



그림으로 설명하고 실생활 예제로 실습하는

자버를 부탁해

학습 로드맵

- 01 자바 시작하기
- 02 변수, 타입, 연산자

- ② 메소드와 조건문
- 03 메소드
- 04 조건문

Part 1 기본 문법 Part 2 객체 문법

- 4 OOP 기초
- 07 객체 지향 프로그래밍
- 08 객체 지향 프로그래밍의 활용
- 09 자바 API

- **6** OOP 심화
- 10 상속
- 11 인터페이스
- 12 예외 처리와 파일 입출력

- ❸ 반복문과 배열
- 05 반복문
- 06 배열

6 프로젝트

13 실전 프로젝트: 영화 예매 관리 프로그램

CH01 자바 시작하기

01 자바 프로그래밍이란

02 자바의 특징

03 자바 개발 환경 만들기

04 자바 프로그램의 기본 작성 흐름

자바 프로그래밍의 과정을 이해한다.

자바의 언어적 특징을 학습한다.

자바 **개발 환경**을 구축한다.

첫 프로그램을 작성한다.

CH01 자바 시작하기

01 자바 프로그래밍이란

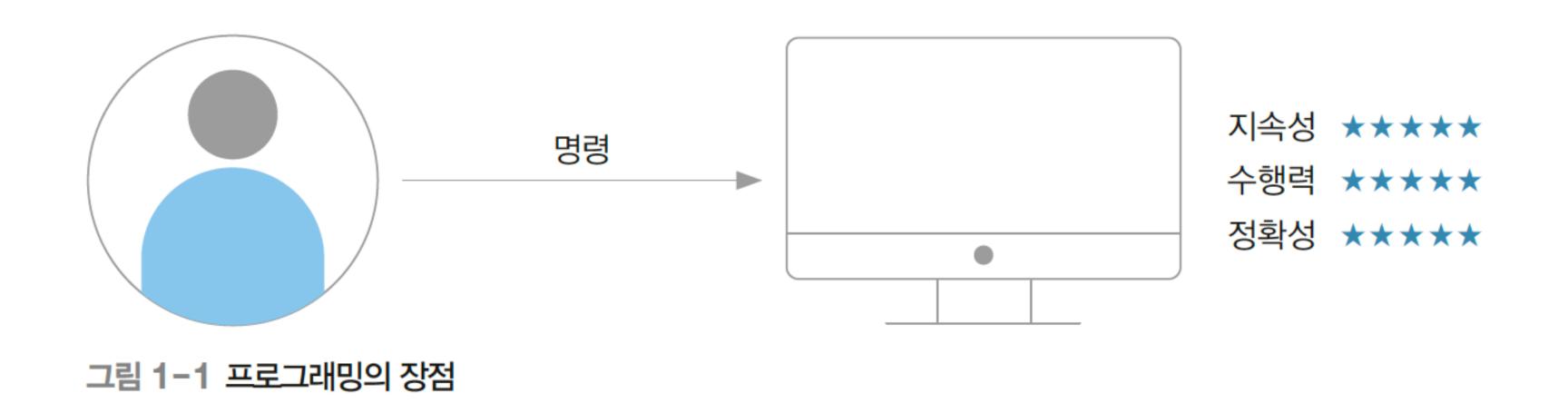
02 자바의 특징

03 자바 개발 환경 만들기

04 자바 프로그램의 기본 작성 흐름

1. 프로그래밍이란

프로그램을 만드는 것 코드를 통해 컴퓨터에게 명령하는 행위 컴퓨터의 장점을 활용한 기술





TIP 코드(code)란 컴퓨터가 수행할 내용을 특정 언어로 작성한 것

Ⅱ. 자바의 역할

사람의 명령을 컴퓨터가 이해할 수 있게 함

- 자연어: 사람이 사용하는 언어(안녕하세요, Hello, 你好吗, ...)
- **기계어**: 0과 1만으로 구성된 이진 언어(0100, 1101, ...)

자바는 프로그래밍 언어로, 코드를 통해 컴퓨터에 명령할 수 있는 언어

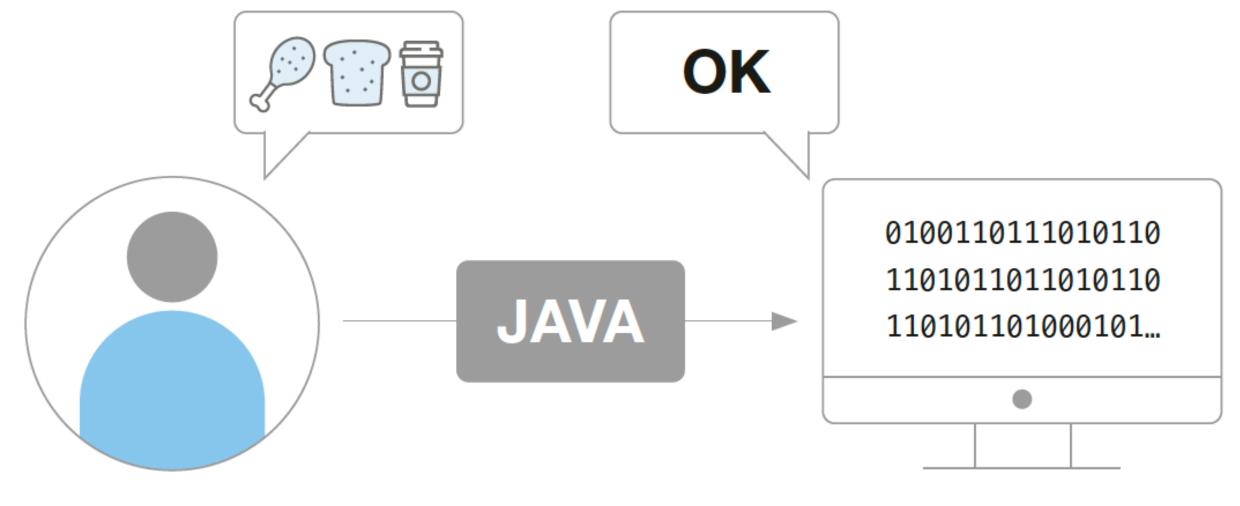


그림 1-3 프로그래밍 언어 자바

Ⅲ. 자바 프로그래밍의 과정

코드 작성 - 컴퓨터에게 시킬 일을 자바 언어로 작성

컴파일링 - 자바 코드가 컴파일러를 통해 바이트코드로 번역

인터프리팅 - 바이트코드가 자바 가상 기계(JVM)에 의해 기계어로 통역

코드 실행 - 컴퓨터가 기계어를 수행

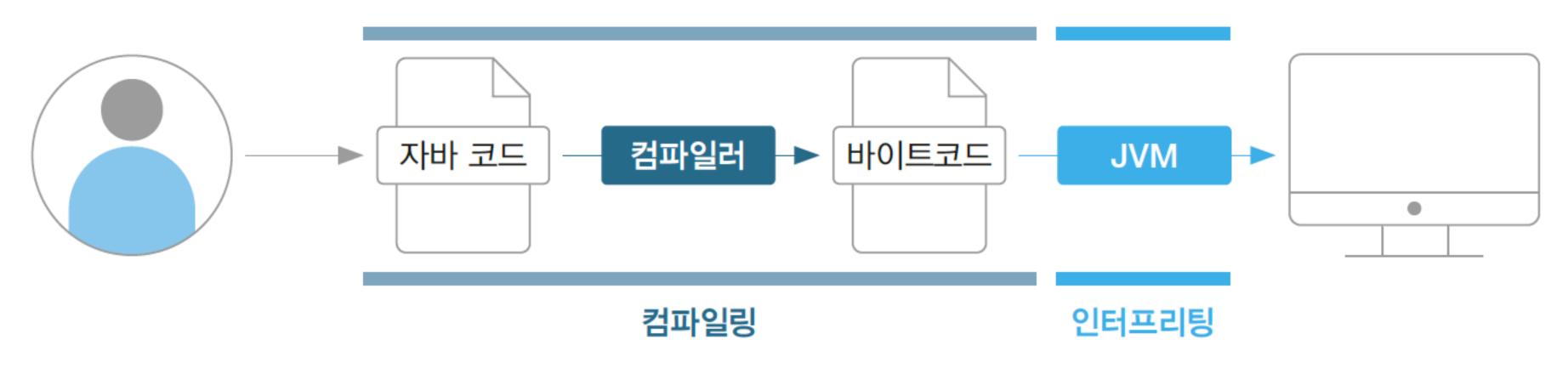


그림 1-4 자바 프로그래밍의 과정: 컴파일링과 인터프리팅

Quiz

01 다음 빈칸에 알맞은 용어를 채우시오.

- _____란 컴퓨터가 수행할 내용을 특정 언어로 작성한 것이다.
- ____는 자바 코드가 컴파일러를 통해 번역된 것이다. 이는 ___을 통해 기계어로 인터프리팅된다.

CH01 자바 시작하기

01 자바 프로그래밍이란

02 자바의 특징

03 자바 개발 환경 만들기

04 자바 프로그램의 기본 작성 흐름

1. 인기 프로그래밍 언어

자바는 국내외 수많은 개발자가 많이 사용하는, 인기 프로그래밍 언어 국내의 경우, 정부와 대기업 주도의 전자정부 표준프레임워크에서 자바가 프로그래밍 언어로 채택

TIOBE 지수에 따르면, 자바는 2021년 9월 기준, 11.12%의 인지도로 상위 랭크

Sep 2021	Sep 2020	Change	Program	nming Language	Ratings	Change
1	1		9	С	11.83%	-4.12%
2	3	^		Python	11.67%	+1.20%
3	2	~	(K)	Java	11.12%	-2.37%
4	4		9	C++	7.13%	+0.01%
5	5		9	C#	5.78%	+1.20%

Ⅱ. 멀티 플랫폼 언어

다양한 환경에서 실행되는 프로그래밍 언어

운영 환경에 따른 JVM을 제공함으로써 높은 이식성(portability)을 가짐

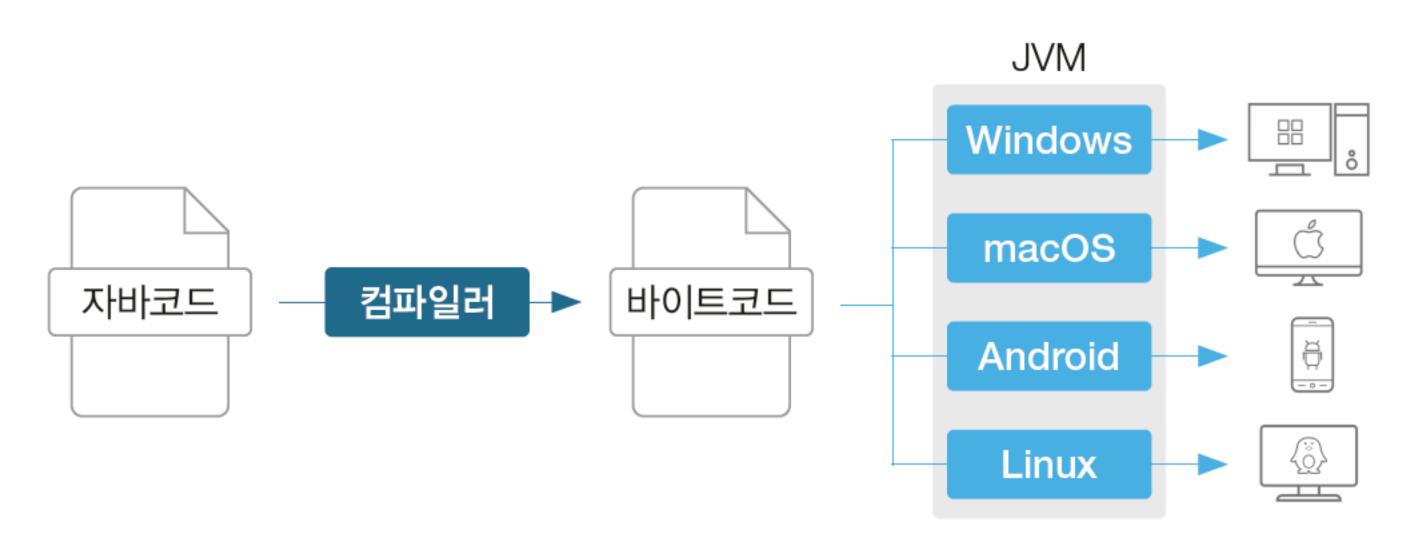


그림 1-7 자바의 장점인 이식성



TP 이식성이란, 다양한 환경에서도 잘 동작하는 프로그램의 특성입니다.

Ⅲ. 객체 지향 언어

객체를 조합하여 전체 프로그램을 만드는 기법(OOP)이 가능한 언어 객체 지향은 간단히 조립식 프로그래밍이 가능한 언어라고 할 수 있음 부품을 조립하듯, 자바는 객체를 조합하여 전체 프로그램을 만들 수 있음

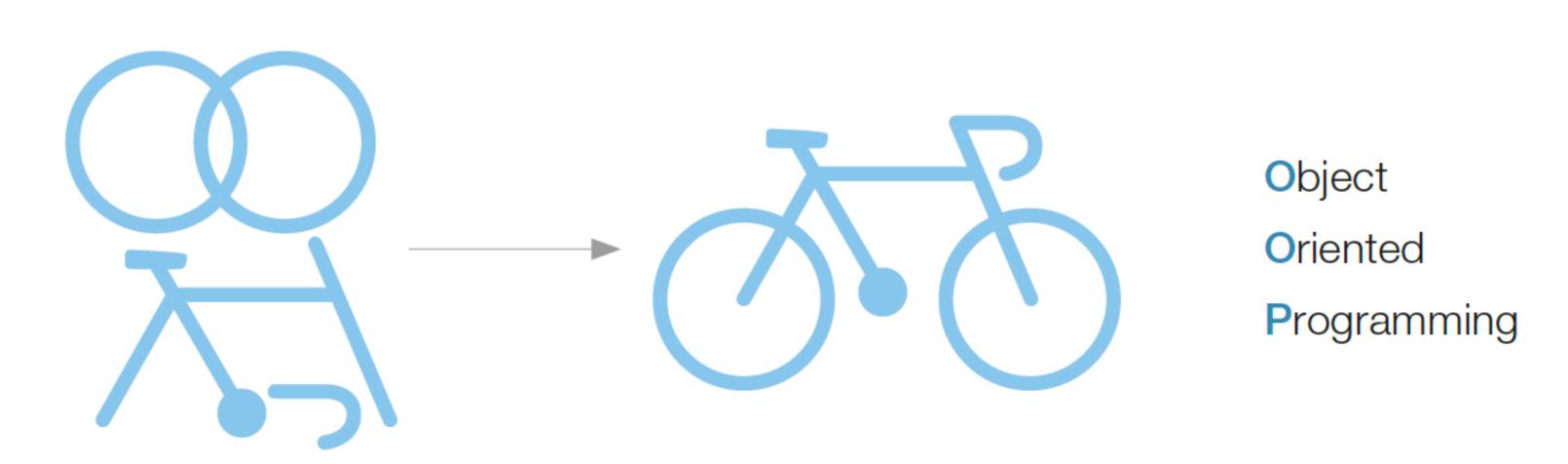


그림1-8 자전거로 본 객체 지향 프로그래밍

Quiz

02 다음 빈칸에 알맞은 용어를 채우시오.

- '____이란 프로그램이 다양한 환경에서도 잘 동작하는 특성이다
- ____이란 객체를 조합하여 전체 프로그램을 만드는 기법으로 OOP라 불린다

CH01 자바 시작하기

01 자바 프로그래밍이란

02 자바의 특징

03 자바 개발 환경 만들기

04 자바 프로그램의 기본 작성 흐름

I. JDK 설치

자바 프로그래밍을 시작하려면 자바 개발 도구(JDK)가 필요 학습을 위해 기본 에디션인 Java SE를 설치

- Java SE: 표준 개발을 위한 JDK
- Java EE: 웹 서버 개발을 위한 분산 환경용 JDK
- Java ME: 임베디드 시스템 개발을 위한 JDK

JDK는 크게 무료 오픈 소스로 제공되는 OpenJDK 계열과 상업용 OracleJDK 계열로 구분(두 계열 모두 학습용으로 무료 사용이 가능)

여기서 잠깐! JDK의 구조

JDK는 프로그래밍을 위한 다양한 개발 도구와 자바 실행 환경(JRE)을 포함 JRE는 다시 자바 프로그램 실행에 필요한 라이브러리와 JVM을 내포 개발이 목적이 아닌 자바 프로그램 실행만을 원하는 경우, JRE만 따로 설치할수도 있음

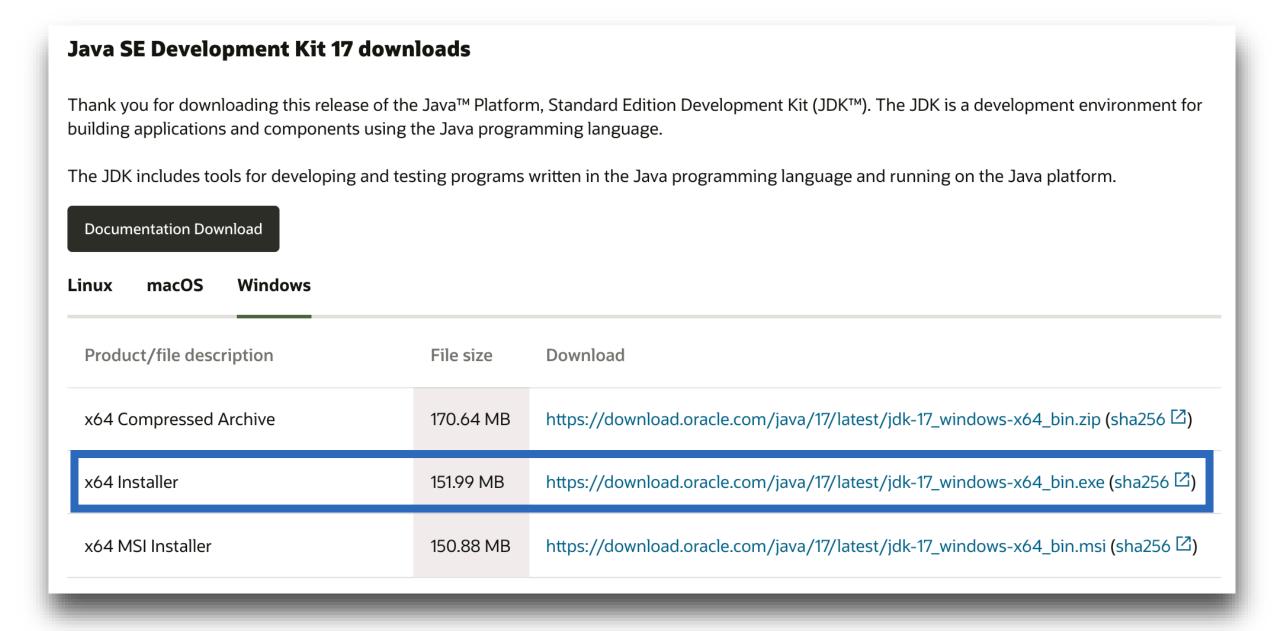


그림 1-9 JDK의 구조

I . JDK 설치

JDK 다운로드

- https://www.oracle.com/java/technologies/downloads에 접속
- Java SE Development Kit 17 설치 파일 다운로드
- 여기서는 64비트 윈도우를 기준으로 진행하므로, [x64 Installer]를 선택



I . JDK 설치

JDK 설치

- 다운로드가 완료되었다면 설치 파일을 실행(설정값은 기본값 그대로 진행)

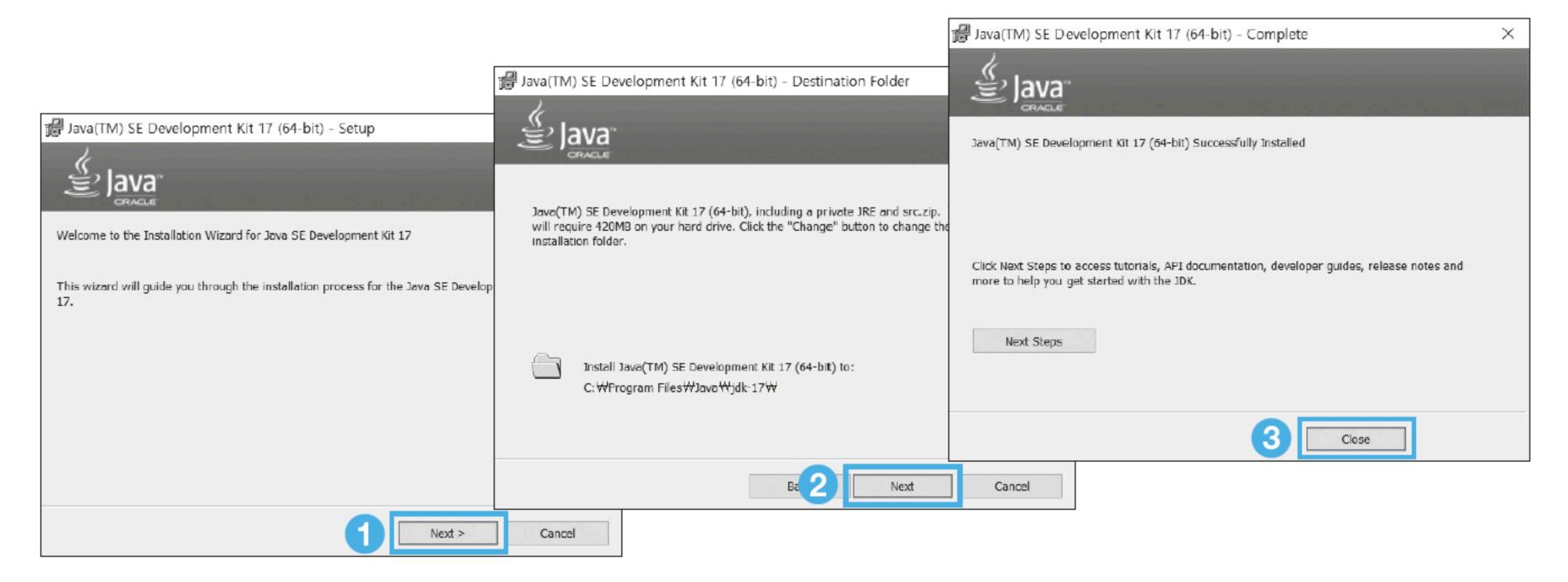


그림 1-11 JDK 설치 진행과정

기본 설정으로 JDK가 설치되었다면,

- C:₩Program Files₩Java₩jdk-17 폴더가 생성

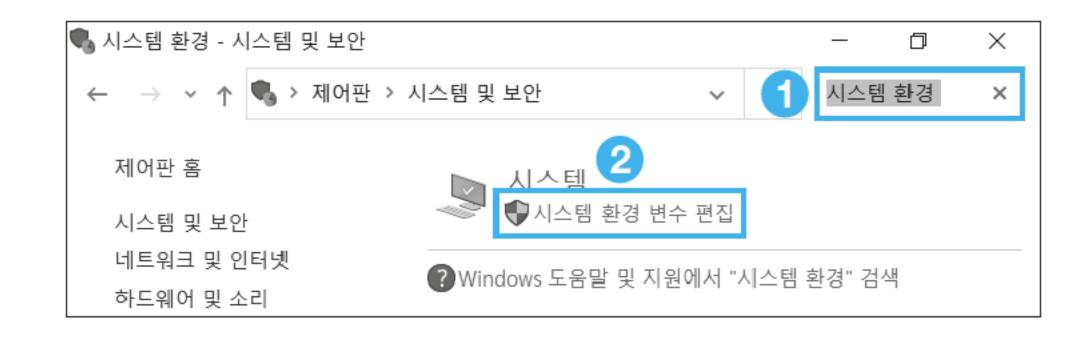


TIP 다른 버전의 JDK를 설치했다면, C:₩Program Files₩Java₩jdk-[버전번호] 폴더가 생성

설치된 폴더의 컴파일러와 JVM을 사용하기 위해, 환경 변수 등록이 필요

환경 변수 창 열기

- ① 제어판에서 '시스템 환경'을 검색해
- ② [시스템 환경 변수 편집]을 클릭
- ③ 시스템 속성창이 열리면 [환경 변수]를 클릭



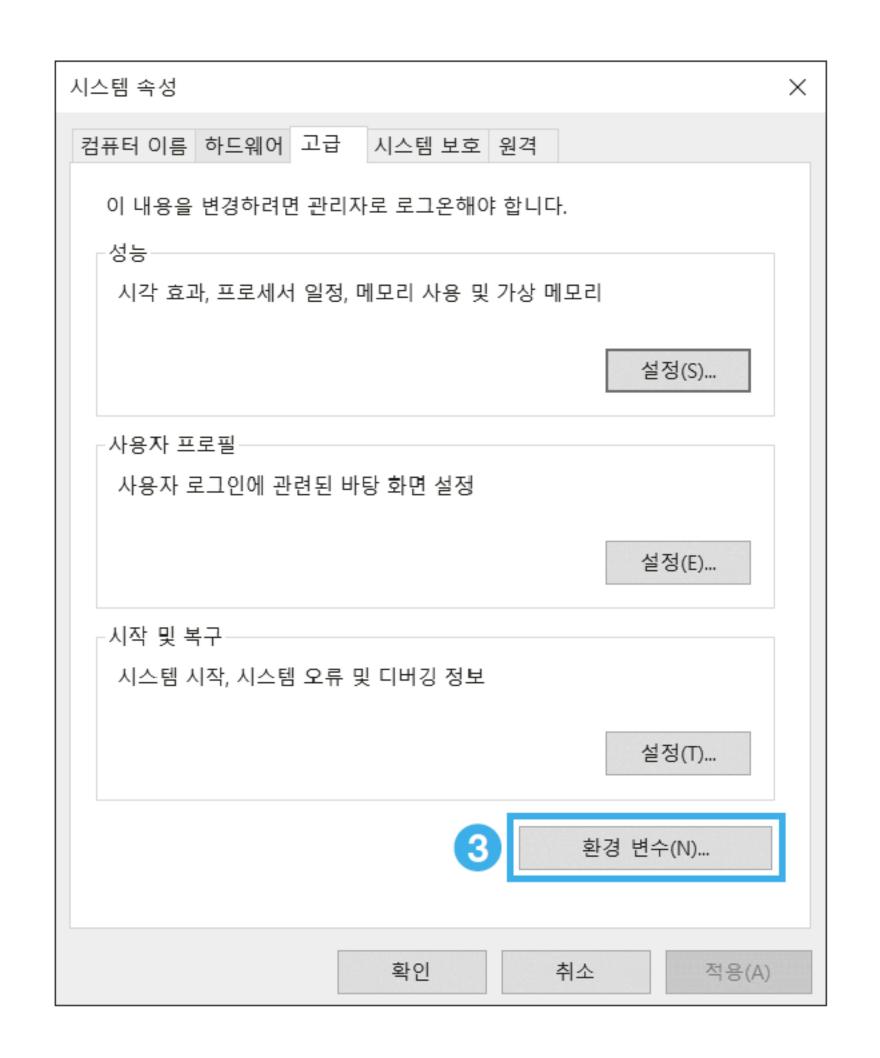


그림 1-12 환경 변수 창 열기

자바 환경 변수 설정

- ① 열려진 환경 변수 창에서 새 환경 변수를 생성
- ② 변수 이름은 JAVA_HOME, 변수 값은 앞서 설치한 JDK 폴더 위치를 입력
- ③ 입력이 끝나면 [확인] 버튼 클릭
- ④ 이어서 시스템 변수를 더블클릭하여 편집
- ⑤ Path에 '%JAVA_HOME% ₩bin'을 추가
- ⑥ 제일 위로 올려준 뒤
- ⑦ [확인]을 울러 설정 종료

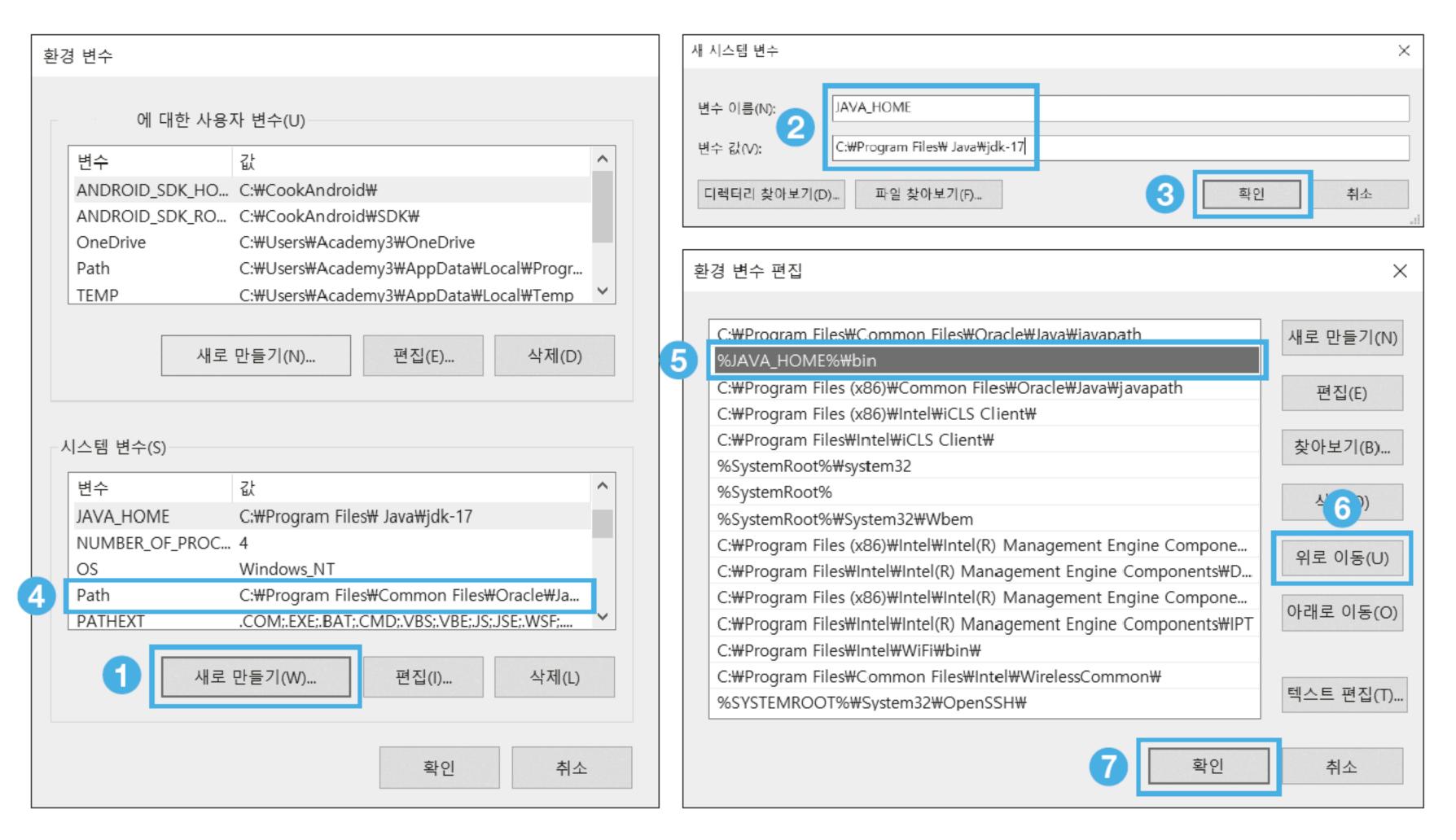
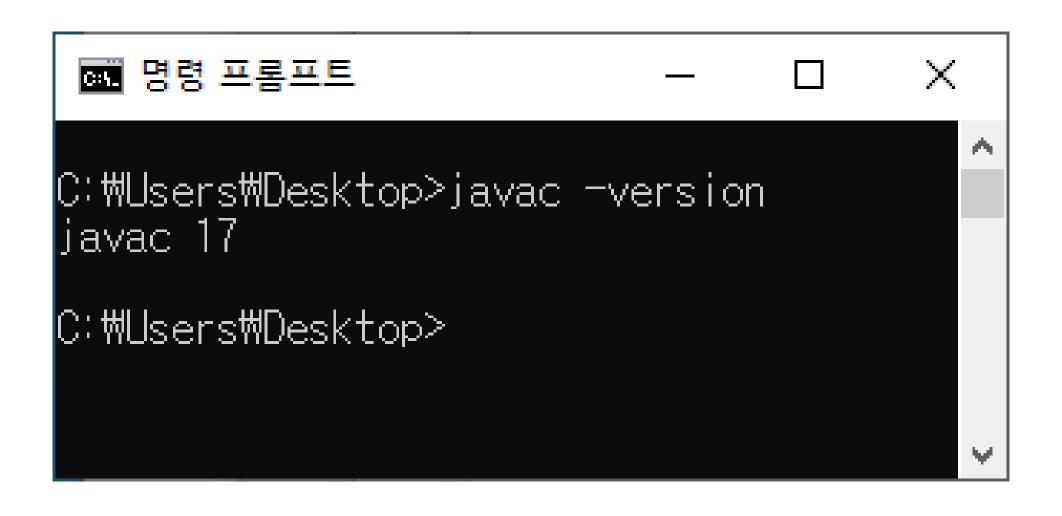


그림 1-13 환경 변수 설정

환경 변수 적용 확인

- 윈도우 검색 창에서 'cmd'를 입력해 새로운 명령 창을 생성
- "javac -version" 명령을 실행하여 앞서 설치한 JDK 버전이 출력되는지 확인



III. IDE 설치

JDK와 환경 변수를 설정했다면, 이어서 통합 개발 환경(IDE)를 설치

- IDE는 개발 효율을 높이는 프로그램으로, 자바 코드의 컴파일과 실행 및 구문 오류 검증 등 개발을 위한 편의 기능을 제공
- 자바의 대표적 IDE로는 이클립스(Eclipse)와 인텔리제이 IDEA(IntelliJ IDEA), 넷빈즈(NetBeans)가 있음

Ⅲ. IDE 설치

이클립스 접속 및 파일 다운로드

- https://www.eclipse.org에 접속하여 설치 파일 다운로드

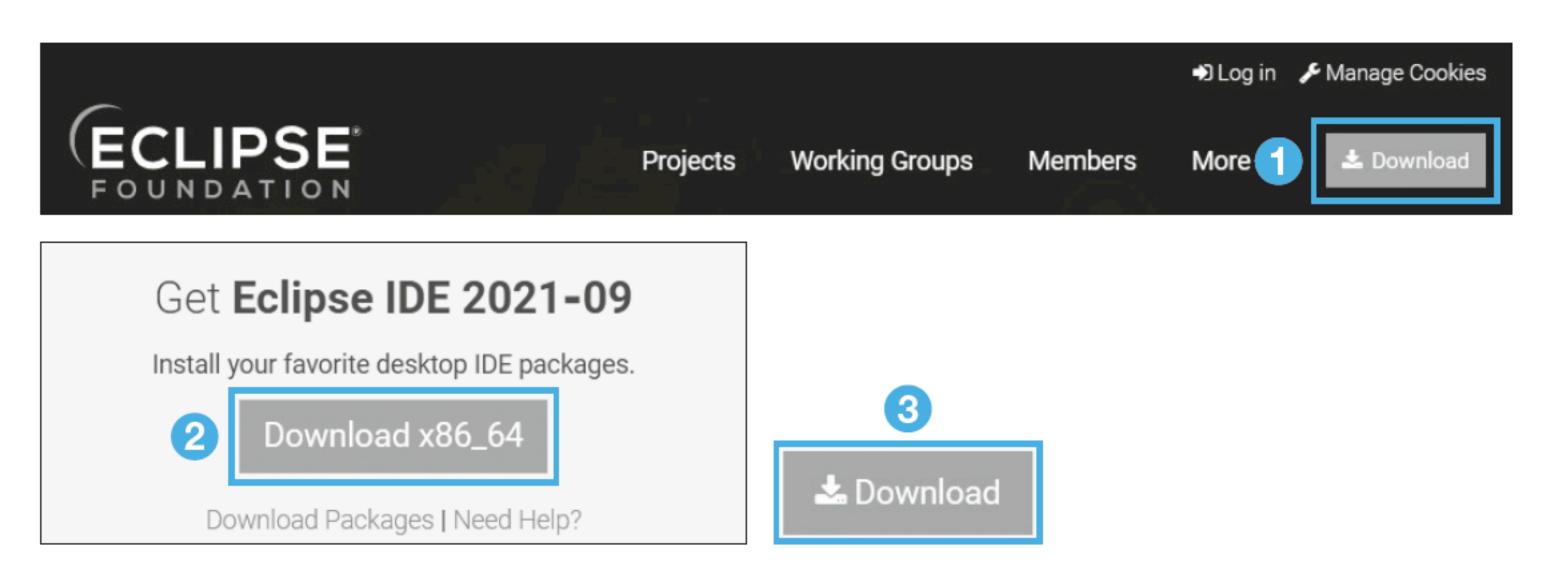
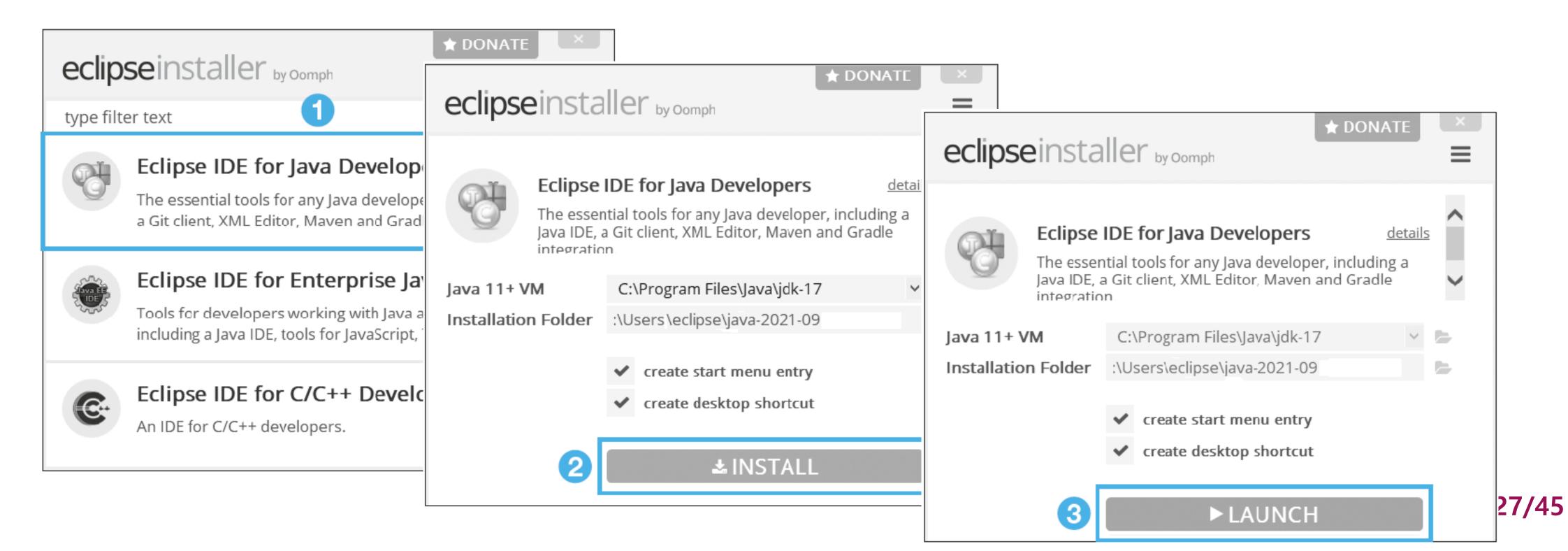


그림 1-15 이클립스 접속 및 파일 다운로드

Ⅲ. IDE 설치

이클립스 설치 진행 및 실행

- 다운로드가 완료되었다면, 설치할 이클립스 ① 버전 선택 → ② 설치 진행
 - → ③ 설치 완료 순서대로 설치 과정을 진행



Ⅲ. IDE 설치

이클립스 작업 공간 설정

- 이클립스가 실행되면, 작업 공간 설정 창이 열림
- 원하는 작업 공간 폴더를 지정해도 되지만, 기본값 그대로 진행

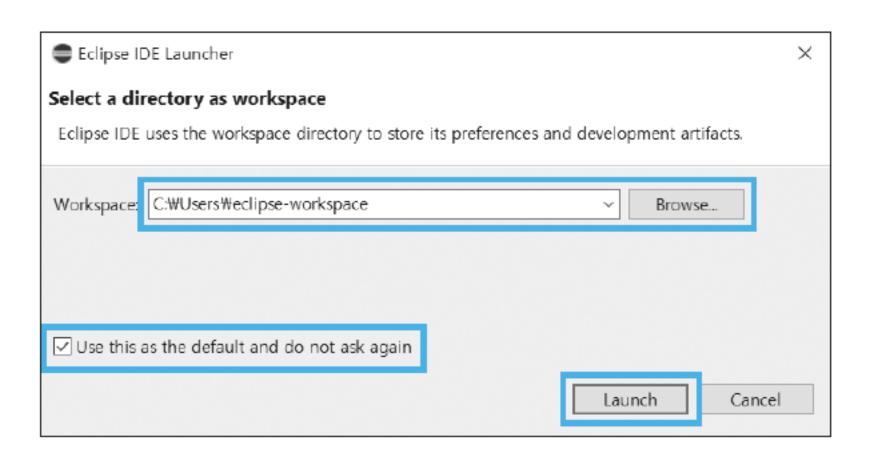


그림 1-17 이클립스 작업 공간 설정

III. IDE 설치

이클립스 동작 확인

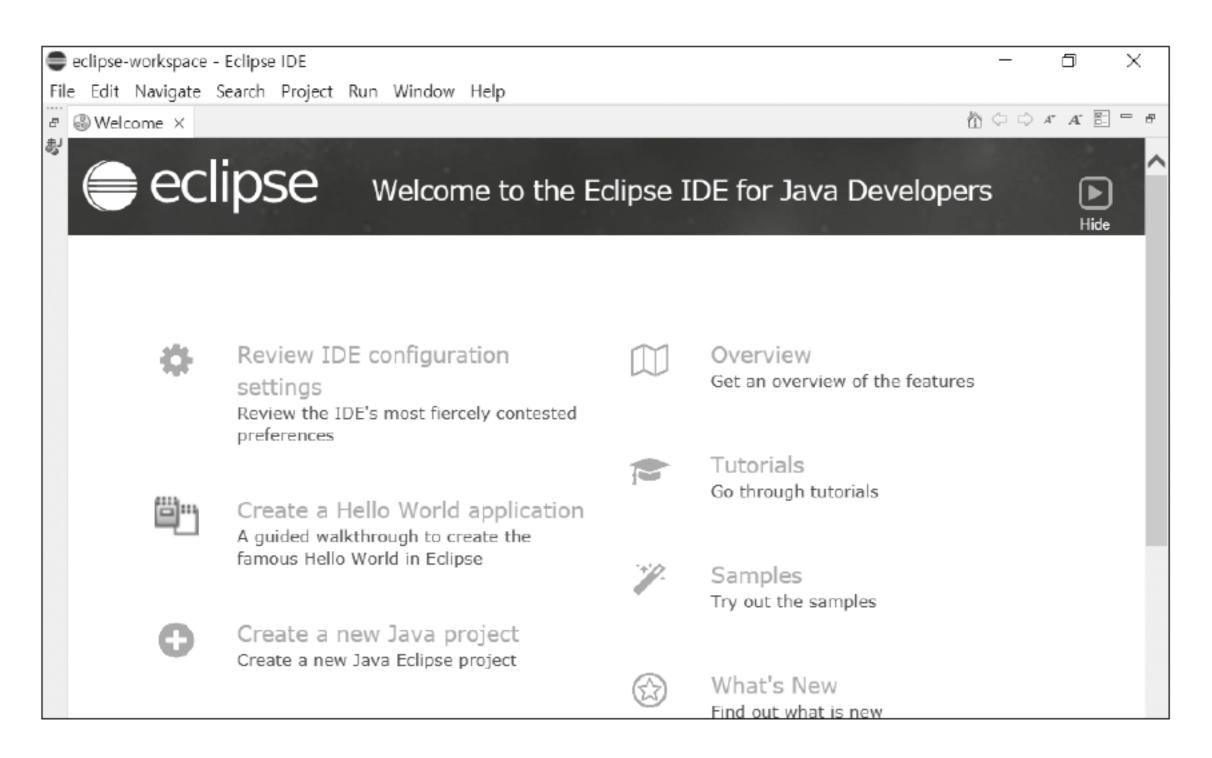


그림 1-18 이클립스 실행 화면

Ⅳ. 헬로 월드 출력하기

자바 개발 환경 동작 확인을 위한 "헬로 월드" 출력 프로그램 작성

자바 프로젝트 생성

- ① 왼쪽 상단 메뉴바의 [File]-[New]- [Java Project]를 클릭
- ② 프로젝트 이름을 입력
- ③ [Finish] 버튼 클릭
- ④ 모듈 생성은 하지 않는 것으로 설정
- ⑤ 프로젝트 생성

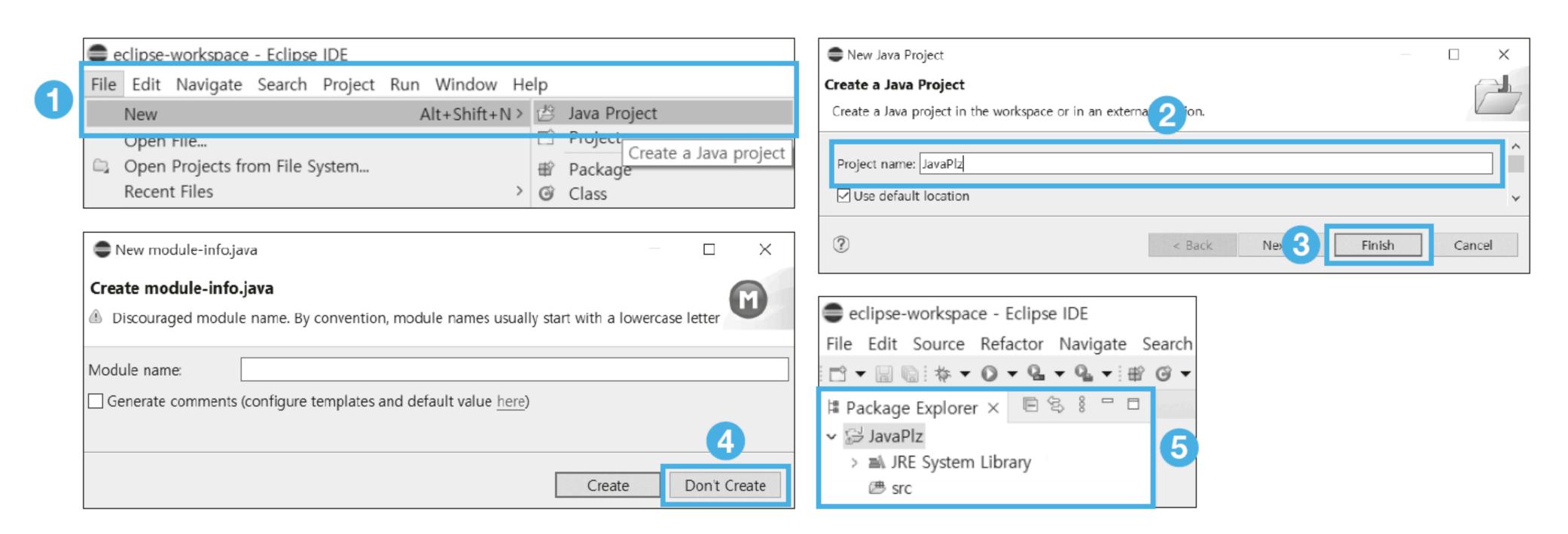


그림 1-20 자바 프로젝트 생성

Ⅳ. 헬로 월드 출력하기

패키지 생성

- ① [Package Explorer] 창에서
- ② src 폴더를 우클릭하여 [New]-[Package] 메뉴를 클릭
- ③ 패키지 이름을 설정하고 [Finish] 버튼을 클릭
- ④ 새 패키지 생성

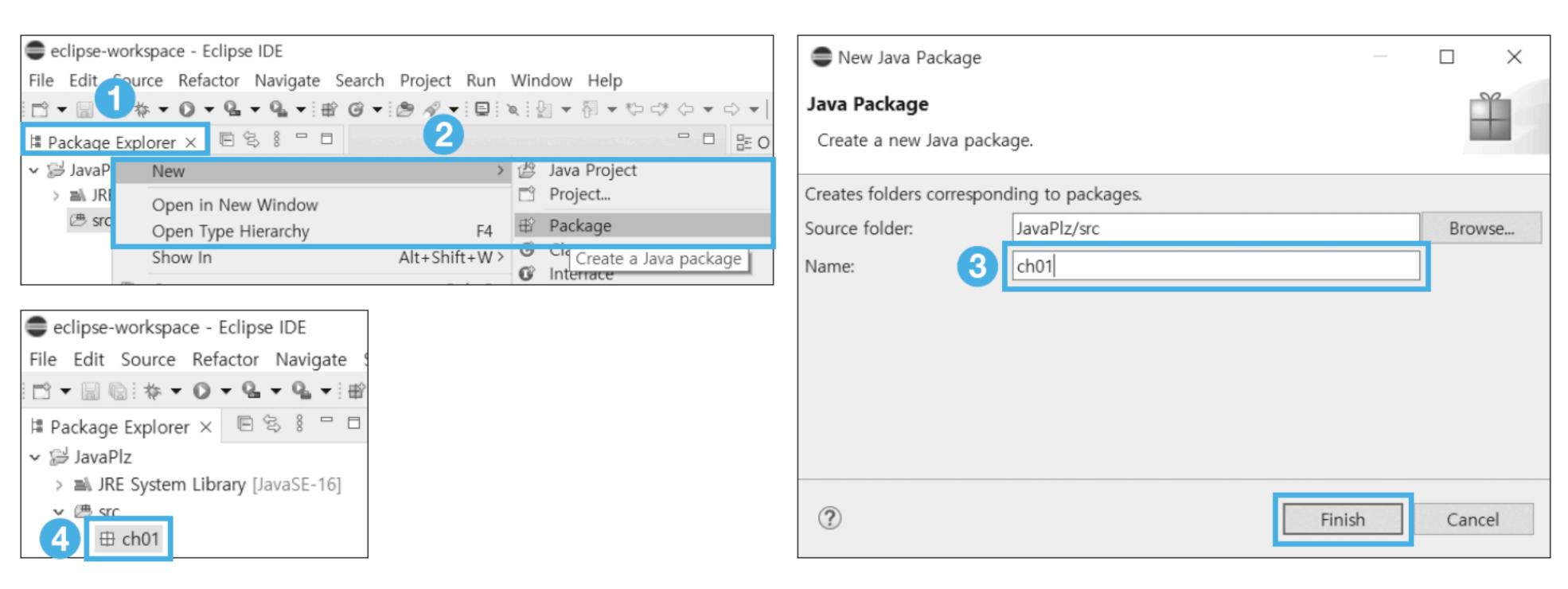
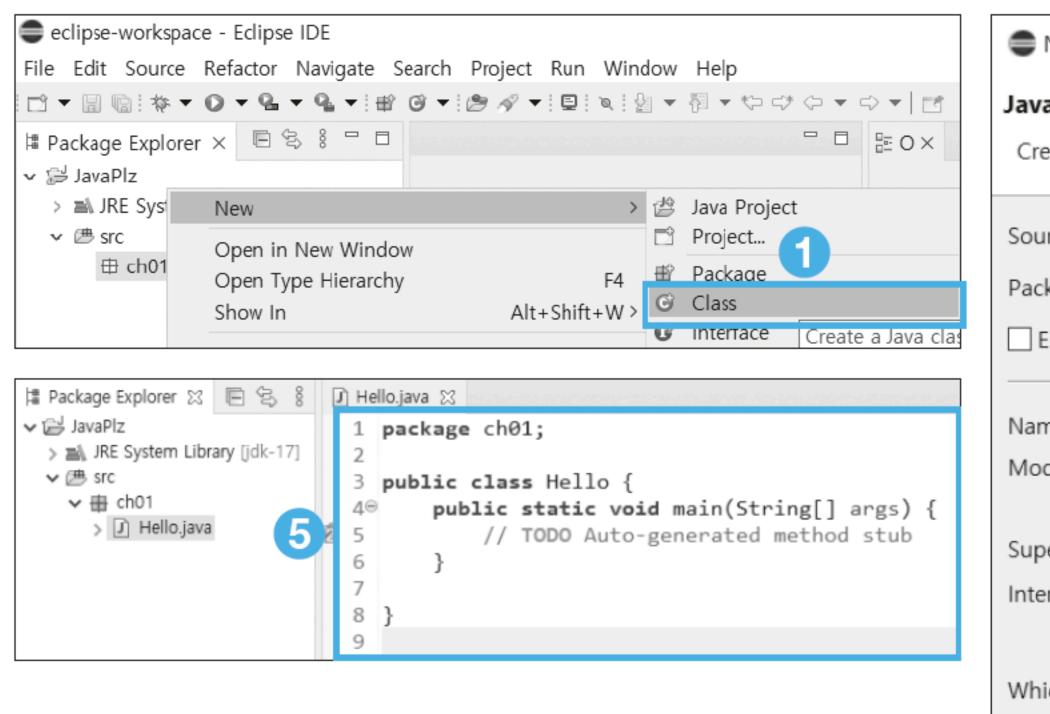


그림 1-21 패키지 생성

IV. 헬로 월드 출력하기

클래스 생성

- ① 앞서 생성한 패키지를 우클릭하여 클래스를 만들고
- ② 클래스 이름을 'Hello'로 작성
- ③ 추가로 main 메소드를 자동 생성을 체크
- ④ [Finish] 버튼을 클릭
- ⑤ Hello 클래스가 Hello.java 파일로 자동 생성



New Java Class							
Java Class Create a new Java							
Create a riew save	i Class.						
Source folder:	JavaPlz/src		Browse				
Package:	ch01		Browse				
Enclosing type:	Enclosing type:						
Name: Modifiers:	Hello public opackage oprivate prote abstract final static	ected					
Superclass:	java.lang.Object		Browse				
Interfaces:			Add				
white the state of							
Which method stubs would vou like to create? ✓ public static void main(String[] args)							
	Constructors from superclass						
	✓ Inherited abstract methods						
Do you want to add comments? (Configure templates and default value here)							
	Generate comments						
?	4 Finish		Cancel				

Ⅳ. 헬로 월드 출력하기

문자열 출력

- ① 출력을 위한 메소드 System.out.println()을 사용해 코드를 작성
- ② 파일을 저장
- ③ 이어서 실행 버튼을 클릭해 프로그램 수행
- ④ 헬로 월드!가 콘솔창에 출력

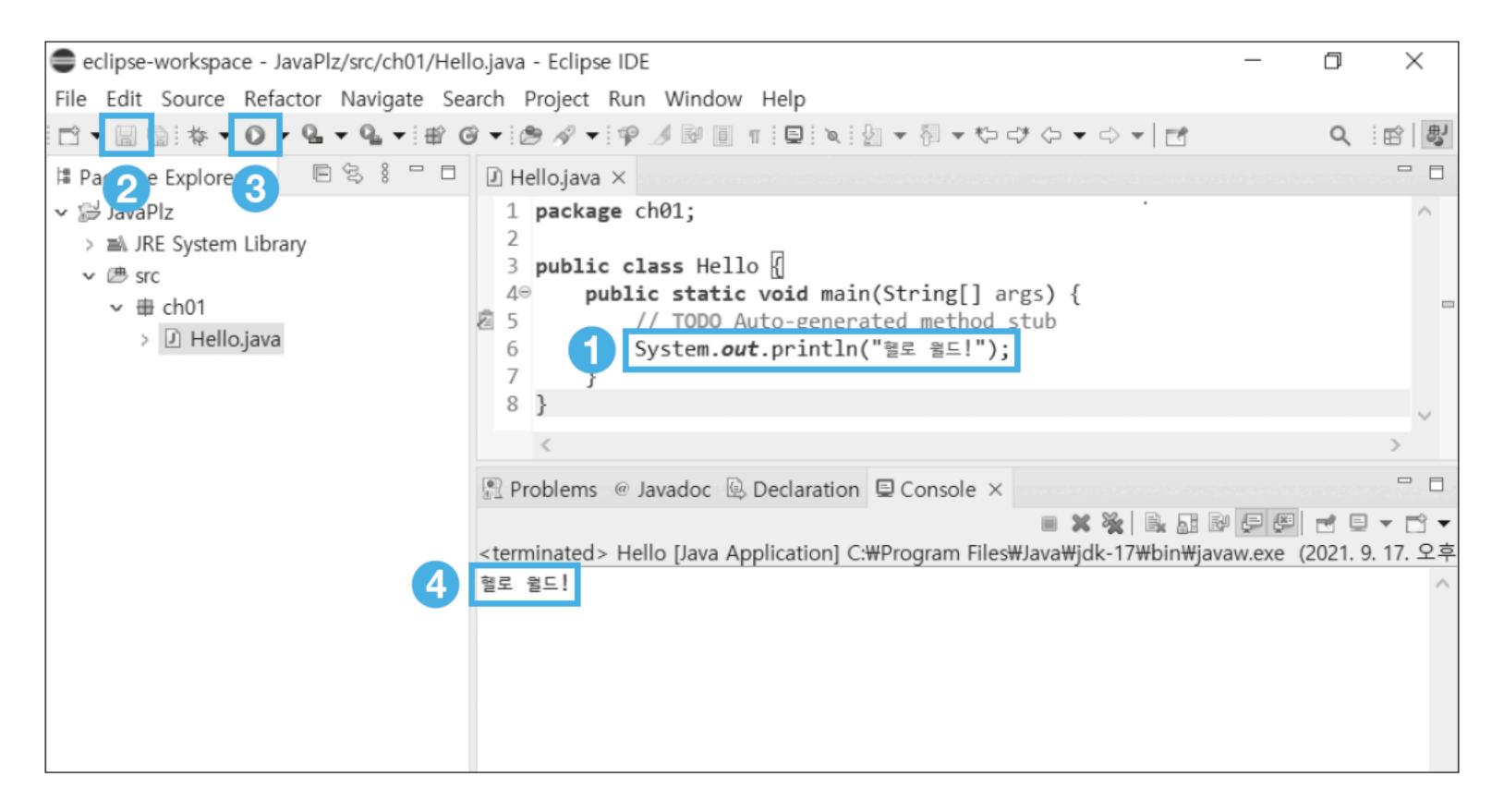


그림 1-23 문자열 출력

여기서 잠깐! 이클립스를 활용한 자바 프로그래밍

- 이클립스에서 자바 프로그램을 만들려면 먼저 프로젝트를 생성해야 함생성된 프로젝트는 폴더로 만들어지는데,
 - **자바 코드**는 src 폴더에서 작성해야 하며, 이를 컴파일한 **바이트코드**는 bin 폴더에서 자동으로 생성
 - bin 폴더는 이클립스에서 보이지 않지만, 윈도우 탐색기를 통해 확인할 수 있음

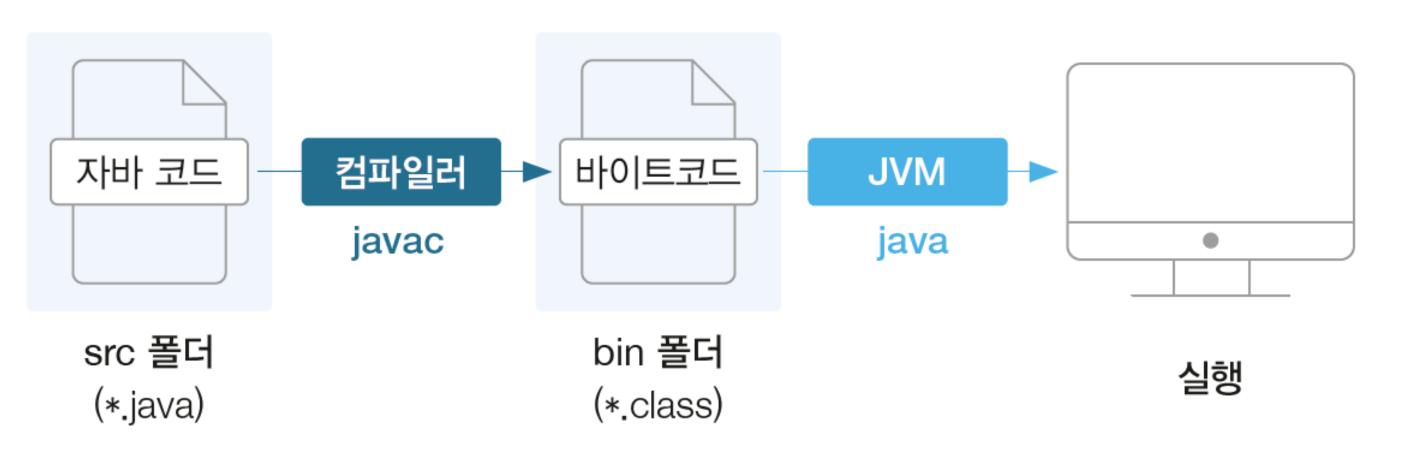


그림 1-24 이클립스를 활용한 자바 프로그래밍

CH01 자바 시작하기

01 자바 프로그래밍이란

02 자바의 특징

03 자바 개발 환경 만들기

04 자바 프로그램의 기본 작성 흐름

1. 클래스 작성

클래스(class)란 자바 프로그램의 최소 단위

자바 프로그램은 적어도 하나의 클래스를 가져야 함

클래스는 이름의 첫 글자를 대문자로 하는 것이 관례이며, 중괄호 내부에 여러 코드 를 담을 수 있음

이클립스에서 클래스를 만들면 클래스가 자동 생성되는데, 다음은 이름이 Hello인 클래스의 코드

```
    public class Hello { ... }
    클래스 내부 코드

    클래스를 정의하는 키워드
```



TIP 클래스 이름은 자바 코드의 파일명과 일치해야만 컴파일할 수 있음

Ⅱ. 메인 메소드 추가

메인 메소드(main method)란 자바 프로그램의 시작점으로,

- 중괄호 내부에 실행문을 담음
- 메인 메소드가 있어야 프로그램을 실행할 수 있음

```
public class Hello {
    public static void main(String[] args) {
        ··· 실행될 코드가 작성되는 위치
    }
}
```

TIP 메소드(method)란 특정 동작을 수행하는 구문('3장 메소드' 참조)

Ⅲ. 실행문 작성

- 실행문(statement)이란 프로그램 실행을 서술하는 코드
- System.out.println() 메소드는 화면에 문자열을 출력하는 실행문
 - 문자열이란 "헬로 월드!", "abcd"처럼 큰따옴표로 둘러싸인 문자
- 각 실행문은 세미콜론(;)을 기준으로 구분

```
헬로 월드! 문자열 출력
실습 1-1
                                                                 ch01/ex1_1/Hello.java
   public class Hello {-
                                                 ---클래스명
       public static void main(String[] args) {
02
           System.out.println("헬로 월드!"); ----출력할 문자열
03
04
05
 실행 결과
                                                        🗕 실습 코드
                                                        https://github.com/cloudstudying-
헬로 월드!
                                                        kr/JavaPlz/blob/master/src/part1/ch01/e
                                                       x1_1/Hello_iava
```

Ⅳ. 주석 달기

주석(comment)이란 코드 설명을 위한 문법으로, 주석 처리된 코드는 실행되지 않음

- **한 줄 주석**(line comment): 주석의 시작인 // 을 기준으로 해당 행의 끝까지를 주석 처리
- **범위 주석**(block comment): 주석의 시작과 끝을 **/*** 와 ***/**로 감싸 코드를 주석 처리

```
// 한줄 주석: 글자 앞쪽에 슬래시를 2회 사용

/* 블럭 주석(여러 줄 주석):
글자 앞뒤를
슬래시와 별표시로
감싼다. */
```

```
실습 1-2
           주석이 추가된 헬로 월드! 문자열 출력
                                                           ch01/ex1_2/Hello.java
01
   /*
02
    * 헬로 월드! 문자열 출력 프로그램
                                                   범위 주석
03
    * 파일명: Hello.java
04
    */
05
   // 1 클래스 작성
   public class Hello {
                                                ··········한 줄 주석
       // 2 메인 메소드 추가
80
       public static void main(String[] args) {
09
           // 3 문자열 출력
10
          System.out.println("헬로 월드!");
11
12
13 }
 실행 결과
                                                  □ 실습 코드
                                                  https://github.com/cloudstudying-
헬로 월드!
                                                  kr/JavaPlz/blob/master/src/part1/ch01/e
                                                  x1_2/Hello java
```

Thank You!

