# ▶ 프로그래밍 언어

프로그램을 작성하기 위한 언어체계, 사람이 컴퓨터와 소통하게 하는 요소

## TIOBE Index

- 프로그래밍 언어를 이용하는 개발자 & 업체의 수

Jan 2025	Jan 2024	Change	Programming Language	Ratings	Change
1	1		 Python	23.28%	+9.32%
2	3	▲	 C++	10.29%	+0.33%
3	4	▲	 Java	10.15%	+2.28%
4	2	▼	 C	8.86%	-2.59%
5	5		 C#	4.45%	-2.71%
6	6		 JavaScript	4.20%	+1.43%
7	11	▲	 Go	2.61%	+1.24%
8	9	▲	 SQL	2.41%	+0.95%
9	8	▼	 Visual Basic	2.37%	+0.77%
10	12	▲	 Fortran	2.04%	+0.94%
11	13	▲	 Delphi/Object Pascal	1.79%	+0.70%
12	10	▼	 Scratch	1.55%	+0.11%
13	7	▼	 PHP	1.38%	-0.41%
14	19	▲	 Rust	1.16%	+0.37%
15	14	▼	 MATLAB	1.07%	+0.09%

## PYPL

- 프로그래밍 언어 튜토리얼이 검색된 수

Worldwide, Jan 2025 :				
Rank	Change	Language	Share	1-year trend
1		Python	29.8 %	+1.7 %
2		Java	15.35 %	-0.5 %
3		JavaScript	7.96 %	-1.0 %
4		C/C++	7.17 %	+0.3 %
5		C#	6.36 %	-0.3 %
6		R	4.63 %	+0.0 %
7		PHP	3.87 %	-0.6 %
8		TypeScript	2.84 %	-0.0 %
9	▲▲	Rust	2.8 %	+0.5 %
10		Objective-C	2.65 %	+0.2 %
11	▼▼	Swift	2.48 %	-0.2 %
12		Go	2.11 %	-0.0 %
13		Kotlin	1.9 %	+0.1 %
14		Matlab	1.59 %	+0.0 %
15	▲	Ada	1.21 %	+0.2 %

# ▶ 자바 언어 특징

운영체제에 독립적(이식성이 높음)

사용하기 쉬운 언어

- 능률적이고 명확한 코드 작성 가능
- 다른 언어의 단점 보완  
(포인터, 메모리 관리)

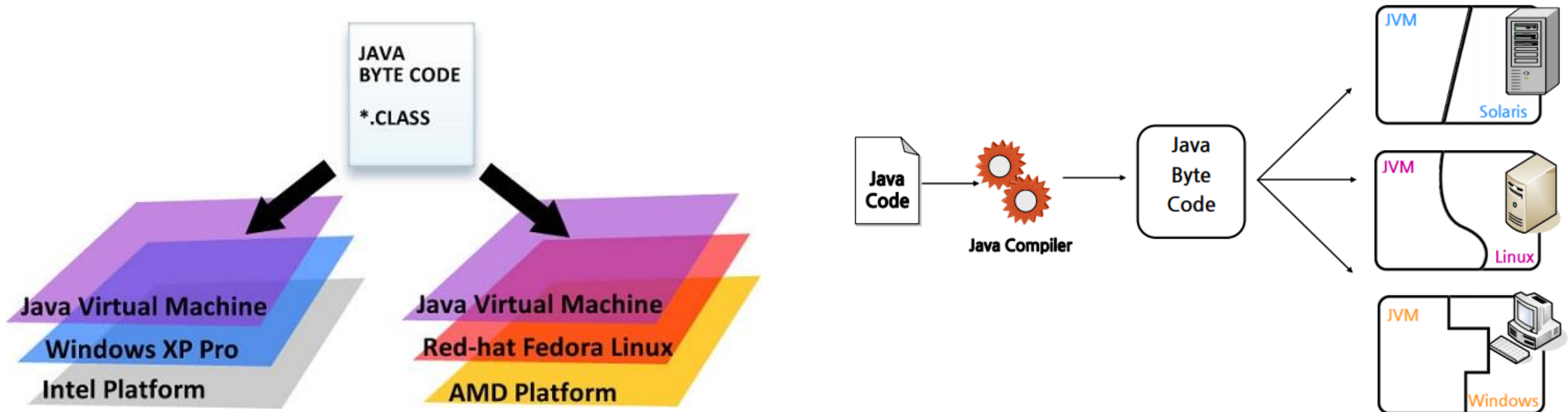


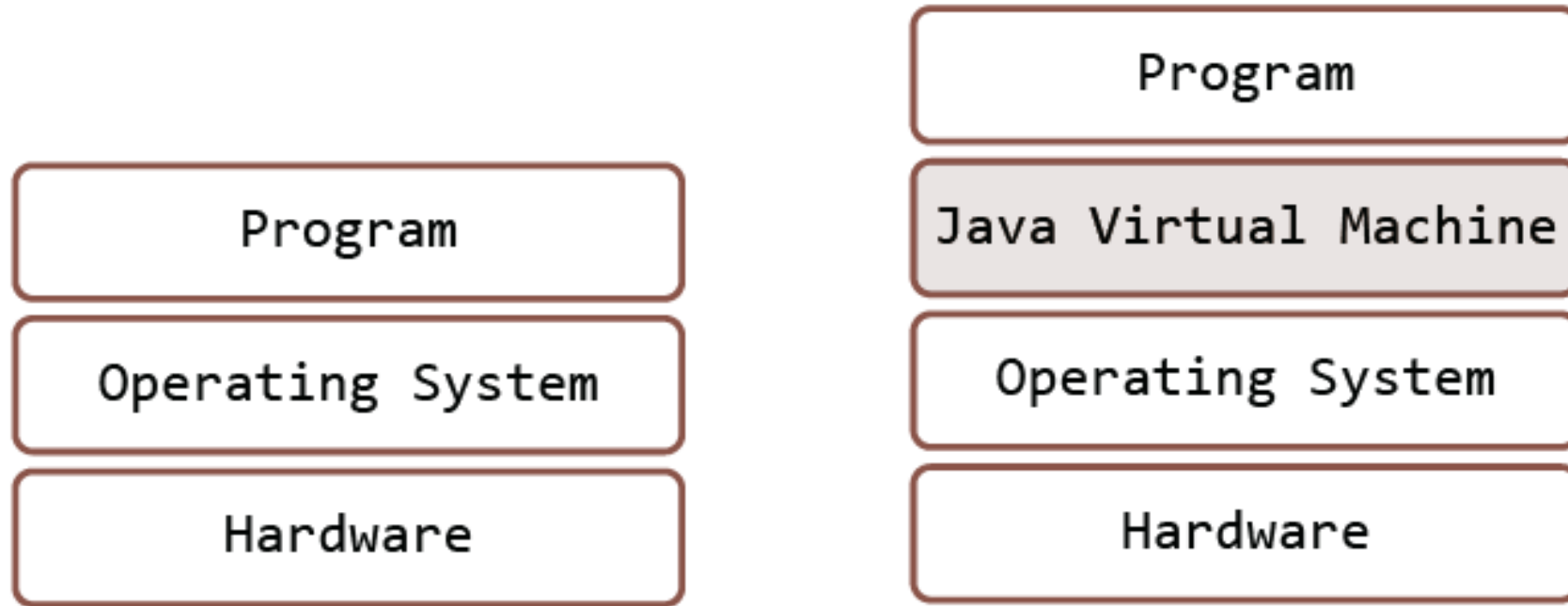
객체 지향 언어

자동 메모리 관리  
(Garbage Collection)

# ▶ JVM(Java Virtual Machine)

자바를 실행하기 위한 가상 기계로 플랫폼에 의존적  
byte code(class파일)를 해석하고 실행하는 interpreter





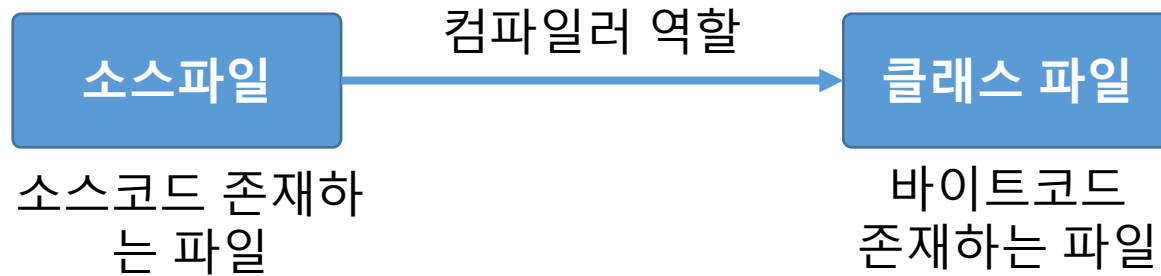
일반적인 프로그램과 자바 프로그램의 차이



운영체제에 따른 자바 가상머신의 차이

## ■ 자바 컴파일러와 자바 바이트코드

### ▶ 자바 컴파일러 (javac.exe)



### ▶ 자바 런처 (java.exe)

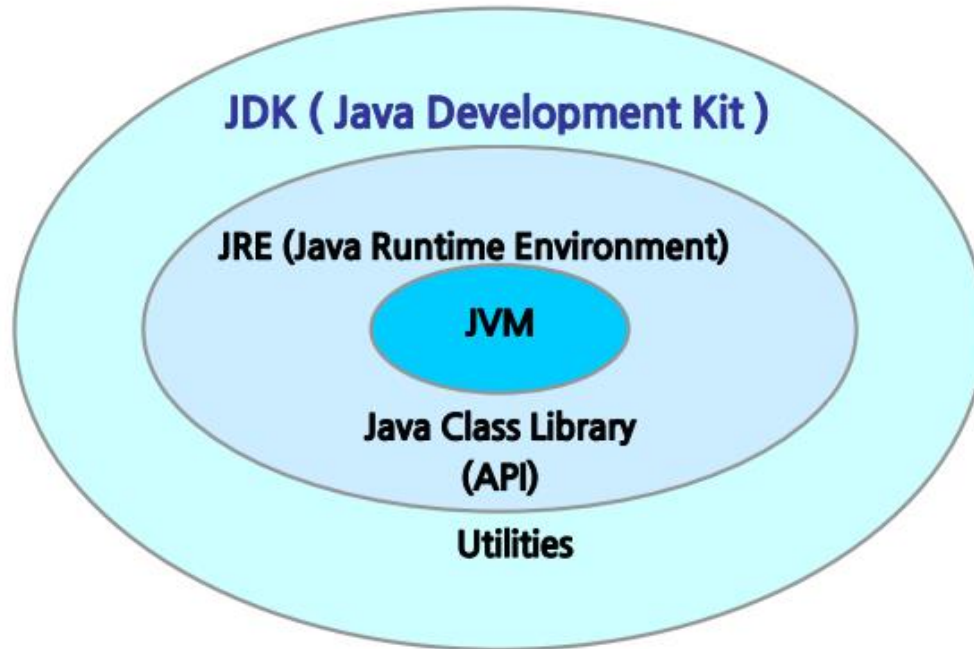
- 자바 프로그램과 자바 가상머신을 처음 구동하는 소프트웨어
- 클래스 파일을 대상으로 구동을 시작한다.



# ▶ 자바 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

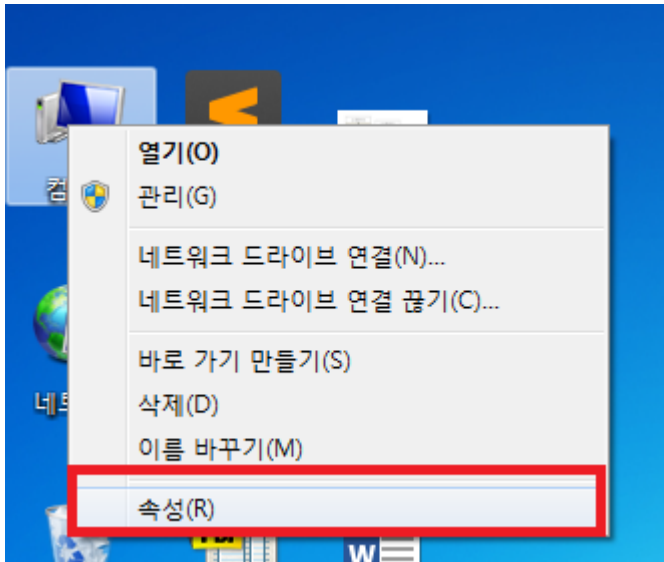
## ▶ 자바 설치

Oracle 홈페이지 접속([www.oracle.com](http://www.oracle.com)) – 다운로드 클릭

Java 11 (JDK 11) 설치 – 오라클 계정 필요

# ▶ 자바 설정

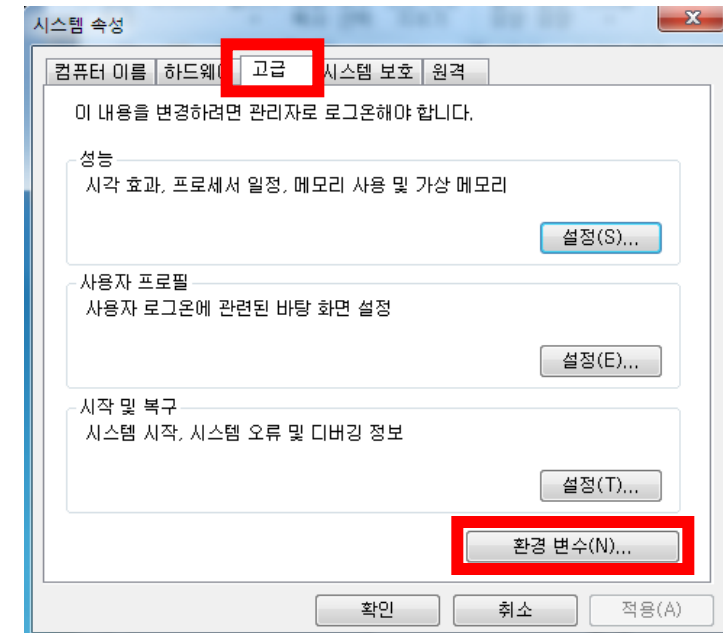
‘내 컴퓨터’ 오른쪽 클릭 - 속성



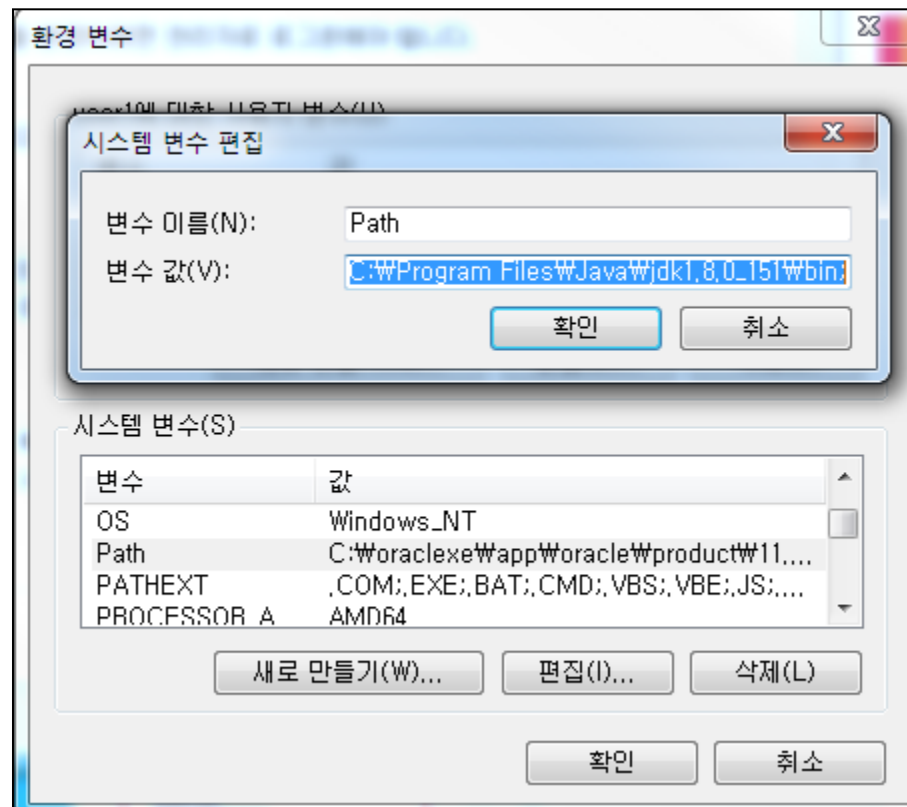
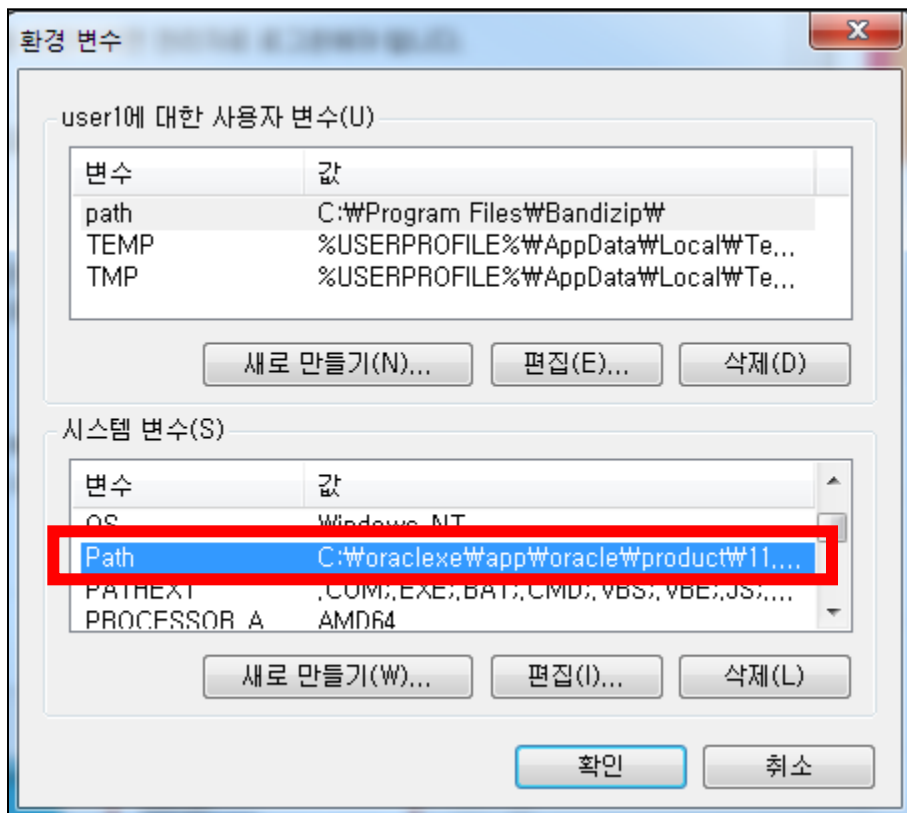
고급시스템 설정



고급 - 환경변수



# ▶ 자바 설정



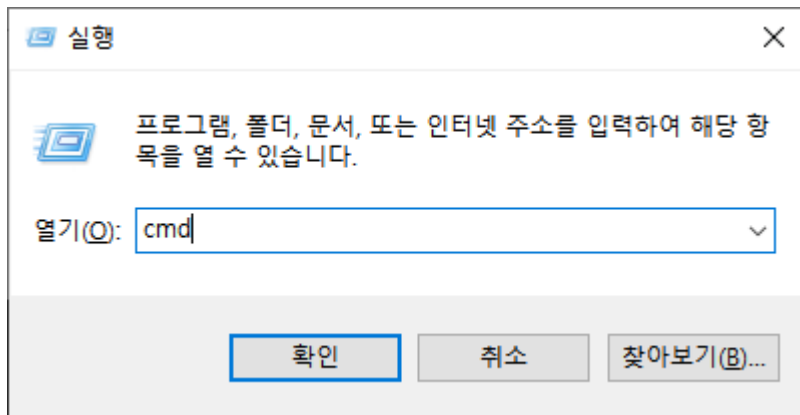
변수 값 : JDK가 설치된 폴더 경로 추가

(Java.exe / javac.exe 파일이 있는 경로)

(C:\Program Files\Java\jdk1.8.0\_202\bin)

# ▶ 자바 설정

## ✓ 환경변수 테스트



윈도우키 + R 버튼 누르고 실행창에서 cmd 입력 후 확인

```
C:\Users\User1>java -version
java version "11.0.25" 2024-10-15 LTS
Java(TM) SE Runtime Environment 18.9 (build 11.0.25+9-LTS-256)
Java HotSpot(TM) 64-Bit Server VM 18.9 (build 11.0.25+9-LTS-256, mixed mode)

C:\Users\User1>javac -version
javac 11.0.25

C:\Users\User1>_
```

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 HelloWorld{  
    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 : HelloWorld.java

파일 형식 : 모든파일

### 4. cmd창 열기

시작 – 실행 – cmd 입력( 또는 윈도우 키 + r)

### 5. cmd 창에서 workspace 경로 찾아가기

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

# ▶ 자바 개발 환경

## ✓ Compile Test

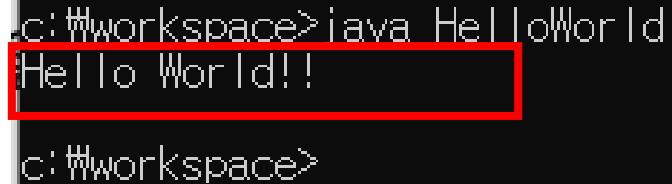
### 6. Compile 하기

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

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

### 7. 실행하기 (HelloWorld!! 메시지 출력 확인)

```
C:\workspace> java HelloWorld
```



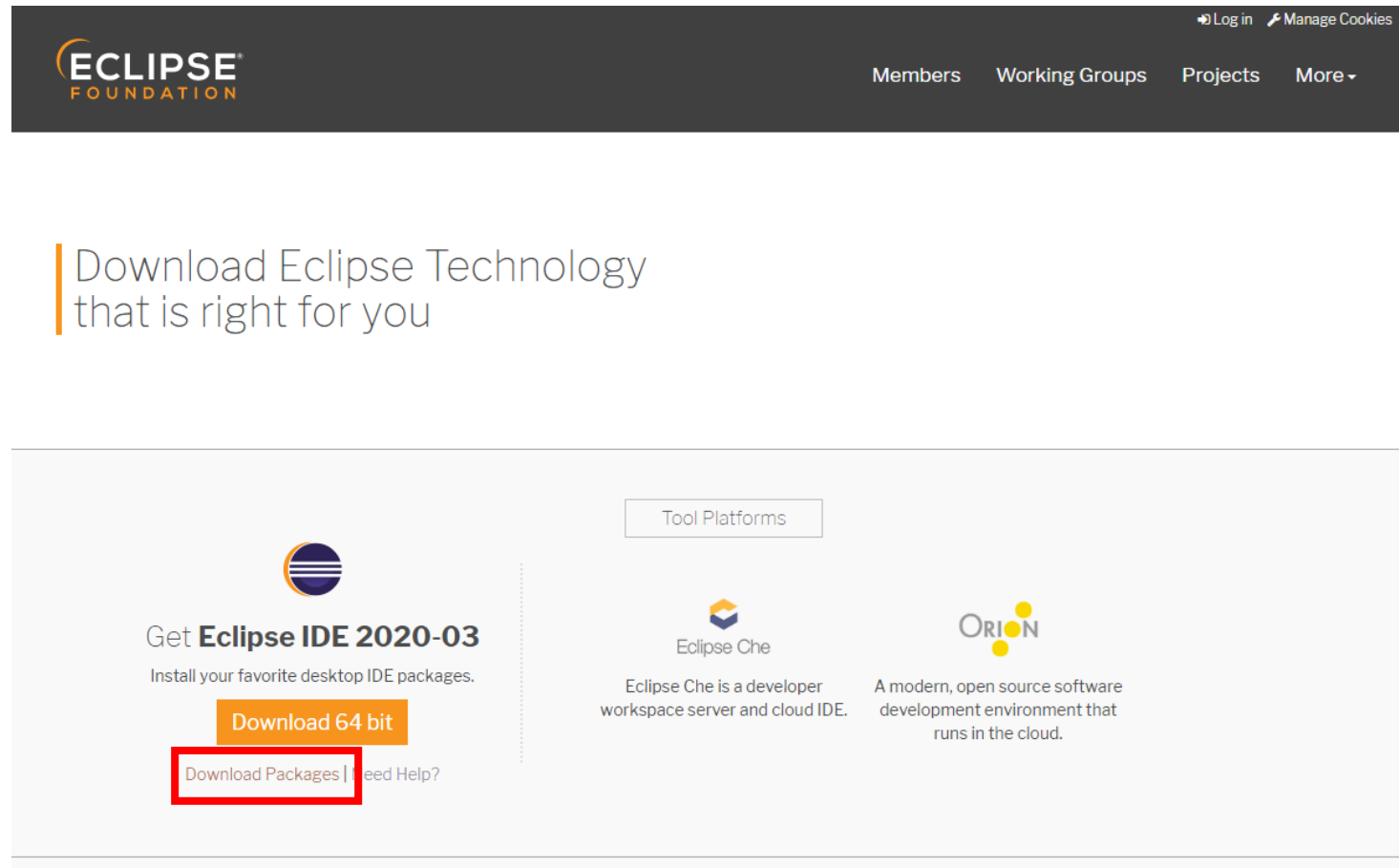
```
c:\workspace> java HelloWorld
Hello World!!
c:\workspace>
```



## ▶ 이클립스 설치

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


- Download Package 클릭



# ▶ 이클립스 설치


## MORE DOWNLOADS – Eclipse 2019-03(4.11) 클릭

### Eclipse IDE 2020-03 R Packages




**Eclipse IDE for Java Developers**  
196 MB 167,304 DOWNLOADS  
The essential tools for any Java developer, including a Java IDE, a Git client, XML Editor, Mylyn, Maven and Gradle integration

Windows 64-bit  
Mac Cocoa 64-bit  
Linux 64-bit




**Eclipse IDE for Enterprise Java Developers (includes Incubating components)**  
400 MB 137,489 DOWNLOADS  
Tools for developers creating Java Enterprise and Web applications, including a Java IDE, tools for Web Services, JPA and Data Tools, JSF, Mylyn, Maven and Gradle, Git, and more.  
Click here to file a bug against Eclipse Web Tools Platform.  
Click here to file a bug against Eclipse Platform.  
Click here to file a bug against Maven integration for web projects.  
Click here to report an issue against Eclipse Wild Web Developer (incubating).

Windows 64-bit  
Mac Cocoa 64-bit  
Linux 64-bit




**Eclipse IDE for C/C++ Developers (includes Incubating components)**  
235 MB 29,421 DOWNLOADS  
An IDE for C/C++ developers with Mylyn integration.

Windows 64-bit  
Mac Cocoa 64-bit  
Linux 64-bit



**Eclipse IDE for Eclipse Committers**  
362 MB 16,829 DOWNLOADS  
Package suited for development of Eclipse itself at Eclipse.org; based on the Eclipse Platform adding PDE, Git, Marketplace Client, source code and developer documentation.  
Click here to file a bug against Eclipse Platform.  
Click here to file a bug against Eclipse Git team provider.

Windows 64-bit  
Mac Cocoa 64-bit  
Linux 64-bit



### Get Eclipse IDE 2020-03

Install your favorite desktop IDE packages.

**Download 64 bit**

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

#### RELATED LINKS

- [Compare & Combine Packages](#)
- [New and Noteworthy](#)
- [Install Guide](#)
- [Documentation](#)
- [Updating Eclipse](#)
- [Forums](#)
- [Simultaneous Release](#)


#### MORE DOWNLOADS

- [Other builds](#)
- [Eclipse 2019-12 \(4.14\)](#)
- [Eclipse 2019-09 \(4.13\)](#)
- [Eclipse 2019-06 \(4.12\)](#)
- [Eclipse 2019-03 \(4.11\)](#)
- [Eclipse 2018-12 \(4.10\)](#)
- [Eclipse 2018-09 \(4.9\)](#)
- [Eclipse Photon \(4.8\)](#)
- [Eclipse Oxygen \(4.7\)](#)
- [Eclipse Neon \(4.6\)](#)
- [Older Versions](#)

# ▶ 이클립스 설치

## Eclipse IDE for Enterprise Java Developers - 해당되는 운영체제 클릭 - Download 클릭


Eclipse IDE 2019-03 R Packages



**Eclipse IDE for Enterprise Java Developers**  
340 MB | 549,185 DOWNLOADS

Tools for Java developers creating Enterprise Java and Web applications, including a Java IDE, tools for Enterprise Java, JPA, JSF, Mylyn, Maven, Git and more.  
[Click here to file a bug against Eclipse Web Tools Platform.](#)  
[Click here to file a bug against Eclipse Platform.](#)  
[Click here to file a bug against Maven integration for web projects.](#)


Windows 64-bit  
Mac Cocoa 64-bit  
Linux 64-bit



**Eclipse IDE for Java Developers**  
189 MB | 358,809 DOWNLOADS

The essential tools for any Java developer, including a Java IDE, a Git client, XML Editor, Mylyn, Maven and Gradle integration


Windows 64-bit  
Mac Cocoa 64-bit  
Linux 64-bit



**Eclipse IDE for C/C++ Developers**  
229 MB | 252,247 DOWNLOADS

An IDE for C/C++ developers with Mylyn integration.

Windows 64-bit  
Mac Cocoa 64-bit  
Linux 64-bit



**Eclipse IDE for Eclipse Committers**  
316 MB | 47,009 DOWNLOADS

Package suited for development of Eclipse itself at Eclipse.org; based on the Eclipse Platform adding PDE, Git, Marketplace Client, source code and developer documentation.

Windows 64-bit  
Mac Cocoa 64-bit  
Linux 64-bit

[Click here to file a bug against Eclipse Platform.](#)

**ECLIPSE FOUNDATION** Members Working

Home / Downloads / Eclipse downloads - Select a mirror

All downloads are provided under the terms and conditions of the Eclipse Foundation Software User Agreement unless otherwise specified.

[Download](#)

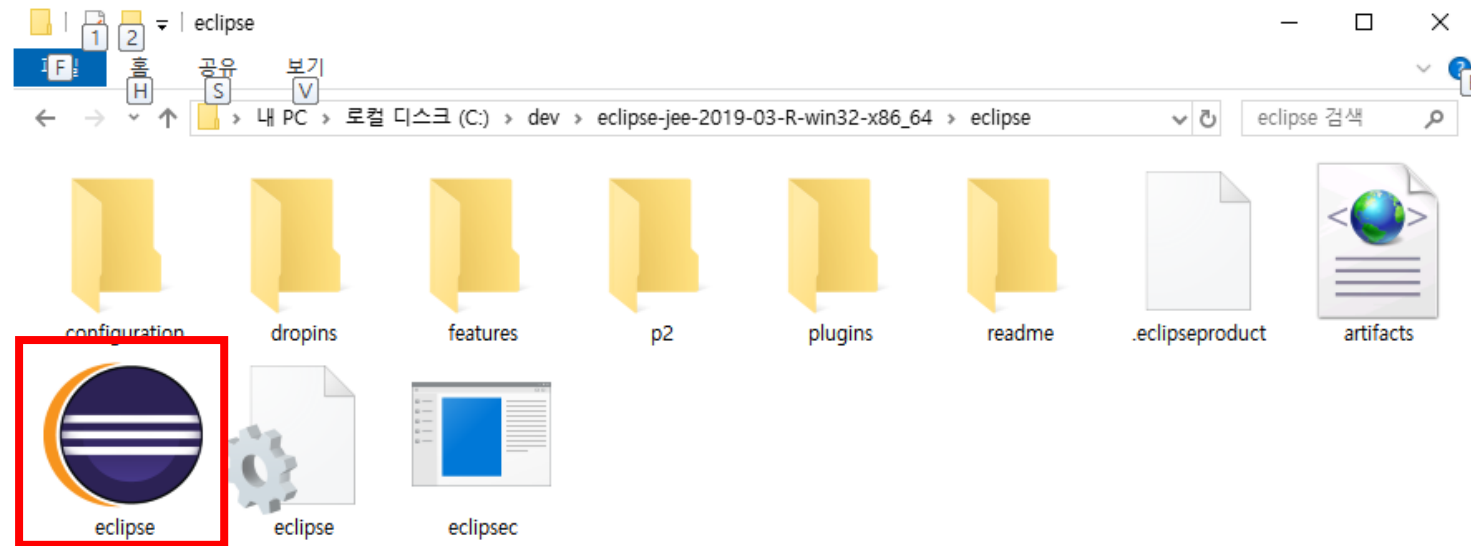
**Download from:** Japan - Japan Advanced Institute of Science and Technology (http)

**File:** eclipse-jee-2019-03-R-win32-x86\_64.zip **SHA-512**

[>> Select Another Mirror](#)

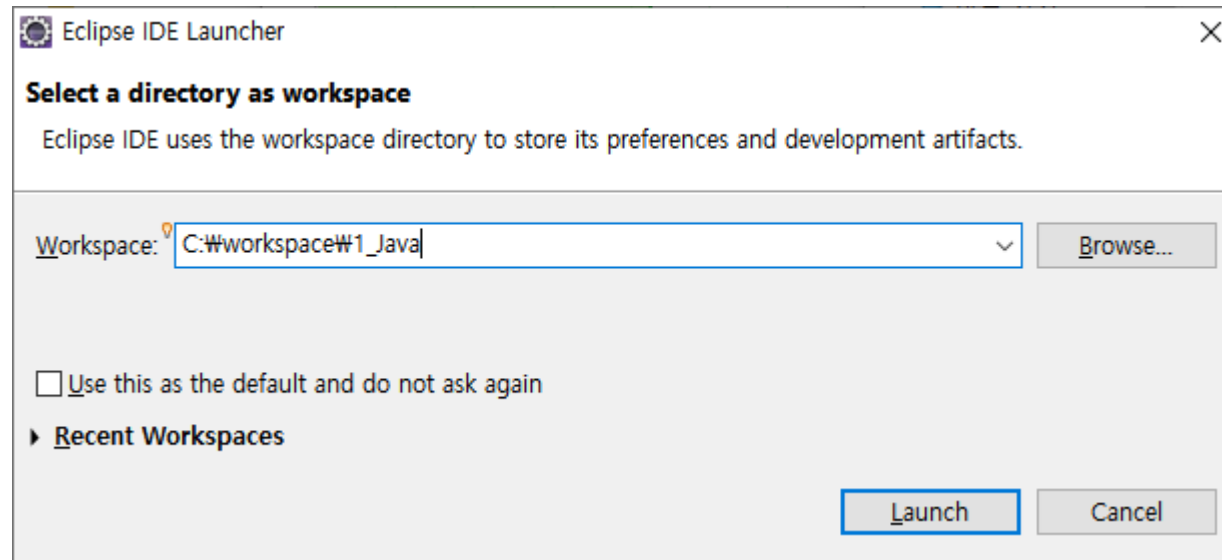
## ▶ 이클립스 설치

C드라이브에 dev 폴더를 생성하여 다운로드된 파일을 이동  
- 압축 해제 - eclipse.exe 실행



## ▶ 이클립스 설치

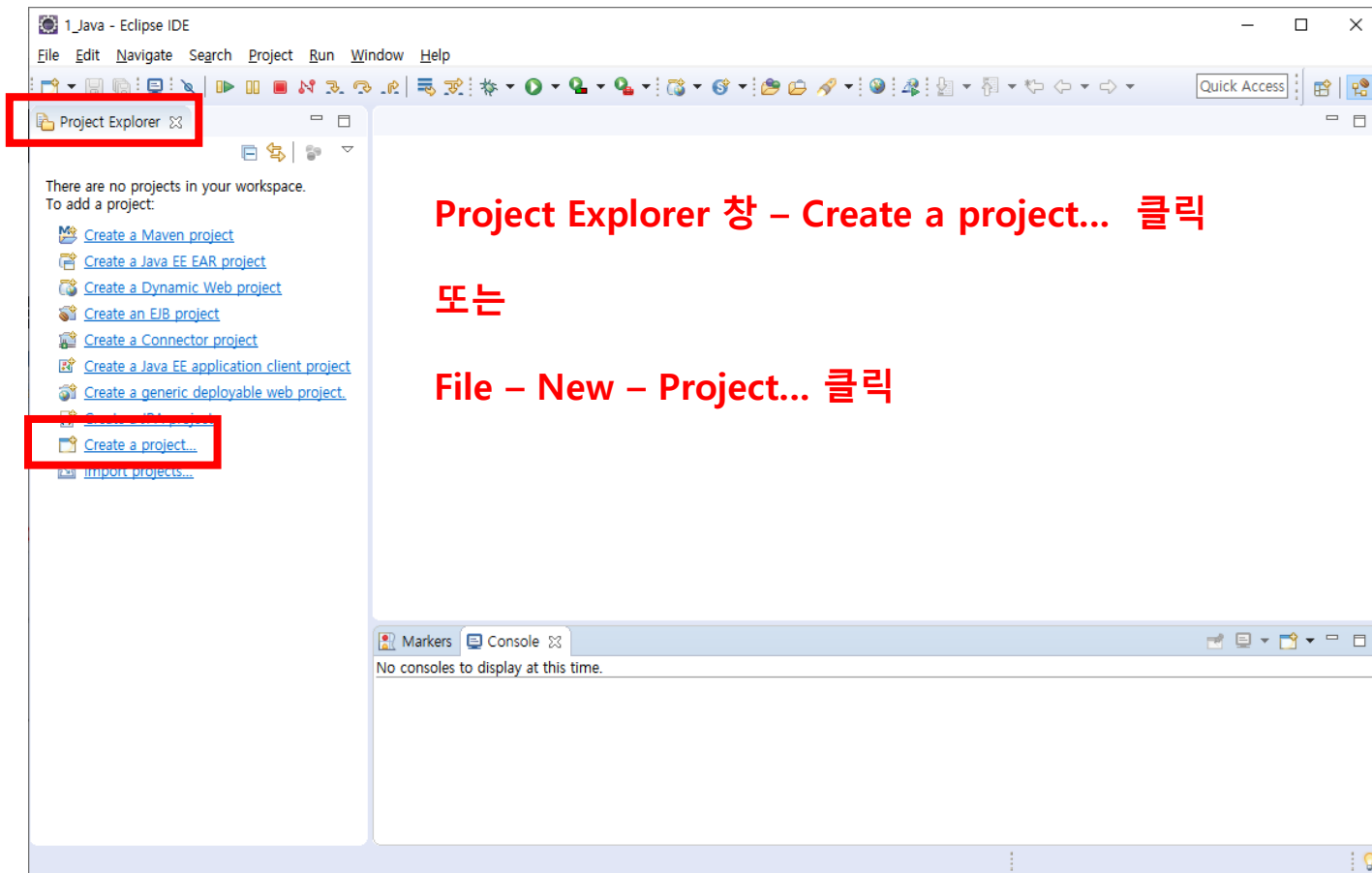
Workspace 입력란에 C:\workspace\1\_Java 작성 후 Launch 클릭



# ▶ 자바 프로그래밍 순서

## ✓ Eclipse 환경

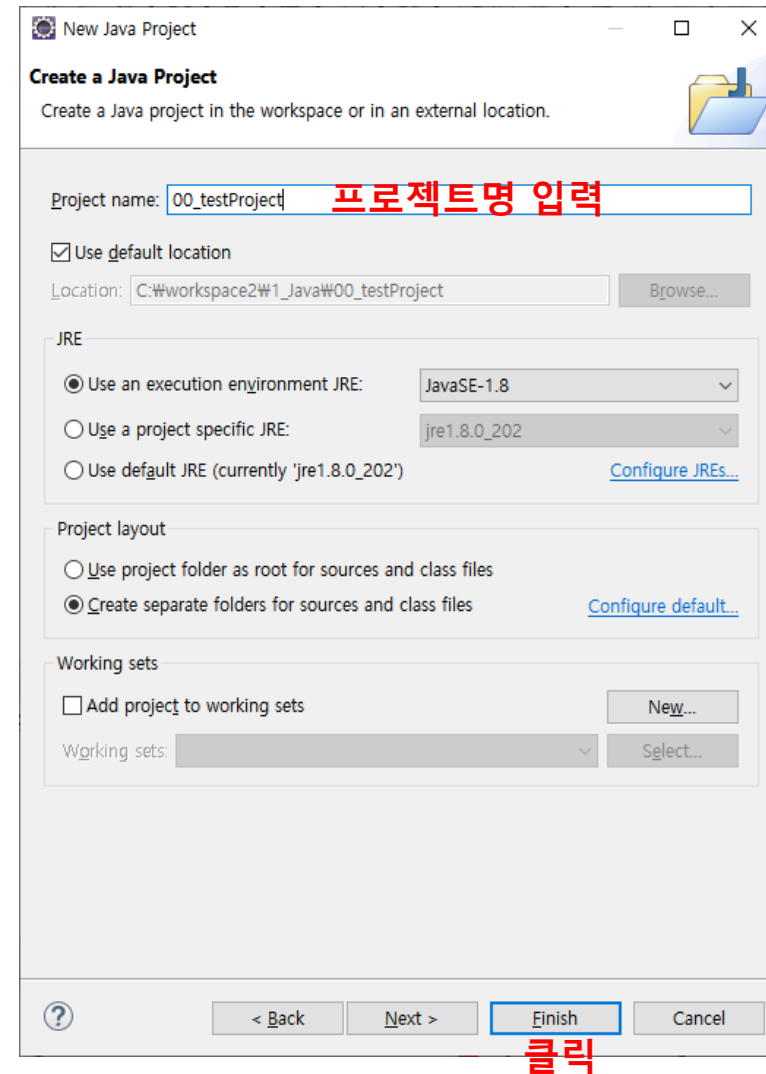
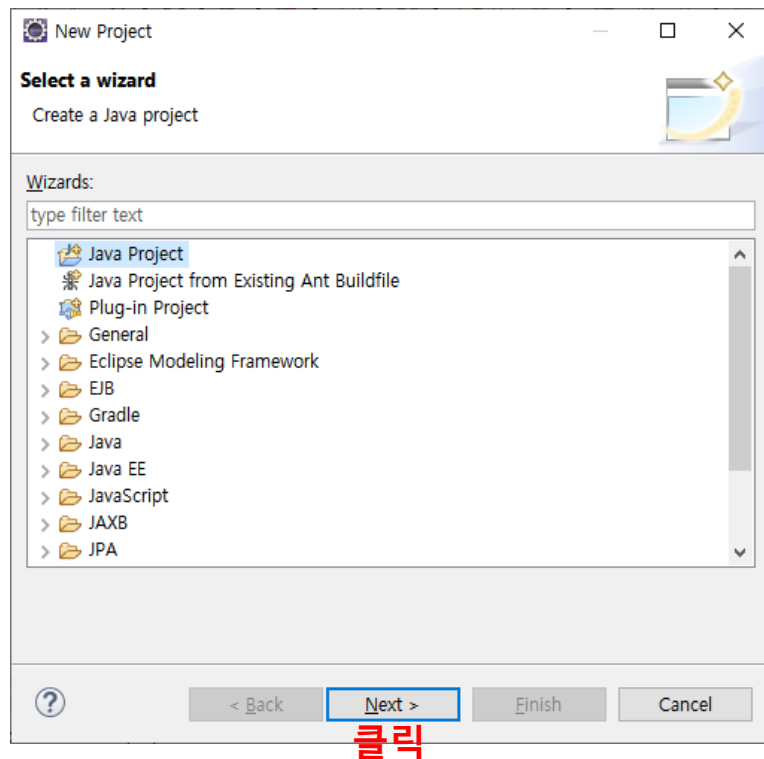
### 1. project 만들기



# ▶ 자바 프로그래밍 순서

## ✓ Eclipse 환경

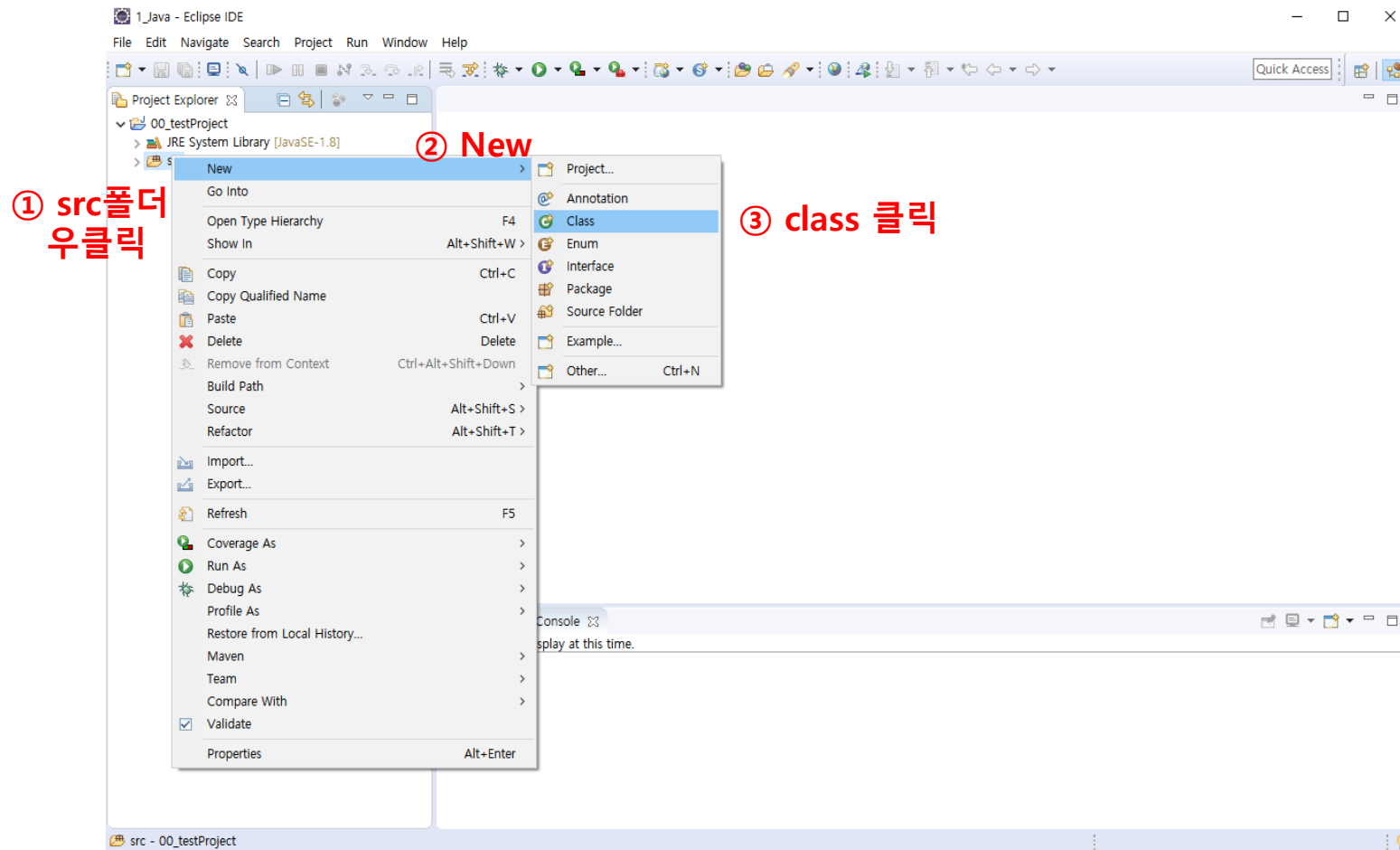
### 1. project 만들기



# ▶ 자바 프로그래밍 순서

## ✓ Eclipse 환경

### 2. Class 만들기

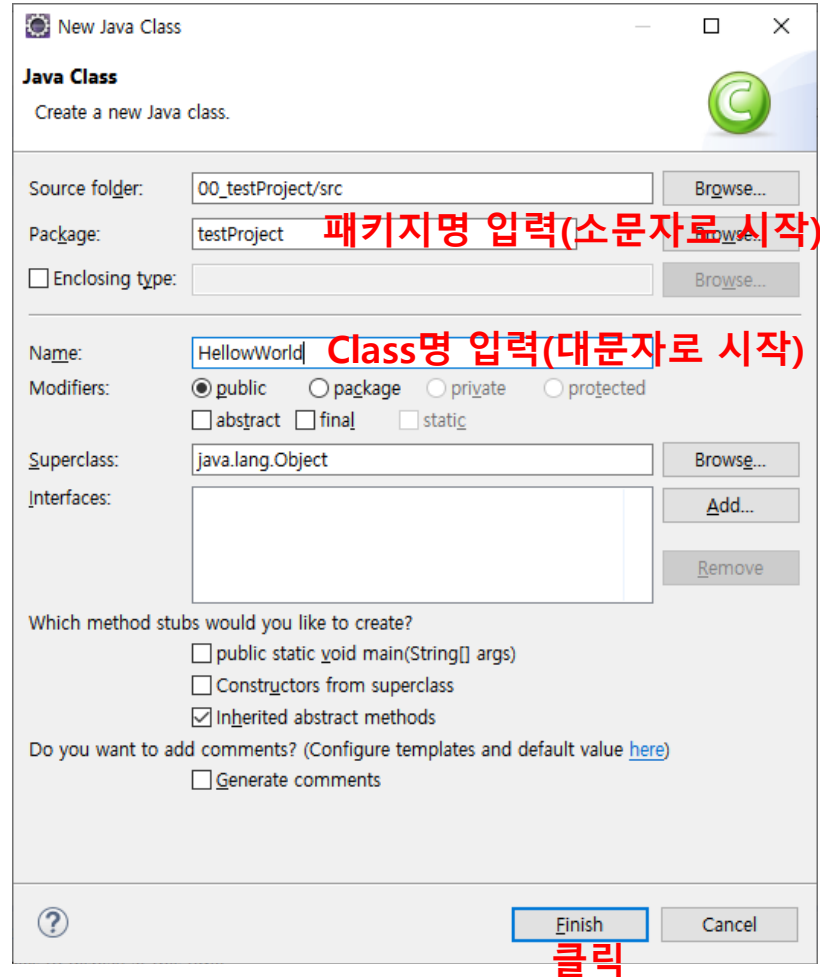




# ▶ 자바 프로그래밍 순서

## ✓ Eclipse 환경


### 2. Class 만들기

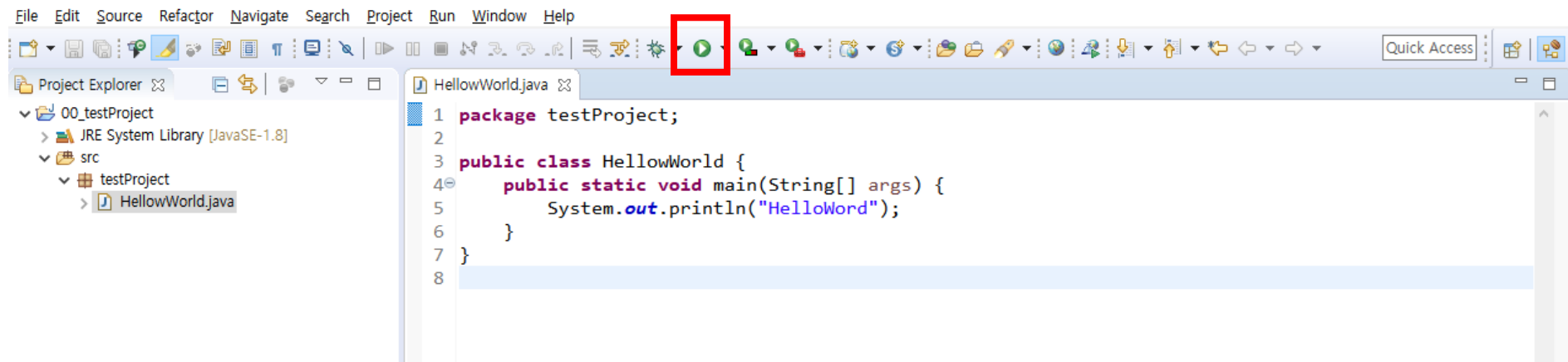


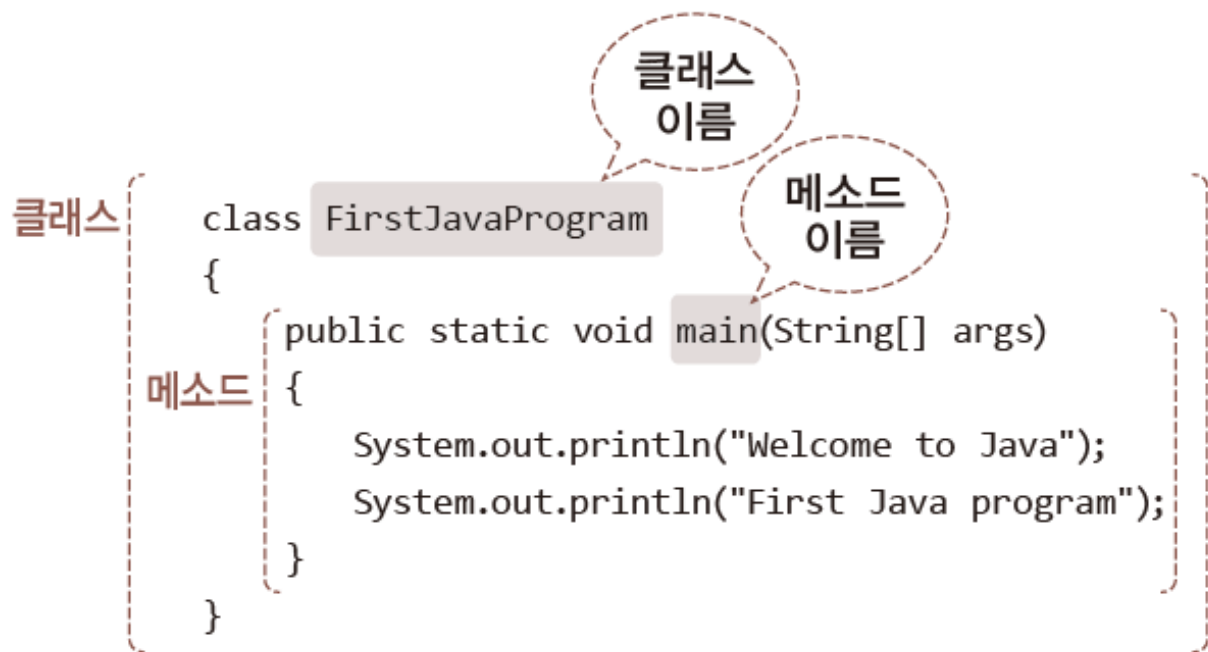
# ▶ 자바 프로그래밍 순서

## ✓ Eclipse 환경

### 3. Class 작성 후 실행

Class 작성 완료 후  클릭 또는 Ctrl + F11을 눌러 실행





- 중괄호를 이용해서 클래스와 메소드의 영역을 구분
- 문장의 끝에는 세미콜론을 붙여서 문장의 끝 표시

```
명령 프롬프트
C:\JavaStudy>java FirstJavaProgram
Welcome to Java
First Java program
C:\JavaStudy>
```

- 프로그램 실행 시 main 메소드 안 문장들 순차적 실행
- System.out.println의 괄호 안에 출력 내용 큰따옴표로 묶어서 표시
- System.out.println 실행 이후 자동 개행

## 프로그램의 골격과 구성

# ▶ 자바 프로그래밍 기본

## ✓ class

객체를 만들기 위한 **일종의 설계도**.

자바에서 모든 코드는 반드시 클래스 안에 존재해야 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 뛴다

`/* */` : 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 형태의 메서드 선언부**로

Java Application을 실행하는데 필요한 메서드 (프로그램 실행 시 `java.exe`에 의해 호출됨)

모든 클래스가 main메서드를 가지고 있어야 하는 것은 아니지만 **하나의 Java애플리케이션에는 main메서드를 포함한 클래스가 반드시 하나는 있어야 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;`  
`}`

} (멤버) 메서드

`... 이하 생략...`

`}`

01-4.

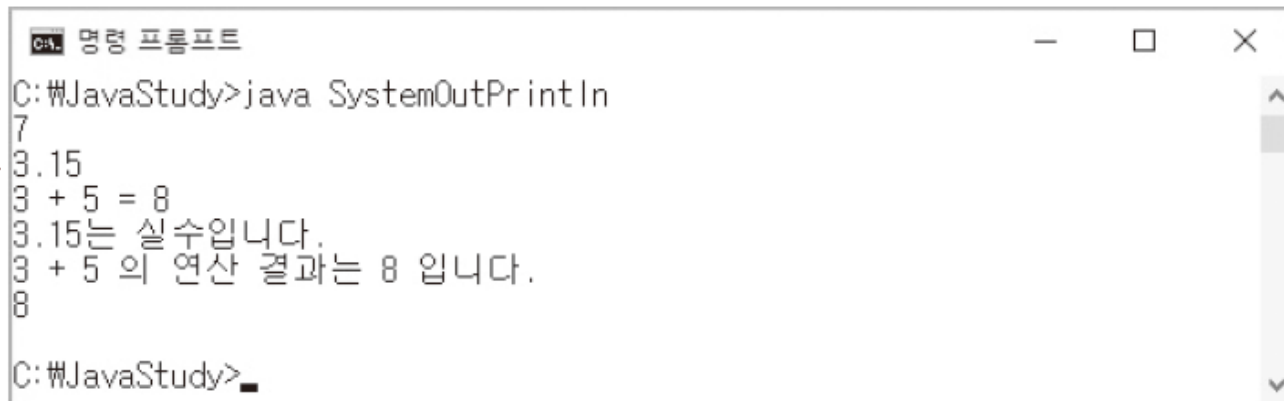
들여쓰기와 컴파일의  
대상에서 제외되는 주석!



# ■ System.out.println에 대한 다양한 활용

## ◆ SystemOutPrintIn.java

```
1. class SystemOutPrintIn
2. {
3.     public static void main(String[] args)
4.     {
5.         System.out.println(7);
6.         System.out.println(3.15);
7.         System.out.println("3 + 5 = " + 8);
8.         System.out.println(3.15 + "는 실수입니다.");
9.         System.out.println("3 + 5" + " 의 연산 결과는 8입니다.");
10.        System.out.println(3 + 5);
11.    }
12. }
```



```
C:\JavaStudy>java SystemOutPrintIn
7
3.15
3 + 5 = 8
3.15는 실수입니다.
3 + 5 의 연산 결과는 8 입니다.
8
C:\JavaStudy>
```

## ◆ BlockComment.java

```
1.  /*
2.   파일이름: BlockComment.java
3.   작 성 자: 홍길동
4.   작 성 일: 2019년 7월 25일
5.   목 적: System.out.println 메소드의 기능 테스트   주석 처리 영역
6.  */
7.
8.  class BlockComment
9.  {
10.     public static void main(String[] args)
11.     {
12.         /* 다음은 단순한 정수의 출력 */   주석 처리 영역
13.         System.out.println(7);
14.
15.         System.out.println(3.15);
16.         System.out.println("3 + 5 = " + 8);
17.         System.out.println(3.15 + "는 실수입니다.");
18.         System.out.println("3 + 5" + " 의 연산 결과는 8입니다.");
19.
20.         /* 다음은 덧셈 결과의 출력 */   주석 처리 영역
21.         System.out.println(3 + 5);
22.     }
23. }
```

블록 단위 주석

## ◆ BlockComment2.java

```
1.  /*
2.   * 파일이름: BlockComment2Java
3.   * 작 성 자: 홍길동
4.   * 작 성 일: 2019년 7월 25일
5.   * 목   적: System.out.println 메소드의 기능 테스트
6.  */
7.
8.  class BlockComment2
9.  {
```

블록 단위 주석의 다른 사례

## ◆ LineComment.java

```
1. // 파일이름: LineComment.java
2. // 작 성 자: 홍길동
3. // 작 성 일: 2019년 7월 25일
4. // 목   적: System.out.println 메소드의 기능 테스트
5.
6. class SystemOutPrintln
7. {
8.     public static void main(String[] args)
9.     {
10.         System.out.println(7);    // 다음은 단순한 정수의 출력
11.
12.         System.out.println(3.15);
13.         System.out.println("3 + 5 = " + 8);
14.         System.out.println(3.15 + "는 실수입니다.");
15.         System.out.println("3 + 5" + " 의 연산 결과는 8입니다.");
16.
17.         System.out.println(3+5);   // 덧셈 결과의 출력
18.     }
19. }
```

행 단위 주석

```
1. class SystemOutPrintln
2. {
3.     // 4칸 정도 오른쪽으로 들여쓰기 되었다.
4.     public static void main(String[] args)
5.     {
6.         // main 메소드 내에서 4칸 정도 오른쪽으로 들여쓰기 되었다.
7.         System.out.println(7);
8.         ....
9.     }
10. }
```

들여 쓰기

```
1. class SystemOutPrintln {
2.     public static void main(String[] args) {
3.         System.out.println(7);
4.         ....
5.     }
6. }
```

자바에서 권고 및 추천되는 방식

```
1. class SystemOutPrintln
2. {
3.     public static void main(String[] args)
4.     {
5.         System.out.println(7);
6.         ....
7.     }
8. }
```