

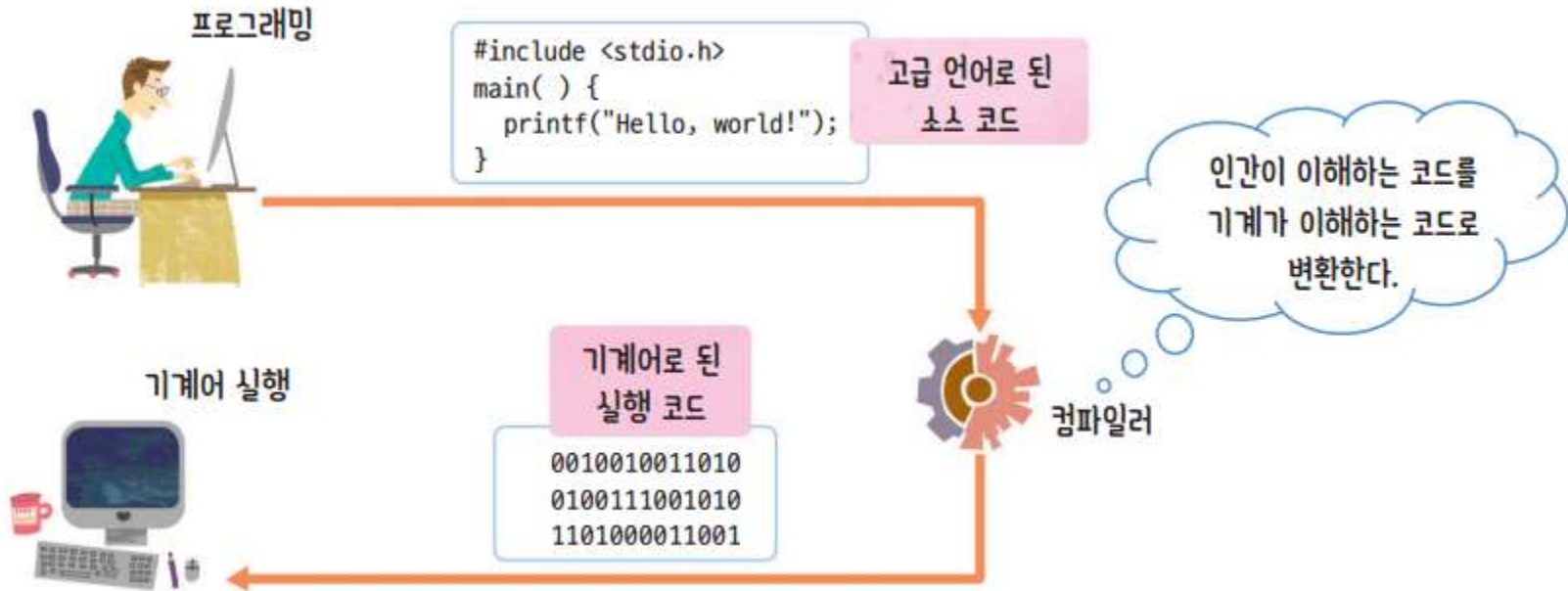
자바 시작하기

프로그래밍 언어

- 컴퓨터와 인간 사이를 연결할 수 있는 매개체이며 컴퓨터에 작업을 시키기 위한 도구
- 기계어인 저급 언어부터 자연어에 가까운 고급 언어까지 다양
- 보통 프로그래밍 언어라고 하면 고급 언어를 의미하는데 C, C++, 자바(Java) 등
- 고급 언어로 작성한 소스 코드는 컴파일러(Compiler)라는 시스템 소프트웨어를 이용해 먼저 기계어로 구성된 실행 코드로 변환

프로그래밍 언어

■ 고급 프로그래밍 언어의 실행 과정



자바 소개

■ 역사

- 선마이크로시스템즈(Sun Microsystems)의 제임스 고슬링(James Gosling)
- 자바의 전신은 1991년 그린 프로젝트(Green Project)로 탄생한 객체 지향 언어인 오크(Oak)
- 1993년 인터넷과 웹이 엄청난 속도로 발전
- 인터넷 환경에 적합하도록 오크를 새롭게 설계한 후 1995년 Java라는 이름으로 발표
- 1996년 자바 1.0을 발표
- 자바 1.2~1.4를 자바 2라고도 하는데, 자바 1.5부터는 1을 빼고 단순하게 자바 5로, 자바 1.6은 자바 6 등으로 지칭



자바 소개

■ 버전별 특징

자바 버전	특징
자바 7	타입 추론 기능, Switch 문에 문자열, try-catch문의 자원 자동 닫기
자바 8	람다식, 인터페이스의 디폴트 메서드, Optional 타입, DateTime
자바 9	모듈 시스템, JShell, 인터페이스의 private 메서드
자바 10	지역 변수 타입 추론 가능한 var
자바 11	첫 LTS 버전, ZGC, Nest 기반 접근 컨트롤, HTTP 클라이언트 표준화
자바 12	경미한 개선
자바 13	경미한 개선
자바 14	Switch 표현식

자바의 특징

단순하다

객체 지향 언어이다.

함수형 코딩을 지원한다.

플랫폼 독립적이다.

분산 처리를 지원한다.

견고하다.

안전하다.

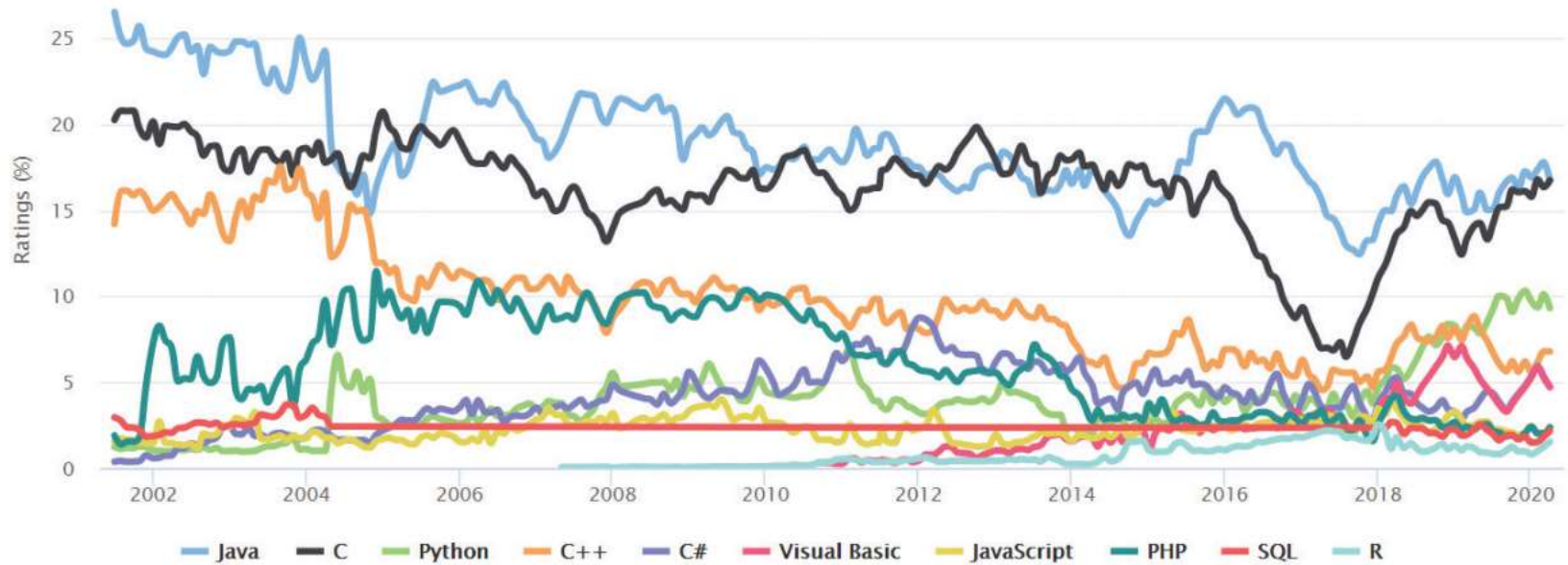
이식성이 좋다.

멀티스레딩을 지원한다.

동적이다.

자바 소개

■ 프로그램 점유율



출처: www.tiobe.com

자바 소개

■ 활용 분야

- 웹 애플리케이션은 물론 CRM, ERP, SCM 등 기업용 애플리케이션
- 빅데이터, 클라우드, 소셜, 모바일, 사물인터넷 등 혁신 기술에서 커넥티드 카, 스마트폰 및 비디오 게임까지 일상생활과 업무 환경 전반에 밀접하게 연관
- 심지어 전 세계 해양 정보 수집, 인간 두뇌와 근골격계 연구 등에도 다양하게 활용



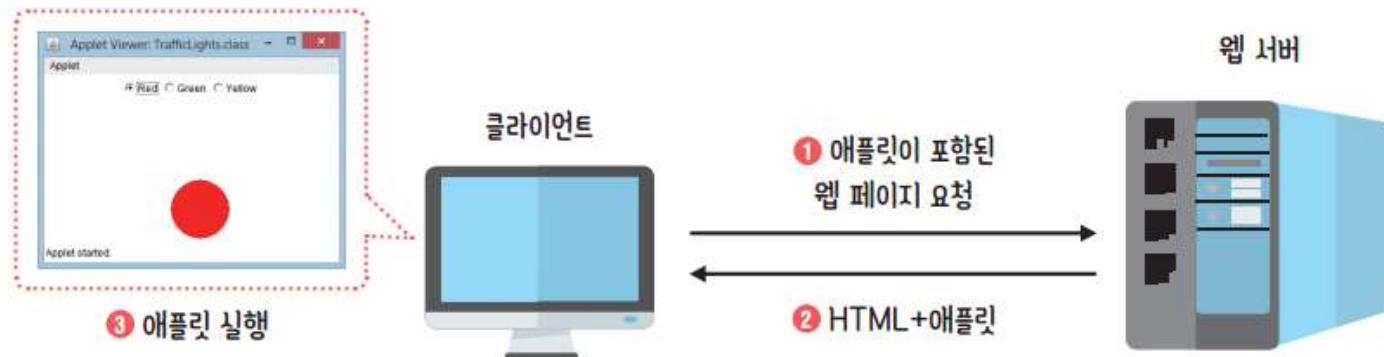
자바 소개

■ 자바 프로그램 종류

● 자바 애플리케이션



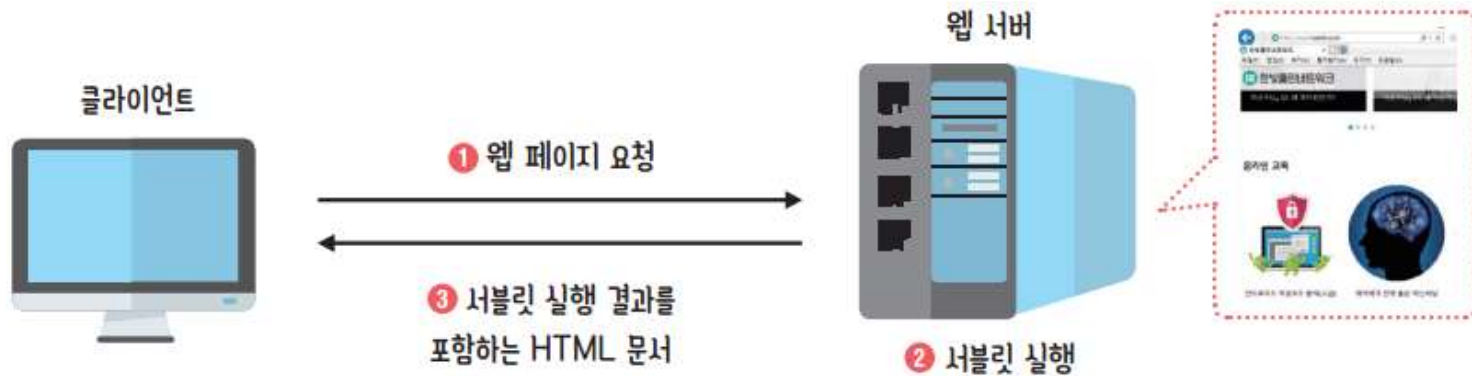
● 자바 애플릿



자바 소개

■ 자바 프로그램 종류

- 자바 서버릿



- JSP
- 자바 빈즈
- ...

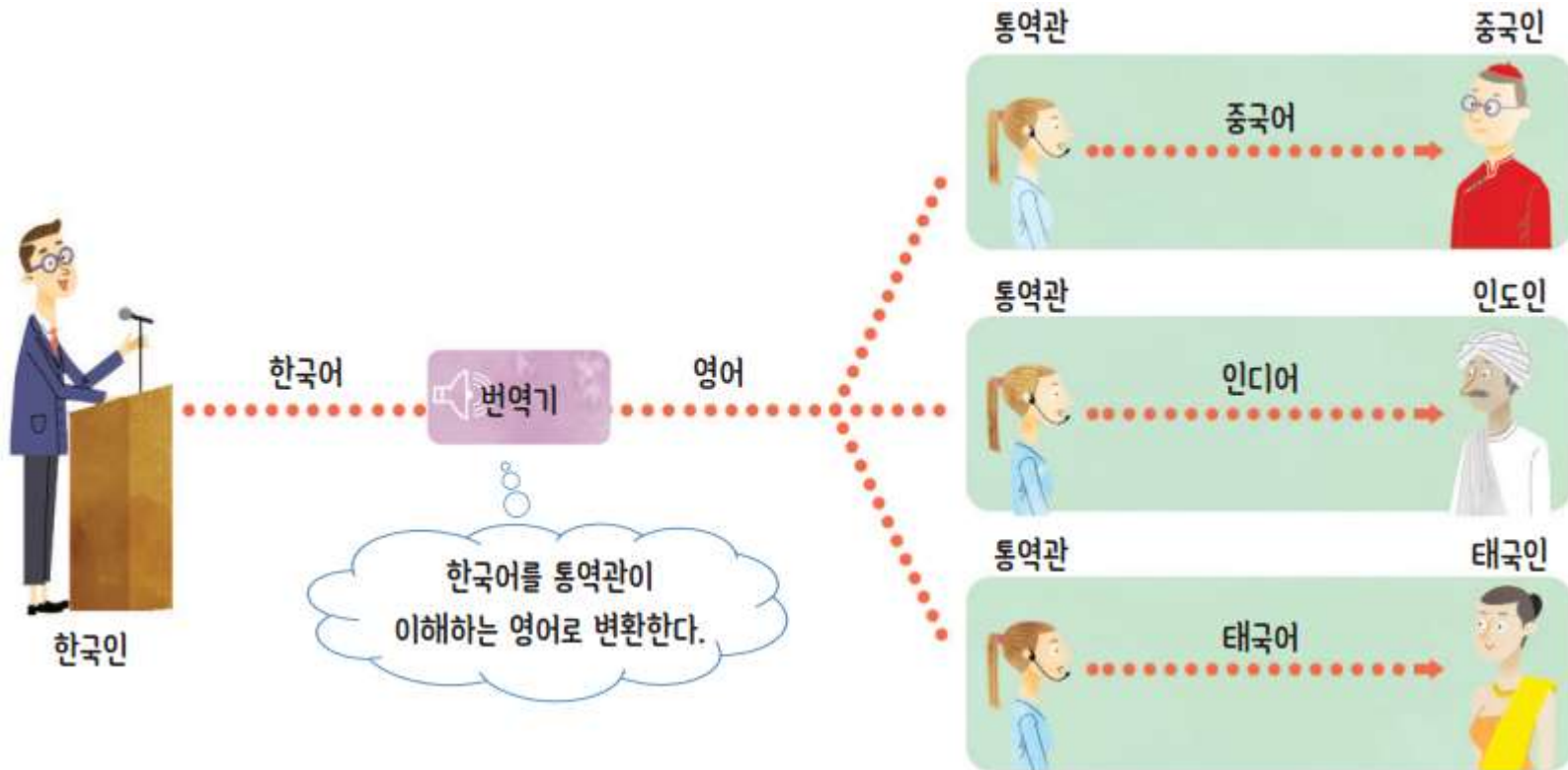
자바 가상 머신

■ 필요성

- 다른 프로그래밍 언어는 플랫폼 종속
 - 하드웨어 아키텍처(CPU)에 따라 사용하는 기계어 종류가 다르다.
 - 운영체제마다 사용하는 응용 프로그래밍 인터페이스(API, Application Programming Interface)와 실행 파일 형식이 다르다.
 - 프로그램을 실행하려면 운영체제가 제어하는 메모리를 사용해야 하는데, 운영체제마다 메모리를 관리하는 방식이 다르다.
- 자바는 자바 가상 머신(JVM)을 사용하기 때문에 플랫폼 독립

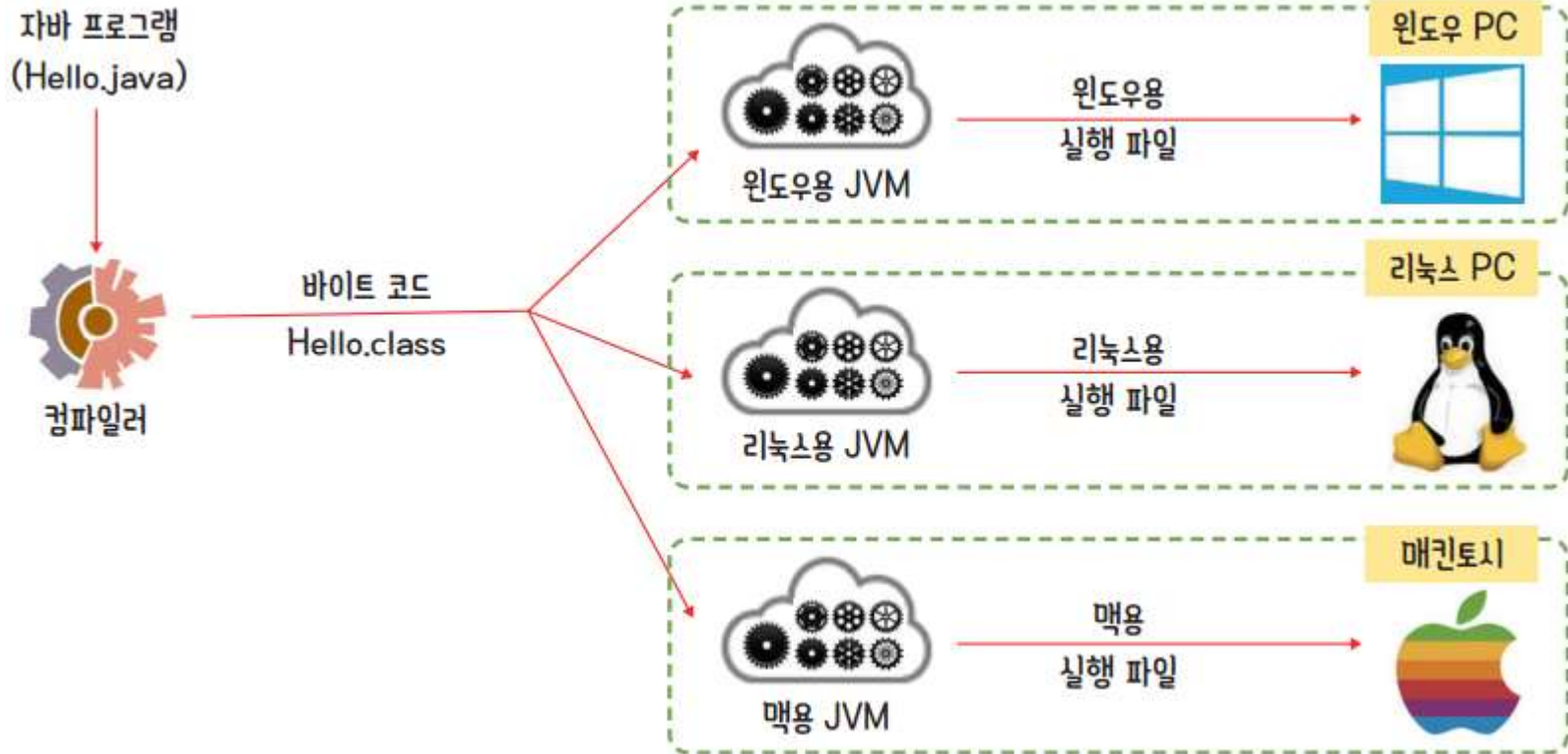
자바 가상 머신

■ 자바 가상 머신은 통역관 역할



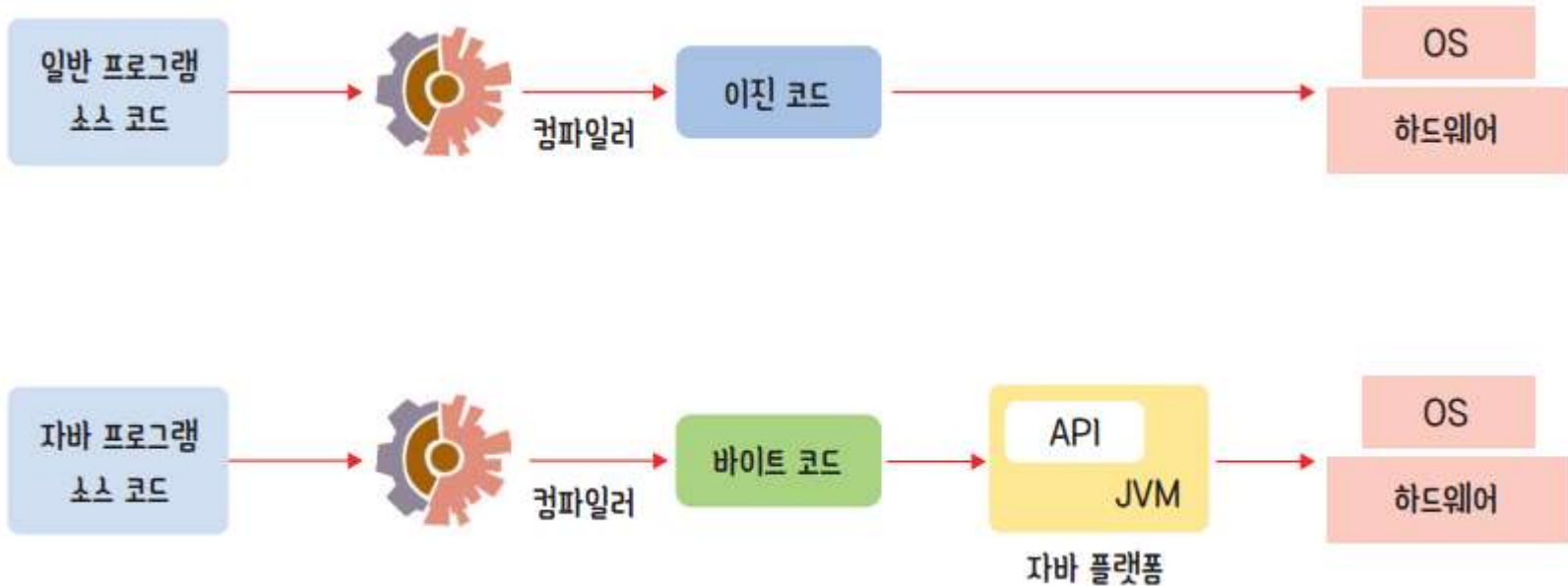
자바 가상 머신

■ 영어에 해당하는 바이트 코드



자바 가상 머신

■ 통역관에 대응하는 JVM



JDK 설치

■ 대표적인 자바 에디션



JDK 설치

■ JDK와 JRE

JDK

컴파일러, 디버거, 애플릿뷰어 등

JRE

클래스 로더, 자바 API, 실행 시간 라이브러리 등

JVM

JIT 컴파일러, 자바 인터프리터 등

JDK 설치

- 오라클 다운로드 사이트에 접속하기

<https://www.oracle.com/kr/java/technologies/downloads/>

Java 19 Java 17

Java SE Development Kit 19.0.2 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**

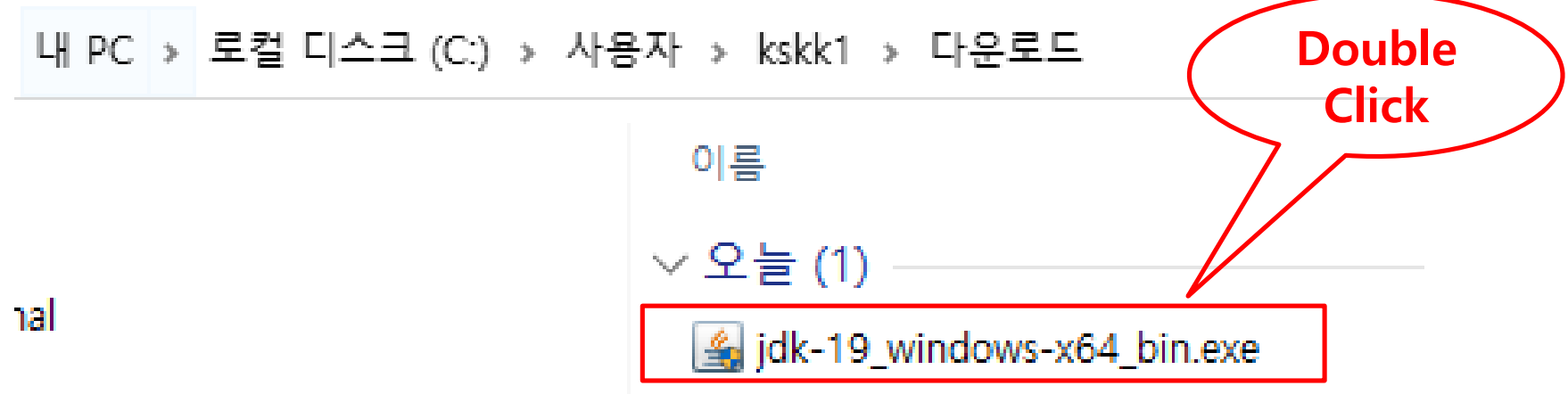
Product/file description	File size	Download
x64 Compressed Archive	179.13 MB	https://download.oracle.com/java/19/latest/jdk-19_windows-x64_bin.zip (sha256)
x64 Installer	158.91 MB	https://download.oracle.com/java/19/latest/jdk-19_windows-x64_bin.exe (sha256)
x64 MSI Installer	157.76 MB	https://download.oracle.com/java/19/latest/jdk-19_windows-x64_bin.msi (sha256)

Click

JDK 설치

■ 오라클 다운로드 사이트에 접속하기

- “다운로드” 폴더에 내려받은 jdk 설치 파일 확인 후 더블클릭



JDK 설치

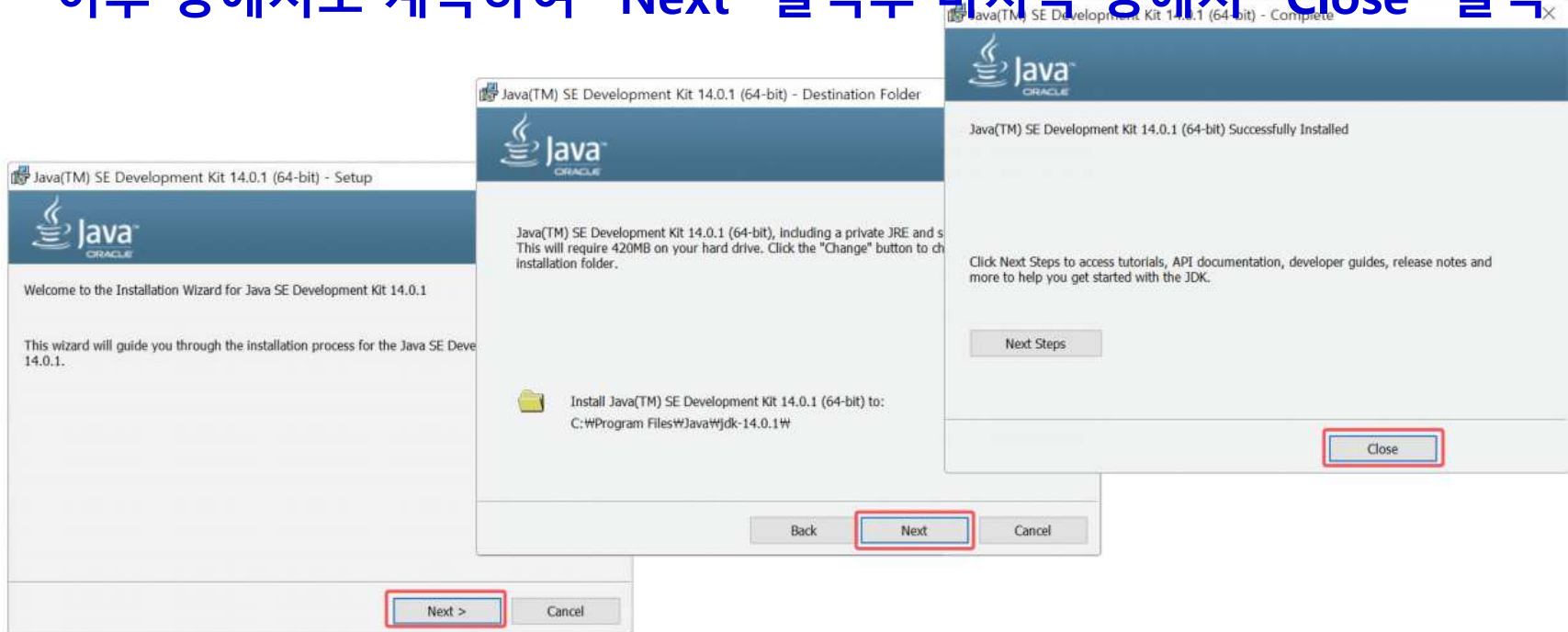
- 오라클 다운로드 사이트에 접속하기
 - Setup 창에서 "Next" 버튼 클릭



JDK 설치

■ 설치 하기

이후 창에서도 계속하여 “Next” 클릭후 마지막 창에서 “Close” 클릭



자바 환경 변수 설정

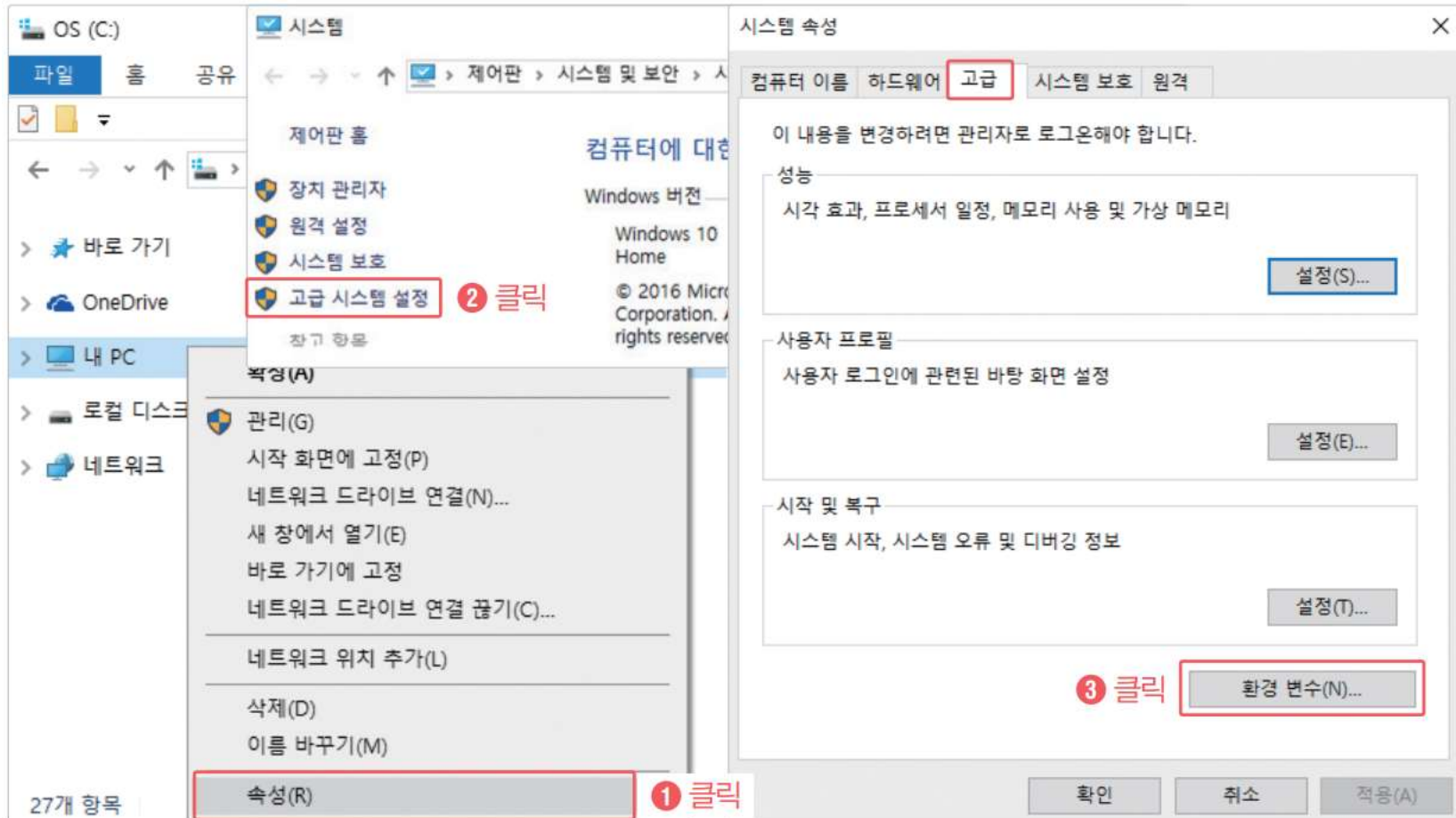
- JDK가 설치된 폴더

- C:\Program Files\Java\jdk버전번호

- 어떤 폴더에서도 자바를 사용하려면 JDK가 설치된 폴더를 환경 변수로 등록해야 함

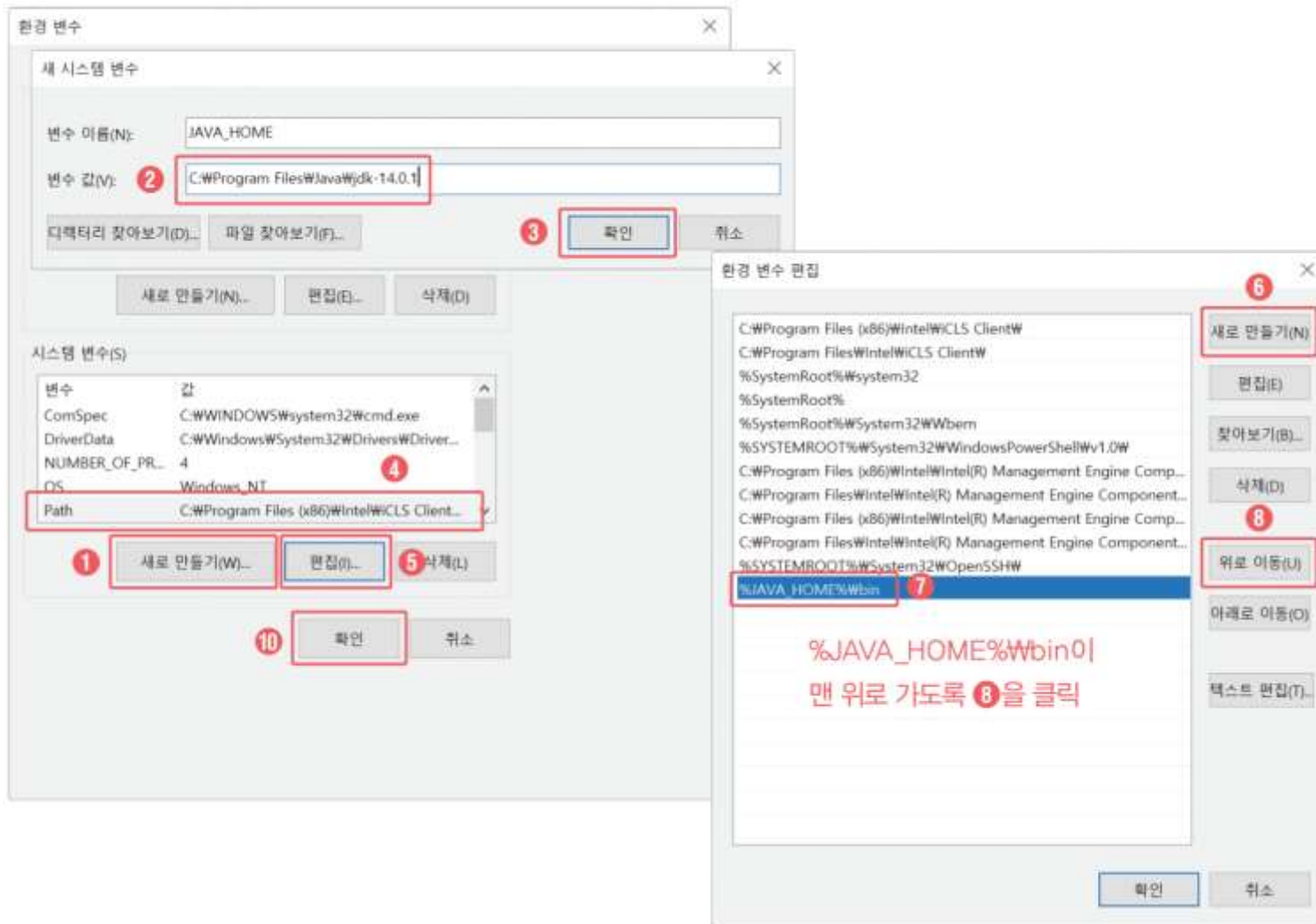
자바 환경 변수 설정

■ [환경 변수] 대화상자 열기



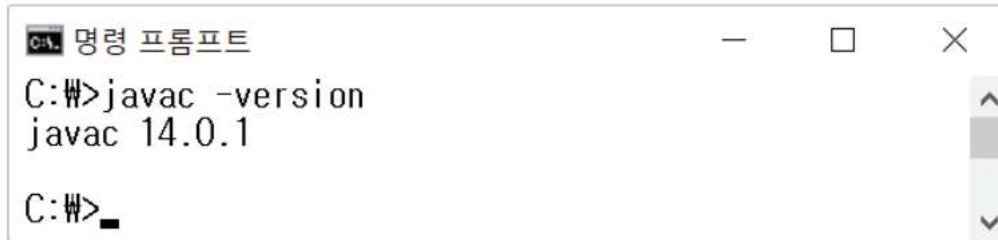
자바 환경 변수 설정

■ 자바 환경 변수 설정하기



자바 환경 변수 설정

■ 자바에 설정한 환경 변수 확인하기



```
명령 프롬프트
C:\W>javac -version
javac 14.0.1
C:\W>_
```

통합 개발 환경

■ 필요성

- 매우 간단한 자바 프로그램은 메모장에서 .java 파일로 작성한 후 컴파일 가능
- 하지만 여러 개의 파일로 된 프로젝트라면 메모장만으로는 어렵다.
- 메모장 대신에 통합 개발 환경(IDE, Integrated Development Environment)을 이용하면 효율적으로 관리 가능
- 대표적인 자바 통합 개발 환경은 이클립스, 넷빈즈(NetBeans), 인텔리J 아이디어(IntelliJ IDEA) 등
- 여기서는 인텔리J 아이디어를 사용
- 인텔리J 아이디어는 저장 버튼이 없고 편집한 코드는 자동으로 저장

통합 개발 환경

■ 이클립스 설치

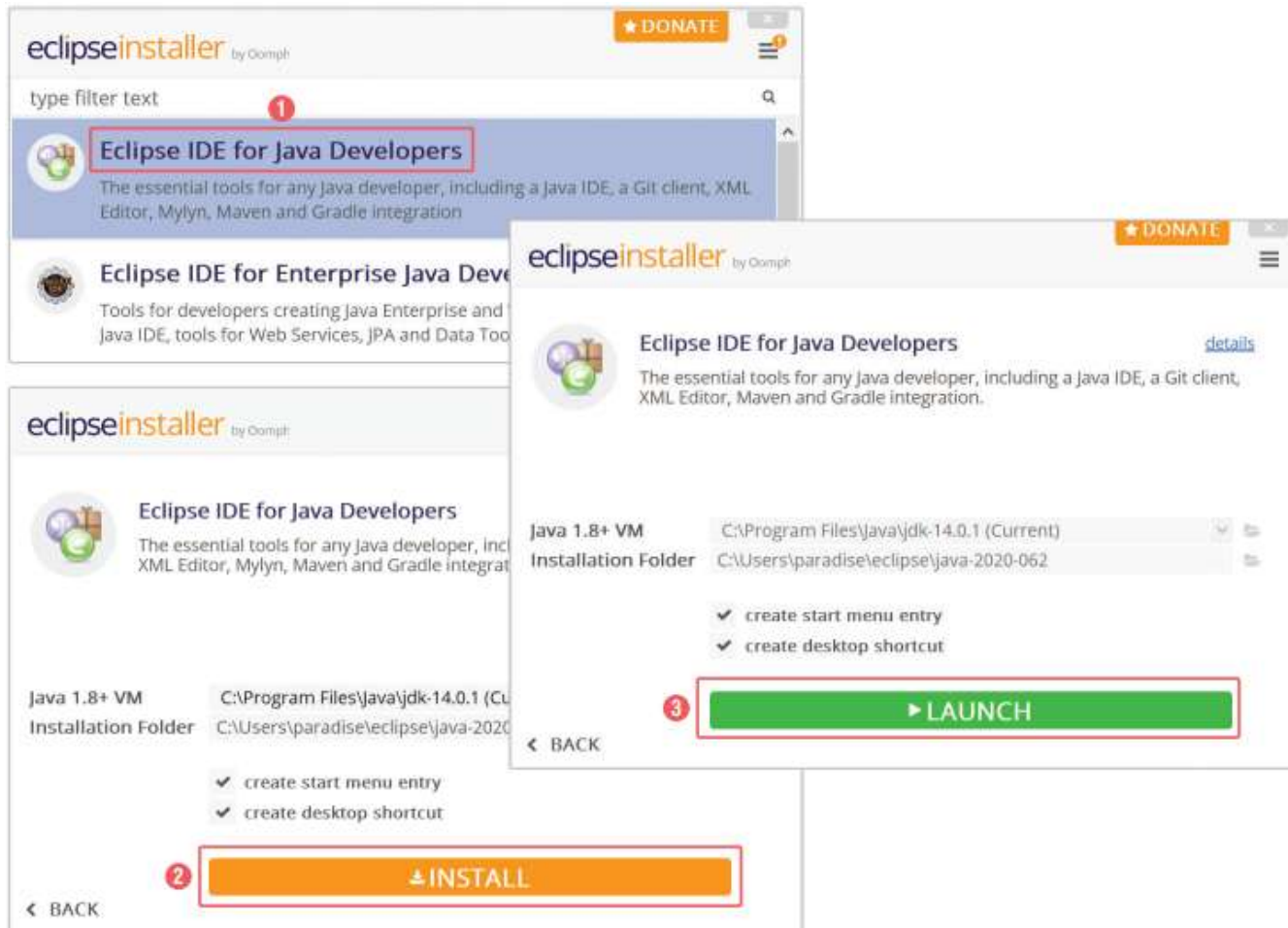
- 사이트(<https://www.eclipse.org>)에 접속하여 설치 파일 다운로드



통합 개발 환경

■ 이클립스 설치

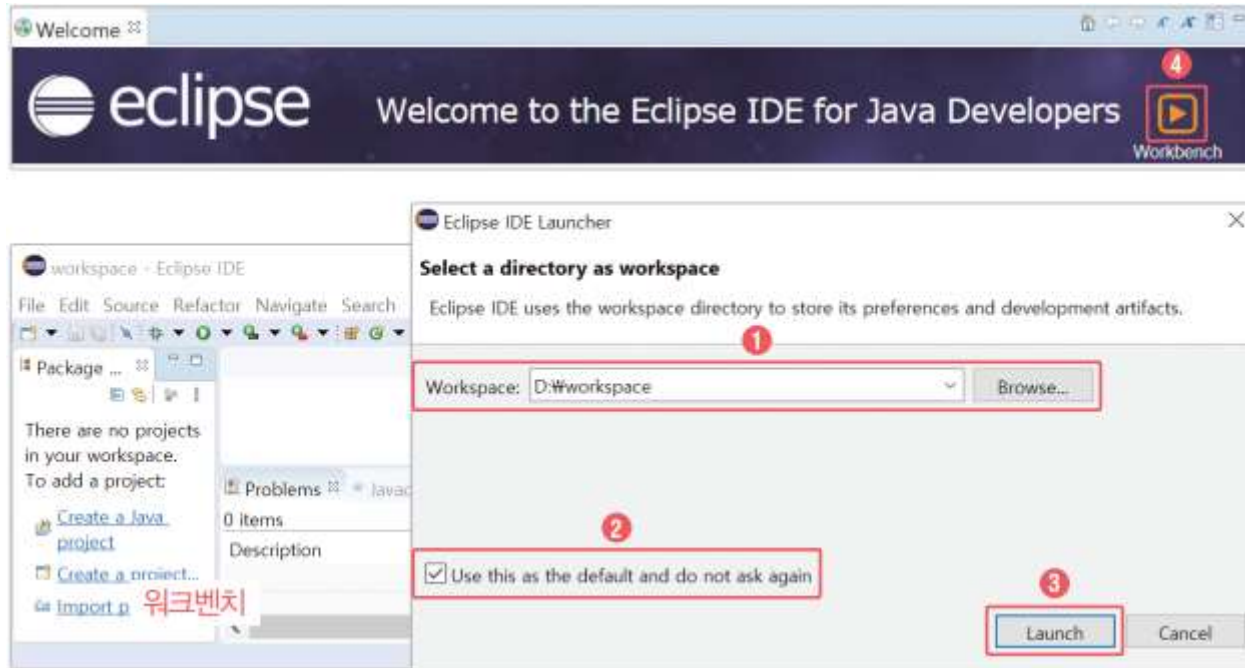
- 다운로드한 파일의 실행



통합 개발 환경

■ 이클립스 설치

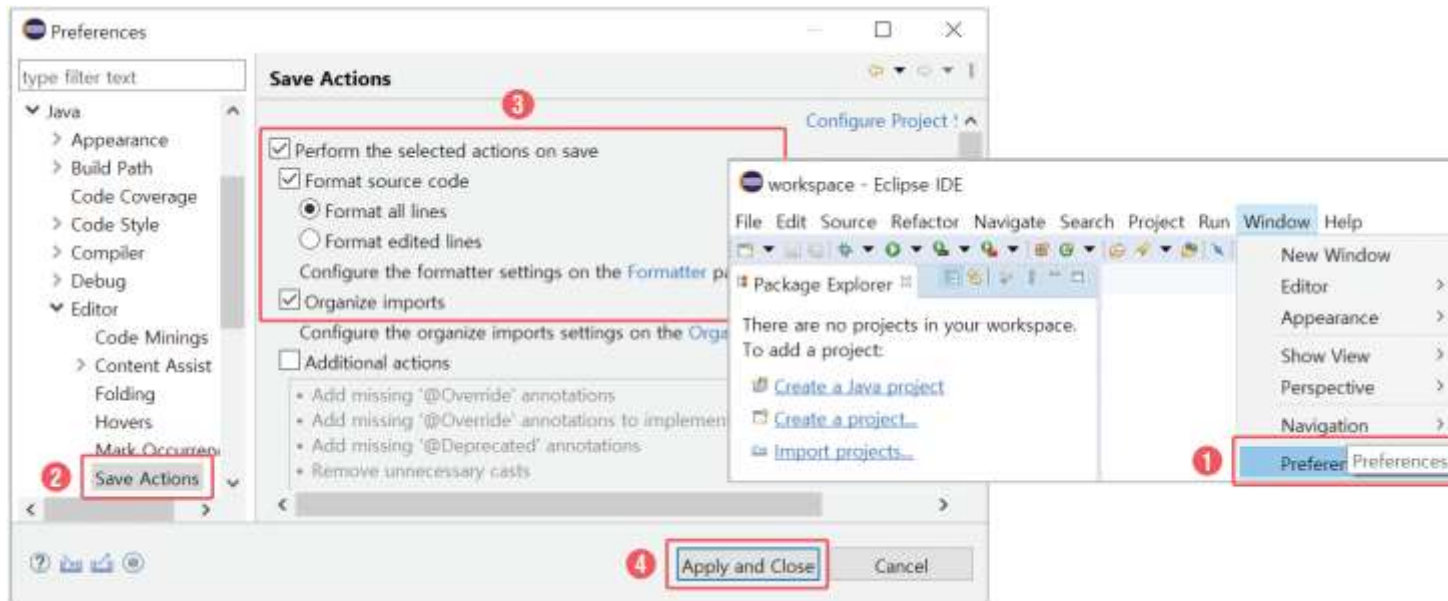
- 이클립스 작업공간 설정



통합 개발 환경

■ 이클립스 설치

- 필요한 라이브러리 클래스의 자동 импорт 및 작성한 소스 코드를 자동 포매팅
- [Window]-[Preferences]를 클릭



통합 개발 환경

■ 이클립스 퍼스펙티브

현재 퍼스펙티브 표시

퍼스펙티브 선택 버튼

메뉴바

툴바

Task List

Outline

Package Explorer

```
1 package chap01;
2
3 public class Hello {
4
5     public static void main(String[] args) {
6         // TODO Auto-generated method stub
7         System.out.println("안녕!");
8     }
9
10 }
11
```

Problems Javadoc Declaration Console

<terminated> Hello [Java Application] C:\Program Files\Java\jdk-14.0.1\bin\javaw.exe (2020)

안녕!

chap01.Hello.main(String[] args) : void - chap01/src

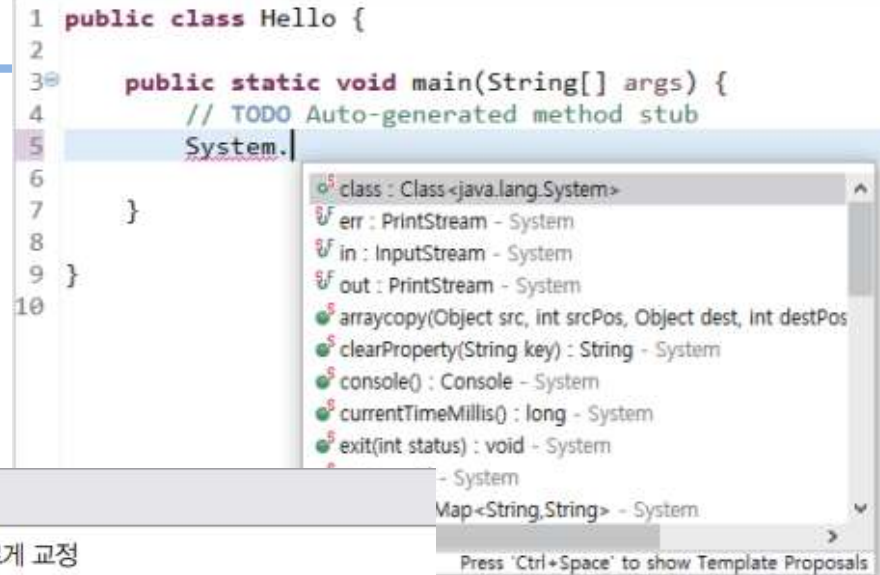
패키지 탐색기 패키지 내에 포함된 소스 파일을 계층 구조로 표시해 파일을 생성하고 삭제할 수 있다.

에디터 자바 코드를 편집할 수 있으며, 문법에 따라 코드가 다양한 색상으로 나타난다.

콘솔 프로그램에서 표준 장치로 출력하는 결과를 보여 준다. 처음에는 보이지 않다가 자바 프로그램을 실행하면 나타난다.

통합 개발 환경

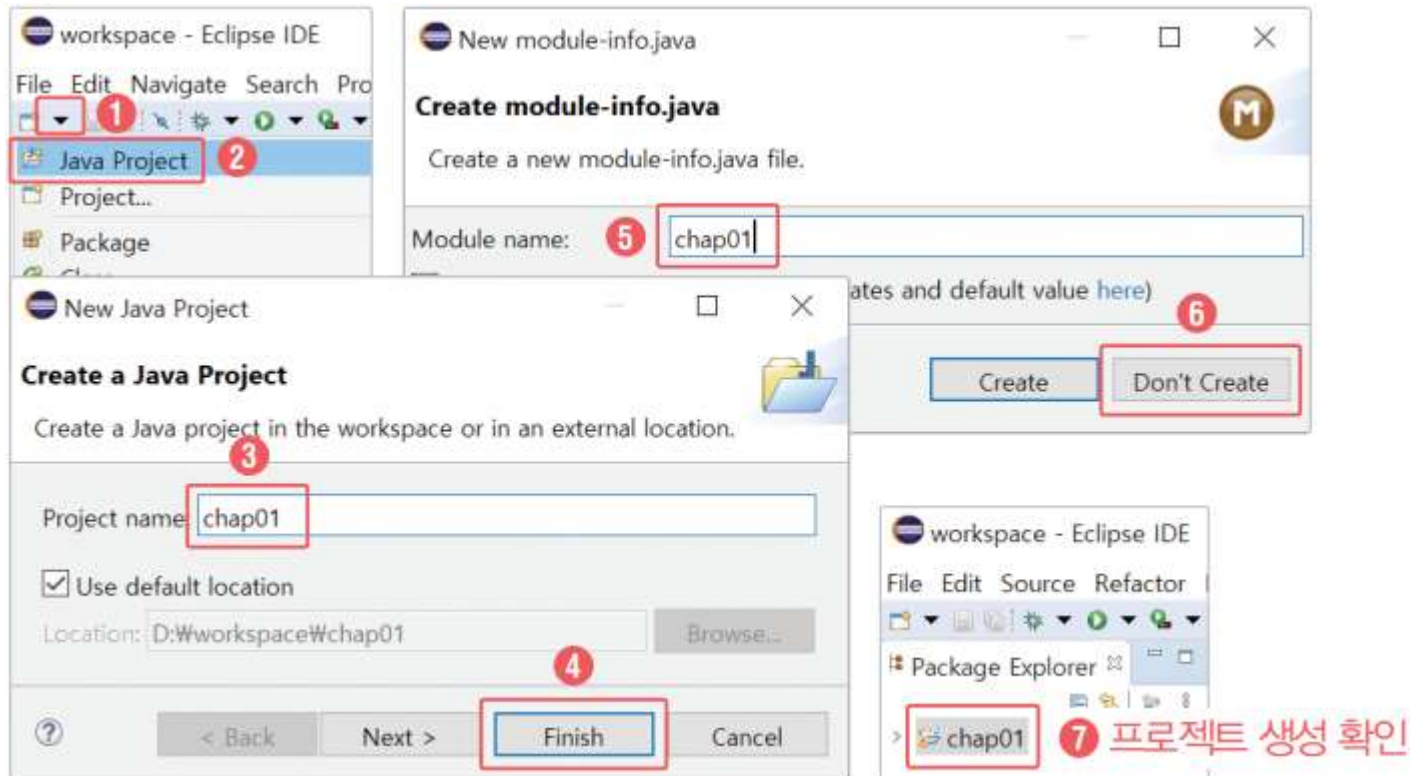
■ 이클립스 코드 박스 활용 및 단축키



단축키	설명
Ctrl + 1	코드를 문맥에 맞게 빠르게 교정
Ctrl + Shift + O	자동으로 import 문 추가
Ctrl + Space Bar	주어진 단어나 문장으로 시작하는 메서드나 변수 표시
Ctrl + /	주석 처리, 해제
Ctrl + Shift + F	코드의 내용을 보기 좋게 포매팅
Ctrl + F11	에디터 뷰에 있는 클래스 실행
Alt + Shift + R	선택한 이름을 일괄 변경(Refactoring)
Alt + Shift + Z	선택한 블록을 try~catch, for, while 문 등으로 자동 완성
F3	메서드변수를 선택하고 누르면 해당 메서드가 선언된 곳으로 이동
'sysout' + Ctrl + Space Bar	System.out.println() 삽입
Ctrl + Shift + X / Ctrl + Shift + Y	소문자를 대문자로, 대문자를 소문자로 치환
Ctrl + d	한 행 삭제

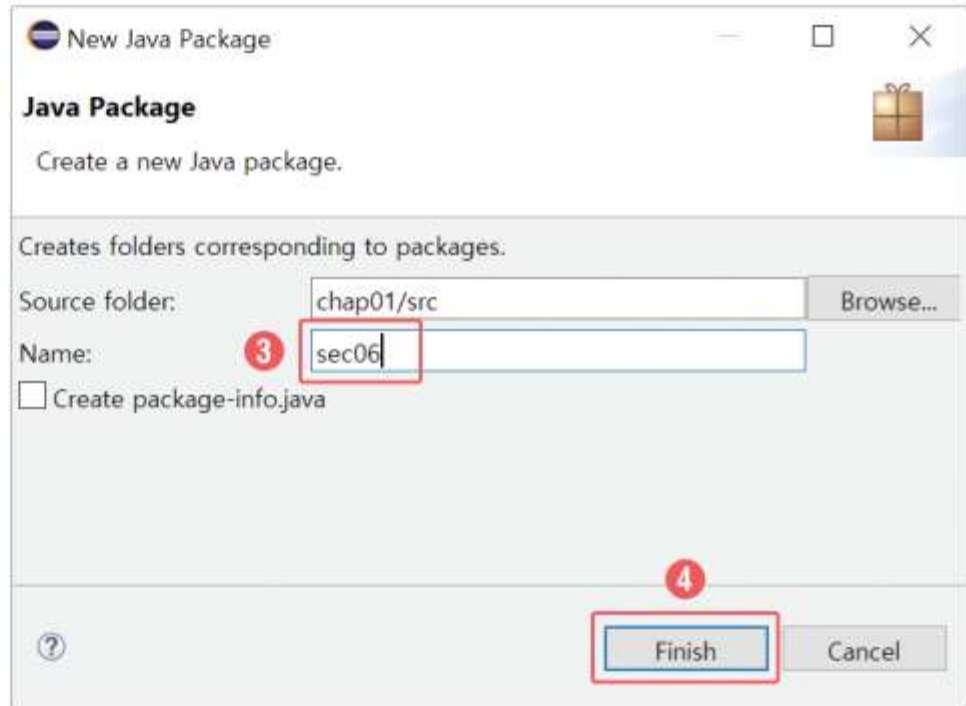
Hello 프로그래밍

■ 프로젝트 생성



Hello 프로그래밍

■ 패키지 생성



Hello 프로그래밍

■ 클래스 생성

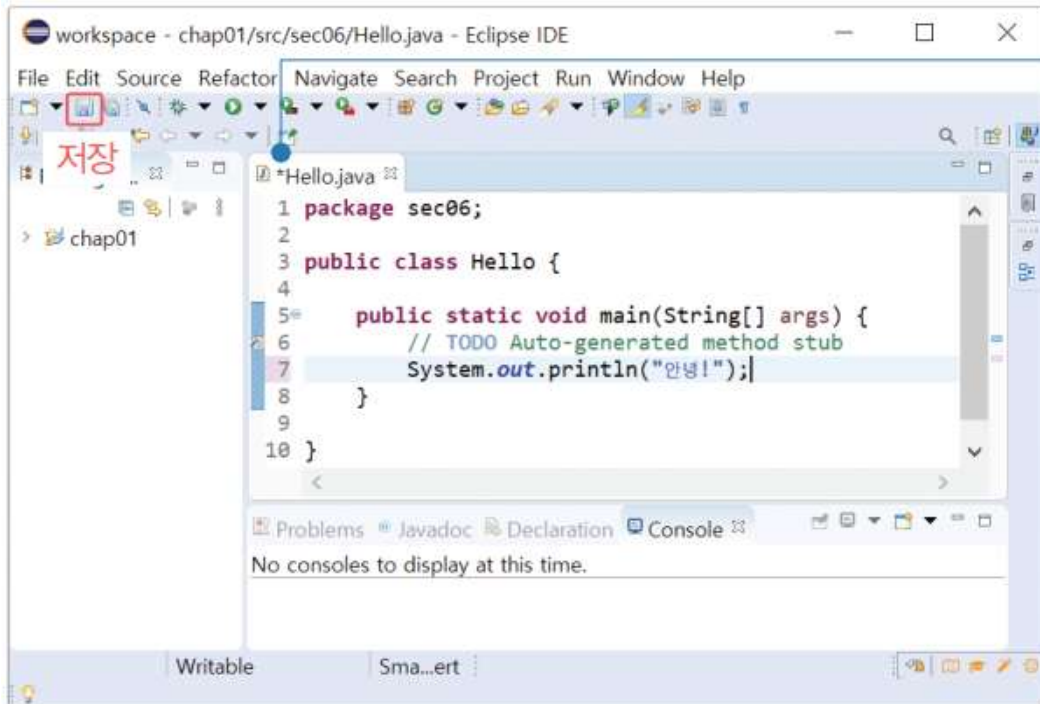
The image shows three screenshots from the Eclipse IDE illustrating the steps to create a new Java class:

- Left Screenshot: New Java Class Dialog**
 - Source folder:** chap01/src
 - Package:** sec06
 - Name:** Hello (labeled with a red circle 3)
 - Modifiers:** public (selected) (labeled with a red circle 6)
 - Superclass:** java.lang.Object
 - Which method stubs would you like to create?:** public static void main(String[] args) (checked, labeled with a red circle 4)
 - Finish button:** (labeled with a red circle 5)
- Top Screenshot: Package Explorer and Editor**
 - Package Explorer:** Shows the project structure with chap01 > src > sec06 > Hello.java (labeled with a red circle 2).
 - Editor:** Shows the code for Hello.java:

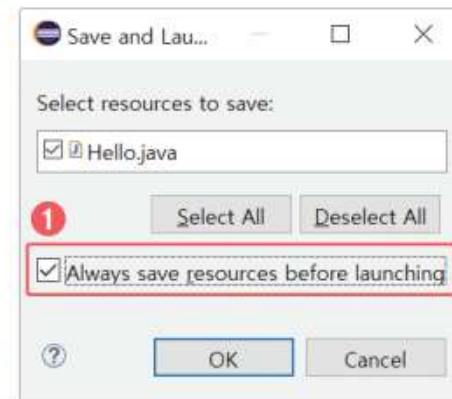
```
1 package sec06;  
2  
3 public class Hello {  
4  
5     public static void main(String[] args) {  
6         // TODO Auto-generated method stub  
7     }  
8  
9 }  
10  
11
```
- Bottom Screenshot: Package Explorer**
 - Package Explorer:** Shows the project structure with chap01 > src > sec06 (labeled with a red circle 1).
 - New Java Class button:** (labeled with a red circle 2)

Hello 프로그래밍

■ 클래스 저장 및 실행



*는 파일을 수정한 후 아직 저장하지 않았다는 의미이다.



Hello 프로그래밍

■ 클래스 저장 및 실행

