

목차

1. 소개
2. 개발환경 구축
3. 프로젝트/패키지 생성
4. 기본 문법
5. 출력 명령어 연습
6. 변수
7. 변수 생성 연습

개발환경 구축

■ JDK18 설치

- JDK는 자바로 프로그램을 개발하는데 필요한 도구를 모아 둔 것
- <https://www.oracle.com/java/technologies/downloads/#java18>로 접속합니다.
- x64 Installer 다운

Java 18 Java 17

Java SE Development Kit 18 downloads

Thank you for downloading this release of the Java™ Platform, Standard Edition Development Kit (JDK™). The JDK is a development environment for building applications and components using the Java programming language.

The JDK includes tools for developing and testing programs written in the Java programming language and running on the Java platform.

Linux macOS **Windows**

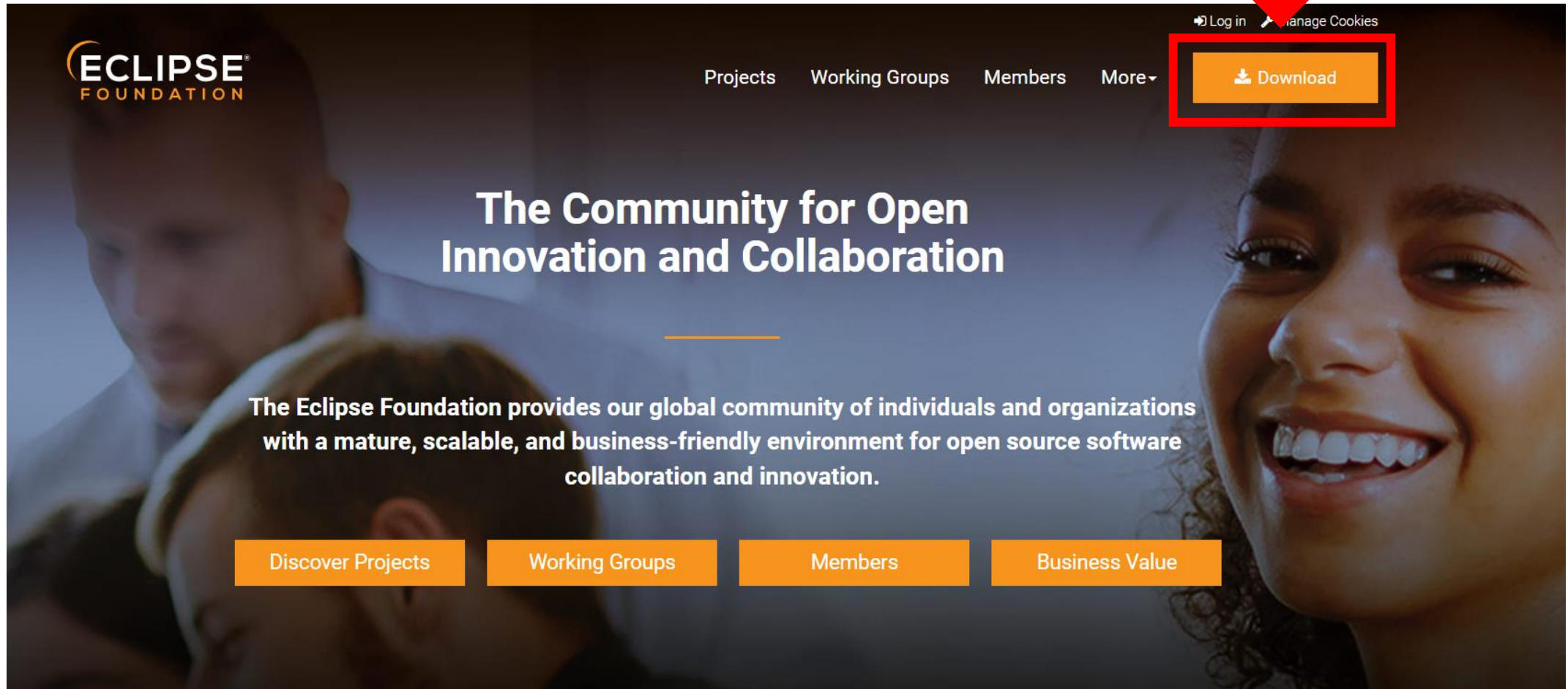
Product/file description	File size	Download
x64 Compressed Archive	172.54 MB	https://download.oracle.com/java/18/latest/jdk-18_windows-x64_bin.zip (sha256 ↗)
x64 Installer	153.2 MB	https://download.oracle.com/java/18/latest/jdk-18_windows-x64_bin.exe (sha256 ↗)
x64 MSI Installer	152.08 MB	https://download.oracle.com/java/18/latest/jdk-18_windows-x64_bin.msi (sha256 ↗)



개발환경 구축

■ Eclipse 설치

- <https://www.eclipse.org/> 로 접속합니다.
- 오른쪽 상단의 Download 버튼을 클릭합니다.



개발환경 구축

■ Eclipse 설치

that is right for you

Sponsored Ad

JAKARTA EE 2022 Jakarta EE Developer Survey PARTICIPATE TODAY!

Advertise Here

Advertise Here

Eclipse IDE Tools

OpenJDK Runtimes

Get **Eclipse IDE 2022-03**
Install your favorite desktop IDE packages.

Download x86_64

Download Packages | Need Help?

TEMURIN
by ADOPTIUM

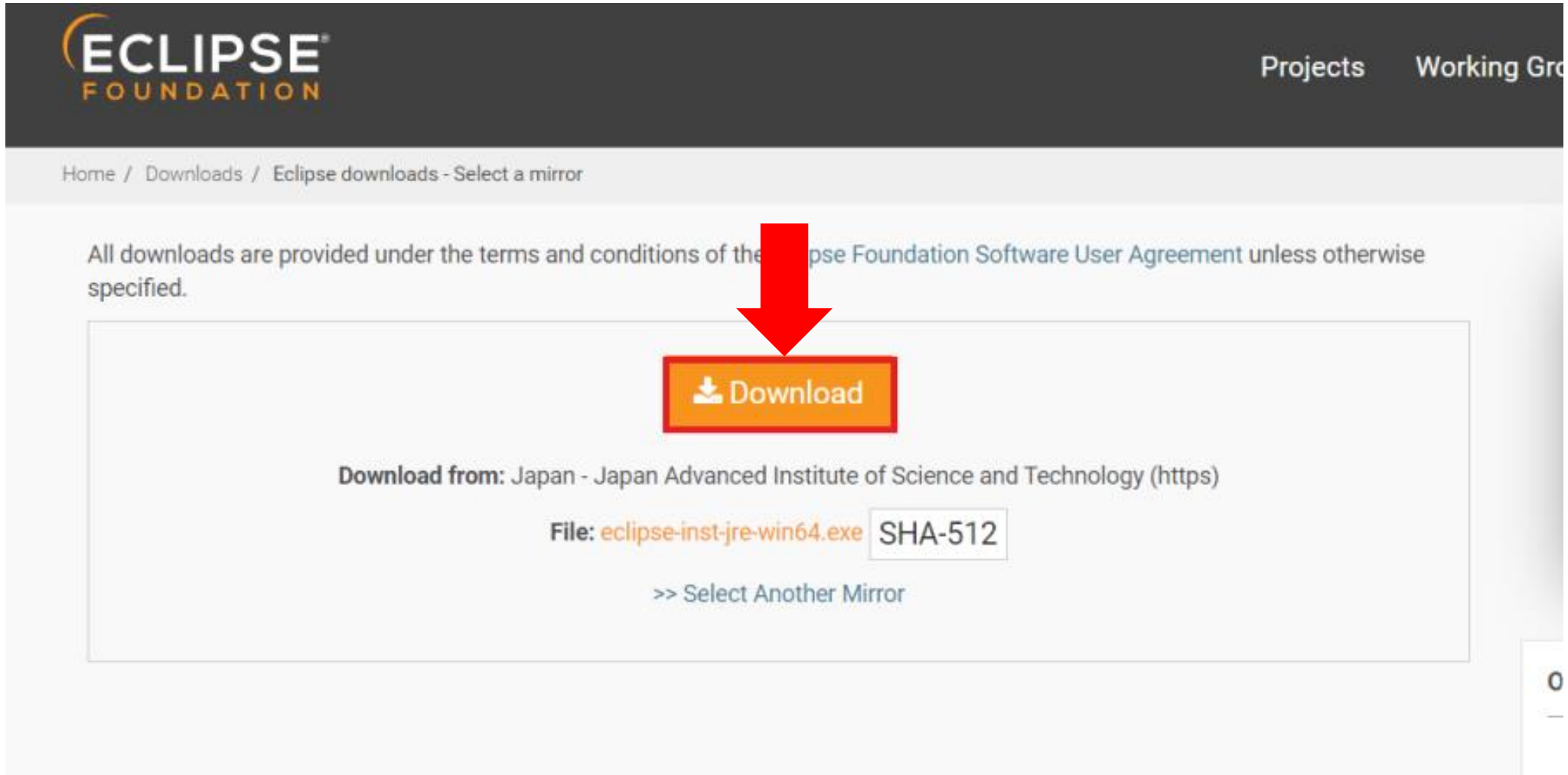
The Eclipse Temurin™ project provides high-quality, TCK certified OpenJDK runtimes and associated technology for use across the Java™ ecosystem.

Download Now

Learn More

개발환경 구축

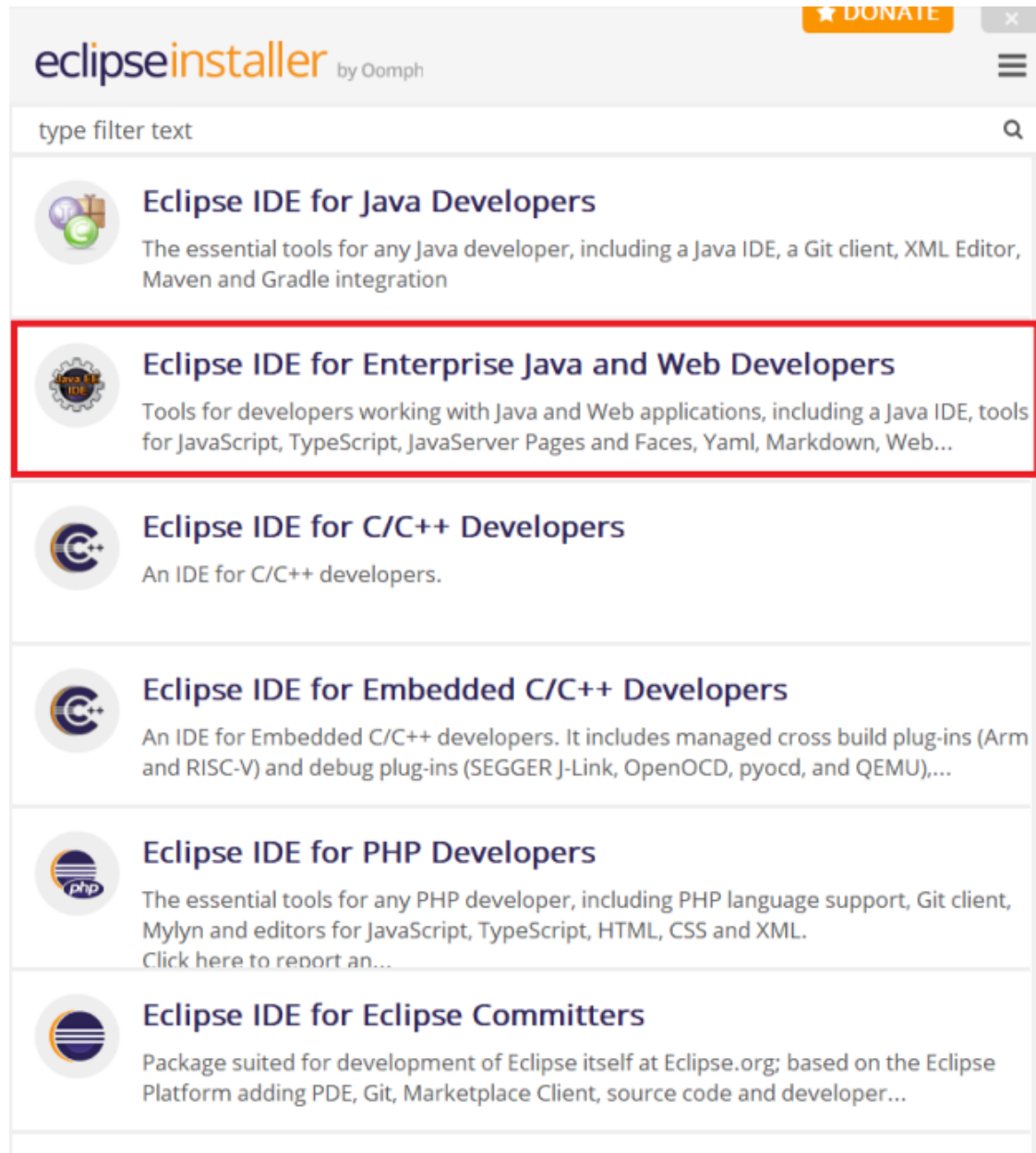
■ Eclipse 설치



개발환경 구축

■ Eclipse 설치

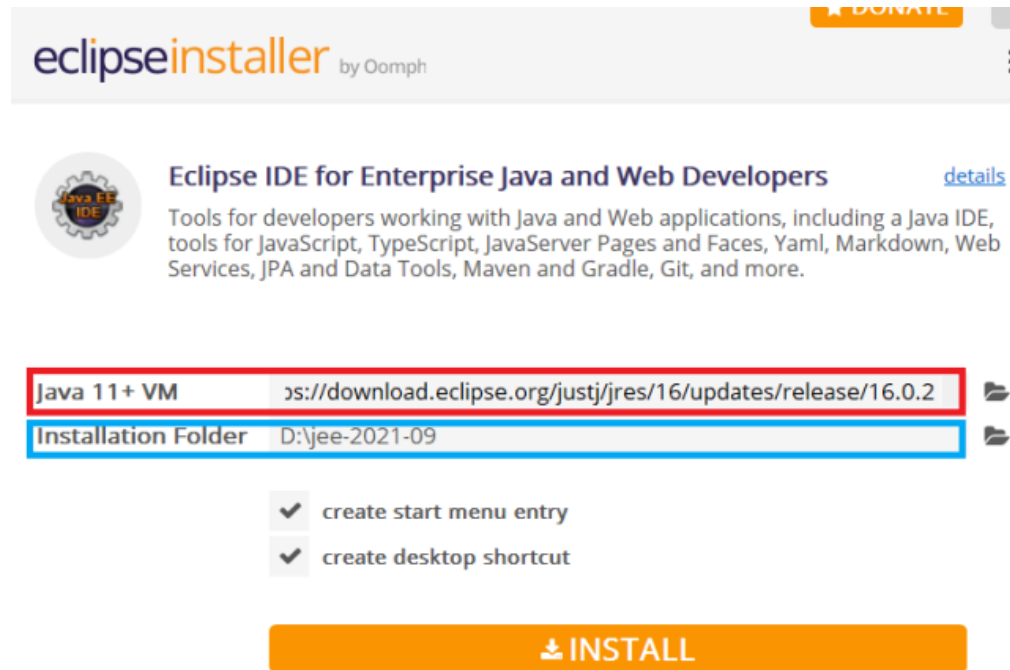
- 설치 파일 다운로드 후 실행
- 실행해서 "Eclipse IDE for Enterprise Java and Web Developers" 선택



개발환경 구축

■ Eclipse 설치

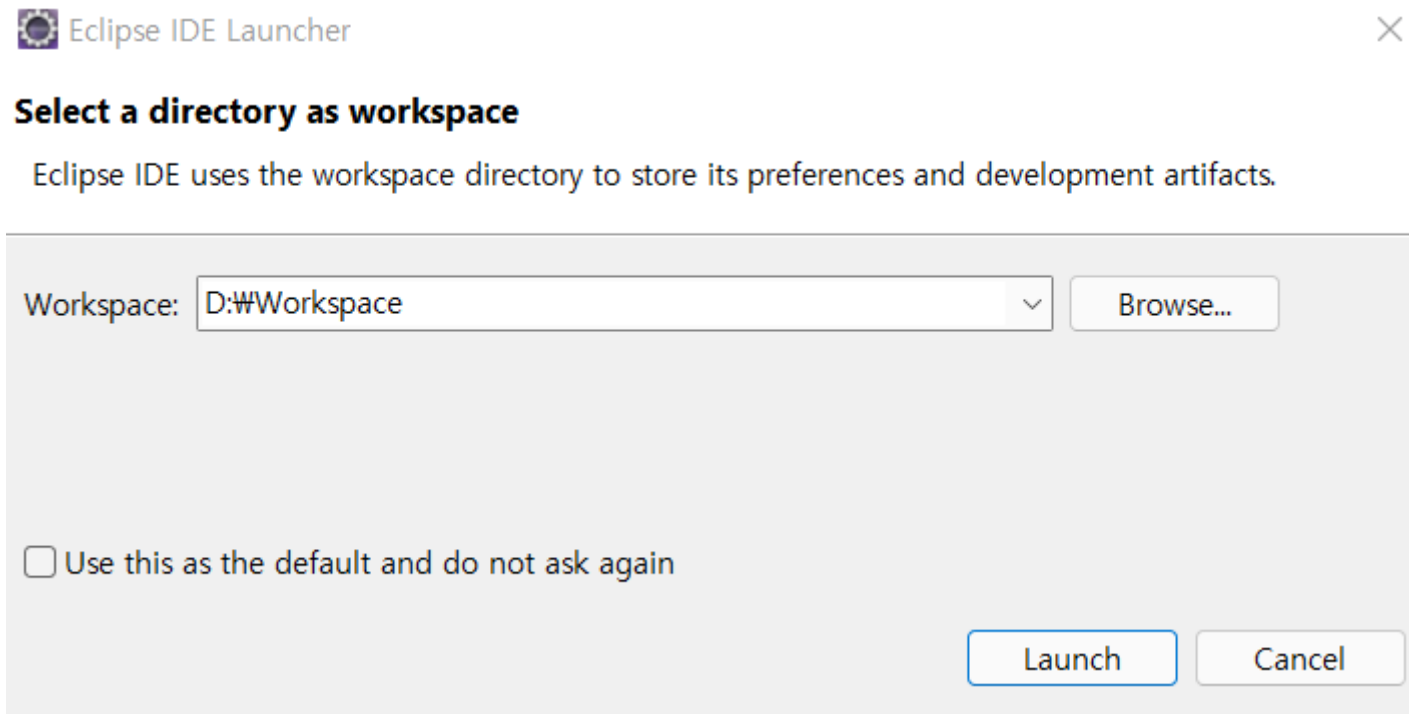
- 빨간 표시된 부분은 설치하는 PC에 Java 11 버전 이상이 없을 시 해당 경로에서 다운로드
- 파랗게 표시된 부분은 이클립스가 설치 될 경로
- 설치할 경로를 지정 후 하단의 INSTALL 버튼을 클릭



개발환경 구축

■ Eclipse 실행

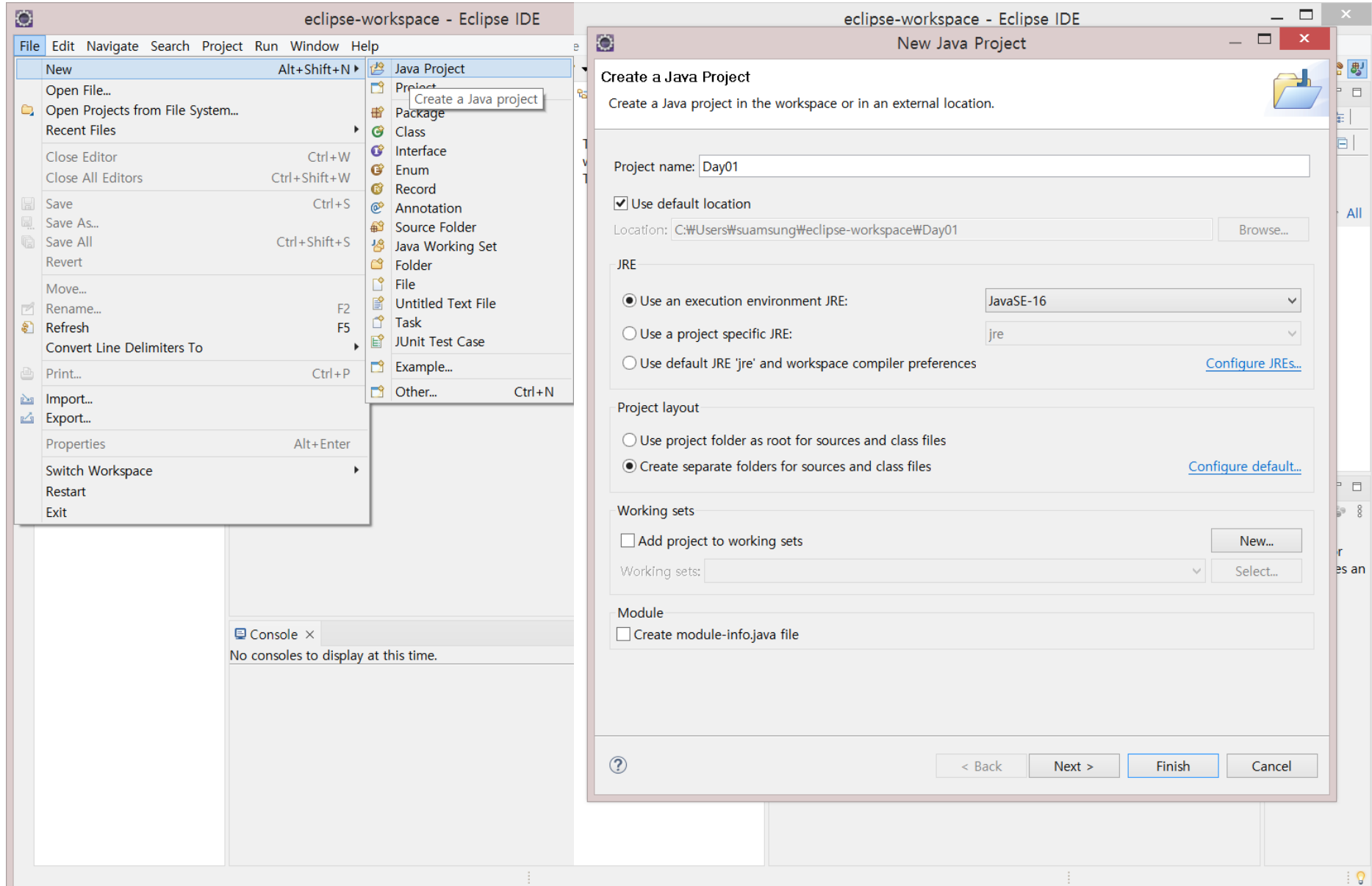
- Workspace 경로 설정
 - ▶ Workspace 경로에 앞으로의 프로젝트들이 저장됨



프로젝트/패키지 생성

■ 프로젝트 생성

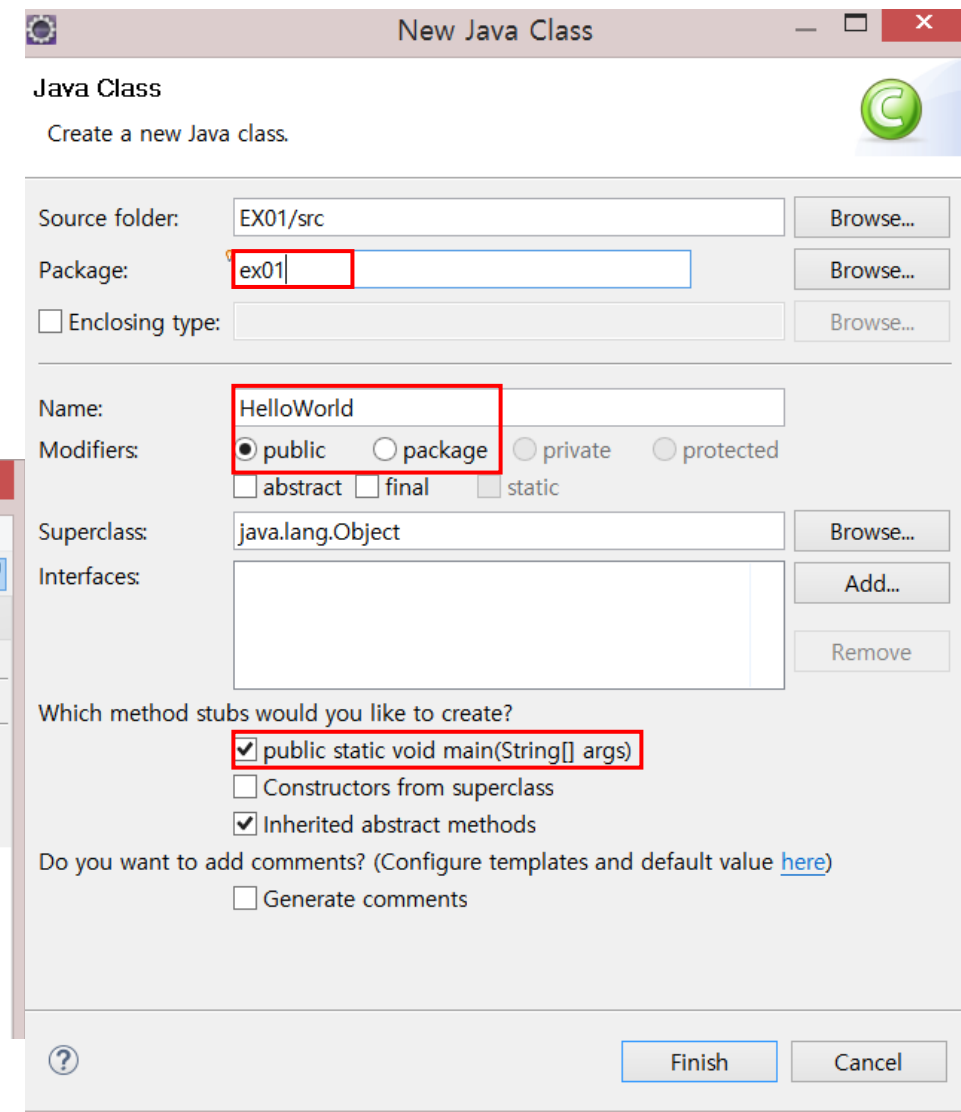
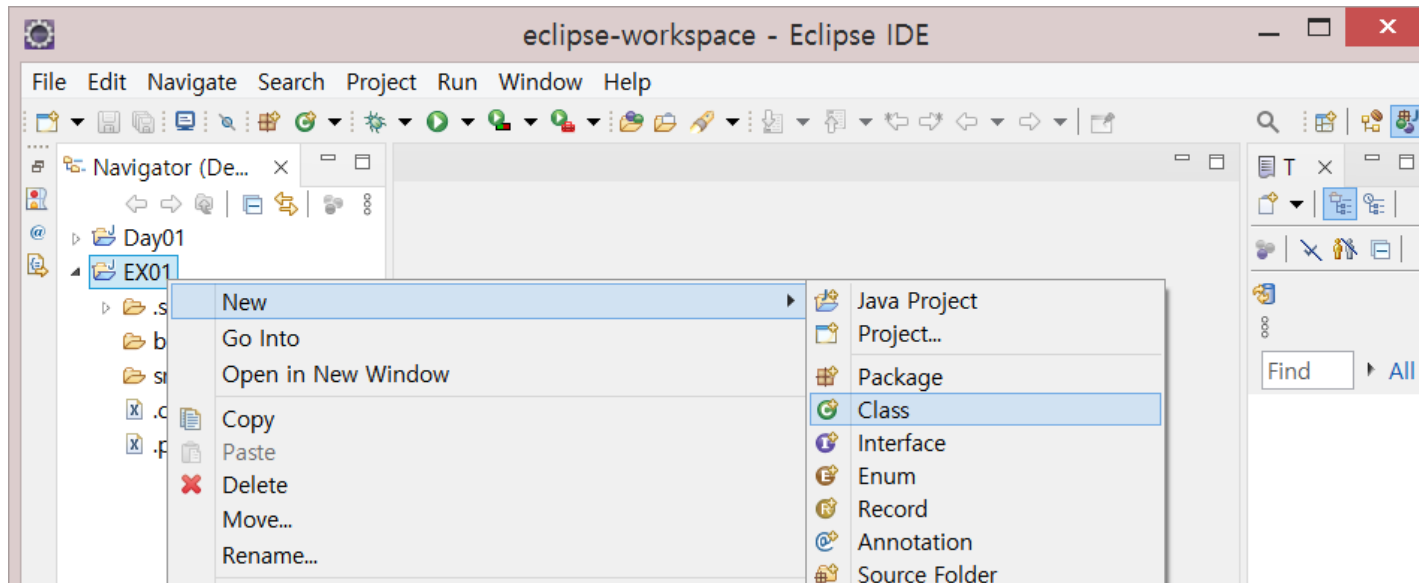
- [File] -> [New] -> [Java Project]
- 수업 프로젝트 이름
▶ Day01
- 실습 프로젝트 이름:
▶ EX01



프로젝트/패키지 생성

■ Class/패키지 생성

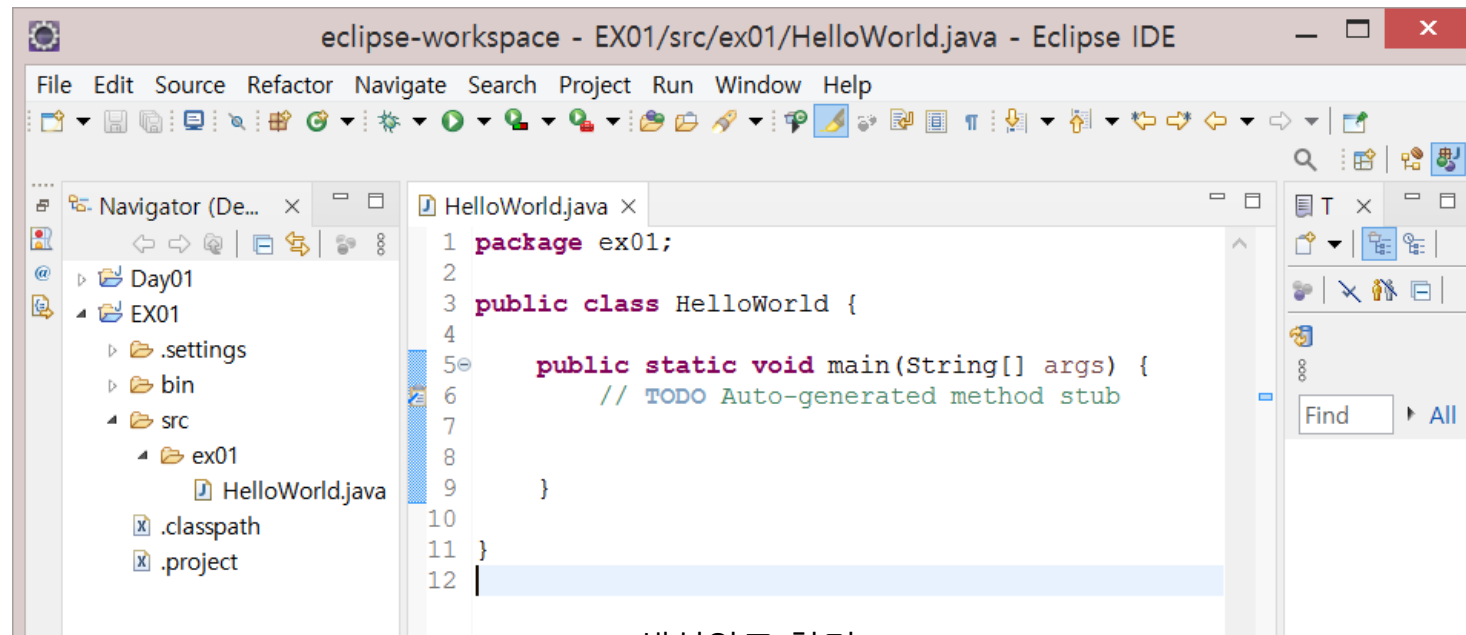
- 패키지는 자바 소스 파일을 비슷한 성격끼리 모아 두는 일종의 폴더 개념
- Class 생성
 - ▶ 프로젝트 선택 후 마우스 오른쪽 버튼 -> [New] -> [Class]
 - ▶ Package: 패키지 이름, ex01로 설정
 - ▶ Name: Class 이름, HelloWorld로 설정
 - ▶ public static void main 항목 체크



프로젝트/패키지 생성

■ Class/패키지 생성

- 자바 프로그램은 여러 개의 Class로 구성
- 그 중 하나의 Class에서 main() 메서드를 사용하여 프로그램을 실행
 - ▶ main() 메서드는 자바 프로그램을 실행
- 대부분의 Class에는 main() 메서드가 없으나 기본 문법을 배우는 당분간은 main()에서 주 프로그램을 작성



<생성완료 화면>

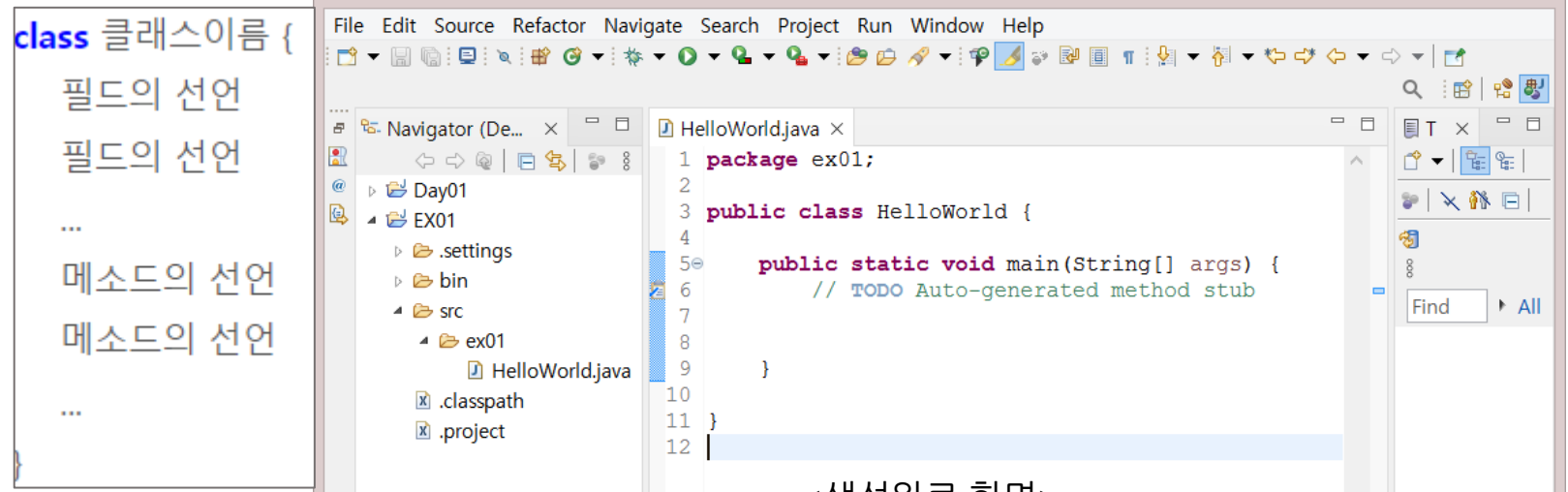
기본 문법

■ 선언

- 자바 프로그램을 작성할 때는 명령어를 통해 선언해주어야 함
- Class 선언시에는 이름 뒤 중괄호, 메서드 선언시에는 이름 뒤 괄호, 중괄호

■ Class/Method

- 자바 프로그램은 한 개 이상의 클래스(class)로 구성
- 클래스는 한 개 이상의 필드(field)나 메서드(method)로 구성
- Class와 Method의 선언은 다음의 예시와 같이 선언
- 중괄호 {}를 사용하여 구분



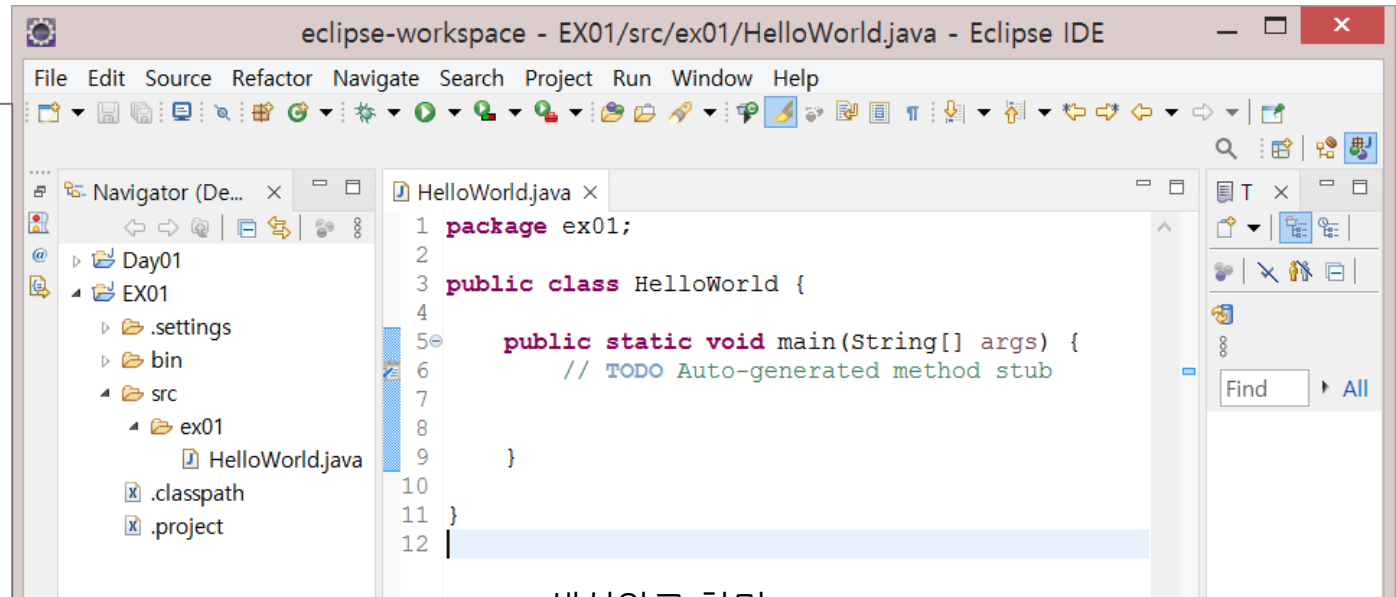
<생성완료 화면>

기본 문법

■ main() Method

- 자바 프로그램이 실행되면 가장 먼저 main() 메서드를 찾아 main()안의 모든 명령문을 순서대로 실행
- 자바에서 main() 메서드는 다음과 같이 선언
 - ▶ `public static void main(String[] args){`
...
`}`
 - ▶ 반드시 `public static void`로 선언 되어야 함

```
class 클래스이름 {  
    필드의 선언  
    필드의 선언  
    ...  
    메소드의 선언  
    메소드의 선언  
    ...  
}
```



<생성완료 화면>

기본 문법

▣ 명령문 (statement)

- 자바 프로그램의 동작을 명시 = 컴퓨터에게 어떤 동작을 하도록 명령
- 모든 명령문은 세미콜론 ; 으로 끝나야 함

▣ 주석 (comment)

- 실제로 컴퓨터에게 동작을 하도록 명령하는 것이 아닌 설명이나 디버깅을 위한 메모
- 컴퓨터의 동작에 영향을 주지 않음
- 예시) `// 한 줄에 대한 주석 처리`

```
/*  
여러 줄에 대한  
주석처리  
*/
```

기본 문법

▣ HelloWorld Class

- "Hello World!!"를 출력하는 프로그램

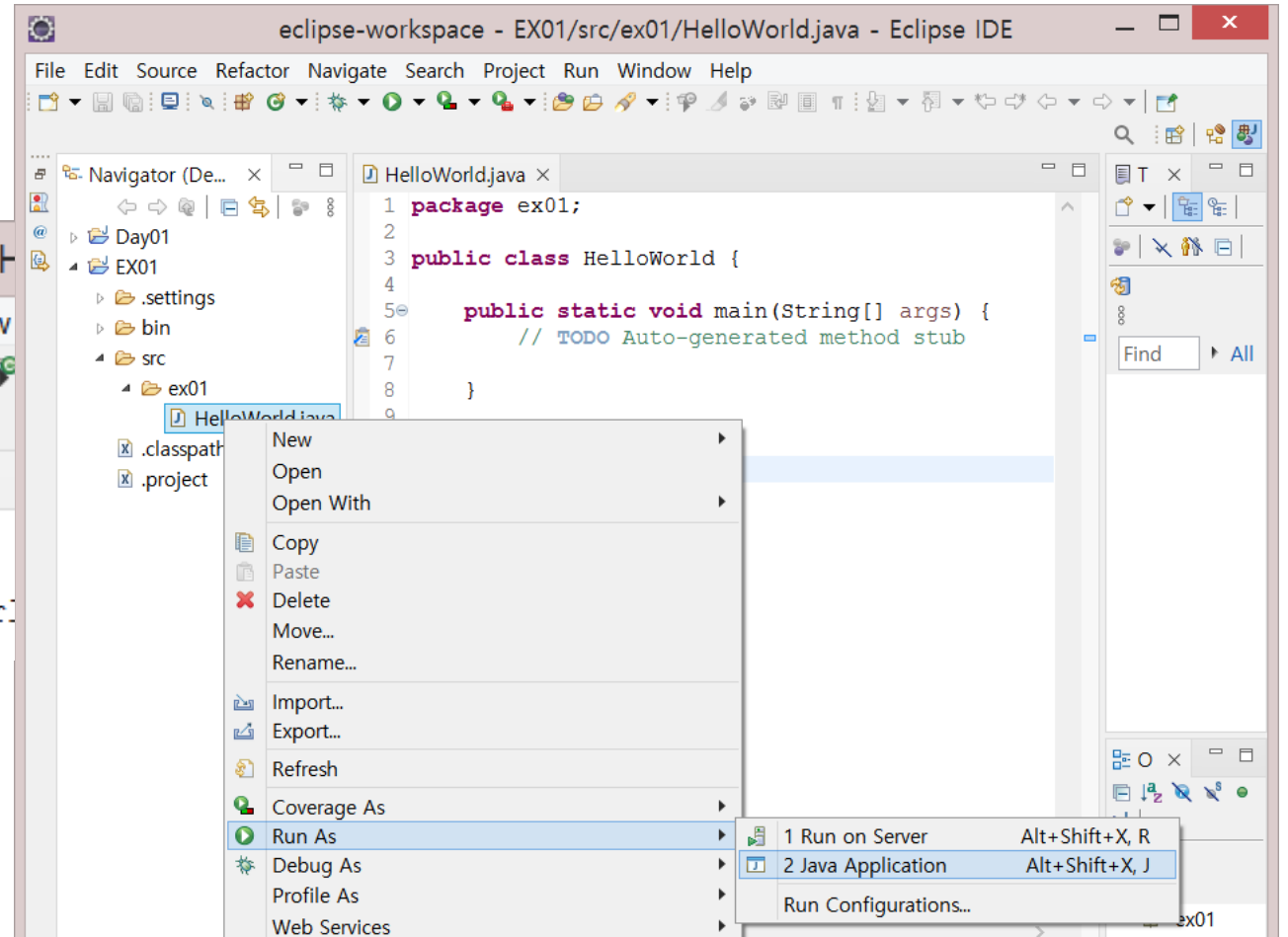
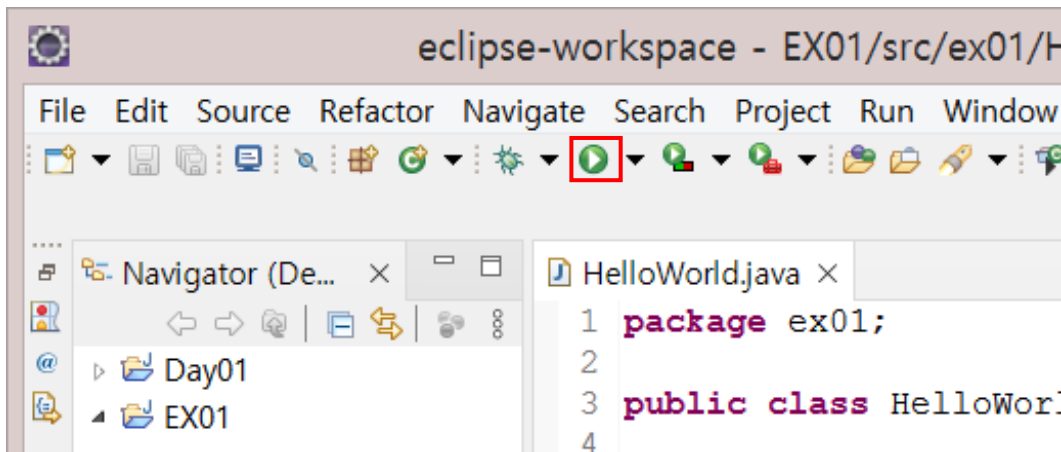
▣ 출력 명령어

- `System.out.println();`
 - ▶ 괄호 안의 내용을 출력하고 줄바꿈
- `System.out.print();`
 - ▶ 괄호 안의 내용을 출력
 - ▶ 괄호 안에 `"/n"`을 넣으면 `println`과 같은 결과를 줄 수 있음

기본 문법

▣ 프로젝트 실행방법

1. Class 선택 마우스 오른쪽 버튼 클릭 -> [Run As] -> [Java Application]
 2. 상단에 실행 아이콘 클릭
 3. 단축키 ctrl+F11
- 결과는 콘솔창으로 출력



출력 명령어 연습

▣ 다음의 여러 타입의 문자들을 출력해봅시다.

- 정수, 실수, 문자열

```
System.out.println(7);
System.out.println(3.15);
System.out.println("3+5=" + 8);
System.out.println(3.15 + "는 실수입니다.");
System.out.println("3+5" + "의 결과는 8입니다.");
System.out.println(3+5);
```

▣ 연습문제 (Practice1) 출력 및 주석 연습

- Practice1 Class를 생성하고 main() 함수에 다음을 구현하시오.
- 숫자 16을 5회 출력하는 프로그램을 작성하시오. 다섯 문장이 모두 같지 않아야 함.(약간만 달라도 됨.)
- 상단에 프로그램 작성 날짜와 작성자의 학번이름을 주석으로 표시하시오.

출력 명령어 연습

▣ 연습문제 (Practice2) 여러 형태의 출력 연습

- Practice2 Class를 생성하고 main() 함수에 다음을 구현하시오.
- 다음 문장들의 출력 결과를 예측하고 확인해보세요.

```
System.out.println("1" + "2")
```

```
System.out.println(true)
```

```
System.out.println("A" + "B")
```

```
System.out.println(1 + 2)
```

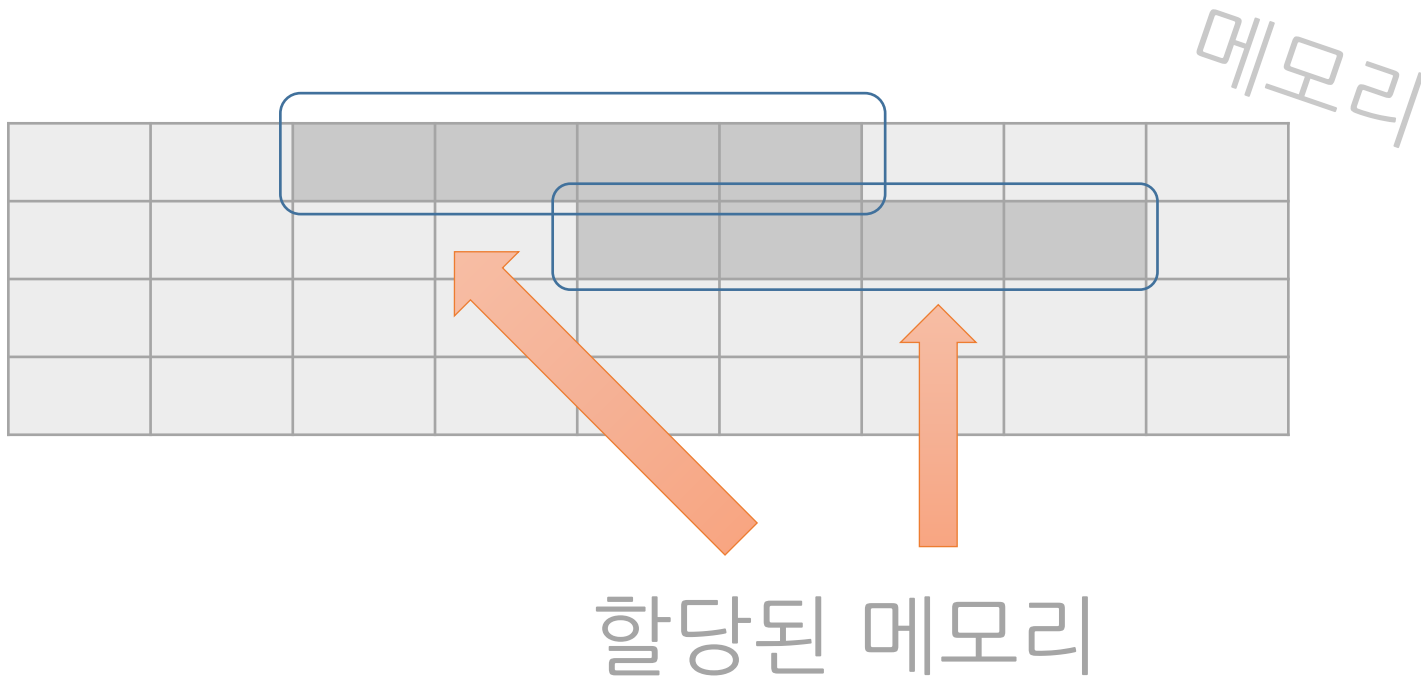
```
System.out.println("1" + "2")
```

```
System.out.println('J' + "ava")
```

변수

▣ 변수란?

- 변수(variable)란 데이터(data)를 저장하기 위해 프로그램에 의해 이름을 할당 받은 메모리 공간을 의미
- 변수 만들기
 - ▶ 변수는 선언문을 통해 만들 수 있음
 - ▶ 필요한 크기의 자료형과 이름을 적어 만듦
- 변수의 데이터 종류에 따라 선언 방식이 달라짐



`int num;`

정수를 저장할
메모리 공간

할당된 메모리
공간의 이름

변수

▣ 변수의 이름 생성 규칙

1. 영문자(대소문자), 숫자, 언더스코어(_), 달러(\$)로만 구성할 수 있습니다.
 2. 숫자로 시작할 수 없습니다.
 3. 변수의 이름 사이에는 공백을 포함할 수 없습니다.
 4. 자바에서 미리 정의된 키워드(keyword)는 사용할 수 없습니다.
- 예시) num1, f1_num, Num, ...

▣ 정수 변수 예시

- Int를 사용하여 선언할 수 있음
 - ▶ 예시) Int num;

▣ 문자열 변수 예시

- String을 사용하여 선언할 수 있음
 - ▶ 예시) String str;

변수 (이후에 더 자세히 배움)

▣ 자료형

- int와 같이 변수에 저장할 데이터의 종류를 알려주는 것을 자료형(data type)이라고 함

▣ 기본형 변수

- 기본형(primitive type) 변수는 실제 연산에 사용되는 변수
- 자바에서는 다음과 같이 8가지 종류의 기본형 변수를 제공

종류/크기	1byte	2byte	4byte	8byte
논리형	boolean			
문자형		char		
정수형	byte	short	int	long
실수형			float	double

변수 생성 연습

▣ 정수형 변수의 생성을 연습해봅시다.

- `Int num = 10;`

▣ 연습문제 (Practice3) 변수의 활용

- Practice3 Class를 생성하고 main()함수에 다음을 구현하시오.
- 10이하의 2의 배수들을 변수로 생성하시오.
- 생성된 변수를 사용하여 다음을 출력하시오.

10이하의 2의 배수는 2, 4, 6, 8, 10입니다.