



자바프로그래밍

박숙영

blue@sookmyung.ac.kr

강의 교재 (참고)

■ 강의 교재

- 천인국, Power JAVA, 인피니티

■ 강의 자료

- 스노보드 자료 참고

■ 강의 평가

- 중간고사 30%
- 기말고사/프로젝트 35%
- 과제 30%
- 출석 5% (결석 -1, 지각 -0.5)



실습 과제

- 매 주 약 6개의 문제가 주어짐

- [평가 항목]

- 필요한 주석 달기(1)
 - 작성자, 날짜, 간단한 문제 설명 (코드 상단에 작성)
 - 각 기능에 대한 필요한 주석 작성 (설명 필요 없으면 생략 가능)
- 코드의 가독성(1)
 - 들여쓰기와 적절한 줄 바꾸기
 - 전체 선택 후 Ctrl + Shift + F 로 코드 정리 가능
- 기능 수행
 - 컴파일 오류 없이 실행되는 지(1)
 - 문제에서 제시한 기능 실행(2)
- 제출 기한 지키기 (어길 시 20% 감점, 24시간 이후 50%, 48시간 이후 0점)

- [제출 파일]

- 소스코드와 실행 결과 캡처한 보고서(pdf로 변환 후 제출)+ 실습 파일들(.java) 함께 제출
- 보고서 미제출시 0점

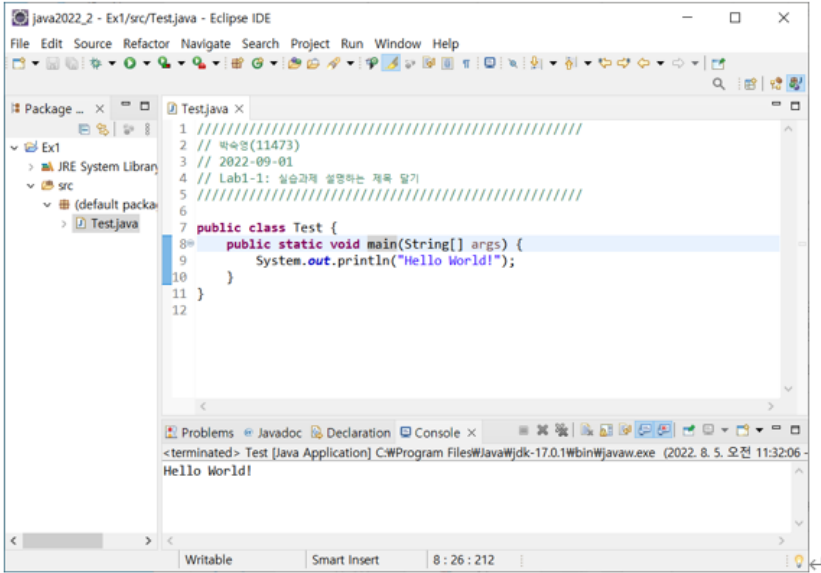
```
////////////////////////////////////  
// 작성자: 박숙영(11473)  
// 작성일: 2022-12-22  
// Lab1-1: 실습과제 주석 설명 예제  
////////////////////////////////////
```

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

실습 결과 보고서

학과: 소프트웨어학부 학번: 12345 이름: <

<실습 1><



The screenshot shows the Eclipse IDE interface. The Package Explorer on the left shows a project named 'Ex1' with a source folder 'src' containing a file 'Test.java'. The Editor window displays the code for 'Test.java':

```
1 //////////////////////////////////////////////////
2 // 박숙영(11473)
3 // 2022-09-01
4 // Lab1-1: 실습과제 설명하는 제목 쓰기
5 //////////////////////////////////////////////////
6
7 public class Test {
8     public static void main(String[] args) {
9         System.out.println("Hello World!");
10    }
11 }
12
```

The Console window at the bottom shows the output: 'Hello World!'.

<실습 2><

.... <

- 파일명: 이름_학번.pdf
- 표지 필요 없음.
- 소스코드가 한 화면에 다 들어가지 않을 경우 나눠서 캡처 진행
- 입력 값에 따라 다양한 실행결과를 보여줘야 할 경우 실행 결과만 추가로 따로 캡처해서 덧붙이기.
- 실행이 제대로 안되었을 경우 또는 필요한 경우 추가 설명 작성 가능



1장 자바소개와 개발도구 설치

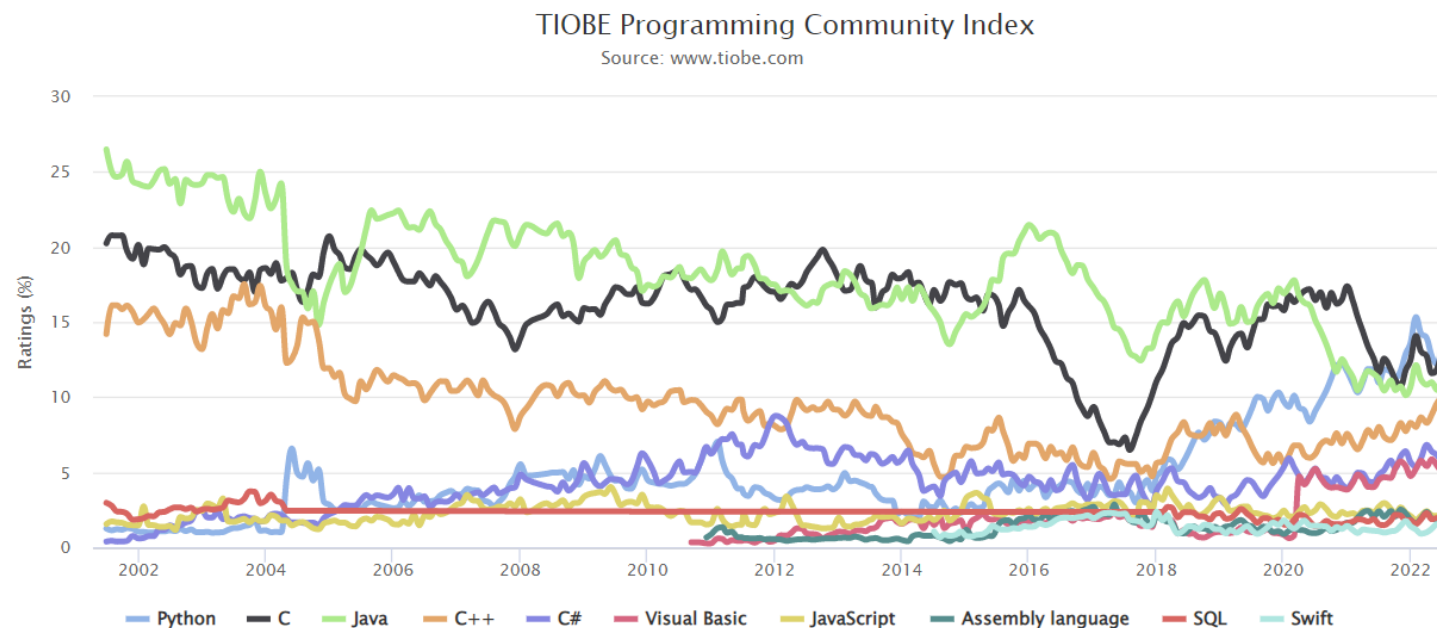
1장의 목표

1. 자바 언어의 특징을 설명할 수 있나요?
2. 자바 프로그램이 실행되는 과정을 설명할 수 있나요?
3. 자바를 설치할 수 있나요?
4. 자바를 명령어 버전으로 컴파일하고 실행할 수 있나요?
5. 이클립스를 이용하여 자바 프로그램을 작성할 수 있나요?



자바란?

- 자바(Java)는 현재 컴퓨터 업계에서 가장 많이 사용되는 프로그래밍 언어 중의 하나



- 자바는 방대한 라이브러리와 다양한 실행 환경을 가지고 있는 하나의 거대한 플랫폼

자바는 누가 만들었을까?

- 1991년에 제임스 고슬링(James Gosling)을 비롯한
썬 마이크로시스템즈 사의 Green 프로젝트팀
 - 가정용 전자 제품에 사용할 수 있는 컴퓨터 언어를 설계
- 목적
 - 플랫폼 호환성 문제 해결
 - 기존 언어로 작성된 프로그램은 PC, 유닉스, 메인 프레임 등 플랫폼 간에 호환성 없음
 - 소스를 다시 컴파일하거나 프로그램을 재 작성해야 하는 단점
 - 플랫폼 독립적인 언어 개발
 - 모든 플랫폼에서 호환성을 갖는 프로그래밍 언어 필요
 - 네트워크, 특히 웹에 최적화된 프로그래밍 언어의 필요성 대두
 - 메모리 사용량이 적고 다양한 플랫폼을 가지는 가전 제품에 적용



자바의 특징

■ 플랫폼 독립성

- 플랫폼 상관없이 어디서든지 실행 가능
 - 자바 컴파일러는 자바 언어로 작성된 프로그램을 바이트 코드 (byte code)라는 특수한 이진파일로 변환
 - JVM(Java Virtual Machine) 이 바이트코드를 실행

■ 객체지향

- 객체별로 코드를 작성하고 객체들을 조합하여 전체 프로그램을 완성하는 프로그램 설계 방법론
- 객체 지향 기법을 사용하면 이미 작성된 코드를 재사용하기가 쉬움
 - 보다 빠르게 신뢰성 있는 프로그램을 만들 수 있음

■ 멀티스레딩

- 여러 가지 작업을 동시에 실행하는 것을 의미
- 자바는 언어 수준에서 멀티스레딩(multithreading)을 지원



자바 응용의 종류

- 자바 애플리케이션(Java application)
 - 독립적으로 실행될 수 있는 일반 응용 프로그램

- 자바 애플릿(Java applet)
 - 웹 브라우저 안에서 실행되는 작은 프로그램

- 자바 서블릿(Java servlet)
 - 웹서버에서 동작하는 서버 모듈로서 클라이언트의 요구를 받아서 그에 대한 처리를 한 후에, 실행 결과를 HTML 문서 형태로 클라이언트 컴퓨터로 전송

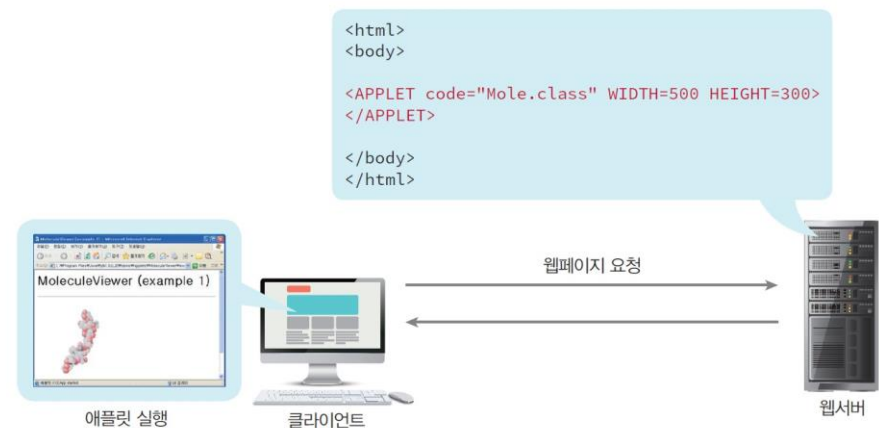


그림 1.10 자바 애플릿의 실행 과정

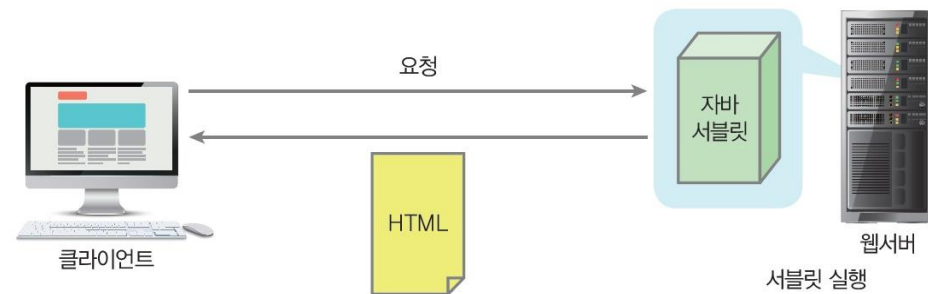
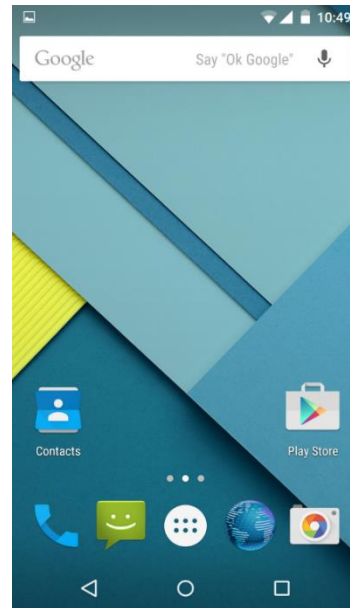


그림 1.11 자바 서블릿의 실행 과정

안드로이드 애플리케이션

- 안드로이드 애플리케이션은 자바로 작성
- 안드로이드 개발자들은 자바의 SE 버전 중에서 AWT와 스윙(swing)을 제외한 거의 모든 패키지를 사용
- 효율성 때문에 자체적인 달빅 가상 머신을 구현



플랫폼 종속성

- C 언어나 C++ 언어를 사용하여 프로그래밍을 하면, 이들 언어는 모두 타겟 컴퓨터의 CPU나 운영체제에 맞춘 코드를 생성
- 따라서 이 프로그램을 다른 CPU나 운영체제에서 실행하려면 다시 컴파일해야 함

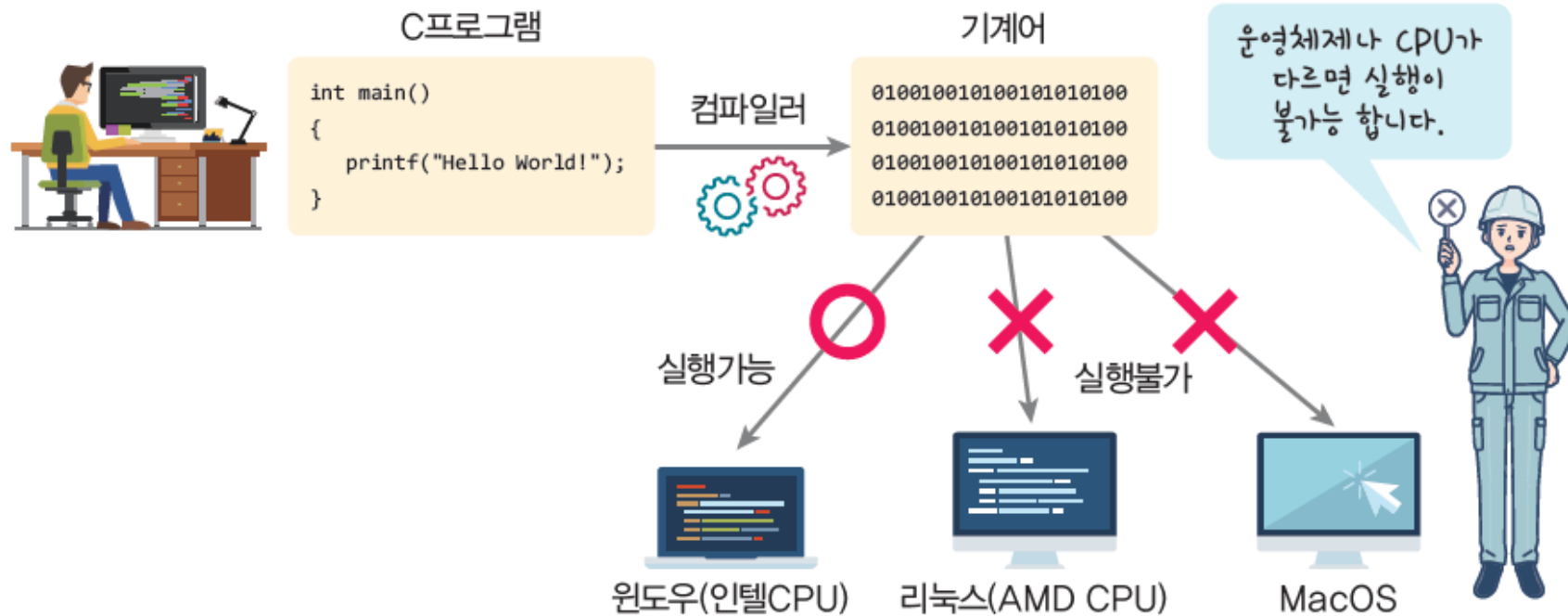


그림 1.7 C 또는 C++ 언어로 컴파일하였을 경우, 운영체제나 CPU가 다르면 실행이 불가능하다.

자바의 플랫폼 독립성

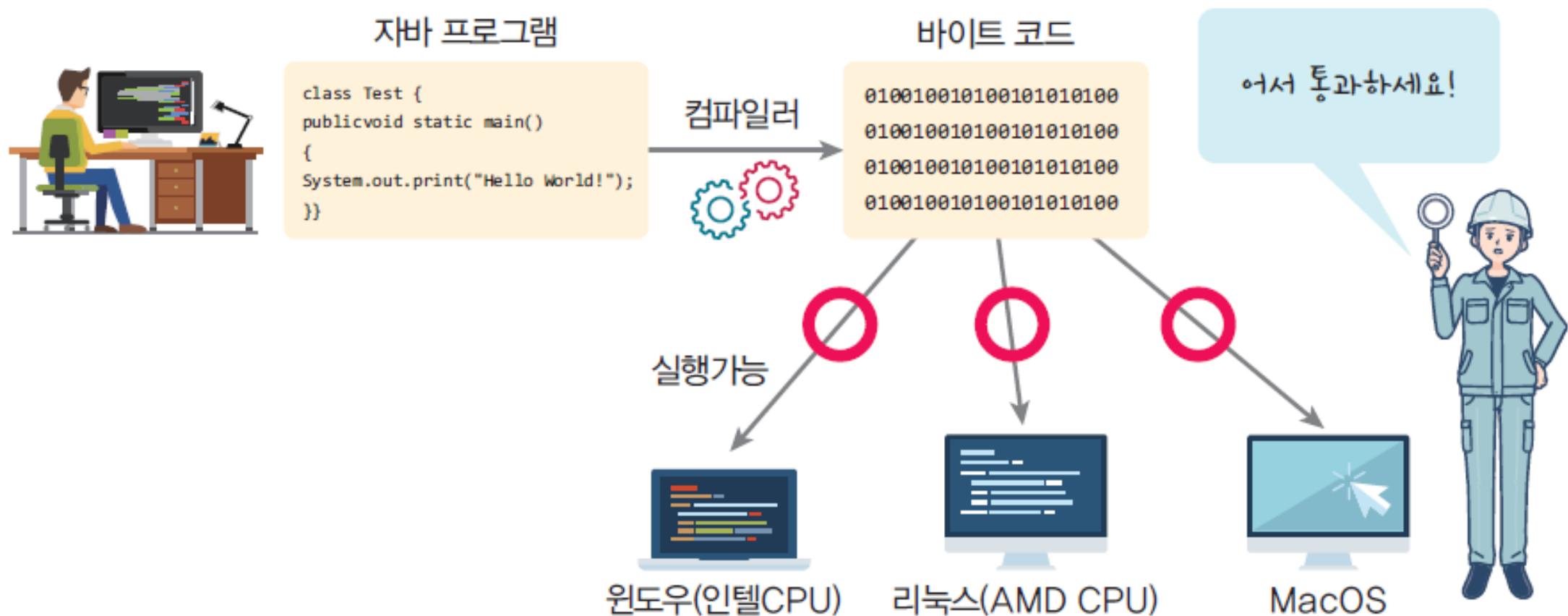
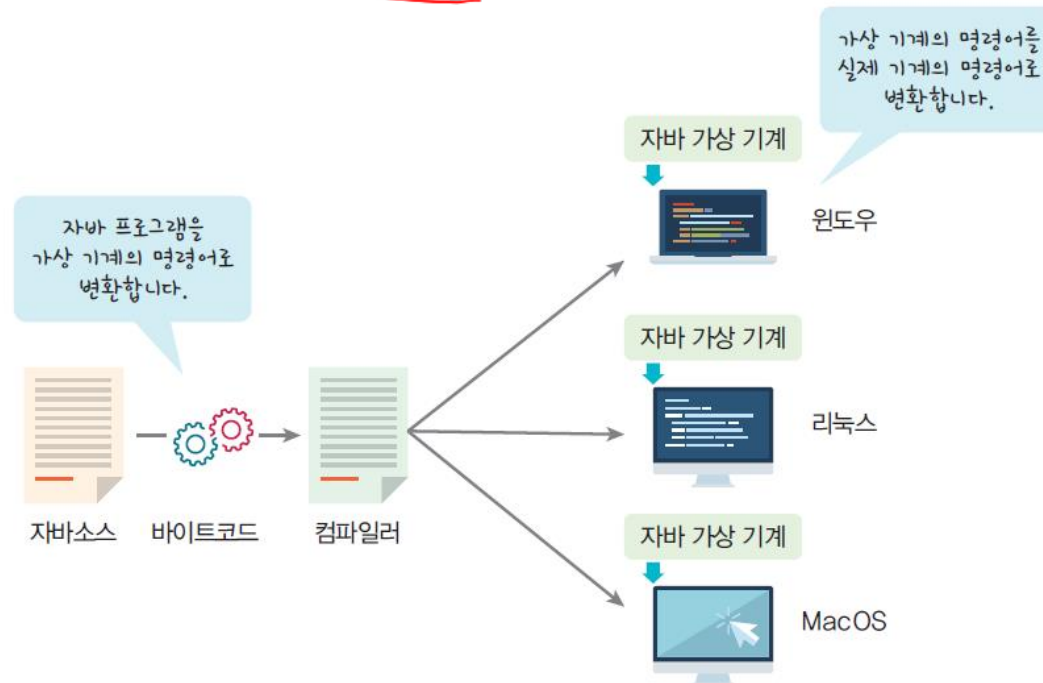


그림 1.8 자바 언어로 컴파일하였을 경우, 운영체제나 CPU에 상관없이 실행이 가능하다.

자바 프로그램의 실행과정

- 자바 컴파일러는 특정한 컴퓨터를 위한 코드를 바로 생성하지 않는다. 대신에 가상 컴퓨터의 기계어인 바이트 코드(byte code)를 생성한다.
- 이 바이트 코드는 자바 가상 기계(JVM: Java Virtual Machine)라고 하는 소프트웨어에 의하여 한 줄씩 해석되면서 실행된다.



JVM을 사용하는 다른 언어들

- JVM 언어를 컴파일하면 자바 가상 기계가 실행할 수 있는 클래스 파일을 생성



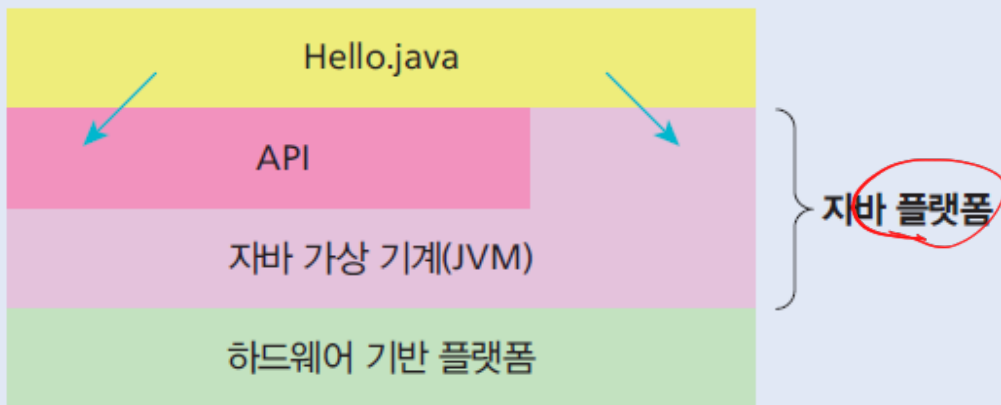
참고

자바 플랫폼

플랫폼(platform)이란 프로그램이 실행되는 하드웨어와 소프트웨어 환경이다. 자바 플랫폼은 두 가지의 요소로 이루어져 있다. 플랫폼만 같으면 바이트 코드를 변경 없이 실행할 수 있다.

- 자바 가상 기계(JVM: Java Virtual Machine)
- 자바 응용 프로그래밍 인터페이스(API: Application Programming Interface)

일반적으로 API란 많은 유용한 기능을 제공하는 라이브러리들의 모임이다. API는 자바 프로그래밍 언어의 핵심 기능을 제공한다. API에는 네트워킹, 보안, XML 생성, 데이터베이스 접근에 필요한 기능들이 포함되어 있다.

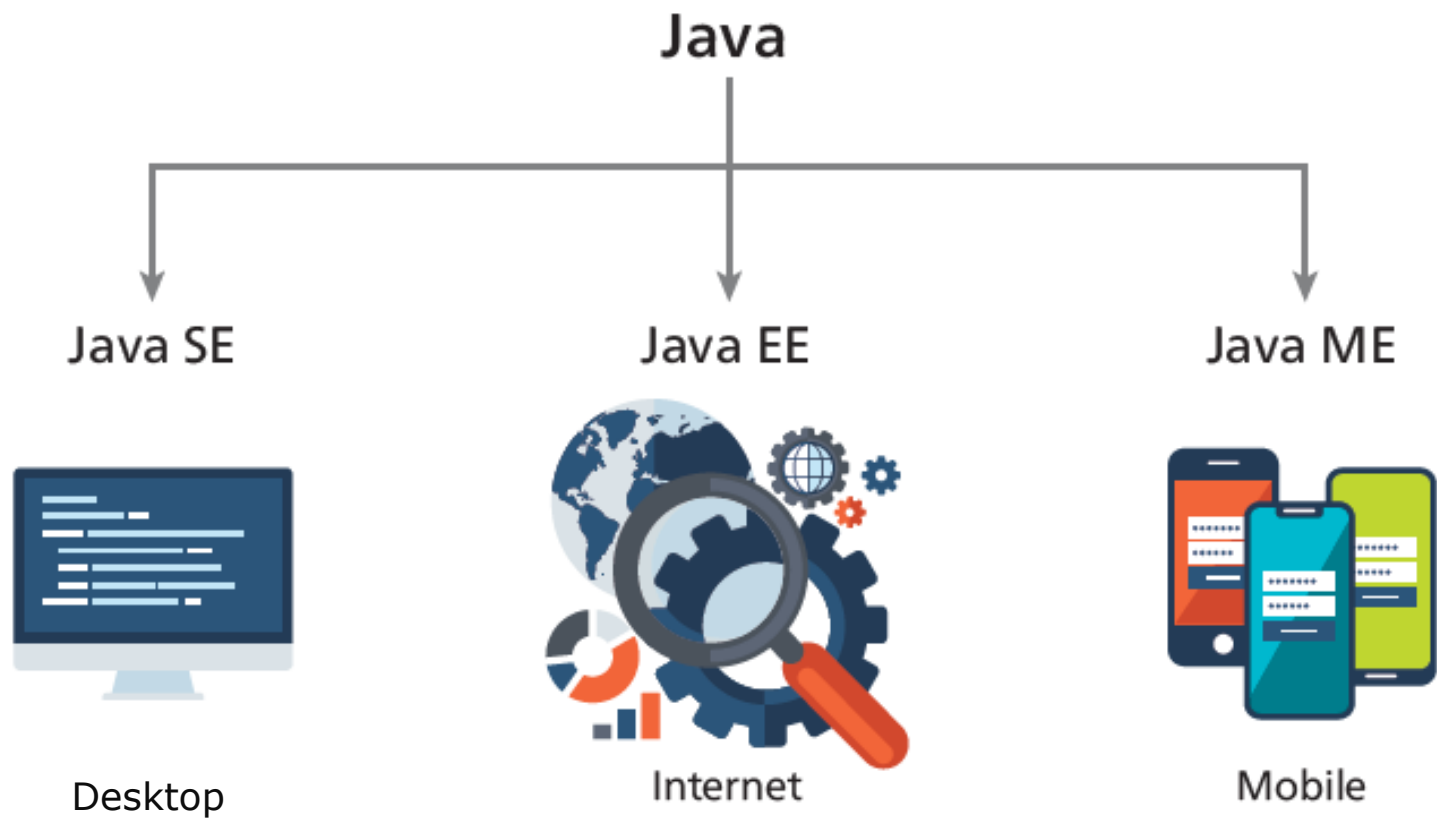


참고



자바의 종류와 버전

- 자바는 개발 분야에 따라 Java SE, Java ME, Java EE, Java FX 플랫폼 등으로 나누어서 제공된다.



자바의 종류와 버전

■ Java SE

- Java SE는 데스크탑과 서버에서 자바 애플리케이션을 개발하고 실행할 수 있게 해주며 임베디드 환경(embedded environment)과 실시간 환경(real-time environments)도 지원한다.

■ Java EE

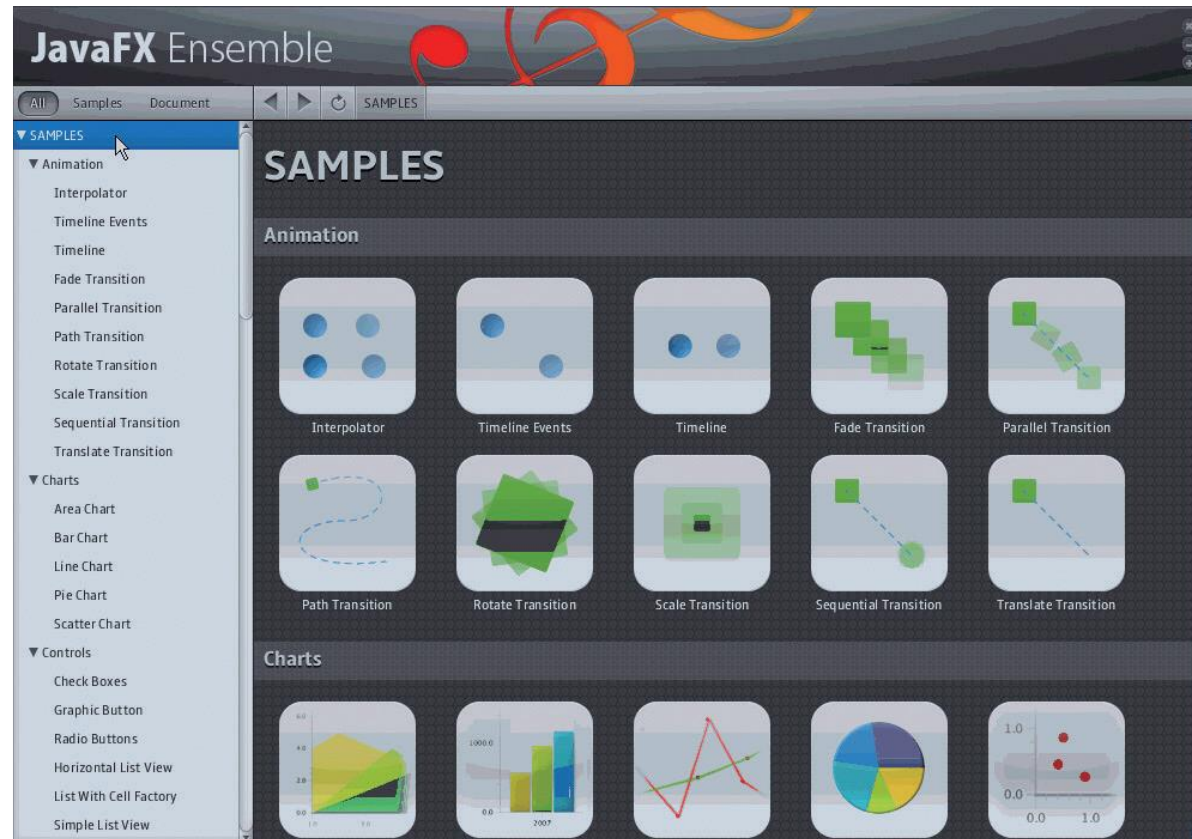
- Java EE는 기업용 애플리케이션을 개발하는 데 필요한 여러 가지 도구 및 라이브러리들을 모아 놓은 것
- 응용 서버, 웹서버, J2EE API, 엔터프라이즈 자바 빈즈(JavaBeans) 지원, 자바 서블릿 API 와 JSP 등을 포함

■ Java ME

- Java ME는 핸드폰, PDA, TV 셋톱박스, 프린터와 같은 모바일 기기나 다른 임베디드 장치들에서 실행되는 애플리케이션을 위한 강인하고 유연한 환경을 제공

JavaFX

- JavaFX 플랫폼은 애플리케이션 개발자들이 더 쉽게 리치-인터넷-애플리케이션(RIA: Rich Internet Application)을 생성하고 배포하기 위한 자바 클라이언트 플랫폼



자바의 버전

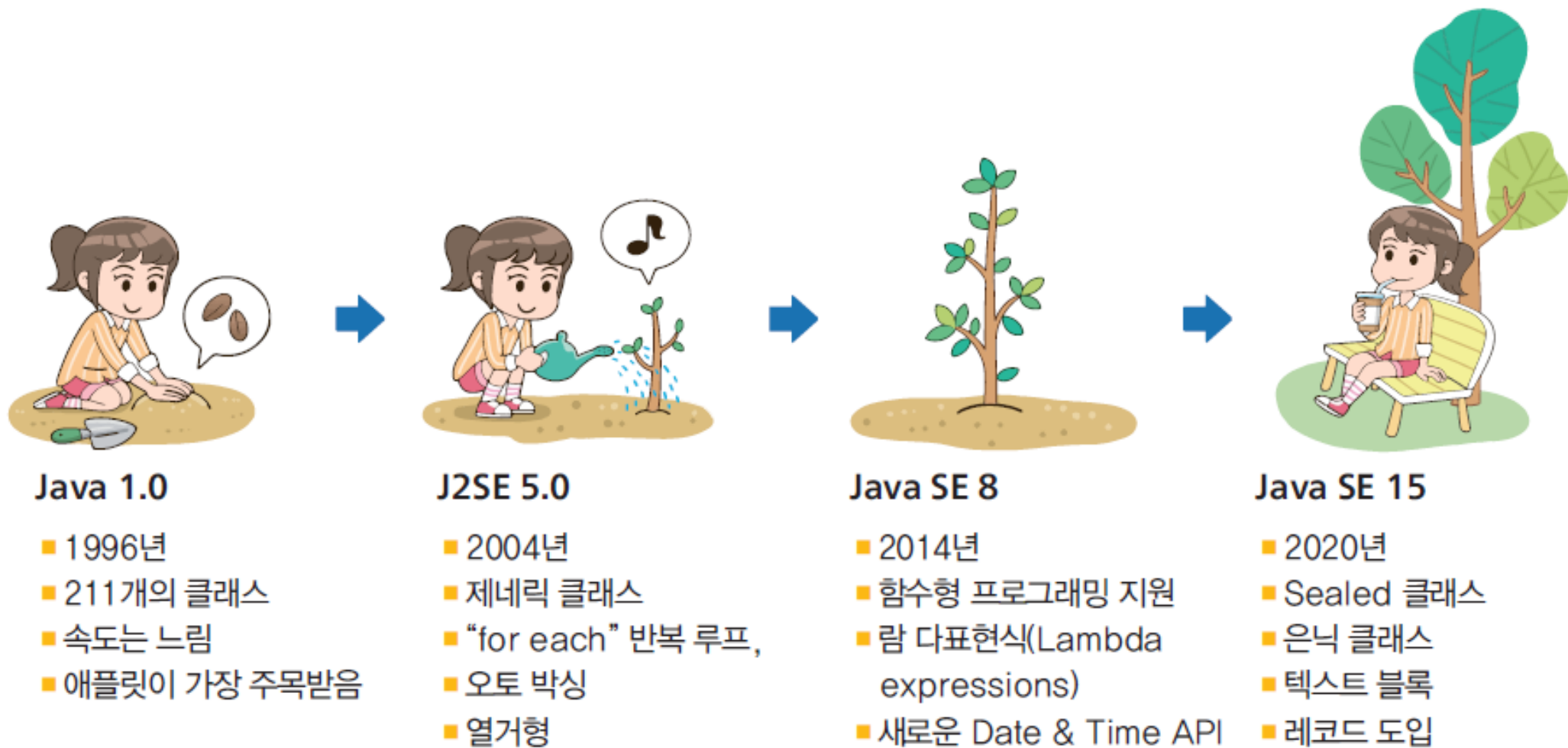


그림 1.14 자바 버전 요약

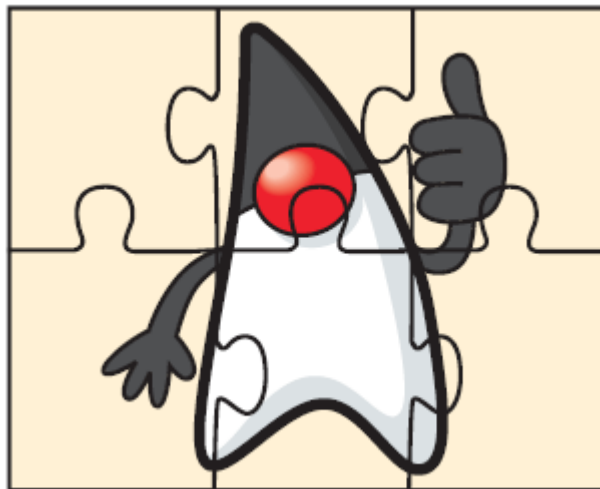
Java SE 8: 함수형 프로그래밍 지원

■ Java SE 8에 추가된 중요한 특징

- 함수형 프로그래밍의 지원
- 람다식(Lambda expressions)
- 작은 가상 기계(VM)
- 병렬 배열 정렬(ParallelArray Sorting)
- 컬렉션을 위한 대용량 데이터 처리
- Base64 인코딩과 디코딩을 위한 표준 API
- 새로운 날짜, 시간 API(Date & Time API)
- 강화된 비밀번호 기반 암호화>Password-Based-Encryption (PBE))

Java SE 9: 모듈화 프로그래밍

- 모듈화는 직소(jigsaw) 프로젝트의 산물
- 모듈화의 목적은 자바 응용 프로그램을 만들 때, 마치 직소 퍼즐을 연결하듯이 필요한 모듈만을 묶어서 사용하려는 것이다.



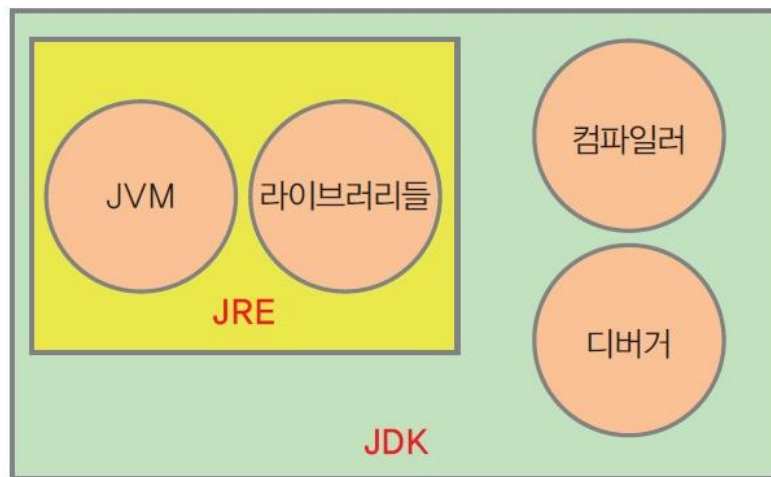
JDK와 JRE

■ JRE(Java Runtime Environment)

- JRE는 자바 프로그램을 실행하기 위한 라이브러리, 자바 가상 기계, 기타 컴포넌트들을 제공한다.

■ JDK(Java Development Kit)

- JDK는 JRE에 추가로 자바 프로그램을 개발하는데 필요한 컴파일러, 디버거와 같은 명령어행 개발 도구를 추가한 것이다.



Oracle JDK vs Open JDK

- 하나는 상업용 코드를 판매하는 Oracle JDK이고 다른 하나는 공개 소스 기반의 Open JDK이다.
- 오라클은 라이선스가 들어있는 코드를 제외한 자바 설치본을 Open JDK를 통하여 배포하고 있다.
- Open JDK는 Oracle JDK와 거의 차이가 없다.

JDK 설치

← → ↻ oracle.com/java/technologies/downloads/#jdk17-windows



ORACLE



Products

Industries

Resources

Support

Events

Developer

Partners



Java 17 available now

Java 17 LTS is the latest long-term support release for the Java SE platform. JDK 17 binaries are free to use in production and free to redistribute, at no cost, under the [Oracle No-Fee Terms and Conditions](#).

[Learn about Java SE Subscription](#)

JDK 17 will receive updates under these terms, until at least September 2024.

Java SE Development Kit 17.0.2 downloads

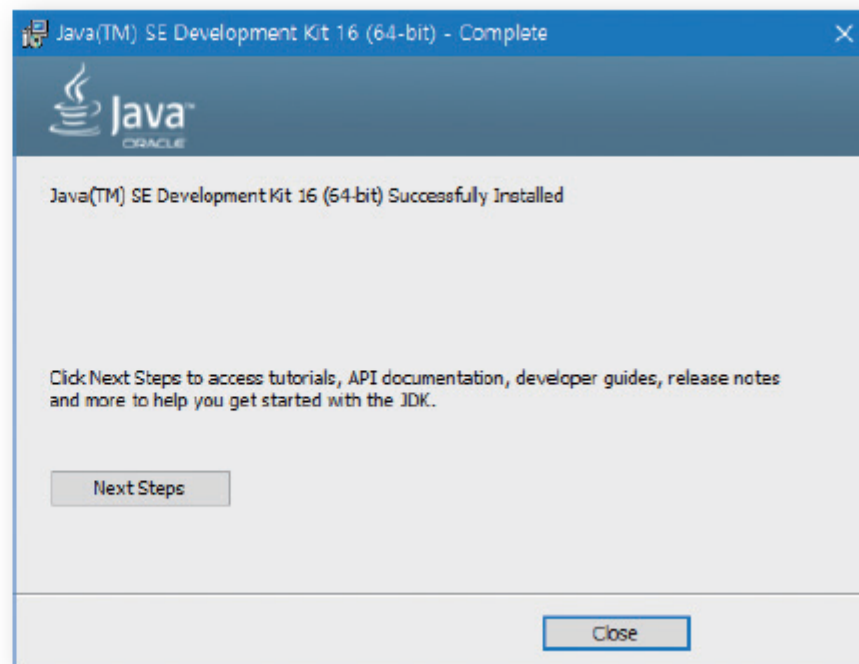
Thank you for downloading this release of the Java™ Platform, Standard Edition Development Kit (JDK™). The JDK is a development environment for building applications and components using the Java programming language.

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	171.34 MB	https://download.oracle.com/java/17/latest/jdk-17_windows-x64_bin.zip (sha256 🔗)
x64 Installer	152.43 MB	https://download.oracle.com/java/17/latest/jdk-17_windows-x64_bin.exe (sha256 🔗)
x64 MSI Installer	151.32 MB	https://download.oracle.com/java/17/latest/jdk-17_windows-x64_bin.msi (sha256 🔗)

JDK 설치



JDK의 폴더

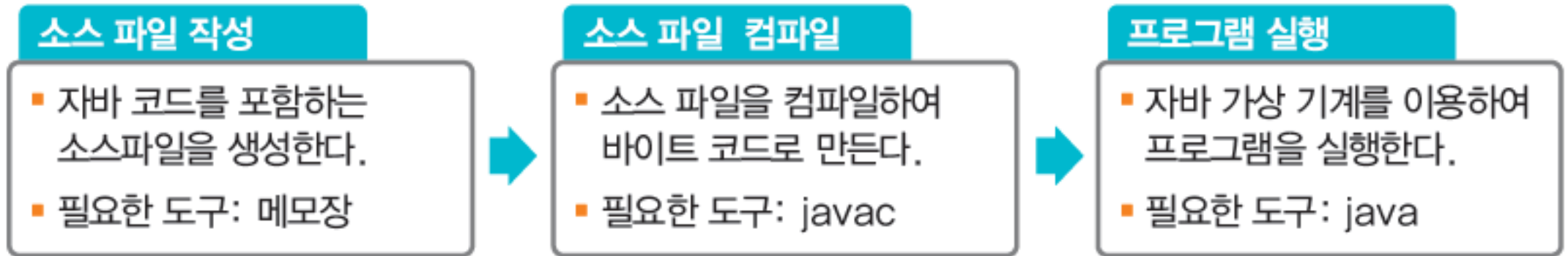
- JDK의 기본 설치 폴더 경로는 c:\Program Files\Java가 된다

표 1.1 JDK의 디렉터리

폴더	설명
bin	컴파일러, 디버거 등의 도구가 들어 있다.
conf	구성 파일이 저장된다. 개발자가 편집할 수 있다.
include	네이티브 코드 프로그래밍을 지원하는 헤더 파일들이다. 이들 파일들은 자바와 C를 동시에 사용하는 프로그램 개발시에 쓰인다.
jmods	컴파일된 모듈이 정의되어 있다.
legal	저작권, 라이선스 파일이 저장된다.
lib	개발 도구들이 필요로 하는 추가적인 클래스 라이브러리와 지원 파일들이다.

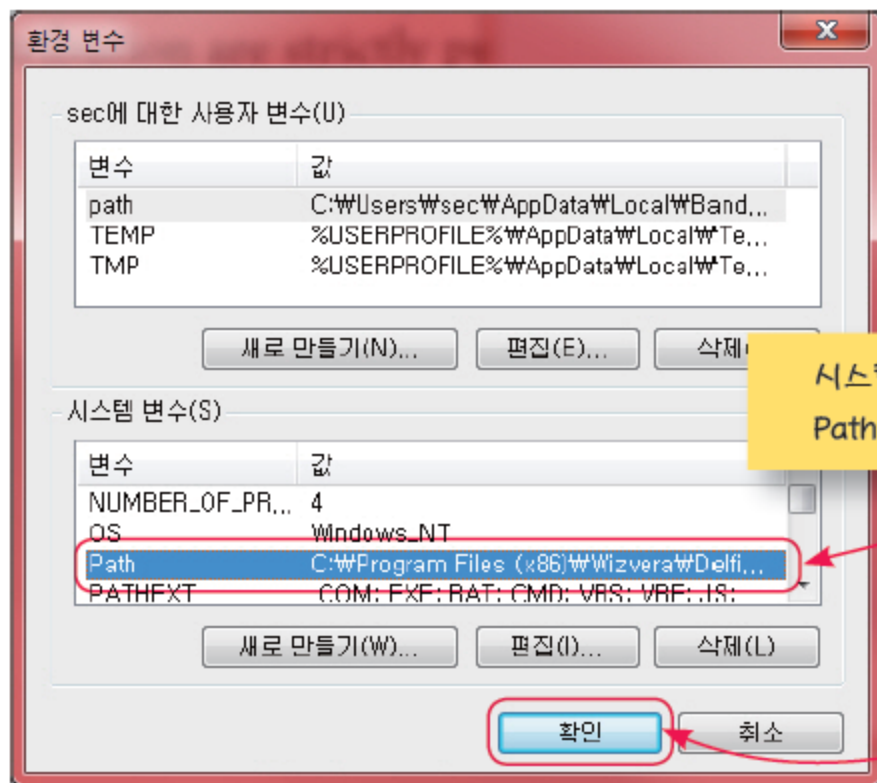
명령어 버전 JDK 사용하기

- 자바 컴파일러인 javac, 바이트코드 실행 명령어 java 등이 포함
- 명령어 도구는 불편하지만 자바 프로그램이 실행되는 원리를 알기 위해서 반드시 한번은 해보아야 한다



경로 설정

- 어디에서나 컴파일러를 실행할 수 있도록 경로(path) 설정하기
- 제어판의 환경 변수 path를 설정한다.



자바 설치 경로 추가 (각 설치된 버전에 따라)
C:\Program Files\Java\jdk-17.0.1\bin

소스 파일 작성

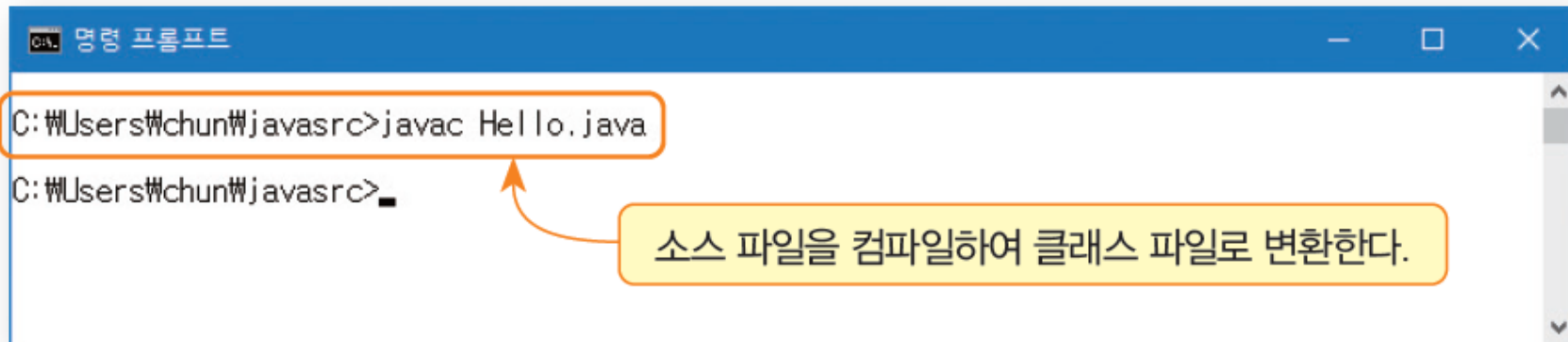
- 메모장 사용해서 **클래스 Hello** 소스코드 작성



- **Hello.java** 이름으로 파일 저장

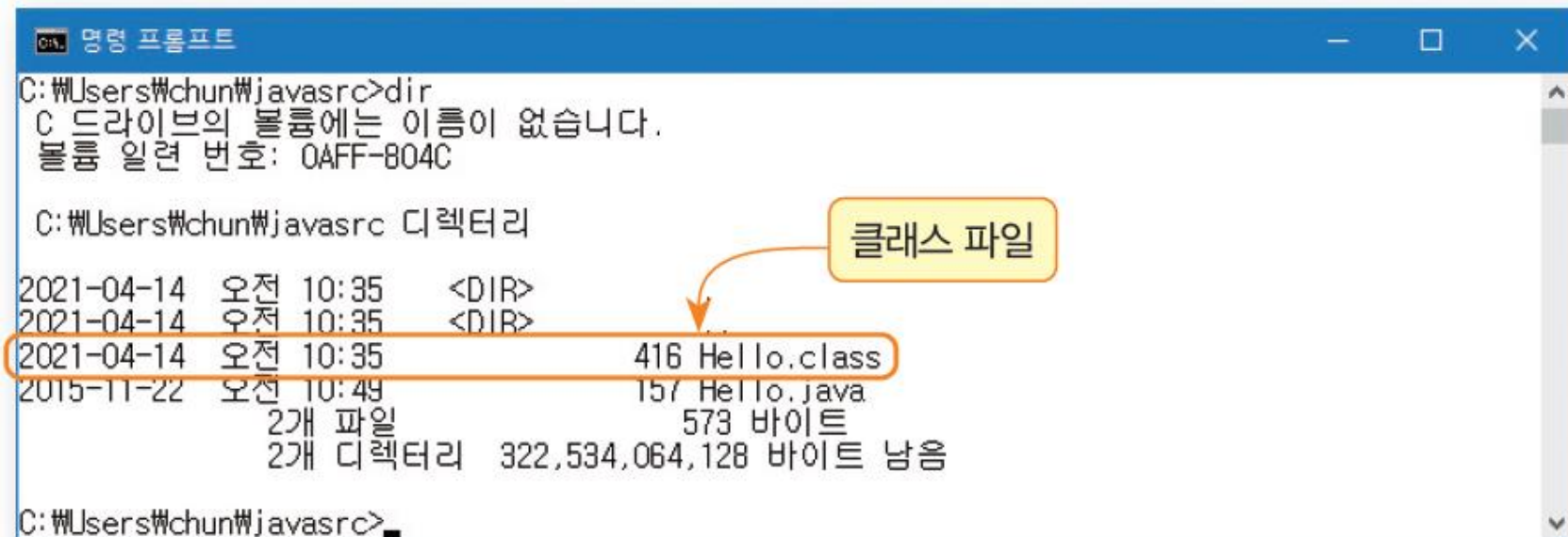
컴파일

- 컴파일러의 명령어 버전 이름은 **javac**이다.



```
명령 프롬프트
C:\Users\chun\javasrc>javac Hello.java
C:\Users\chun\javasrc>_
```

소스 파일을 컴파일하여 클래스 파일로 변환한다.



```
명령 프롬프트
C:\Users\chun\javasrc>dir
C 드라이브의 볼륨에는 이름이 없습니다.
볼륨 일련 번호: 0AFF-804C

C:\Users\chun\javasrc 디렉터리

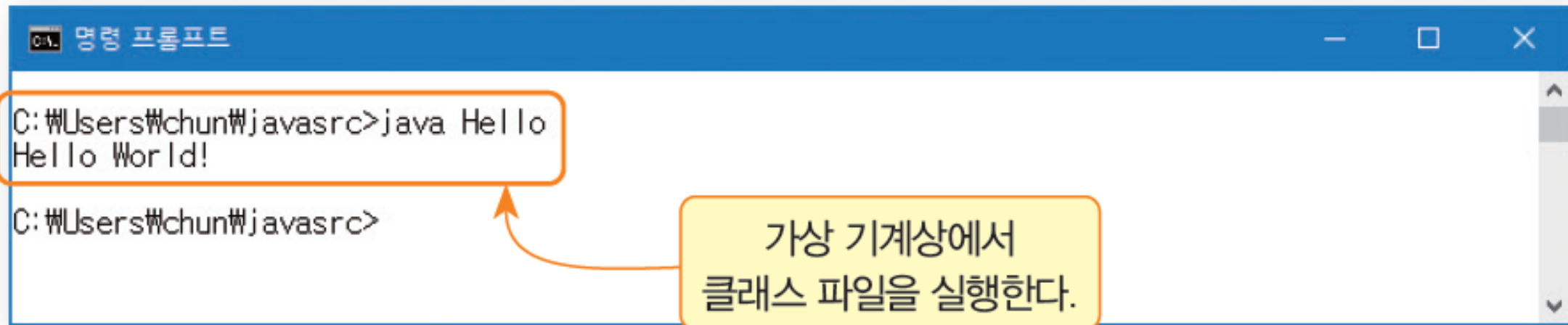
2021-04-14 오전 10:35 <DIR>
2021-04-14 오전 10:35 <DIR>
2021-04-14 오전 10:35 416 Hello.class
2015-11-22 오전 10:49 157 Hello.java
                    2개 파일 573 바이트
                    2개 디렉터리 322,534,064,128 바이트 남음

C:\Users\chun\javasrc>_
```

클래스 파일

실행

- 실행 명령어 **java**를 이용하여 클래스 파일을 실행



The screenshot shows a Windows Command Prompt window titled "명령 프롬프트". The command prompt displays the following text:

```
C:\Users\chun\javasrc>java Hello  
Hello World!  
C:\Users\chun\javasrc>
```

An orange box highlights the command `java Hello` and its output `Hello World!`. A yellow box with the text "가상 기계상에서 클래스 파일을 실행한다." (Execute the class file on the virtual machine) has an arrow pointing to the command `java`.

Hello World 프로그램의 간단한 설명

Hello.java

클래스를 정의하는 문장이다.

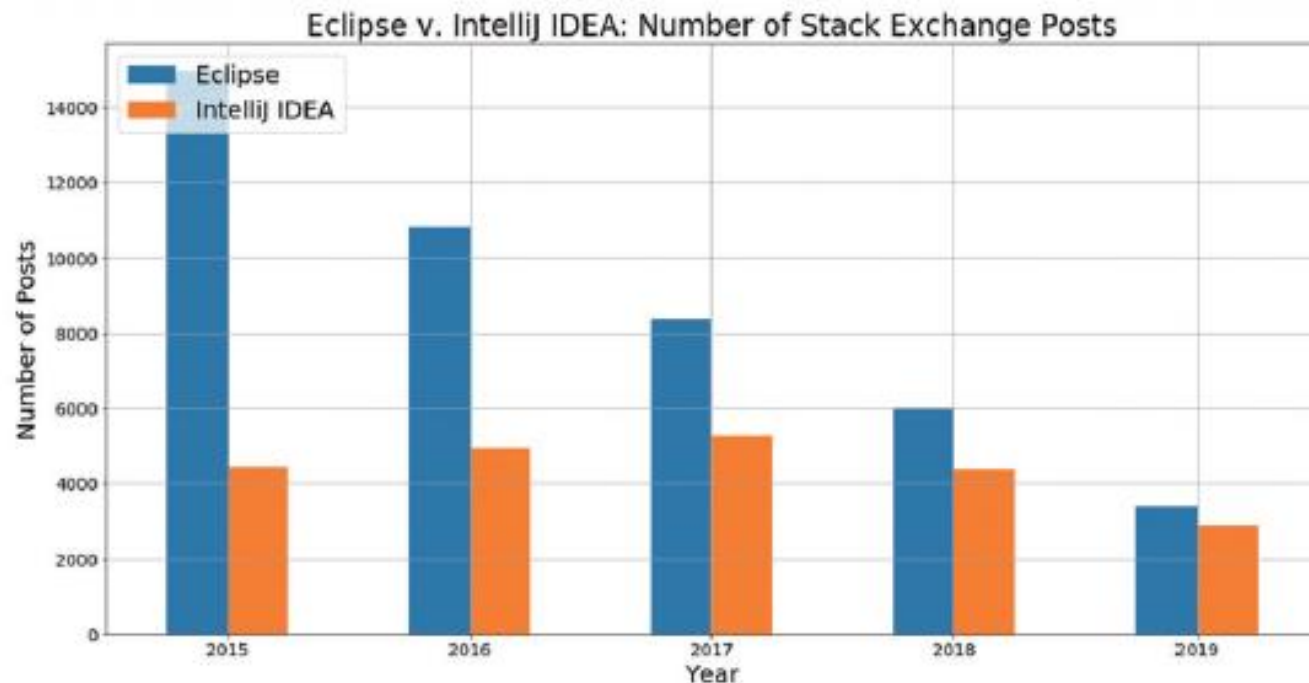
```
01 ① public class Hello {  
02 ②  
03 ③ public static void main(String[] args) {  
04 ④     System.out.println("Hello World!");  
05 ⑤ }  
06 ⑥ }
```

Hello World!

메소드를 정의하는 문장이다.

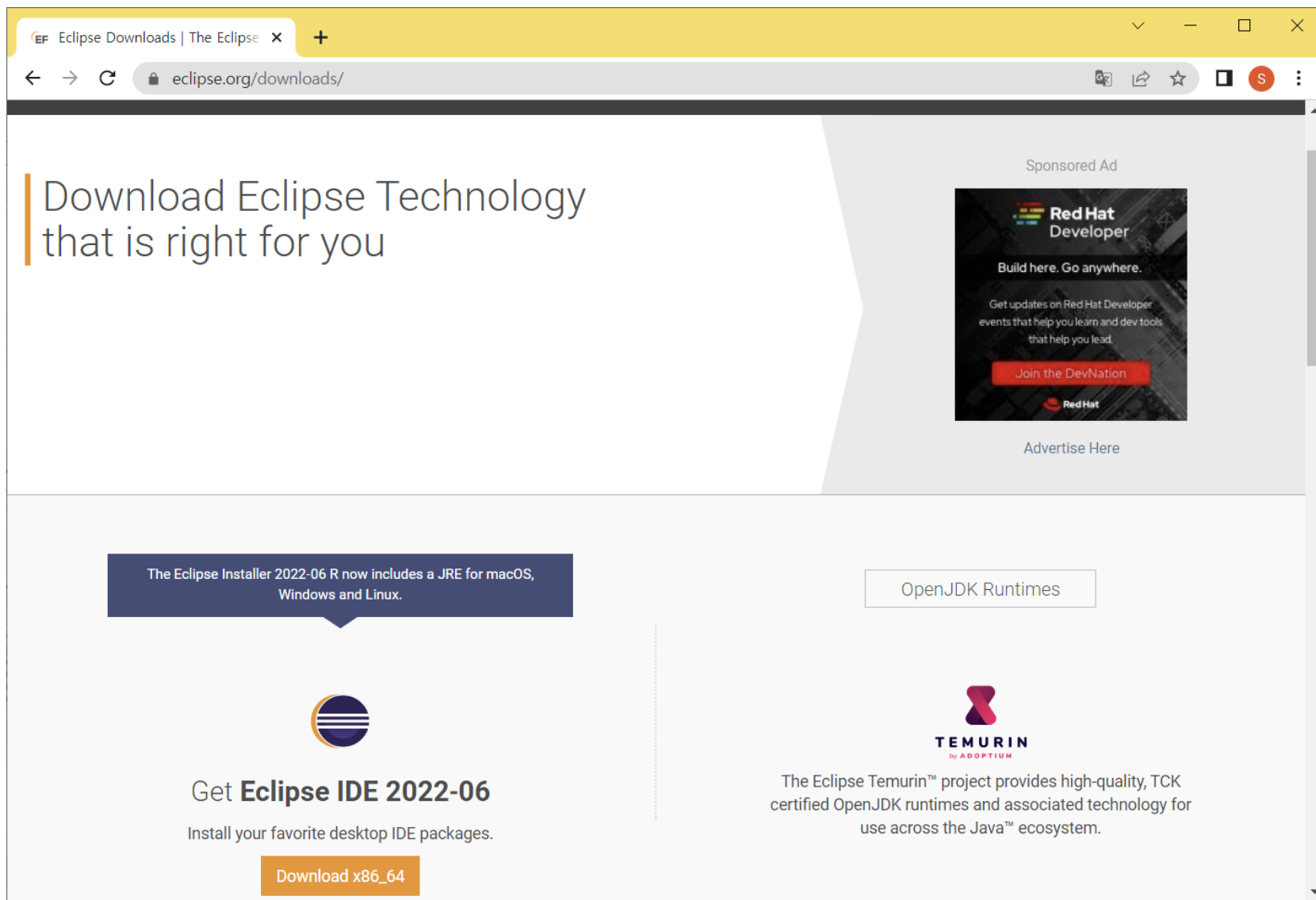
이클립스 소개와 설치

- 개발자들은 다음의 2가지 통합 개발 환경(IDE) 중에서 하나를 선택

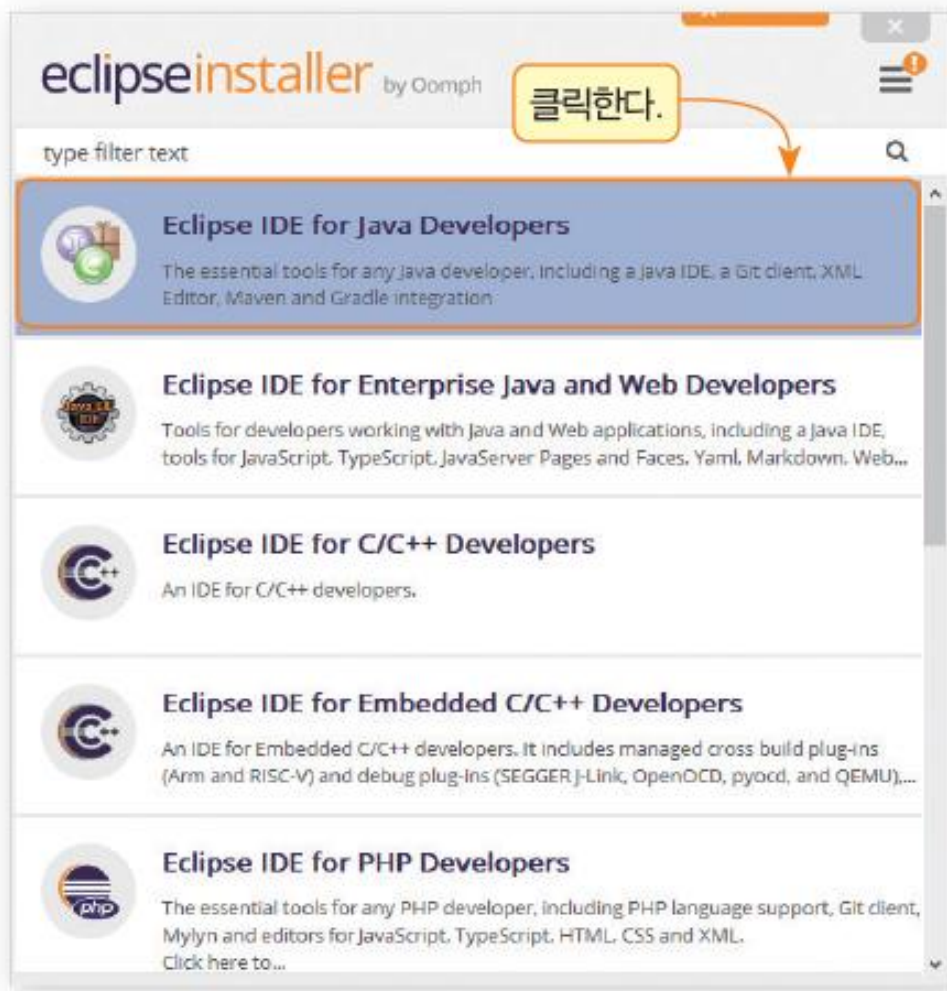


이클립스의 설치

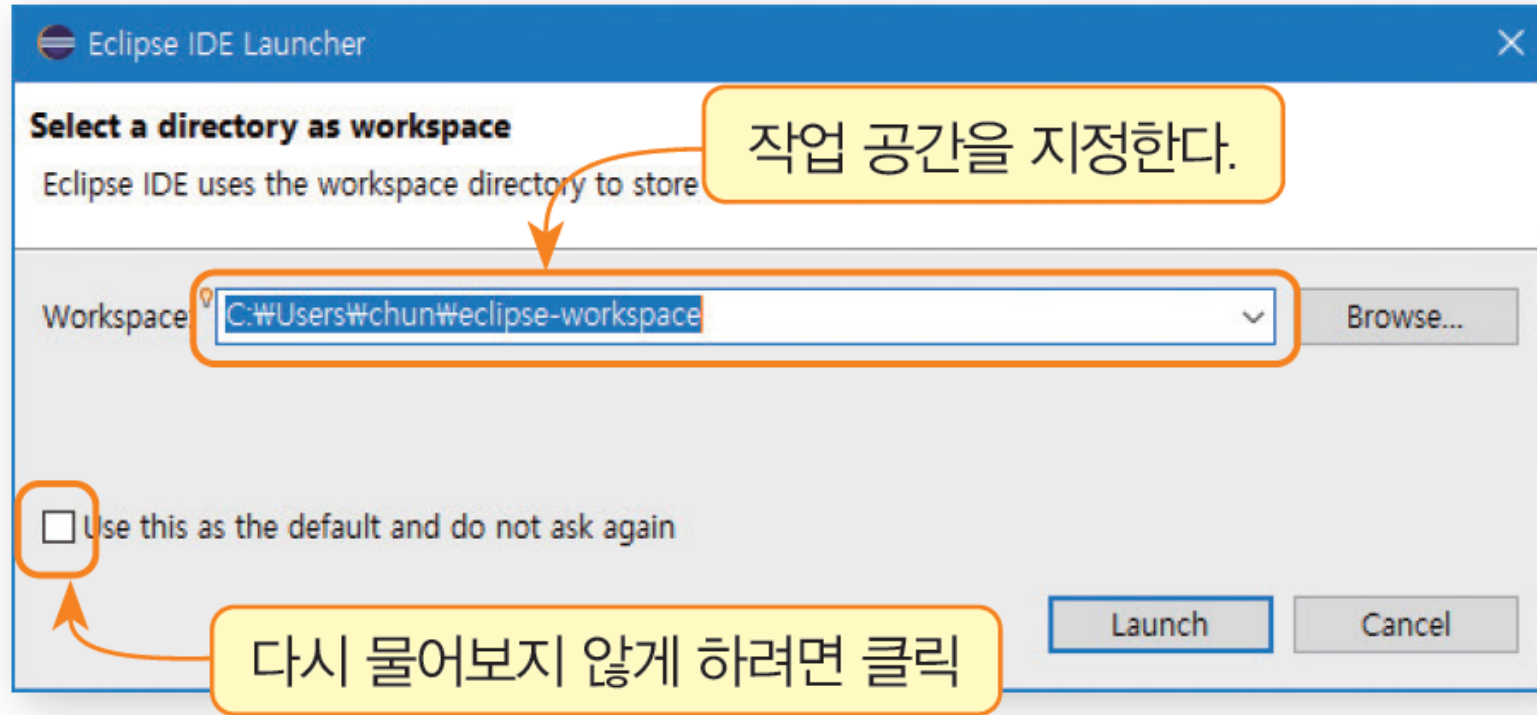
- www.eclipse.org



이클립스의 설치



이클립스 실행

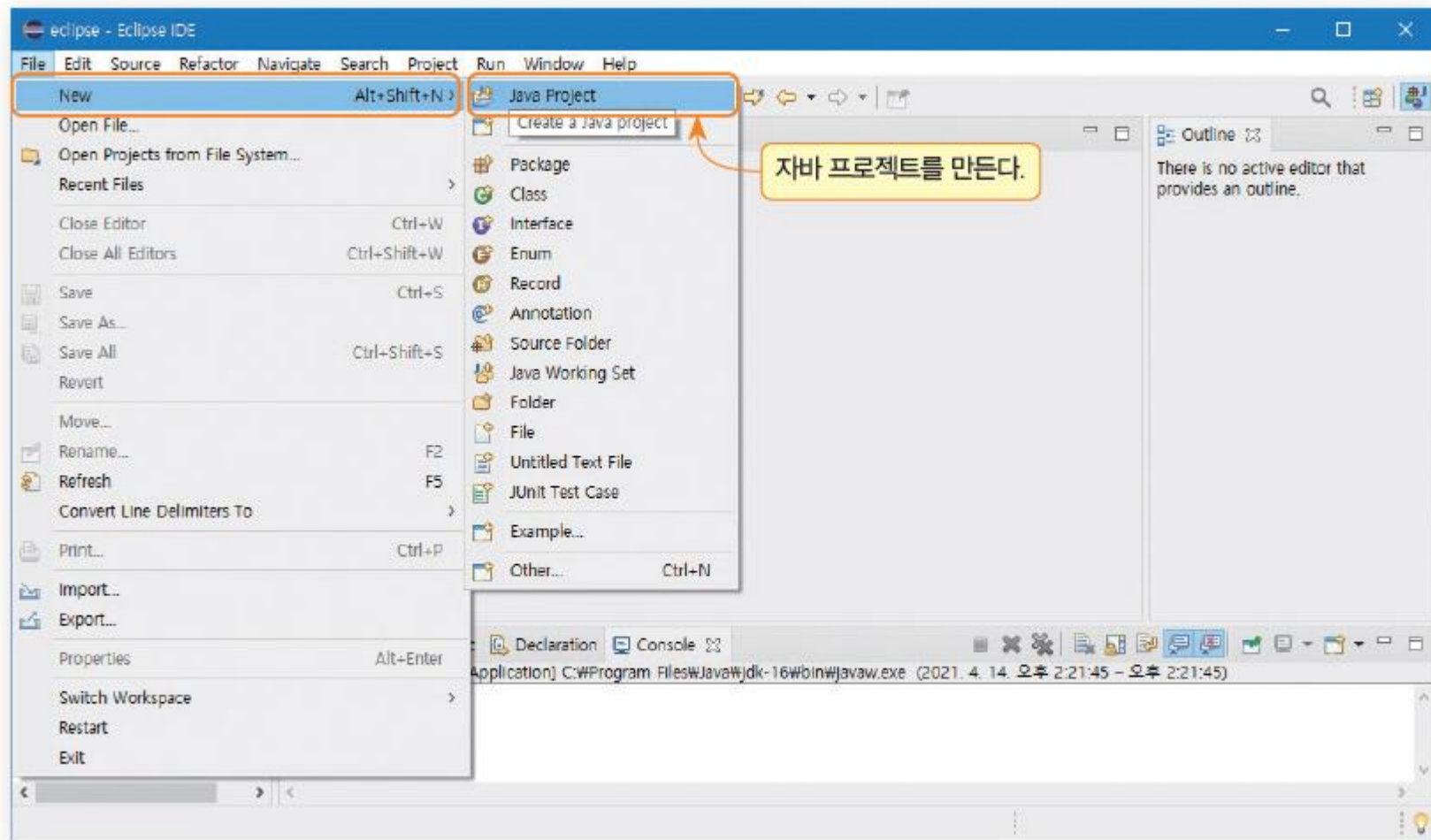


- 시작할 때 작업공간을 다시 물어보게 하려면
 - [General] - [Startup and Shutdown] - [Workspaces]

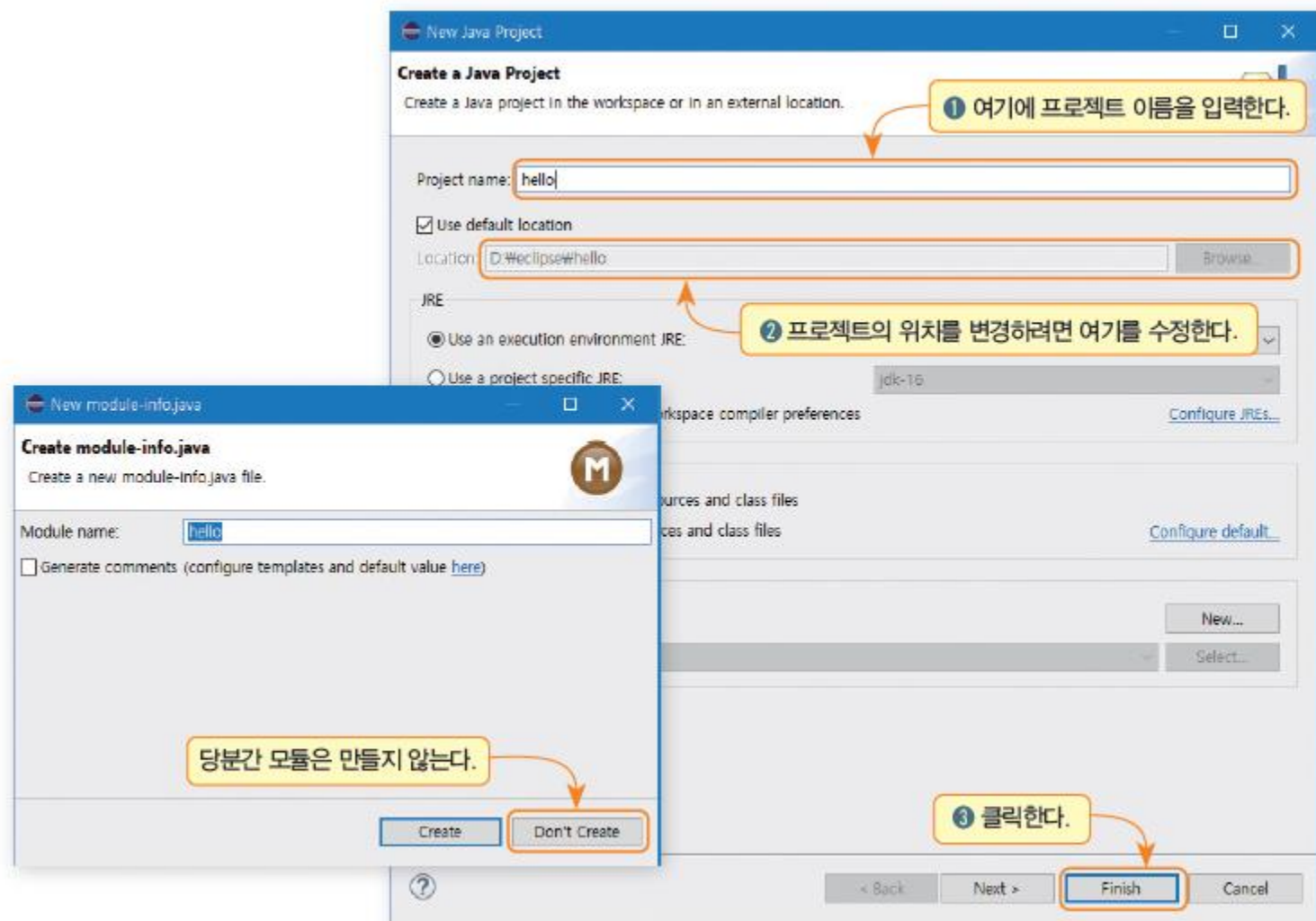
Workspaces

☒ Prompt for workspace on startup

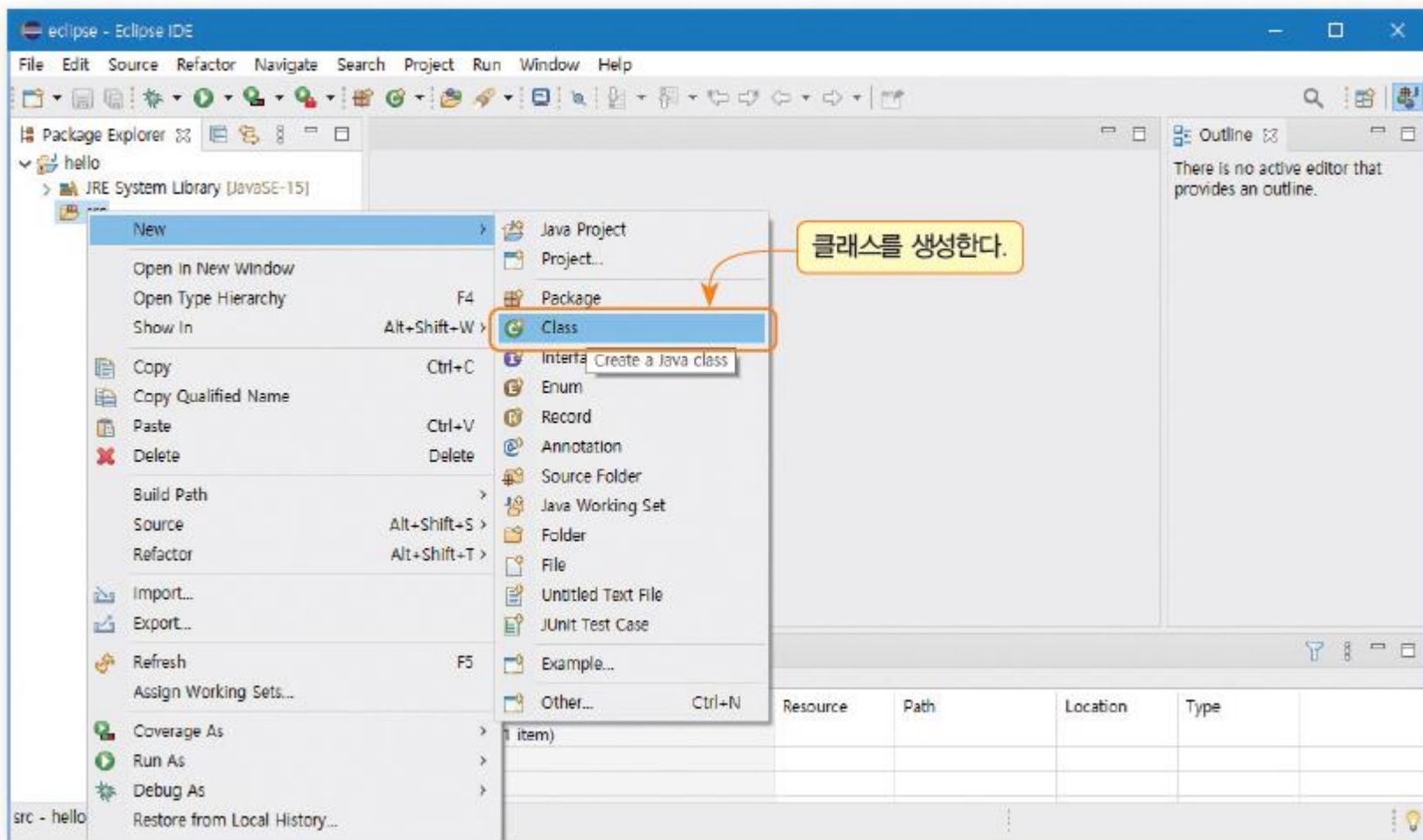
프로젝트 생성



프로젝트 생성



클래스 생성



클래스 생성

New Java Class

Create a new Java class.

Source folder: Browse...

Package: Browse...

☐ Enclosing type: Browse...

Name:

Modifiers: ☒ public ☐ package ☐ private ☐ protected
☐ abstract ☐ final ☐ static

Superclass: Browse...

Interfaces: Add... Remove

Which method stubs would you 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

Finish Cancel

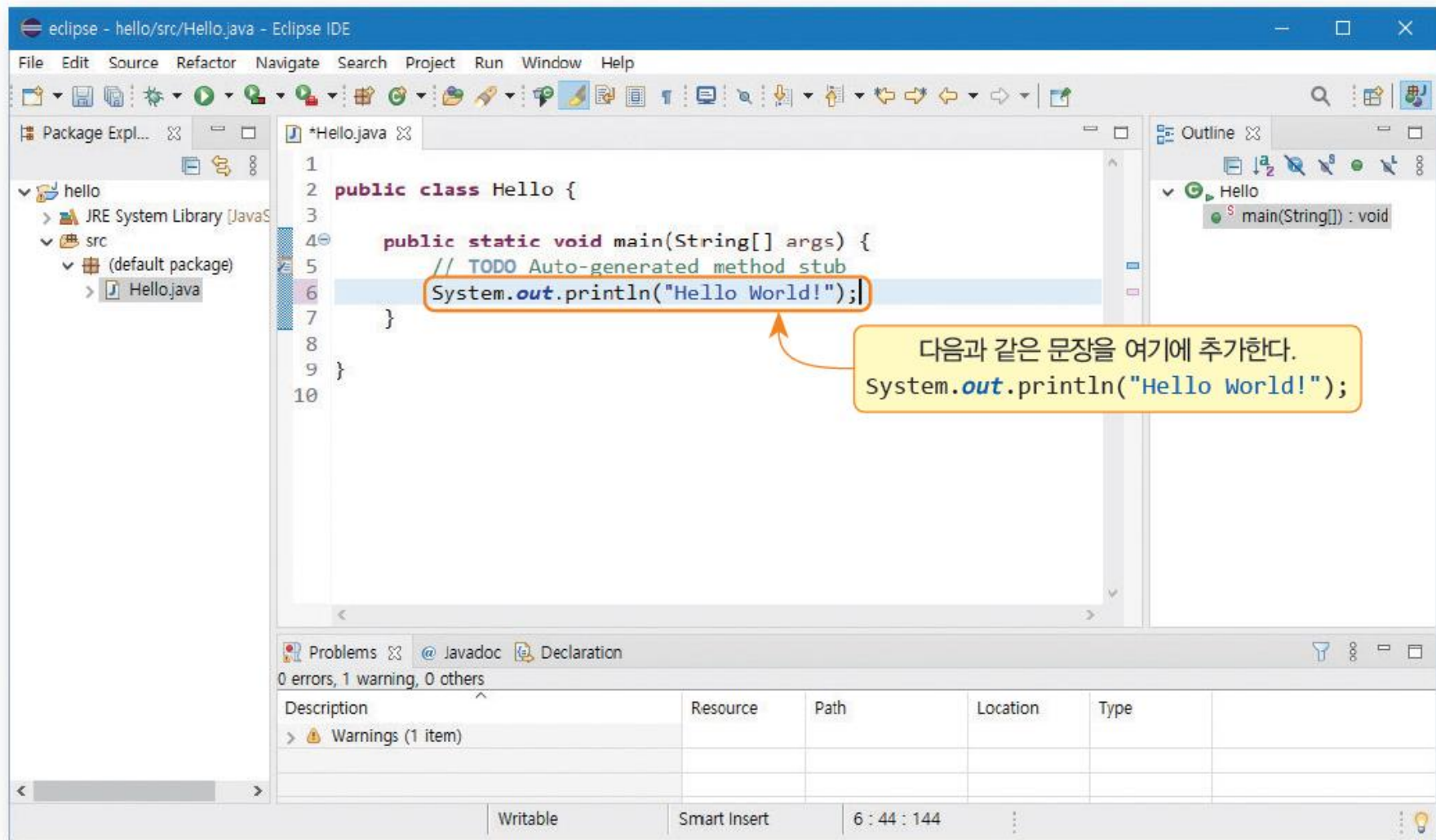
① 제일 처음에 패키지(package) 이름을 입력하는 칸이 있는데 자동으로 입력되는 패키지 이름을 지우도록 하자. 당분간은 패키지를 사용하지 않는 편이 편하다.

② 클래스의 이름은 Hello로 입력한다. 대소문자를 구별하므로 조심하여 입력하기 바란다.

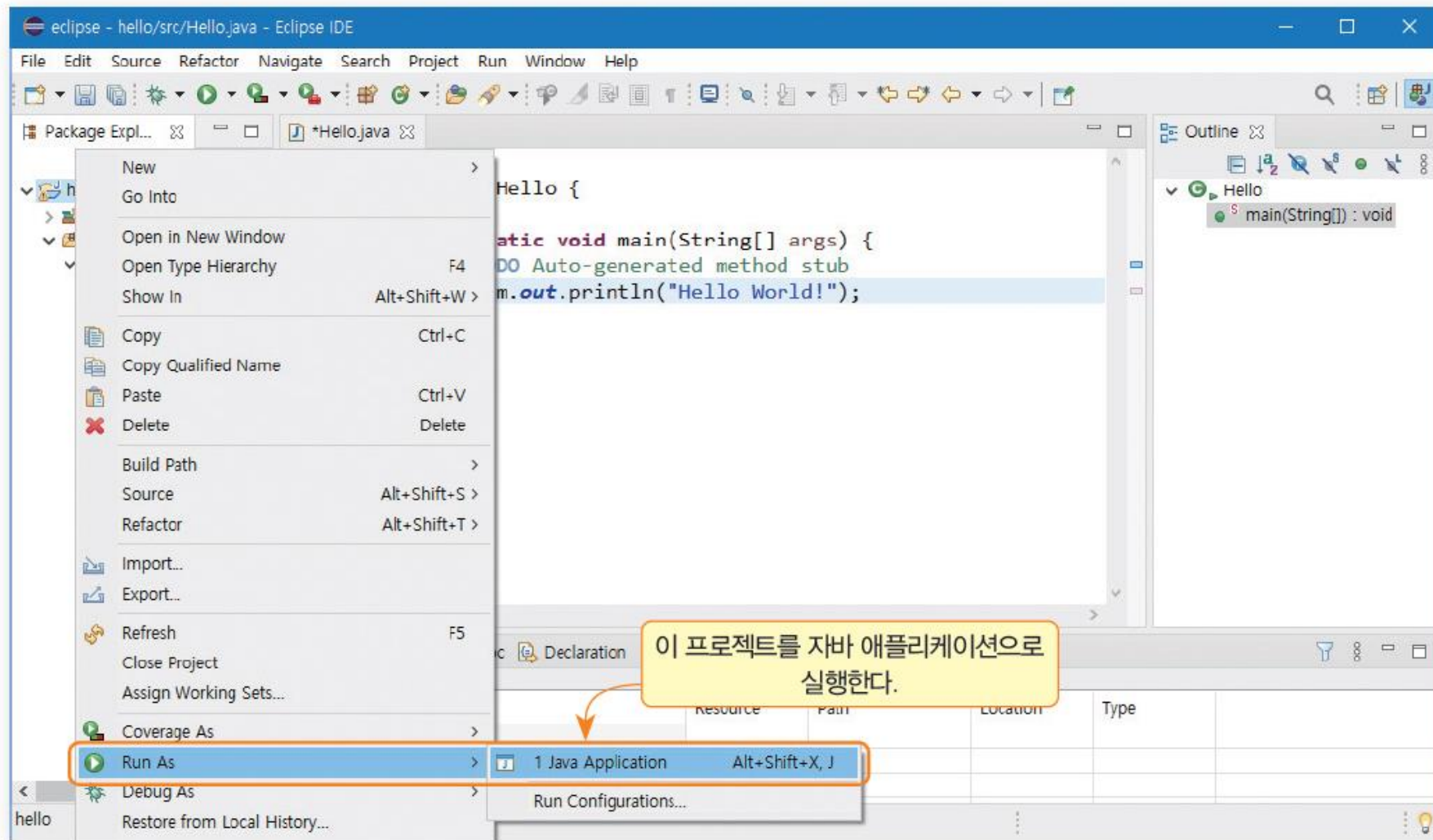
③ 체크박스 중에서 "public static void main(String[] args)"를 체크한다. 이것은 클래스 안에 main()이라는 메소드를 자동으로 생성하라는 의미이다. 자바 프로그램에는 main() 메소드가 반드시 필요하다.

④ [Finish] 버튼을 누르면 Hello.java 파일이 생성되고 에디터를 통하여 편집할 수 있는 상태가 된다.

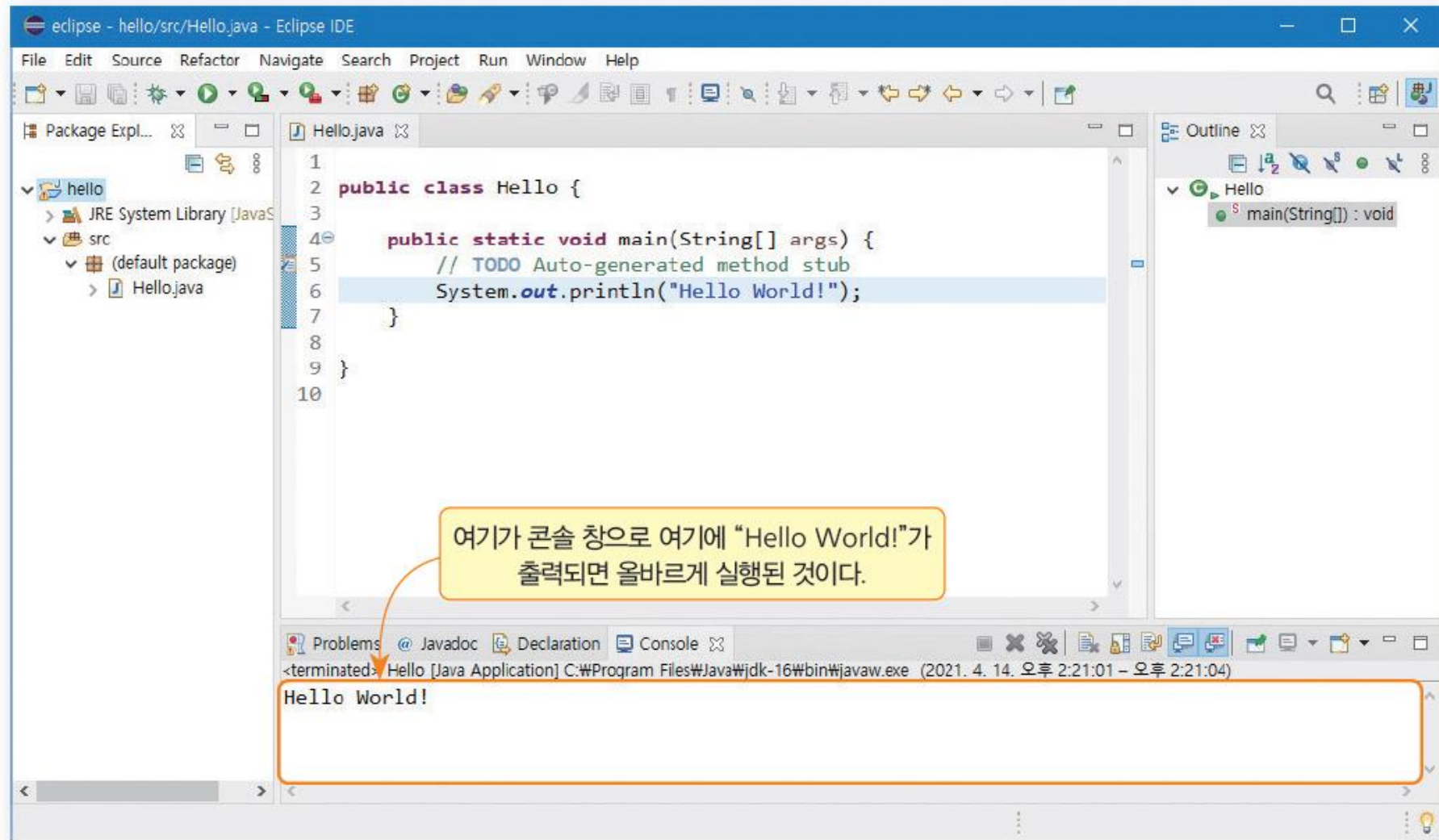
소스 코드 입력



프로그램 실행



프로그램 출력

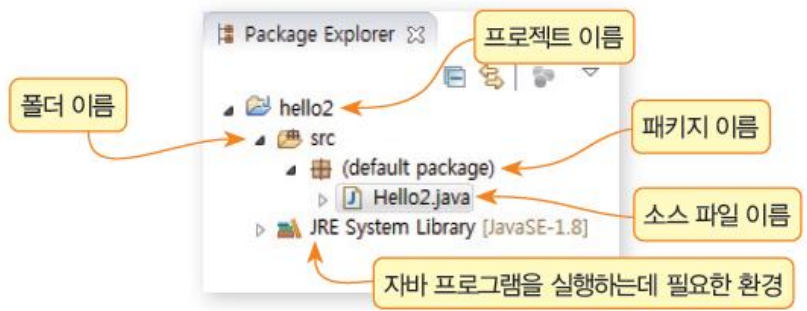
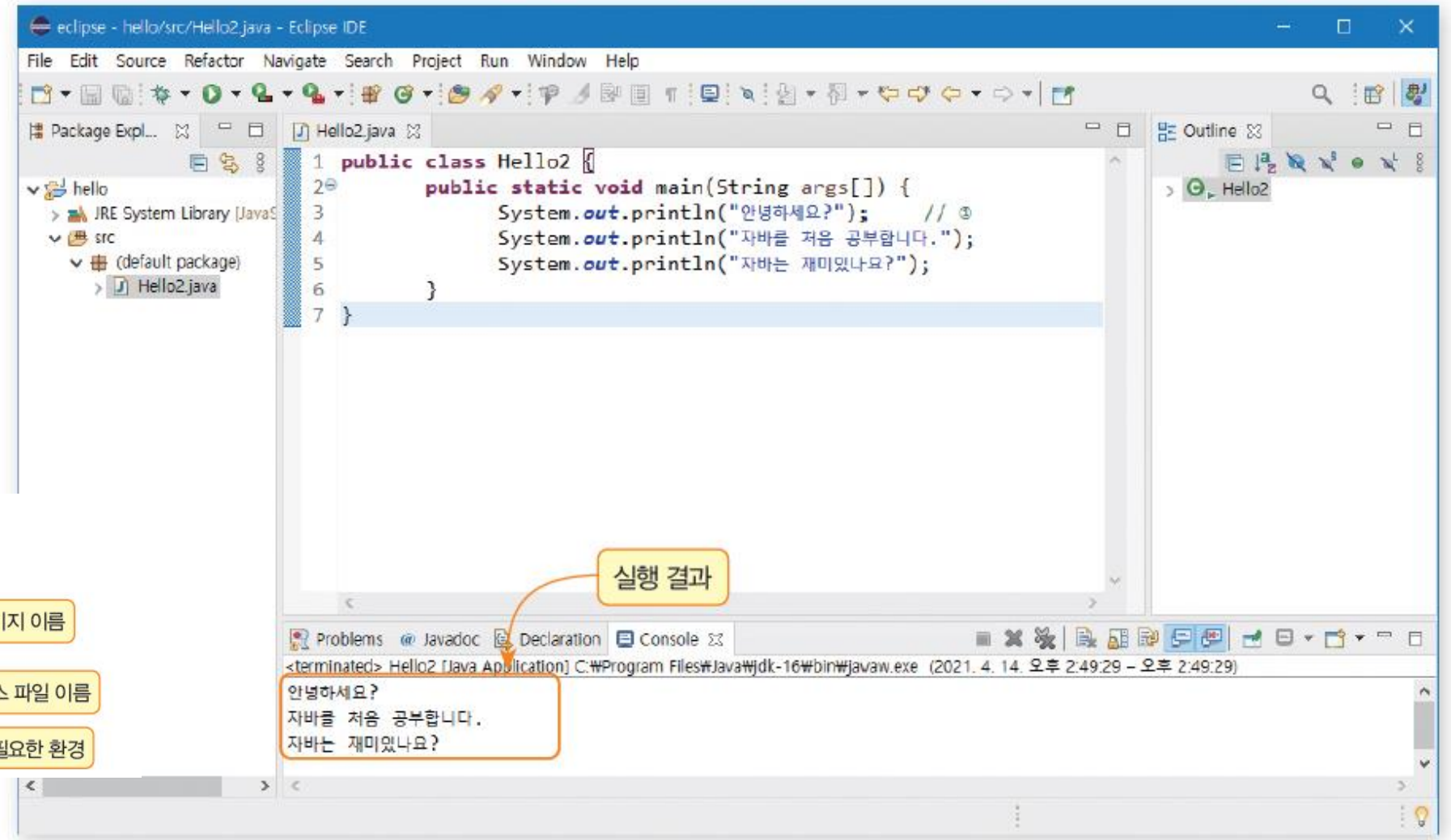
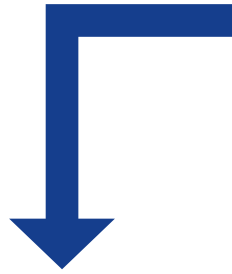


Lab: 이클립스 사용

- 다음 프로그램을 이클립스를 이용하여 컴파일하고 실행하여 보자

```
public class Hello2 {  
    public static void main(String args[]) {  
        System.out.println("안녕하세요?");           // ❶  
        System.out.println("자바를 처음 공부합니다.");  
        System.out.println("자바는 재미있나요?");  
    }  
}
```


Sol.

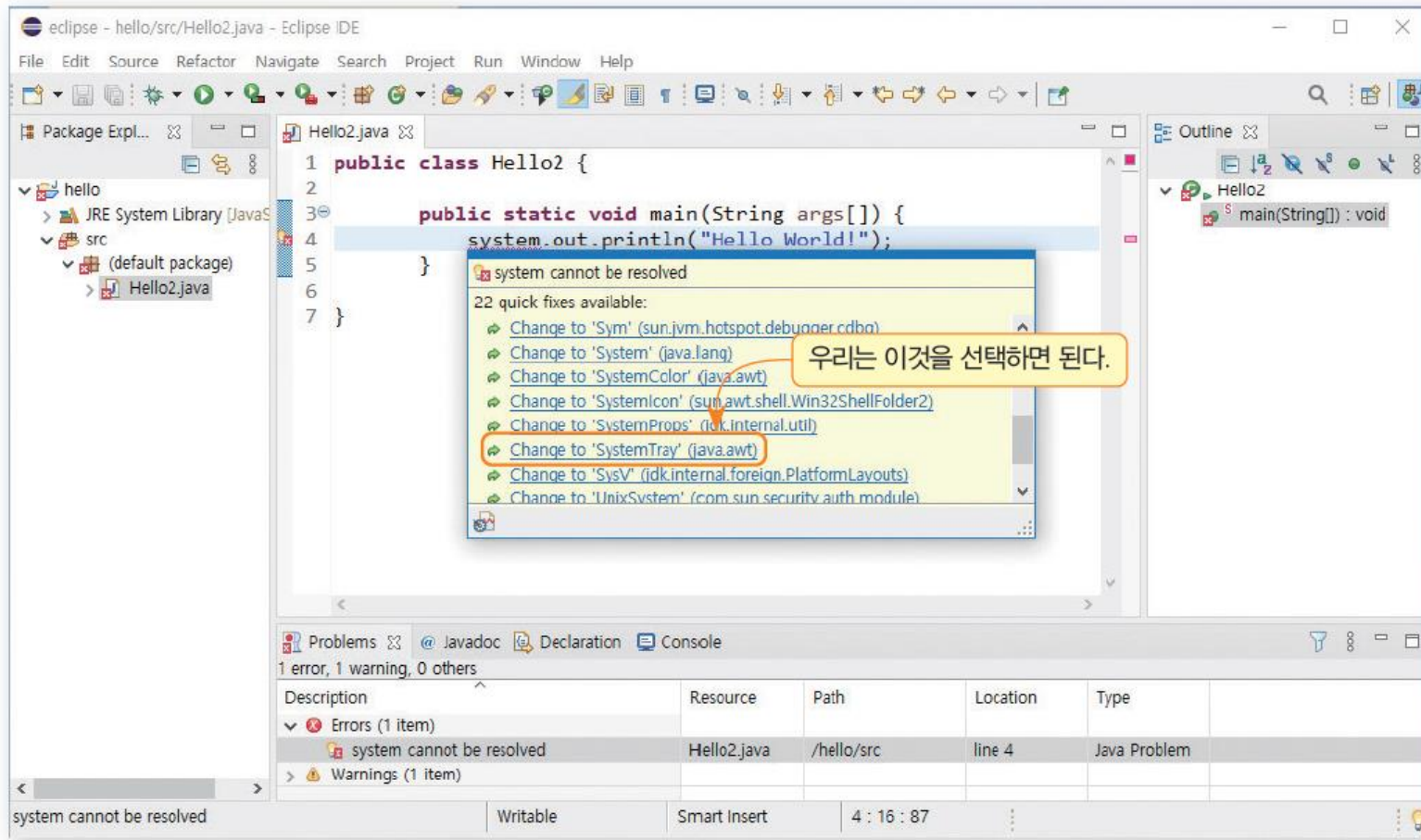


실행 결과

안녕하세요?
자바를 처음 공부합니다.
자바는 재미있나요?

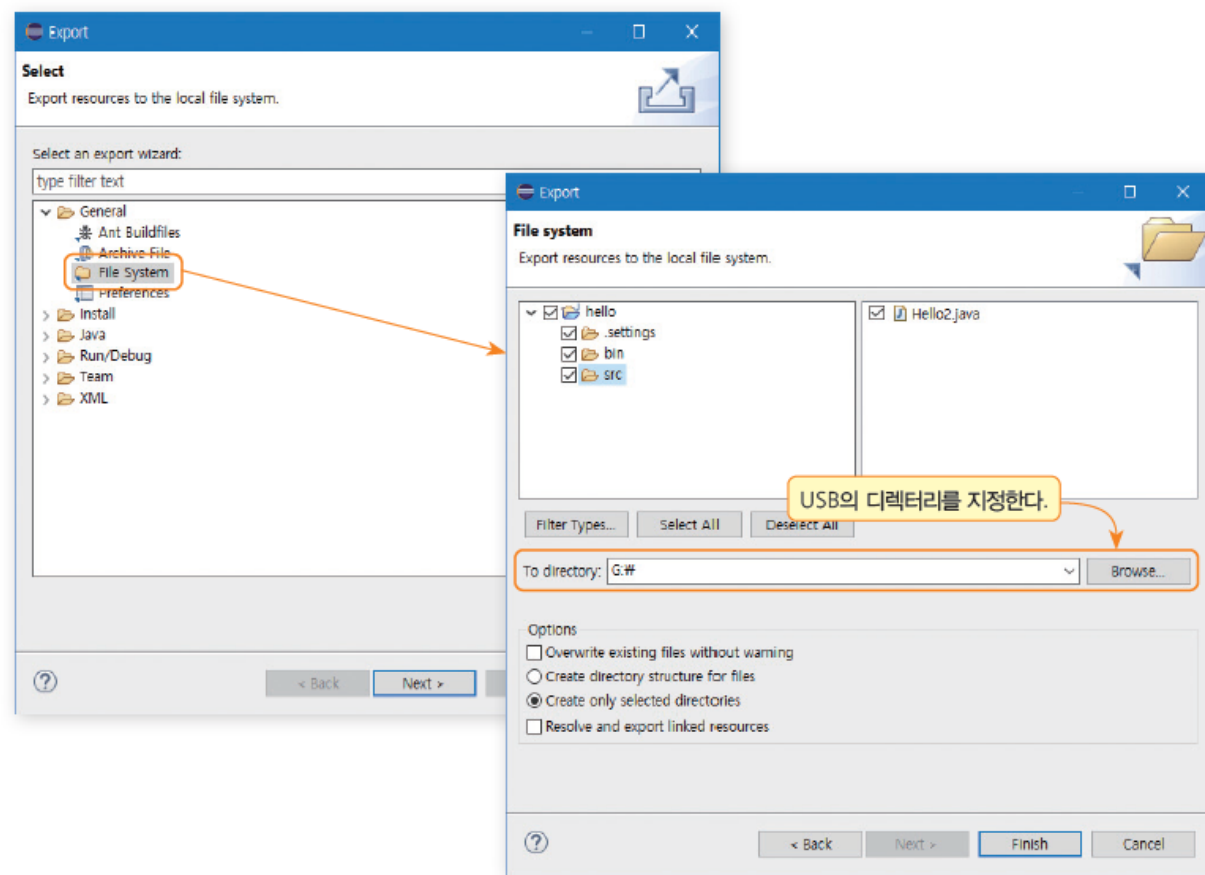
컴파일 오류

■ 이클립스의 Quick Fix 기능



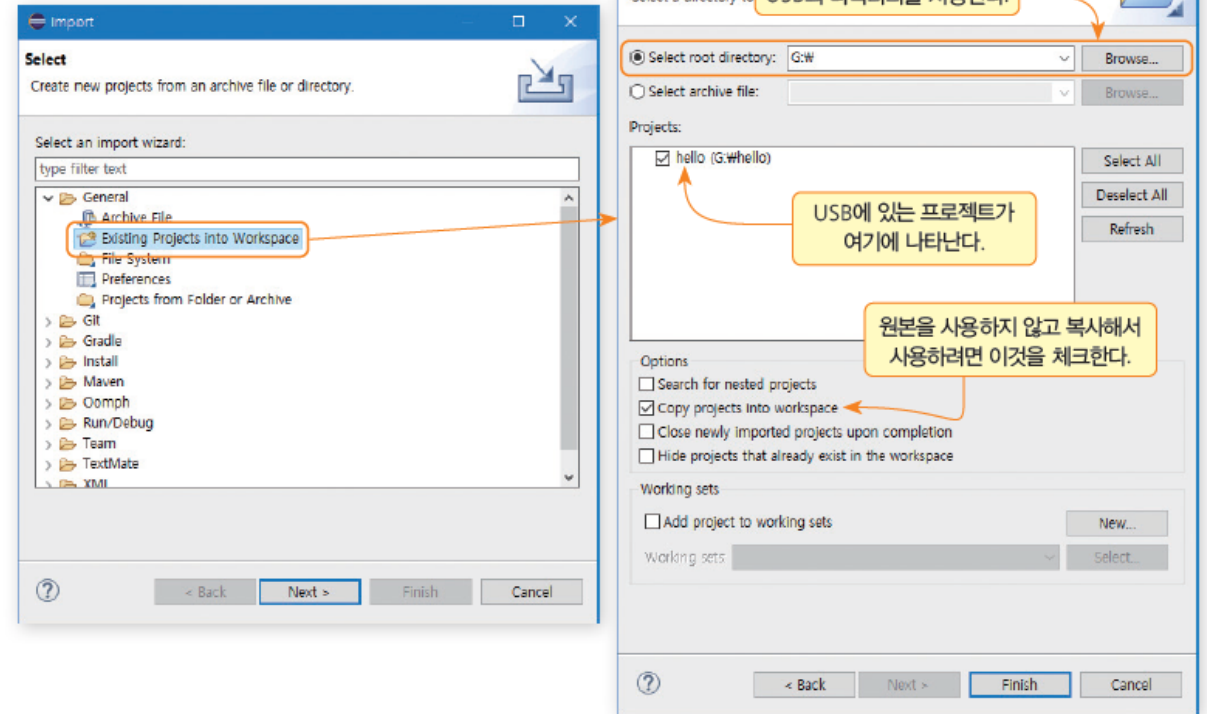
프로젝트를 외부로 내보낼 때

- 앞에서 작성한 hello 프로젝트를 USB에 저장하려면 hello 프로젝트를 선택한 상태에서 [File]-> [Export] 메뉴를 선택한다.
- 대화 상자에서 [General] -> [File System]을 선택한다.
- 대화 상자에서 USB의 디렉터리를 선택하고 [Finish]를 누른다.



외부 프로젝트를 읽을 때

- USB에 저장한 hello 프로젝트를 읽으려면 [File] -> [Import] 메뉴를 선택한다.
- 대화 상자에서 [General] -> [Existing Projects into Workspace]를 선택한다.
- 대화 상자에서 USB의 디렉터리를 선택하고 「Finish」로 나간다



Java API 문서

The screenshot shows the Oracle Java SE 17 & JDK 17 API Specification Overview page. The browser address bar shows the URL `docs.oracle.com/en/java/javase/17/docs/api/index.html`. The page has a navigation bar with tabs: OVERVIEW, MODULE, PACKAGE, CLASS, USE, TREE, PREVIEW, NEW, DEPRECATED, INDEX, and HELP. The current tab is OVERVIEW. Below the navigation bar is a search bar labeled "SEARCH:". The main content area has the title "Java® Platform, Standard Edition & Java Development Kit Version 17 API Specification". Below the title, it states: "This document is divided into two sections: Java SE and JDK". Under "Java SE", it says: "The Java Platform, Standard Edition (Java SE) APIs define the core Java platform for general-purpose computing. These APIs are in modules whose names start with `java`." Under "JDK", it says: "The Java Development Kit (JDK) APIs are specific to the JDK and will not necessarily be available in all implementations of the Java SE Platform. These APIs are in modules whose names start with `jdk`." Below this text is a table with four tabs: All Modules, Java SE, JDK, and Other Modules. The "All Modules" tab is selected. The table has two columns: Module and Description. The rows list modules: `java.base`, `java.compiler`, `java.datatransfer`, `java.desktop`, `java.instrument`, `java.logging`, `java.management`, and `java.management.rmi`.

Overview (Java SE 17 & JDK 17) x +

docs.oracle.com/en/java/javase/17/docs/api/index.html

OVERVIEW MODULE PACKAGE CLASS USE TREE PREVIEW NEW DEPRECATED INDEX HELP Java SE 17 & JDK 17

SEARCH:

Java® Platform, Standard Edition & Java Development Kit Version 17 API Specification

This document is divided into two sections:

Java SE

The Java Platform, Standard Edition (Java SE) APIs define the core Java platform for general-purpose computing. These APIs are in modules whose names start with `java`.

JDK


The Java Development Kit (JDK) APIs are specific to the JDK and will not necessarily be available in all implementations of the Java SE Platform. These APIs are in modules whose names start with `jdk`.

All Modules	Java SE	JDK	Other Modules
Module	Description		
<code>java.base</code>	Defines the foundational APIs of the Java SE Platform.		
<code>java.compiler</code>	Defines the Language Model, Annotation Processing, and Java Compiler APIs.		
<code>java.datatransfer</code>	Defines the API for transferring data between and within applications.		
<code>java.desktop</code>	Defines the AWT and Swing user interface toolkits, plus APIs for accessibility, audio, imaging, printing, and JavaBeans.		
<code>java.instrument</code>	Defines services that allow agents to instrument programs running on the JVM.		
<code>java.logging</code>	Defines the Java Logging API.		
<code>java.management</code>	Defines the Java Management Extensions (JMX) API.		
<code>java.management.rmi</code>	Defines the RMI connector for the Java Management Extensions (JMX) Remote API.		

자바 튜토리얼

The Java™ Tutorials

docs.oracle.com/javase/tutorial/



The Java™ Tutorials

The Java Tutorials have been written for JDK 8. Examples and practices described in this page don't take advantage of improvements introduced in later releases and might use technology no longer available.

See [Java Language Changes](#) for a summary of updated language features in Java SE 9 and subsequent releases.


See [JDK Release Notes](#) for information about new features, enhancements, and removed or deprecated options for all JDK releases.

The Java Tutorials are practical guides for programmers who want to use the Java programming language to create applications. They include hundreds of complete, working examples, and dozens of lessons. Groups of related lessons are organized into "trails".

Trails Covering the Basics

These trails are available in book form as *The Java Tutorial, Sixth Edition*. To buy this book, refer to the box to the right.

- » [Getting Started](#) — An introduction to Java technology and lessons on installing Java development software and using it to create a simple program.
- » [Learning the Java Language](#) — Lessons describing the essential concepts and features of the Java Programming Language.
- » [Essential Java Classes](#) — Lessons on exceptions, basic input/output, concurrency, regular expressions, and the platform environment.
- » [Collections](#) — Lessons on using and extending the Java Collections Framework.
- » [Date-Time APIs](#) — How to use the `java.time` packages to write date and time code.
- » [Deployment](#) — How to package applications and applets using JAR files, and deploy them using Java Web Start and Java Plugin.



Not sure where to start?
See [Learning Paths](#)

Tutorial Contents

Really Big Index

Tutorial Resources

- ▶ Last Updated 2022-03-04
- ▶ [The Java Tutorials' Blog](#) has news and updates about the Java SE tutorials.
- ▶ [Download the latest Java Tutorials bundle.](#)

쿠키 환경설정 | Ad Choices

Mini Project: 사칙 연산 프로그램

- 다음과 같이 피연산자 20과 10을 가지고 사칙 연산한 결과를 출력하는 자바 프로그램을 JDK와 이클립스를 이용하여 컴파일하고 실행하여 보자.

- 실행 결과

```
30  
10  
200  
2
```

- 코드 예

```
System.out.println(10+20);
```