

JAVA

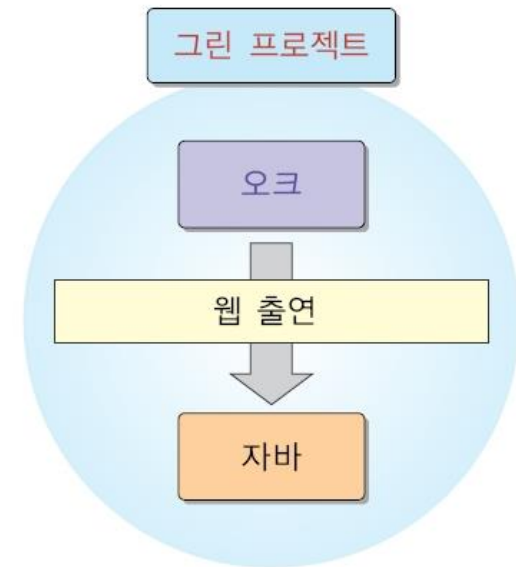
- Overview

JAVA 소개

자바의 소개

• 자바의 탄생

- 썬 마이크로시스템즈사에서 개발
- 인터넷이 확산됨에 따라 기존의 객체지향언어인 오크(OAK)가 **자바**라는 이름으로 새롭게 탄생함



[그림 1-1] 객체지향 언어 오크와 자바



여기서 잠깐

자바의 아버지
제임스 고슬링



제임스 고슬링(James Gosling)은 1956년생으로, 썬의 핵심 개발자이자 그린 프로젝트의 총괄자로 자바를 탄생시킨 장본인이다. 지금은 썬에서 CTO 겸 특별연구원 직책을 맡고 있는데, 아직까지도 개발현장에서 노약장을 과시한다.

개발자 가운데서는 마이크로소프트의 빌 게이츠만큼이나 유명하지만, 개발자 특유의 '몰두', '은둔' 성향 때문에 세상에 널리 알려지지는 않았다. 자바 커피를 하루에도 10여 잔씩 마시는 자바 예찬론자이기도 해서, 자바라는 명칭을 인도네시아 섬 이름으로 유명한 커피 재배지에서 본 따 붙이기도 했다.

자바의 소개

- 자바의 탄생



Windows용 C++ 코드



Window용 실행 코드



Windows



Unix/Linux용 C++ 코드



Unix/Linux용 실행 코드



Unix/Linux



애플 Mac OS X용 C++ 코드



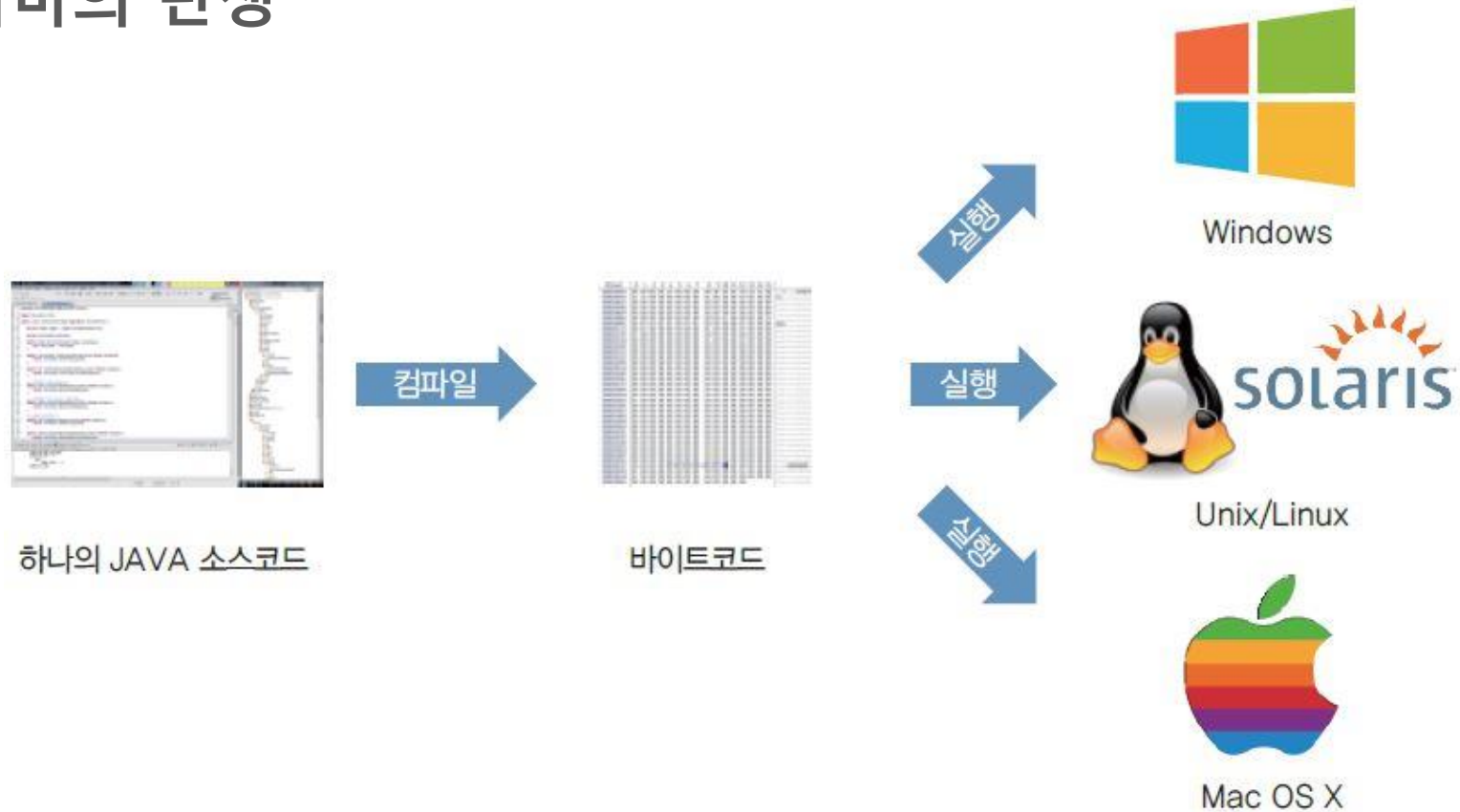
애플 Mac OS X용 실행 코드



Mac OS X

자바의 소개

• 자바의 탄생



– 자바 가상머신(JVM Java Virtual Machine) 개념

- C, C++ 등의 기존 언어는 각 운영체제 별로 서로 다른 코드를 만들고 별도로 컴파일 해야 실행 가능
- JAVA는 하나의 JAVA 코드 만들면 모든 운영체제에서 동일하게 실행 가능(경제적, 효율적)

자바의 소개

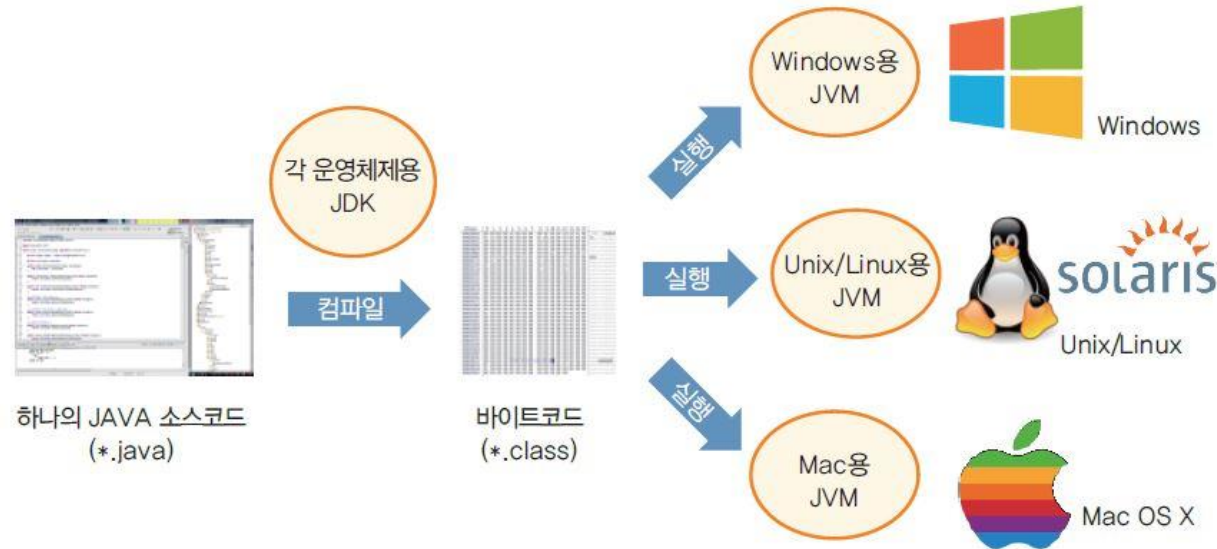
- JAVA언어의 특징

- C++의 객체지향적인 장점을 그대로 가져오면서 동시에 C++의 여러 가지 문제점 보완
- 간결한 프로그래밍 문법 제공
 - 문법이 쉽기 때문에 프로그래밍 초보자가 배우기에 적절
 - C, C++의 강력한 장점이자 문제점으로 꼽히는 포인터 삭제
 - 메모리 관리를 가비지 컬렉터(garbage collector)가 전담하여 프로그래머가 메모리에 대해 고민하지 않고 프로그래밍 로직에 집중

자바의 소개

– 이식성이 매우 좋다

- JAVA 소스코드는 한 번만 작성해놓으면 다양한 운영체제에서 실행 가능



- 메모장이나 이클립스(Eclipse) 등의 텍스트 에디터로 작성, 작성된 소스코드를 javac.exe로 컴파일하면 바이트코드가 생성
- 바이트코드는 모든 운영체제에서 실행이 가능하지만, 단 운영체제에 JAVA 가상 머신인 JVM(Java Virtual Machine)이 미리 설치되어 있어야 한다. JVM은 오라클에서 무료로 배포하고 있다.

자바의 소개

– 완전한 객체지향 언어이다

- 기존 C 언어의 사용하는 구조적인 프로그래밍 기법 그대로 지원,
- 현재 실무에서 사용되는 객체지향 프로그래밍 기법을 100% 지원

– 멀티스레드 프로그래밍을 지원한다

- 하나의 프로세스(process) 안에서 여러 개의 스레드(thread)가 동시에 작동되도록 프로그래밍 가능
- 동시에 작동되는 스레드를 작성하면 병렬 처리가 가능해져 복잡한 대용량 작업을 빠른 시간 내에 처리 가능
- 자체 API에서 멀티스레드 프로그래밍(multi thread programming) 지원하여 쉽게 병렬 프로그래밍

– 다양한 응용 프로그램을 작성할 수 있다

- HWP, 알집, 엑셀 등과 같이 일반 컴퓨터에서 작동하는 '데스크톱 응용 프로그램', 웹 브라우저에서 다운로드된 후 작동하는 '애플릿', 웹 서버에서 작동하는 'JSP(JAVA Server Pages)' 또는 '서블릿(servlet)', TV나 냉장고 등의 가전제품에서 작동하는 '임베디드 프로그램', 안드로이드 스마트폰이나 태블릿에서 작동하는 '모바일 앱'을 JAVA로 작성 가능
- 현재 사용되는 대부분의 IT 환경에서 작동하는 응용 프로그램 작성

자바의 소개

- 많은 오픈 소스 라이브러리가 있다
 - 외부의 다양한 오픈 소스 라이브러리(open source library) 사용 가능
 - 수많은 프로젝트에서 이러한 라이브러리를 오픈 소스로 제공하기 때문에 구현하고자 하는 고급 기능을 누구나 가져와서 사용할 수 있다. 또한 다양한 오픈 소스 프로젝트는 일회성 제공에 그치는 것이 아니라 커뮤니티를 통해 체계적인 소스 관리와 업그레이드가 이뤄지고 있다. 실무에서 JAVA 언어를 많이 사용하는 가장 큰 이유 중 하나는 이러한 오픈 소스 라이브러리를 사용하여 자신의 고급 기능을 빠른 기간 내에 안정적으로 구현할 수 있다는 것이다. 대표적인 오픈 소스 라이브러리로는 Apache Commons, Google Guava, Lucene, Spring, Hadoop, Cassandra 프로젝트 등이 있다.



자바 플랫폼과 가상머신

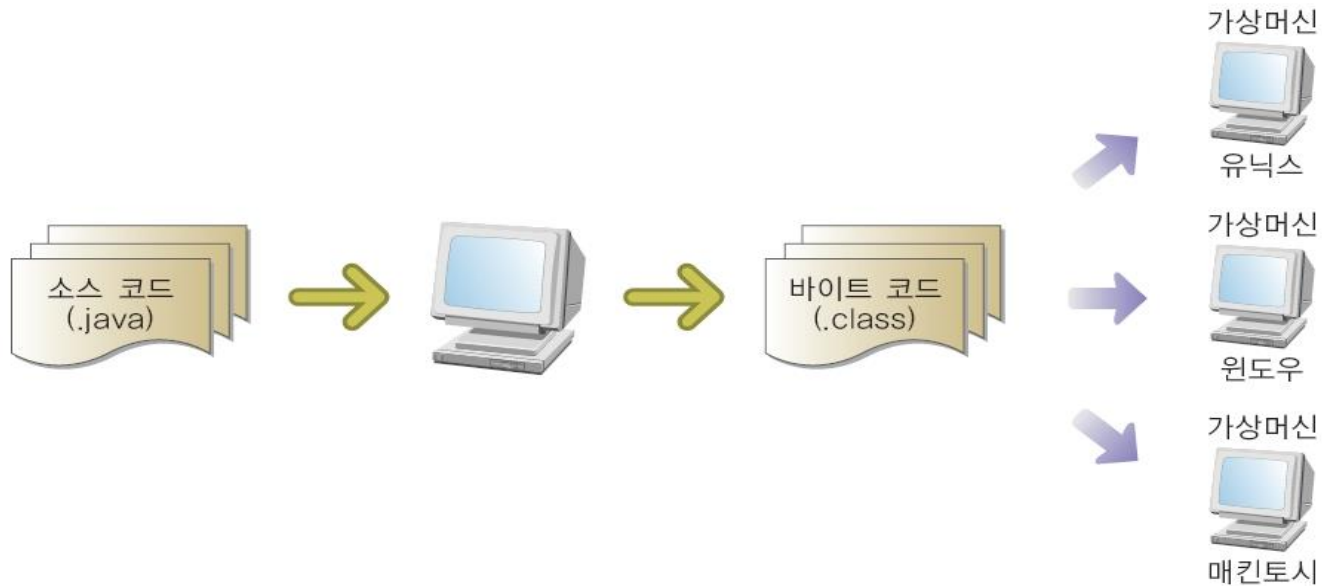
- JAVA 가상 머신

- JVM은 Windows, Unix/Linux, Mac 등 운영체제에 상관없이 한 번 작성한 JAVA 코드를 동일하게 작동시키는 핵심적인 역할
 - JVM에서 '가상 머신(=가상 컴퓨터)'은 실제로는 소프트웨어로 설치되고 작동.
 - Windows에는 Windows용 JVM을, Unix/Linux에는 Unix용 또는 Linux용 JVM을, Mac에는 Mac용 JVM을 설치해야 한다.
 - JVM은 오라클 홈페이지에서 무료로 배포, JAVA와 관련된 프로그램을 실행하면 자동으로 다운로드 후 설치되기도 한다.
 - JVM의 역할을 하는 소프트웨어를 JAVA 소프트웨어 또는 JRE(JAVA runtime environment)라고 한다.
 - JRE는 JAVA 바이트코드를 실행하기 위해 먼저 설치되어야 한다.
- JAVA 초창기에는 이처럼 JRE를 모든 컴퓨터에 설치해야 하는 것에 대해 부정적인 시각도 있었지만, 지금은 컴퓨터에 설치하는 백신, 압축 프로그램, 웹 브라우저 등과 같은 기본적인 프로그램으로 인식되고 있다. 오늘날에는 JAVA로 작성된 프로그램이 별 거부감 없이 사용

자바 플랫폼과 가상머신

• 자바 가상머신

- 실행 형태로 만들기 전 바이트 코드를 먼저 생성
- 가상머신만 있으면 자바 프로그램을 해석할 수 있다.
- 다양한 환경에서 실행가능



자바 플랫폼과 가상머신

- 자바 가상머신의 오해와 진실

- ① 자바 가상머신은 느리다?

초기에는 상당히 느린 것으로 인식되었고 실제로도 성능에 다소 문제가 있었다. 그러나 기술이 발달하면서 JIT(Just-In-Time)와 동적 컴파일 기술인 핫스팟(Hotspot) 기술을 적용해 성능상의 문제점은 거의 해결되었다. 또, 바이트 코드를 최적화하는 서드 파티(Third Party) 제품을 이용해 성능을 어느 정도 향상시킬 수 있다. 인터넷 뱅킹, 온라인 쇼핑몰 등 대규모 온라인 서비스 시스템을 자바로 구축하는 것만 봐도 자바 속도가 느리다는 말은 설득력이 떨어진다.

- ② 모든 자바 프로그램은 다른 플랫폼의 가상머신에서도 동일하게 동작한다

이론적으로는 그렇지만 모니터 크기, 입·출력 장치, 메모리, 디스크 등 물리적인 실행환경에 따라 다를 수 있다. Java ME는 표준 Java SE의 일부 기능만 수용하므로 Java SE로 개발한 프로그램은 Java ME에서 100% 실행할 수 없다. 그러나 같은 자바 플랫폼끼리는 가상머신 제조회사나 운영체제가 달라도 실행하는데 문제가 없다.

자바 플랫폼과 가상머신

- 자바 가상머신의 오해와 진실

- ③ 자바 프로그램은 PC용으로 사용하기 힘들다

자바의 가장 취약점은 애플리케이션의 사용자 인터페이스 부분인데, 기본으로 제공하는 AWT, Swing 등은 성능이나 사용의 편리성 면에서 문제점이 많다. 이 때문에 일반 PC용으로 나온 자바 프로그램은 거의 없다. 그러나 이클립스 프로젝트의 산출물 중 하나인 SWT(Simple Widget Toolkit)는 빠른 속도와 최적화된 운영체제로 이질감 없이 프로그램을 개발할 수 있다. 특히, 이클립스의 RCP(Rich Client Platform)가 확산되면서 앞으로 많은 자바 기반의 PC용 애플리케이션을 볼 수 있을 것으로 기대된다.

정리합시다

- **자바**

- 객체지향 언어인 오크(OAK)에서 시작되어 1995년 인터넷의 급속한 성장으로 인해 '자바'라는 이름으로, 인터넷에 적합한 작고 견고한 언어로서 발전을 거듭하고 있다.

- **자바 플랫폼과 가상머신**

- 다양한 환경에서 동일한 실행결과를 얻을 수 있는 것이 가장 큰 특징
- JAVA EE SE ME 등

- **자바 실행구조와 개발단계**

- 운영체제->소스코드->컴파일->바이트코드->가상머신->실행결과

- **자바의 장단점**

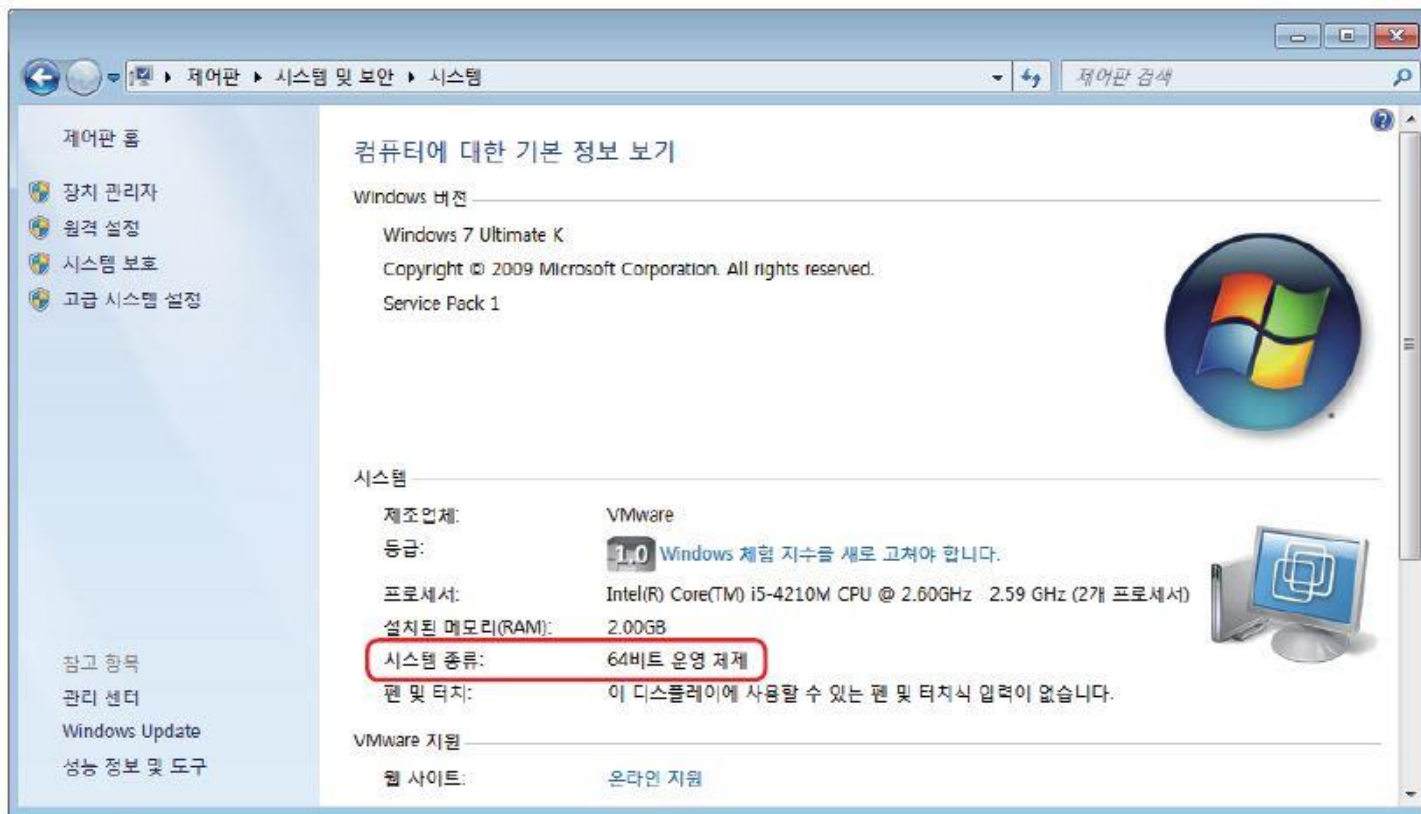
- 객체 지향 언어
- 플랫폼에 독립적
- 윈도우용 클라이언트 프로그램에 상대적으로 취약

JAVA 개발 환경 구축

최소한의 준비 : windows 플랫폼 확인

- Windows 플랫폼 확인

- ① Windows 7 : [시작]-[컴퓨터]에서 마우스 오른쪽 버튼 클릭하고 [속성] 선택
(Windows 8부터는 [시작]에서 마우스 오른쪽 버튼 클릭하고 [시스템] 선택)



최소한의 준비 : JDK 다운로드

- JDK (JAVA Development Kit)




- 자바 프로그램의 개발을 위한 소프트웨어와 라이브러리의 모음
- <https://www.oracle.com/technetwork/java/javase/downloads/index.html>에서 무료로 다운로드

Java SE 8u241

Java SE 8u241 includes important bug fixes. Oracle strongly recommends that all Java SE 8 users upgrade to this release.

- Documentation
- Installation Instructions
- Release Notes
- Oracle License
 - Binary License
 - Documentation License
 - BSD License
- Java SE Licensing Information User Manual
 - Includes Third Party Licenses
- Certified System Configurations
- Readme Files
 - JDK ReadMe
 - JRE ReadMe

Oracle JDK

-  JDK Download
-  Server JRE Download
-  JRE Download
-  Documentation Download
-  Demos and Samples Download














Which Java package do I need?

- **Software Developers: JDK** (Java SE Development Kit): For Java Developers. Includes a complete JRE plus tools for developing, debugging, and monitoring Java applications.
- **Administrators running applications on a server: Server JRE** (Server Java Runtime Environment): For deploying Java applications on servers. Includes tools for JVM monitoring and tools commonly required for server applications, but does not include browser integration (the Java plug-in), auto-update, nor an installer. [Learn more](#)
- **End user running Java on a desktop: JRE** (Java Runtime Environment): Covers most end-users needs. Contains everything required to run Java applications on your system.

Java SE 다운로드

Java SE Development Kit 8u241

This software is licensed under the [Oracle Technology Network License Agreement for Oracle Java SE](#)

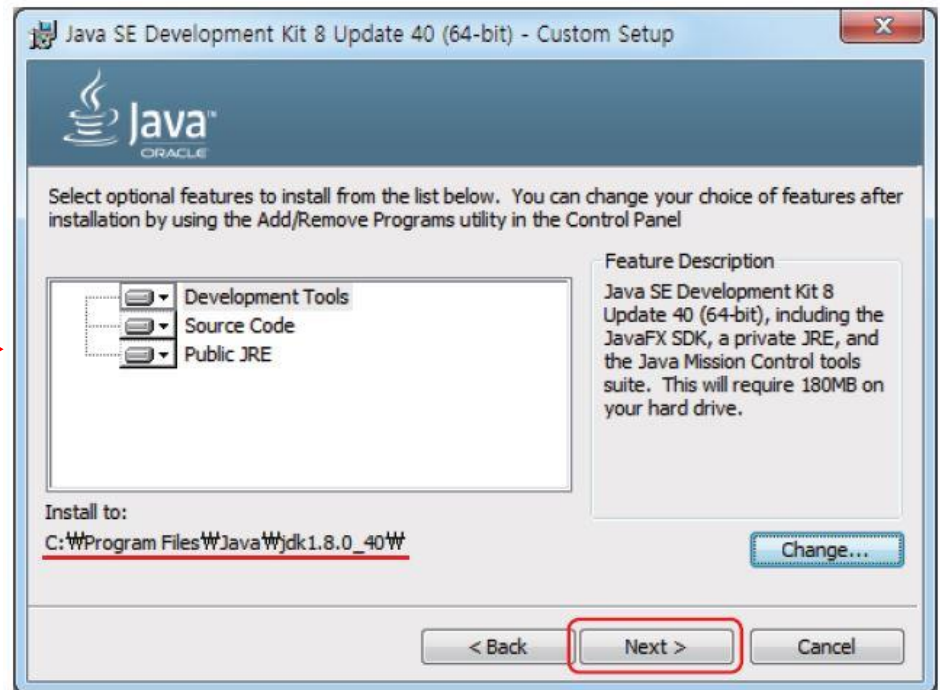
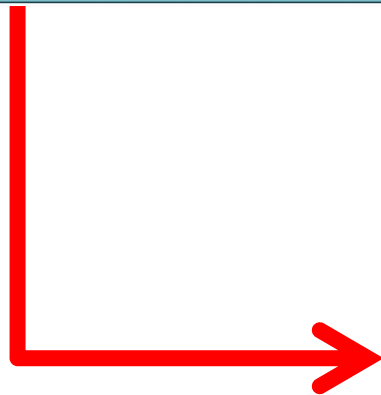
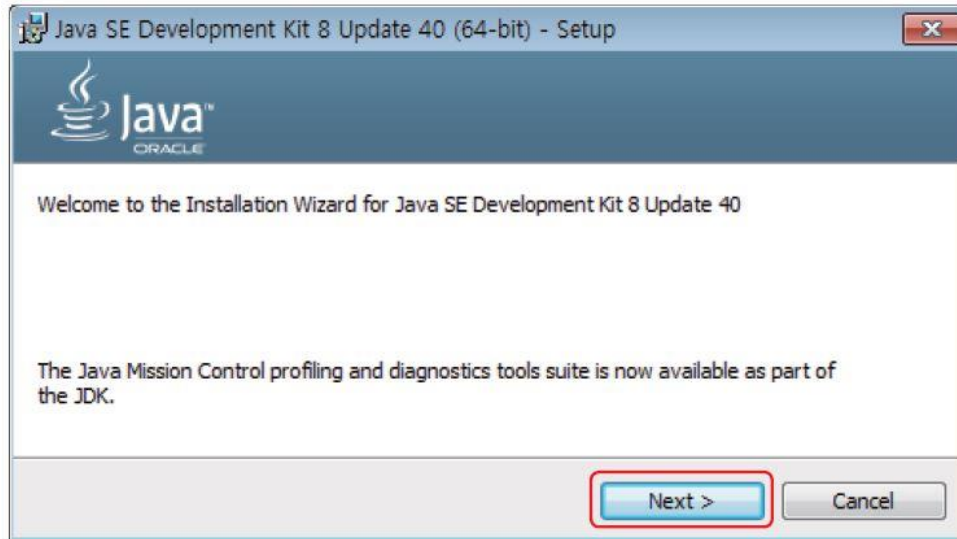
Product / File Description	File Size	Download
Linux ARM 32 Hard Float ABI	72.94 MB	 jdk-8u241-linux-arm32-vfp-hflt.tar.gz
Linux ARM 64 Hard Float ABI	69.83 MB	 jdk-8u241-linux-arm64-vfp-hflt.tar.gz
Linux x86 RPM Package	171.28 MB	 jdk-8u241-linux-i586.rpm
Linux x86 Compressed Archive	186.1 MB	 jdk-8u241-linux-i586.tar.gz
Linux x64 RPM Package	170.65 MB	 jdk-8u241-linux-x64.rpm
Linux x64 Compressed Archive	185.53 MB	 jdk-8u241-linux-x64.tar.gz
macOS x64	254.06 MB	 jdk-8u241-macosx-x64.dmg
Solaris SPARC 64-bit (SVR4 package)	133.01 MB	 jdk-8u241-solaris-sparcv9.tar.Z
Solaris SPARC 64-bit	94.24 MB	 jdk-8u241-solaris-sparcv9.tar.gz
Solaris x64 (SVR4 package)	133.8 MB	 jdk-8u241-solaris-x64.tar.Z
Solaris x64	92.01 MB	 jdk-8u241-solaris-x64.tar.gz
Windows x86	200.86 MB	 jdk-8u241-windows-i586.exe
Windows x64	210.92 MB	 jdk-8u241-windows-x64.exe

Java SE 다운로드

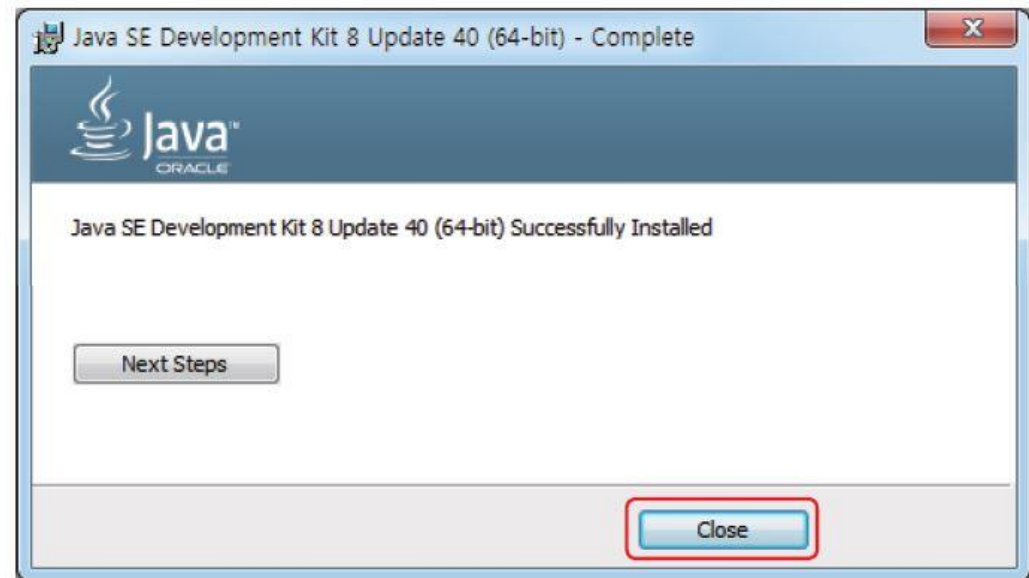
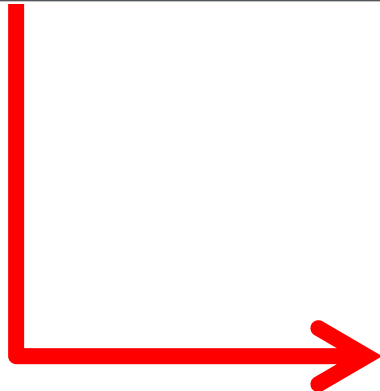
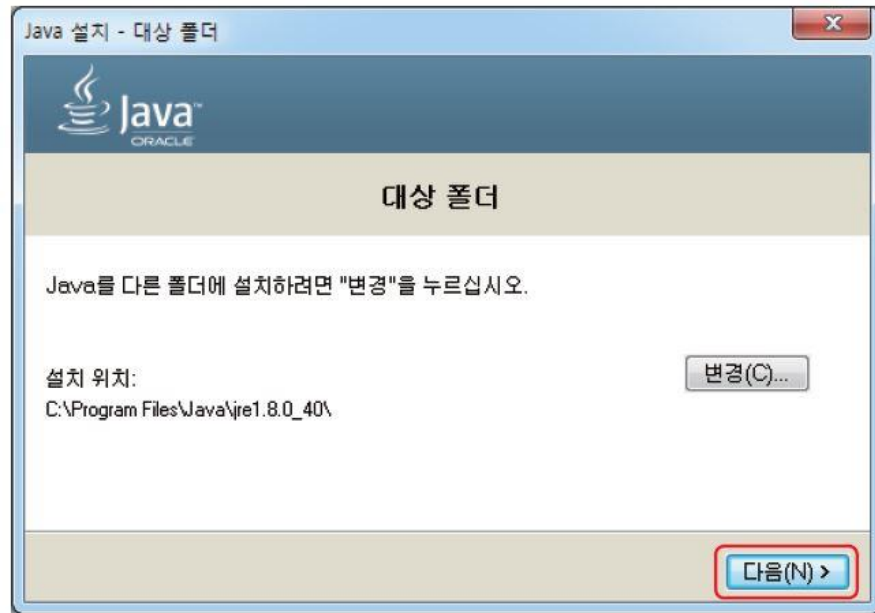
- 'Accept License Agreement' 체크 후, 앞에서 확인한 Windows가 32비트라면 Windows x86용 (파일명 : jdk-8uXX-windows-i586.exe)을, 64비트라면 Windows x64용(파일명 : jdk-8uXX-windows-x64.exe)을 다운로드



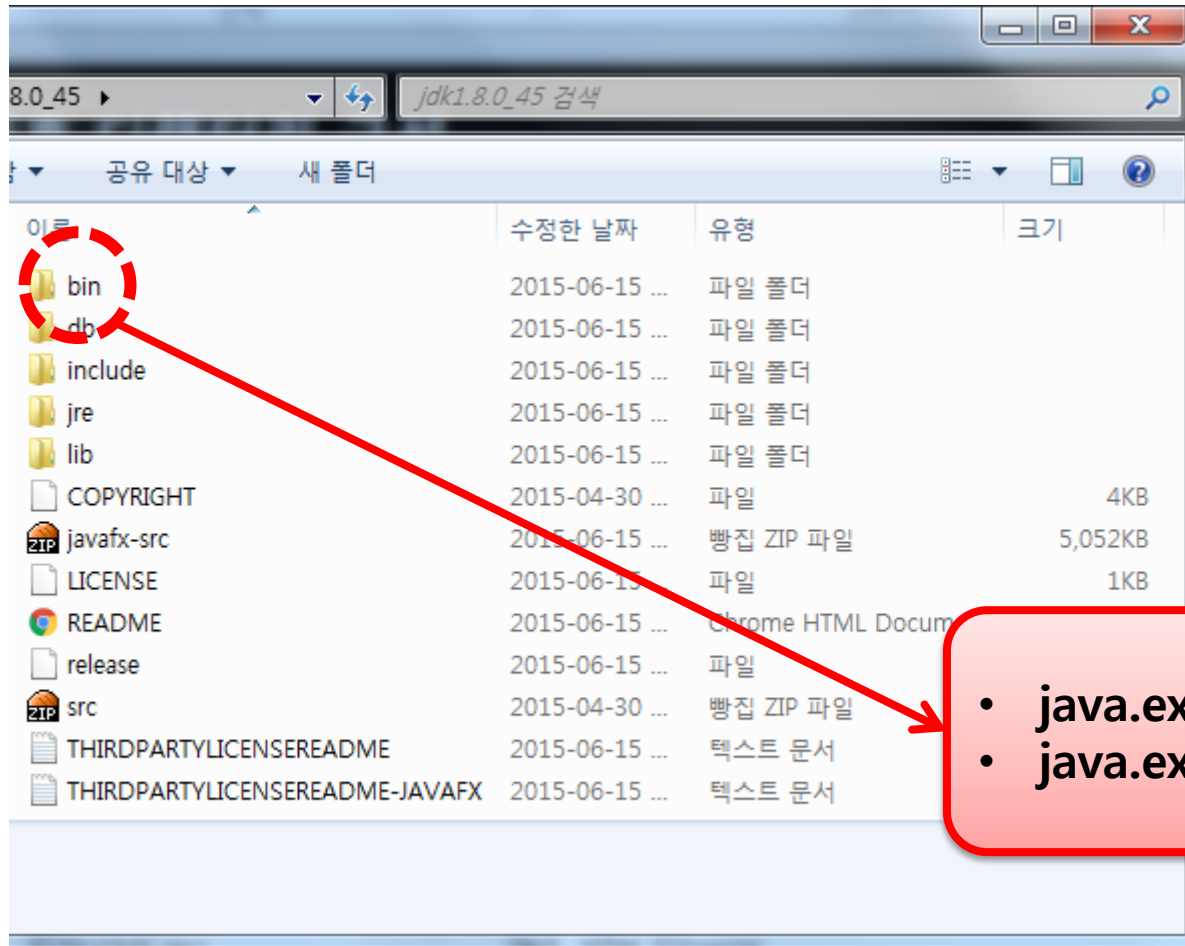
Java SE 다운로드



Java SE 다운로드



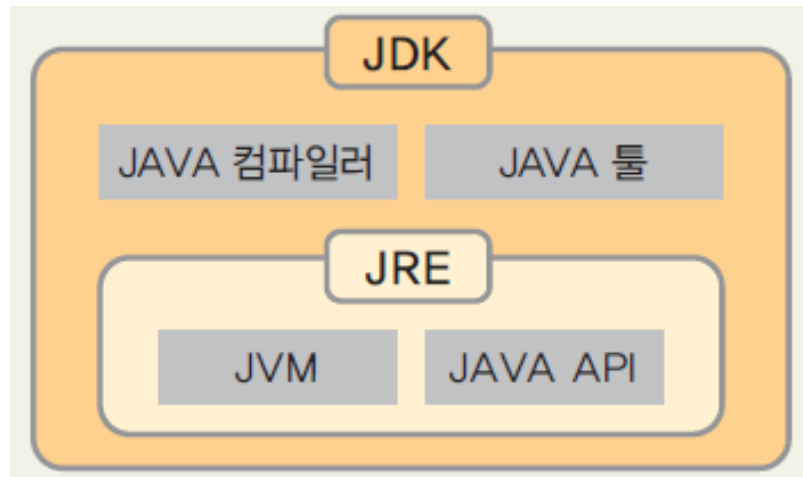
설치 이후 디렉터리 구성



- **java.exe** 자바 컴파일러
- **java.exe** 자바 런처

설치 이후 디렉터리 구성

- JDK와 JRE
 - JAVA 개발 도구인 JDK와 JAVA 실행 환경인 JRE를 명확한 구분
 - JDK : JAVA 프로그램을 작성하고 컴파일하는 데 필요
 - JRE : JAVA를 실행하는 데 필요하다.
 - JDK에는 JRE가 포함(JAVA 개발 없이 컴파일 결과(*.class)만 실행하려면 JRE만 설치)

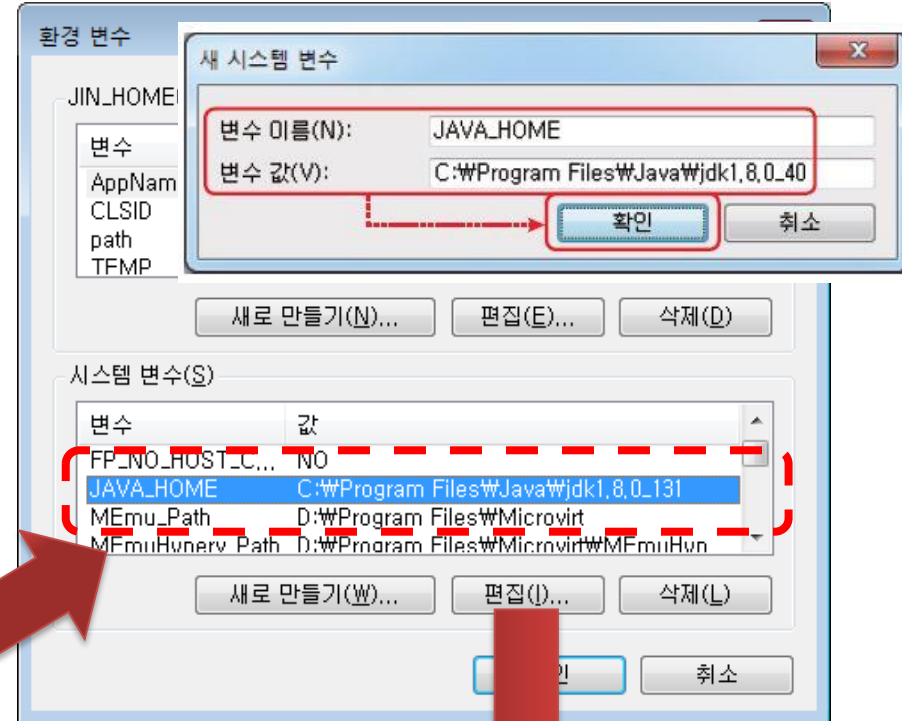
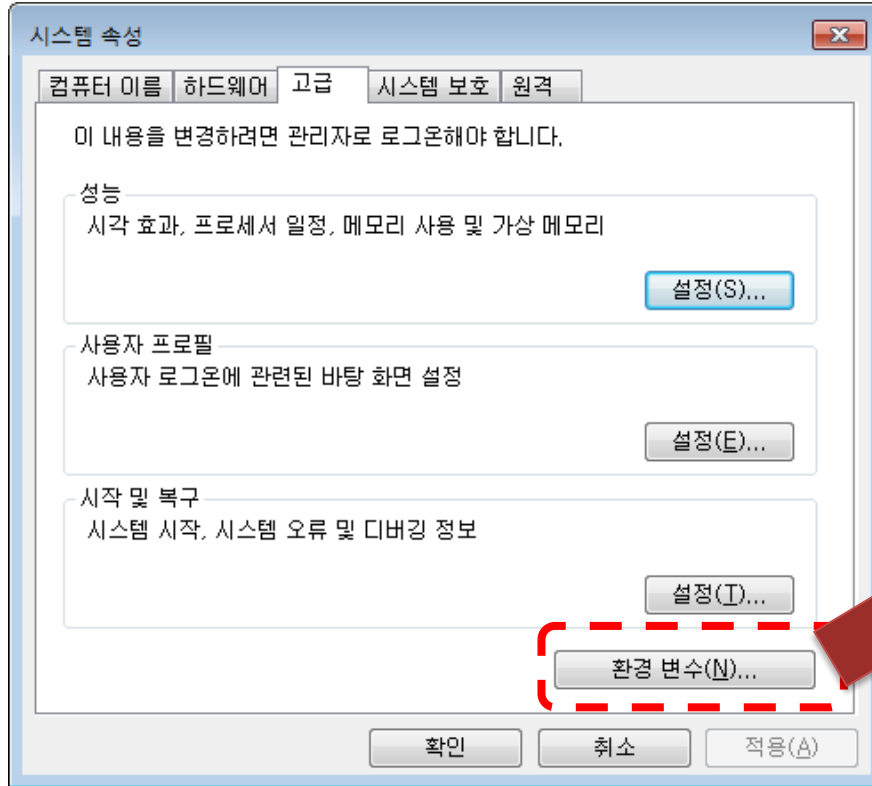


환경변수 설정

- 자바 환경변수 설정.

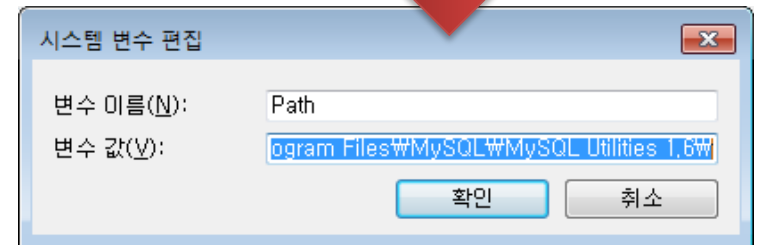
1. 환경변수 설정은 도스기반 디렉토리 시스템을 사용하는 윈도우에서 자바 개발킷이나 클래스의 경로 등을 편리하게 이용하기 위함이고, 나아가 IDE환경의 구축을 용이하게 해 줍니다.
2. 환경변수 설정메뉴는 윈도우7을 기준으로
시작버튼 - 컴퓨터 우클릭 - 속성 - 고급 시스템 설정 - 환경 변수 - 시스템 변수
3. 새로 만들기로 시스템변수 이름을 **JAVA_HOME**이라 만들고 변수 값에는 처음에 JDK를 설치한 폴더 경로를 넣습니다.
예) C:\Program Files\Java\jdk1.8.0_51
4. 그 다음 시스템변수 중 Path를 수정
기존에 있는 변수 값 뒤에 ;%JAVA_HOME%\bin; 라고 덧붙입니다.

설치 이후 추가 과정 : bin 디렉터리 path 설정



JAVA_HOME
C:\Program Files\Java\jdk1.8.0_131

Path
;%JAVAHOME%/bin;



이클립스 설치

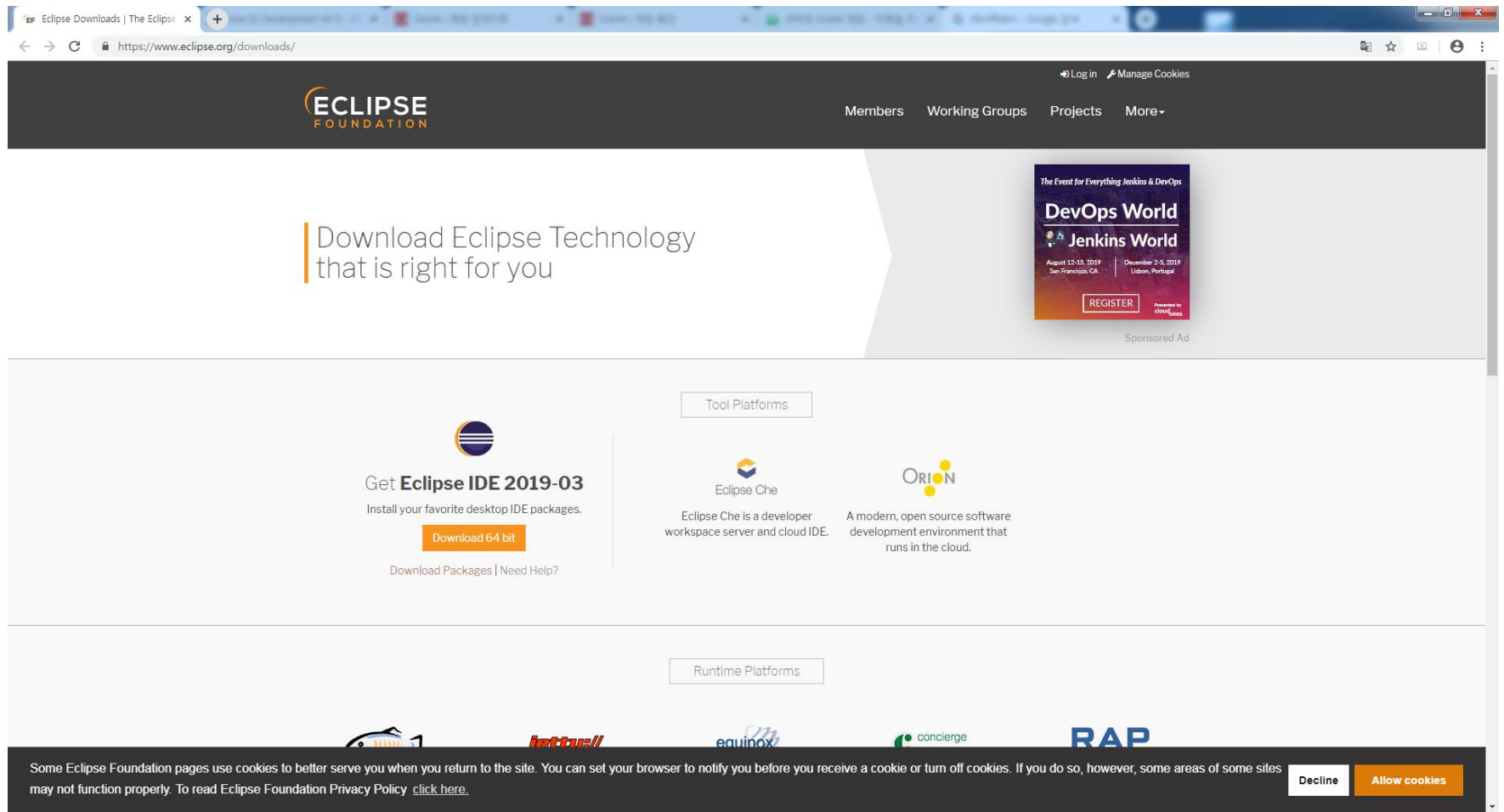
<http://www.eclipse.org/downloads/>

용도에 따라 선택하시면 되는데, 잘 모르겠다면 Eclipse IDE for Java Developers 를 다운로드 합니다.

사실 이클립스는 설치할 것도 없습니다. 그냥 압축 풀면 됩니다.
적절히 편한 곳에 압축 풀어줍니다.

이클립스 설치

<https://www.eclipse.org/downloads/>



The screenshot shows the Eclipse Foundation Downloads page. The browser address bar displays <https://www.eclipse.org/downloads/>. The page features the Eclipse Foundation logo and navigation links: Members, Working Groups, Projects, and More. A main heading reads "Download Eclipse Technology that is right for you". A sponsored advertisement for "DevOps World" and "Jenkins World" is visible on the right. The page is divided into two main sections: "Tool Platforms" and "Runtime Platforms". Under "Tool Platforms", there are three options: "Get Eclipse IDE 2019-03" with a "Download 64 bit" button, "Eclipse Che" described as a developer workspace server and cloud IDE, and "ORION" described as a modern, open source software development environment that runs in the cloud. The "Runtime Platforms" section is partially visible at the bottom, showing logos for Equinox, Concierge, and RAP. A footer banner contains a cookie notice and buttons for "Decline" and "Allow cookies".

Eclipse Downloads | The Eclipse x

Log in Manage Cookies

Members Working Groups Projects More

Download Eclipse Technology that is right for you

DevOps World
Jenkins World
August 12-15, 2019 San Francisco, CA
December 2-5, 2019 Lisbon, Portugal
REGISTER Presented by Cloud Types

Sponsored Ad

Tool Platforms

Get **Eclipse IDE 2019-03**
Install your favorite desktop IDE packages.
[Download 64 bit](#)
[Download Packages](#) | [Need Help?](#)

Eclipse Che
Eclipse Che is a developer workspace server and cloud IDE.

ORION
A modern, open source software development environment that runs in the cloud.

Runtime Platforms

Equinox Concierge RAP

Some Eclipse Foundation pages use cookies to better serve you when you return to the site. You can set your browser to notify you before you receive a cookie or turn off cookies. If you do so, however, some areas of some sites may not function properly. To read Eclipse Foundation Privacy Policy [click here](#).

Decline Allow cookies


이클립스 설치

<https://www.eclipse.org/downloads/packages/>

Eclipse Packages | The Eclipse Foundation

← → ↻ https://www.eclipse.org/downloads/packages/

⌵ ☆ ⓘ ⌵ ⌵ ⌵

MembersWorking GroupsProjectsMoreDownload

Home / Downloads / Packages / Release / Eclipse IDE 2019-03 / R

Eclipse InstallerEclipse PackagesEclipse Developer Builds

Try the Eclipse **Installer** 2019-03 R

The easiest way to install and update your Eclipse Development Environment.

Find out more


1,235,329 Downloads

Download

Mac OS X 64 bit


Windows 64 bit

Linux 64 bit

Sirius

The Easiest Way to Get Your Own Modeling Tool

DOWNLOAD NOW




Get **Eclipse IDE 2019-03**

Install your favorite desktop IDE packages.


Download 64 bit

Download Packages | Need Help?


Eclipse IDE 2019-03 R Packages

Eclipse IDE for Enterprise Java Developers


340 MB | 216,137 DOWNLOADS




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

Eclipse IDE for Java Developers


189 MB | 145,119 DOWNLOADS



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

Eclipse IDE for C/C++ Developers

189 MB | 145,119 DOWNLOADS



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

RELATED LINKS

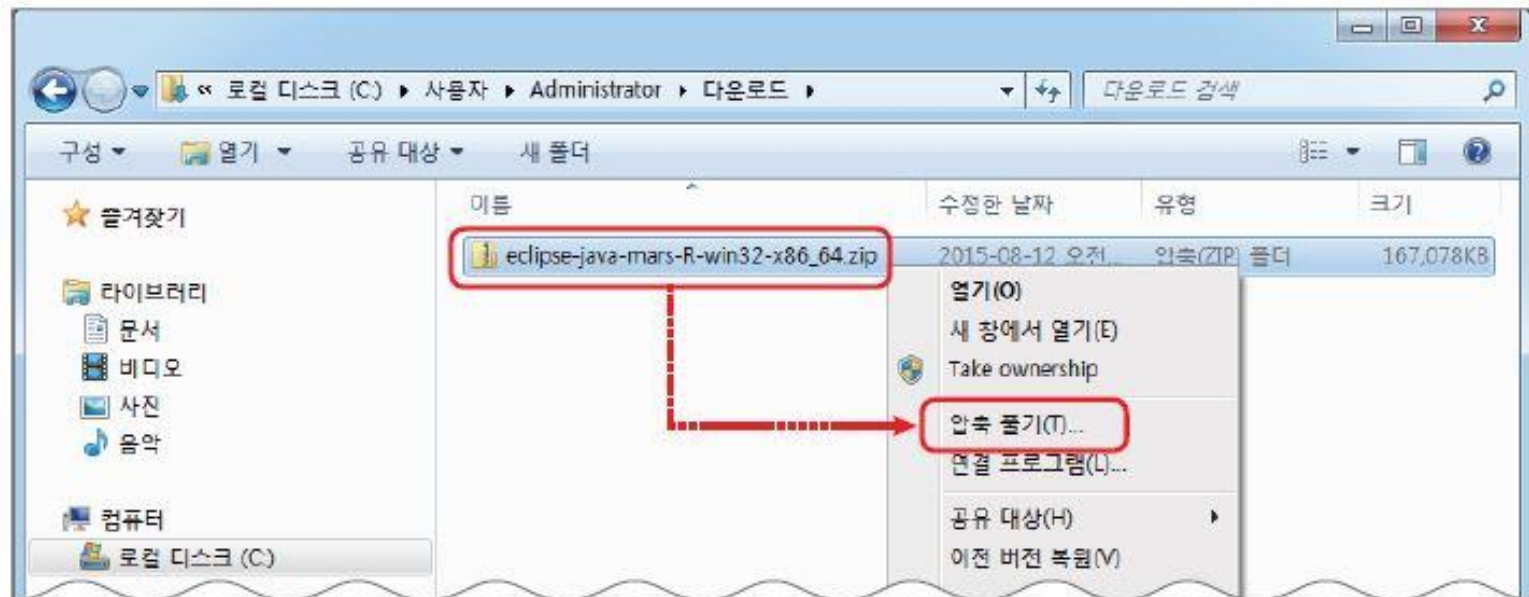
- Compare & Combine Packages
- New and Noteworthy
- Install Guide
- Documentation
- Updating Eclipse
- Forums

Some Eclipse Foundation pages use cookies to better serve you when you return to the site. You can set your browser to notify you before you receive a cookie or turn off cookies. If you do so, however, some areas of some sites may not function properly. To read Eclipse Foundation Privacy Policy [click here.](#)

DeclineAllow cookies

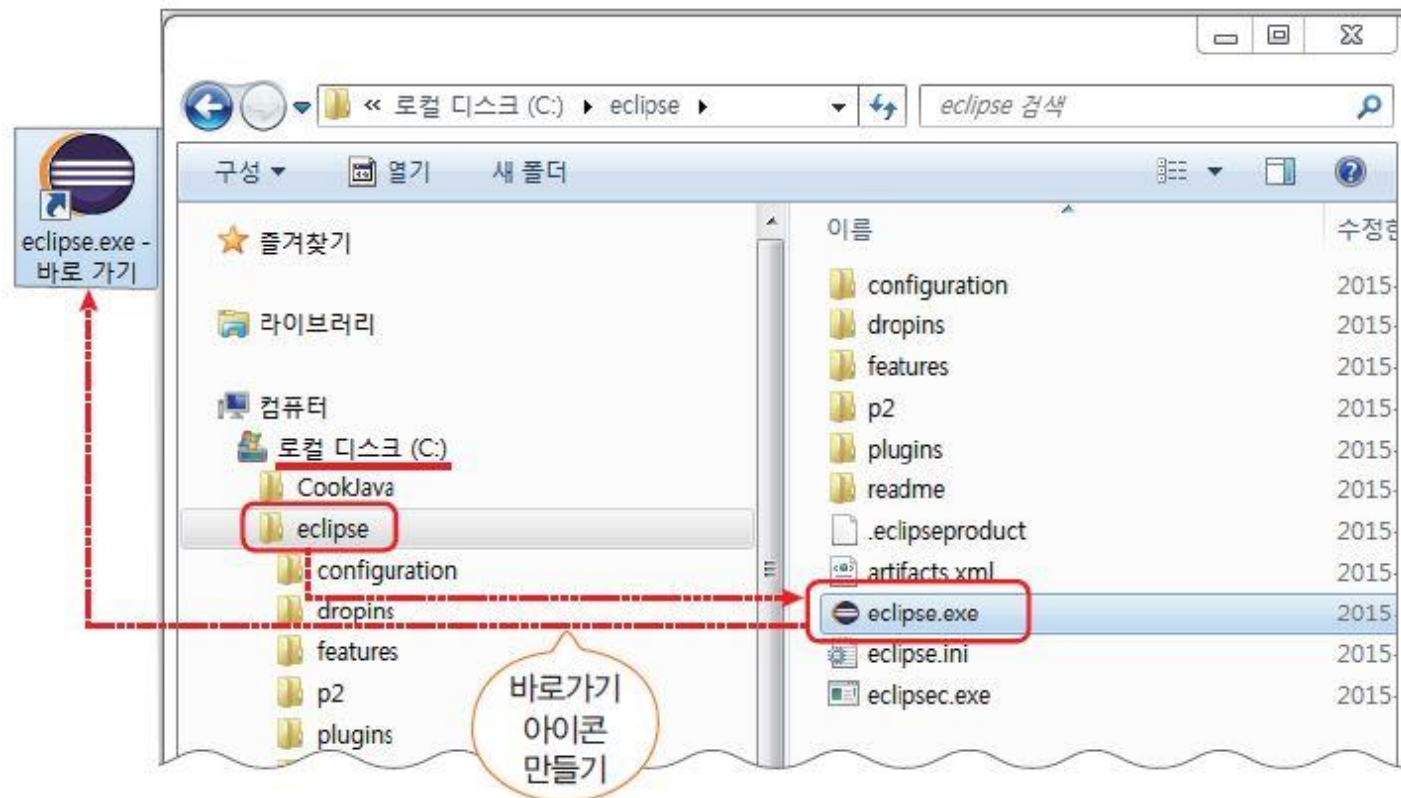
JAVA 개발 환경 (3)

- ③ 파일 탐색기 실행하여 다운로드한 파일 선택, 마우스 오른쪽 버튼 클릭 [압축 풀기] 선택



JAVA 개발 환경 (4)

- ④ 압축이 풀린 폴더 안의 'eclipse' 폴더를 통째로 C:\w 폴더로 옮기면
C:\w\eclipse\workspace.exe 파일 보임(이것이 이클립스의 실행 파일), 이 파일을 바탕화면에 바로가기 아이콘으로 만들기



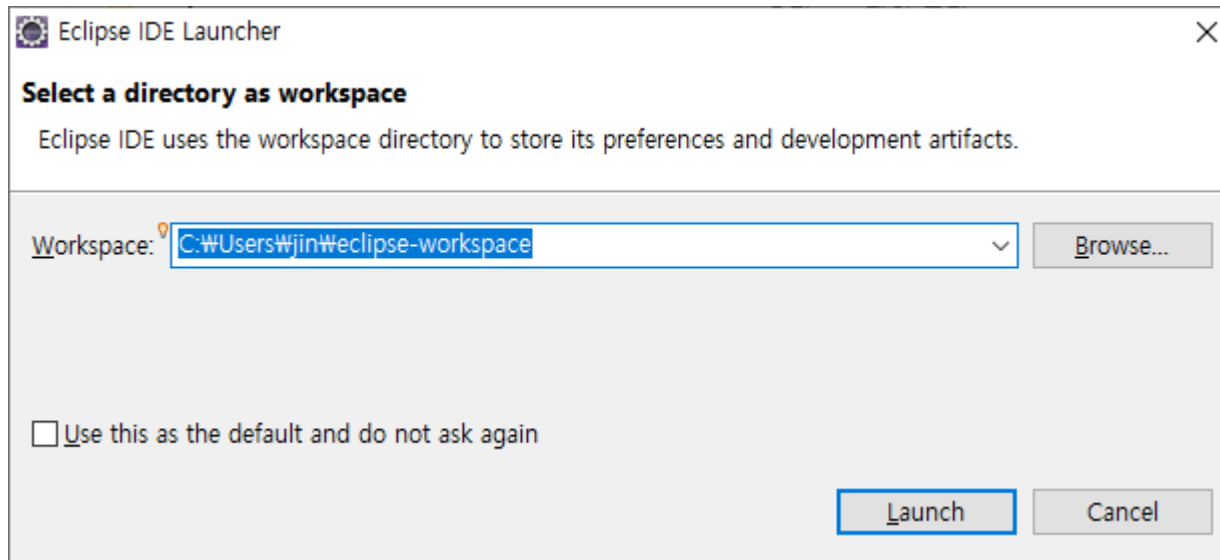
JAVA 개발 환경 (5)

- ⑤ 이클립스 실행 위해 바탕화면의 바로가기 아이콘 더블클릭, 이클립스 로고 화면 나옴



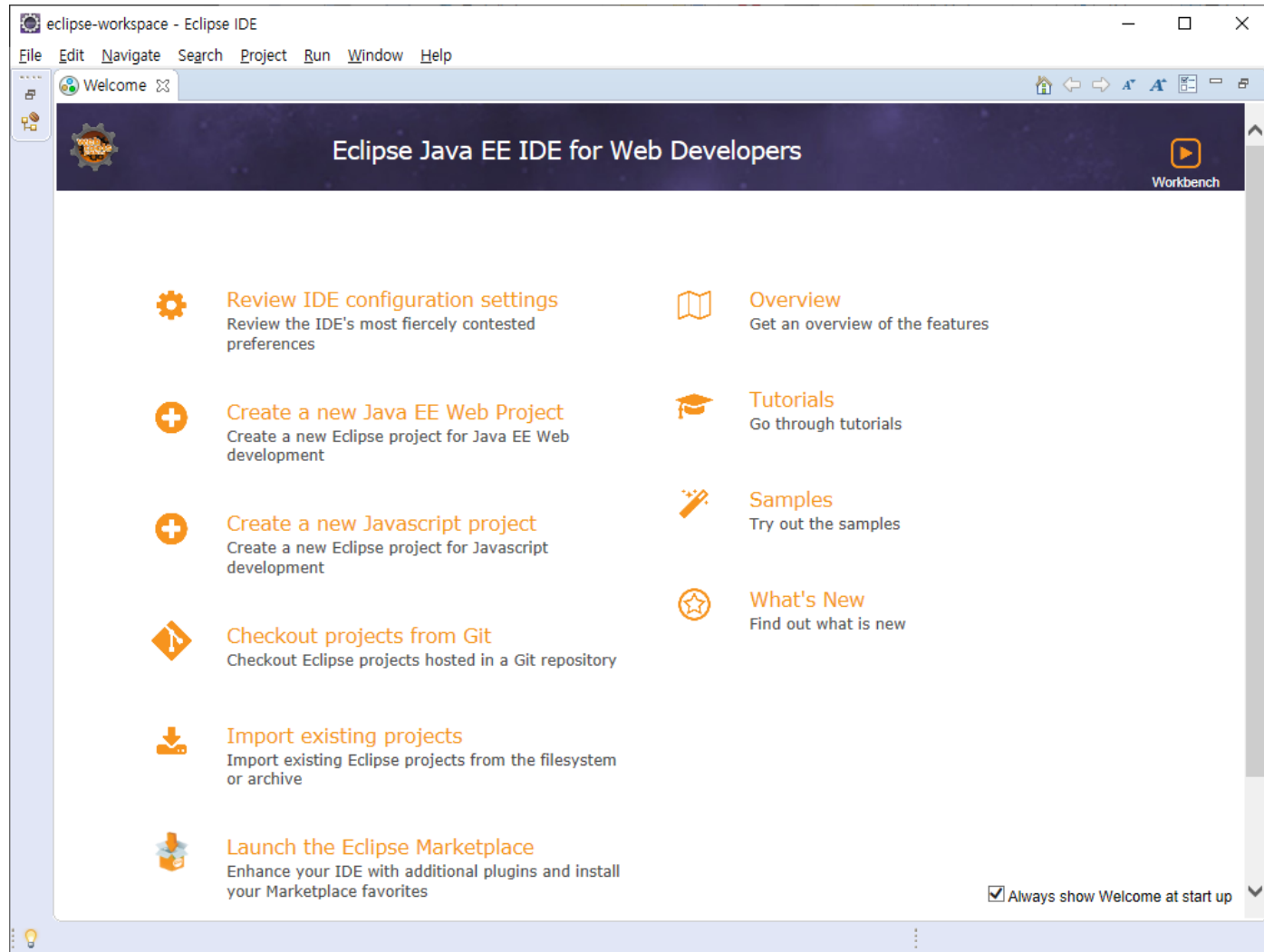
JAVA 개발 환경 (6)

⑥ 'Select a workspace' 창에서 C:\CookJava\라고 직접 써서 변경 후 [OK] 클릭



JAVA 개발 환경 (7)

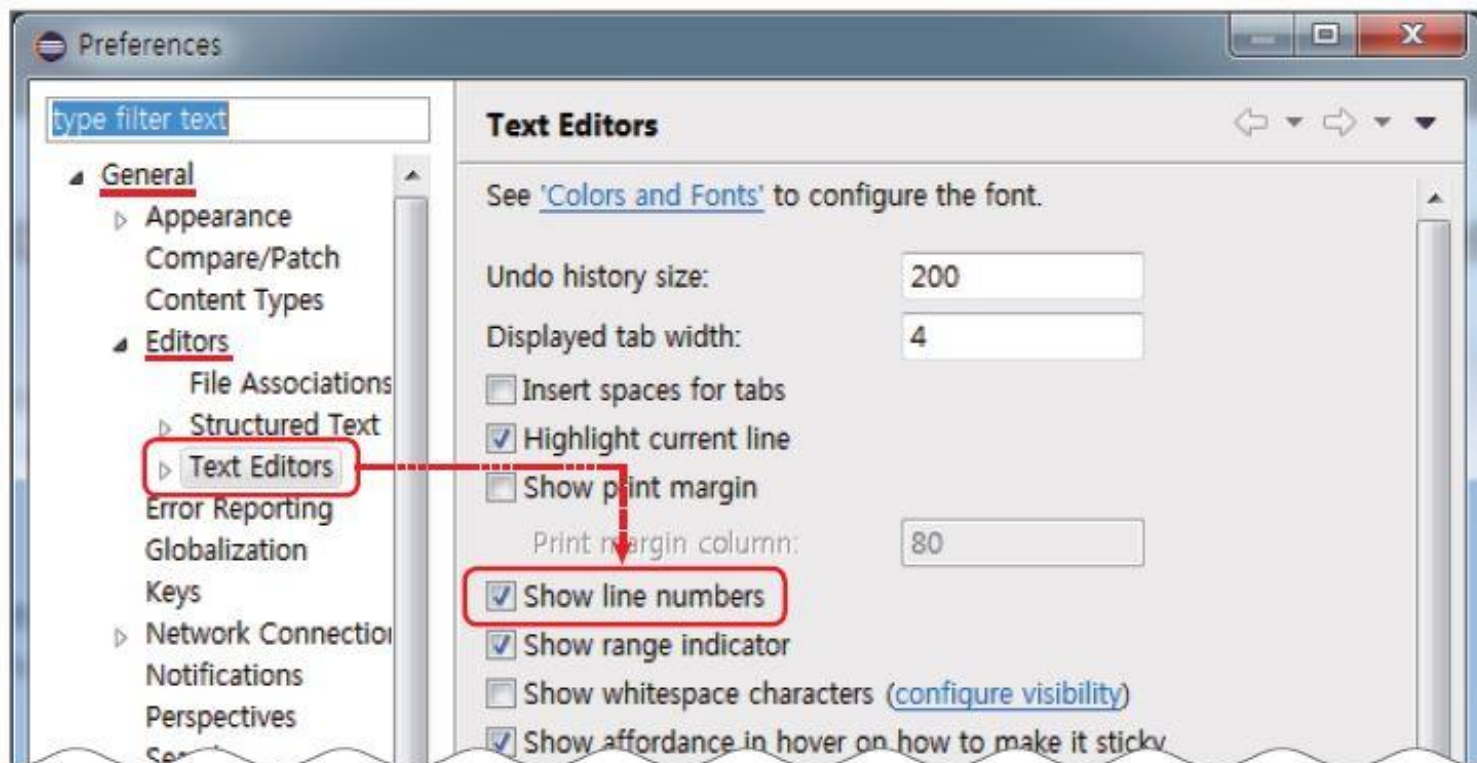
⑦ 이클립스 초기 화면 나눔



JAVA 개발 환경 (8)

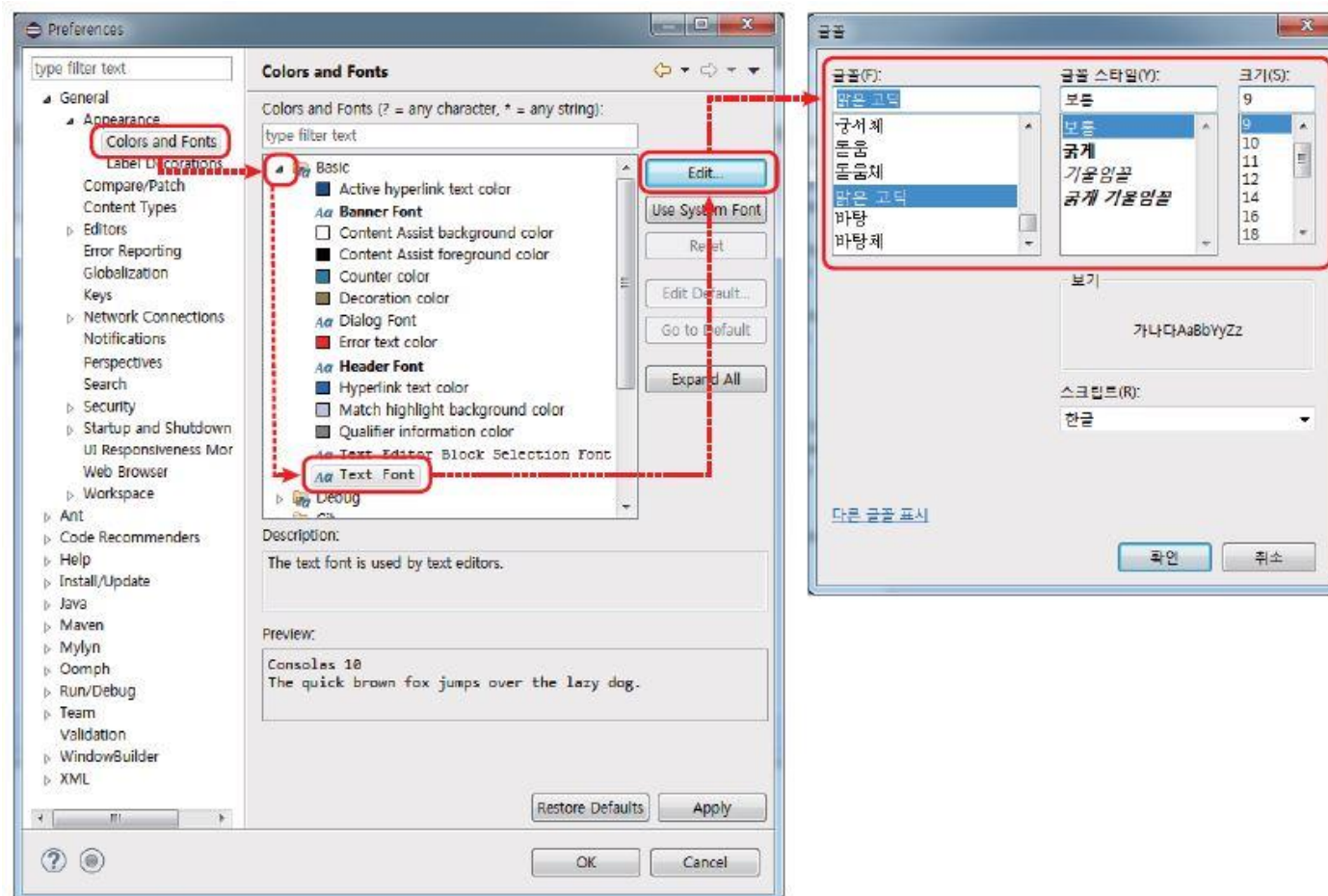
– 개발이 편리하도록 이클립스 설정 변경하기

- ① 행 번호 보이게 하기 : 앞으로 코딩할 소스의 행 번호가 보이도록 설정. Eclipse 메뉴의 [Window]-[Preferences]를 선택, 왼쪽 트리에서 [General]-[Editors]-[Text Editors] 선택, 오른쪽의 'Show line numbers'가 체크되어 있는지 확인. 체크되어 있지 않으면 체크



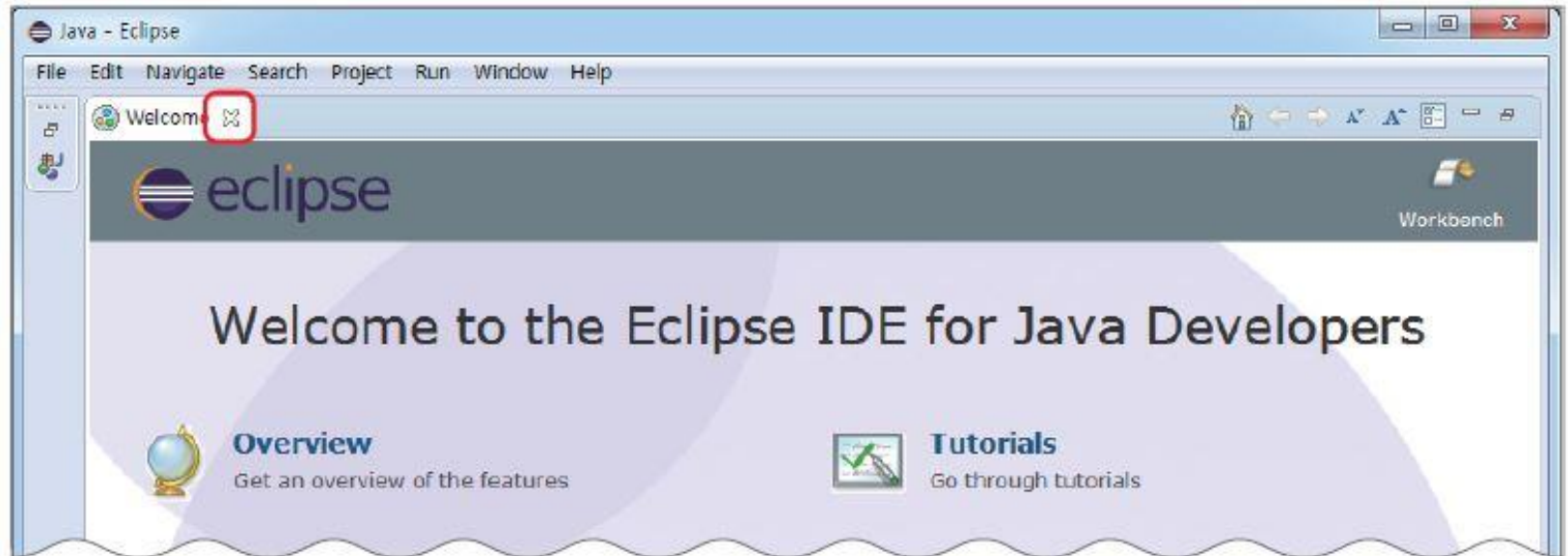
JAVA 개발 환경 (9)

- ② 폰트 변경하기 : 왼쪽 트리의 [General]-[Appearance]-[Colors and Fonts] 선택. 오른쪽 트리의 [Basic]-[Text Font] 선택, [Edit] 클릭, 원하는 글꼴, 크기 등 선택, [확인] 클릭. [OK]를 클릭하여 'Preferences' 창 종료



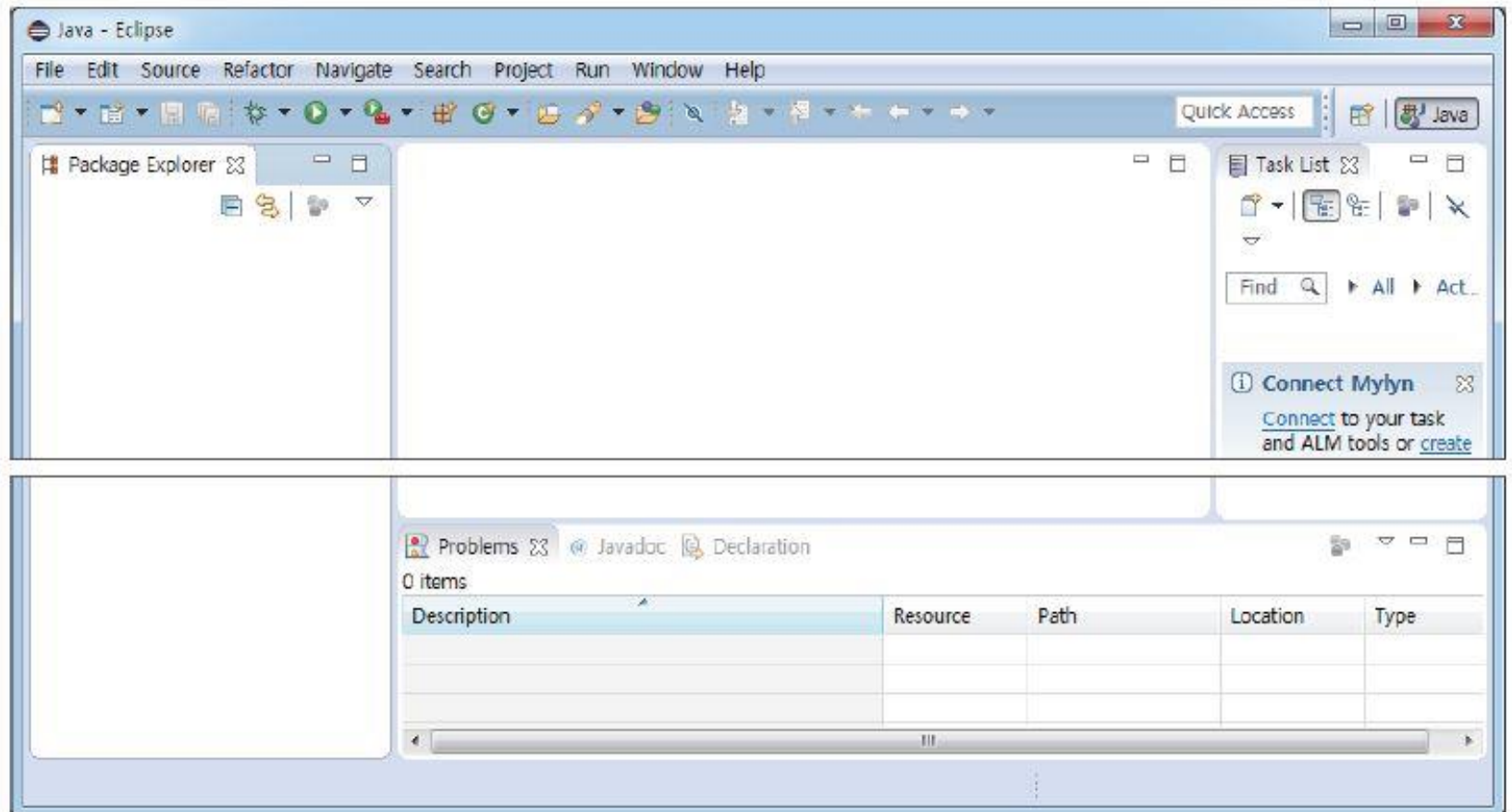
JAVA 개발 환경 (11)

- ④ 초기 Welcome 화면 필요 없으니 'Welcome' 옆의 [x] 클릭하여 창 닫기

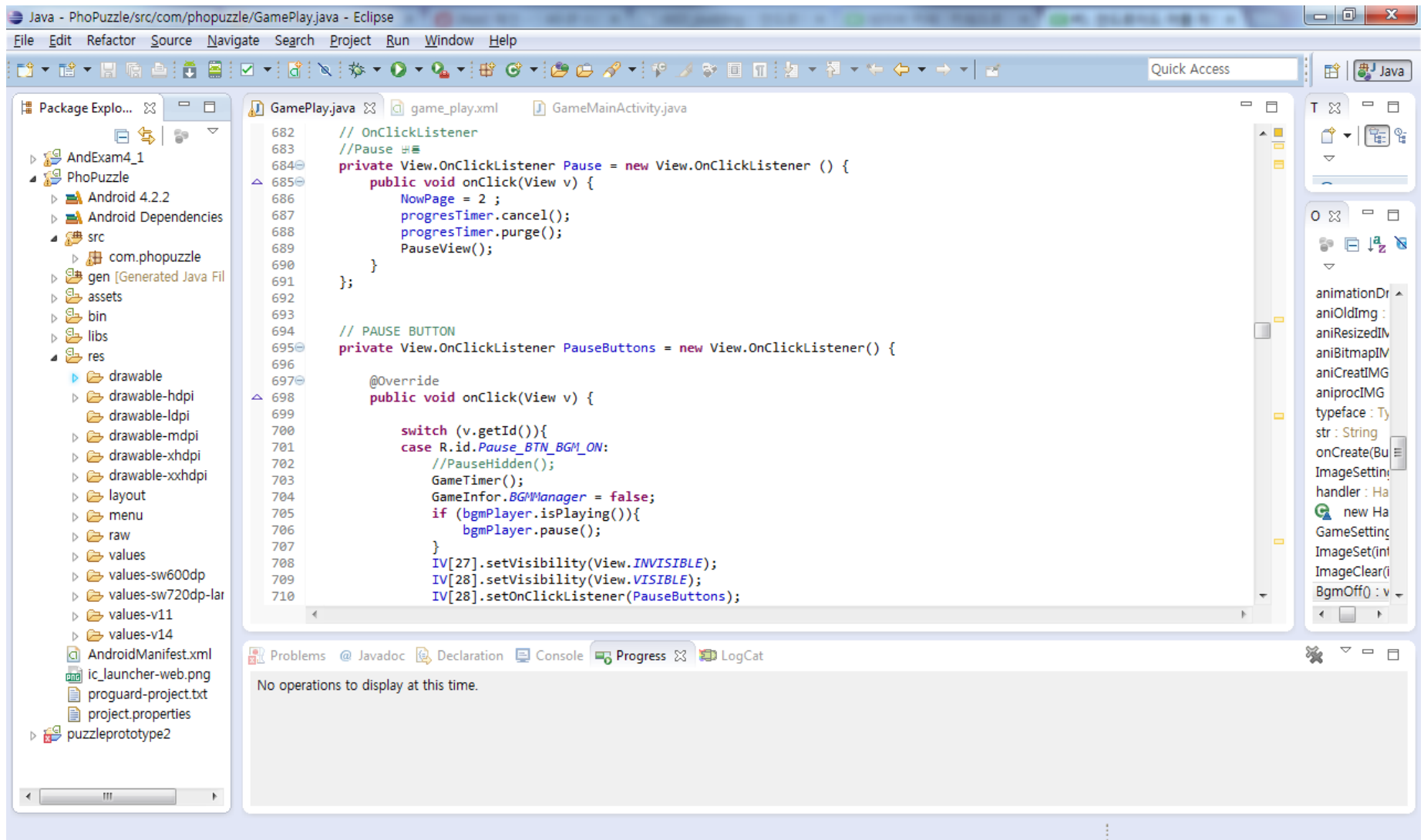


JAVA 개발 환경 (12)

- ⑤ 다음과 같은 프로젝트 창이 나오는데 앞으로는 주로 이 화면에서 JAVA 프로그래밍을 할 것이다. 일단 이클립스 종료



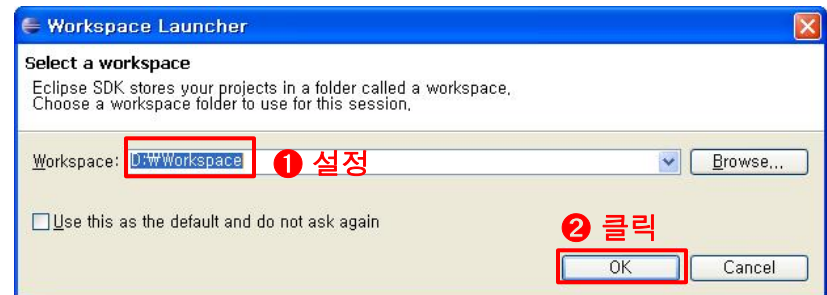
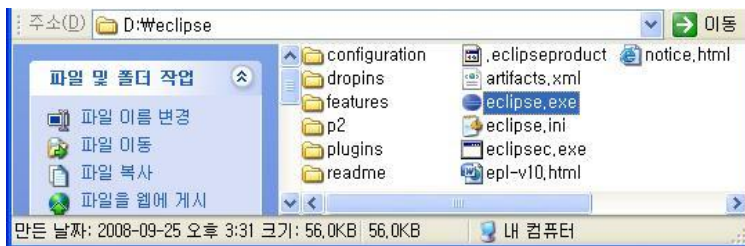
이클립스 살펴보기



실행 해 보기

- 기본 작업 폴더 생성

① 경로를 설정하고 ② <OK> 클릭 (D:\Workspace 폴더 권장)



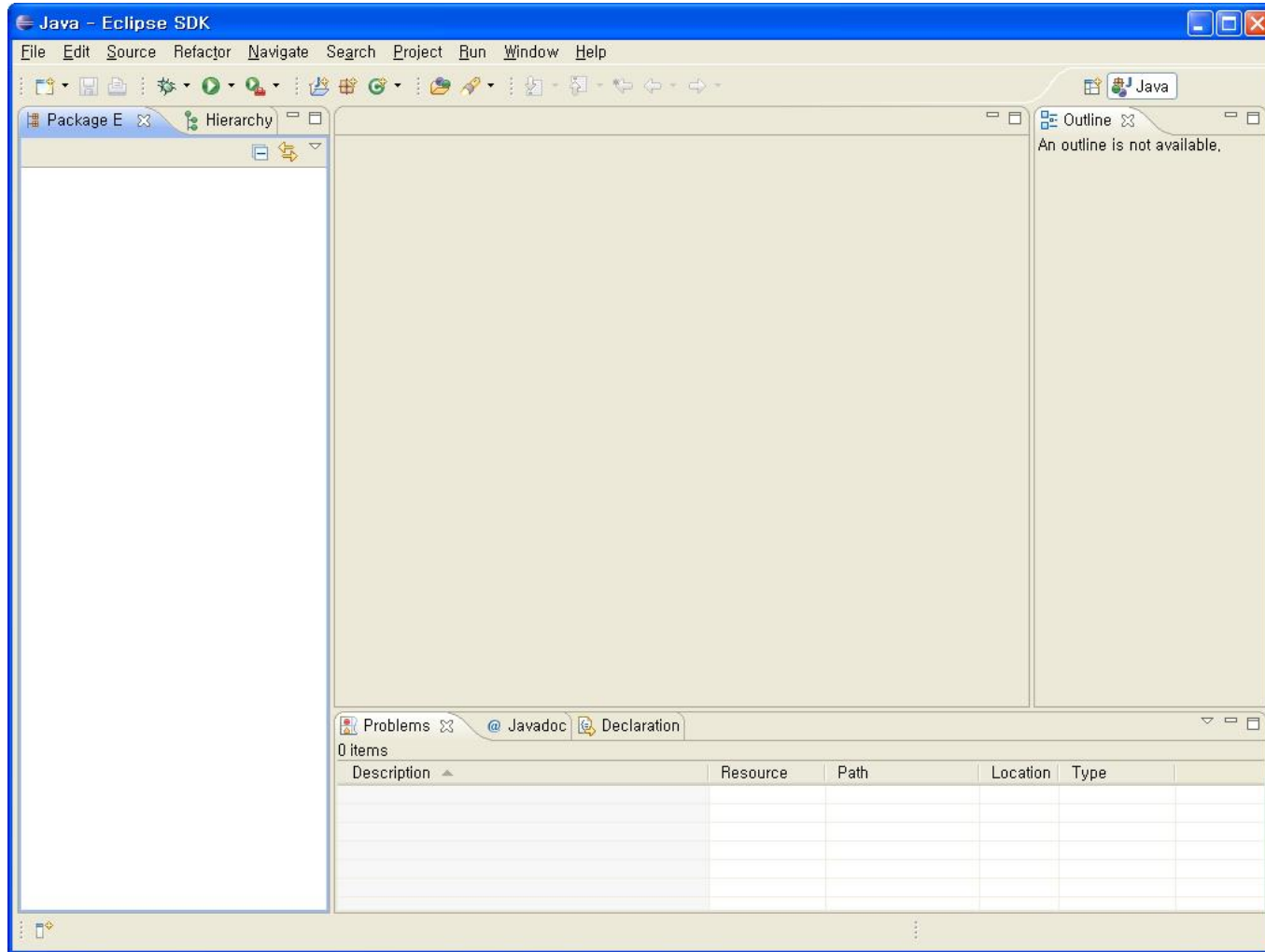
실행 해 보기

- 이클립스 실행 초기 화면

- 이클립스에서 왼쪽 위의



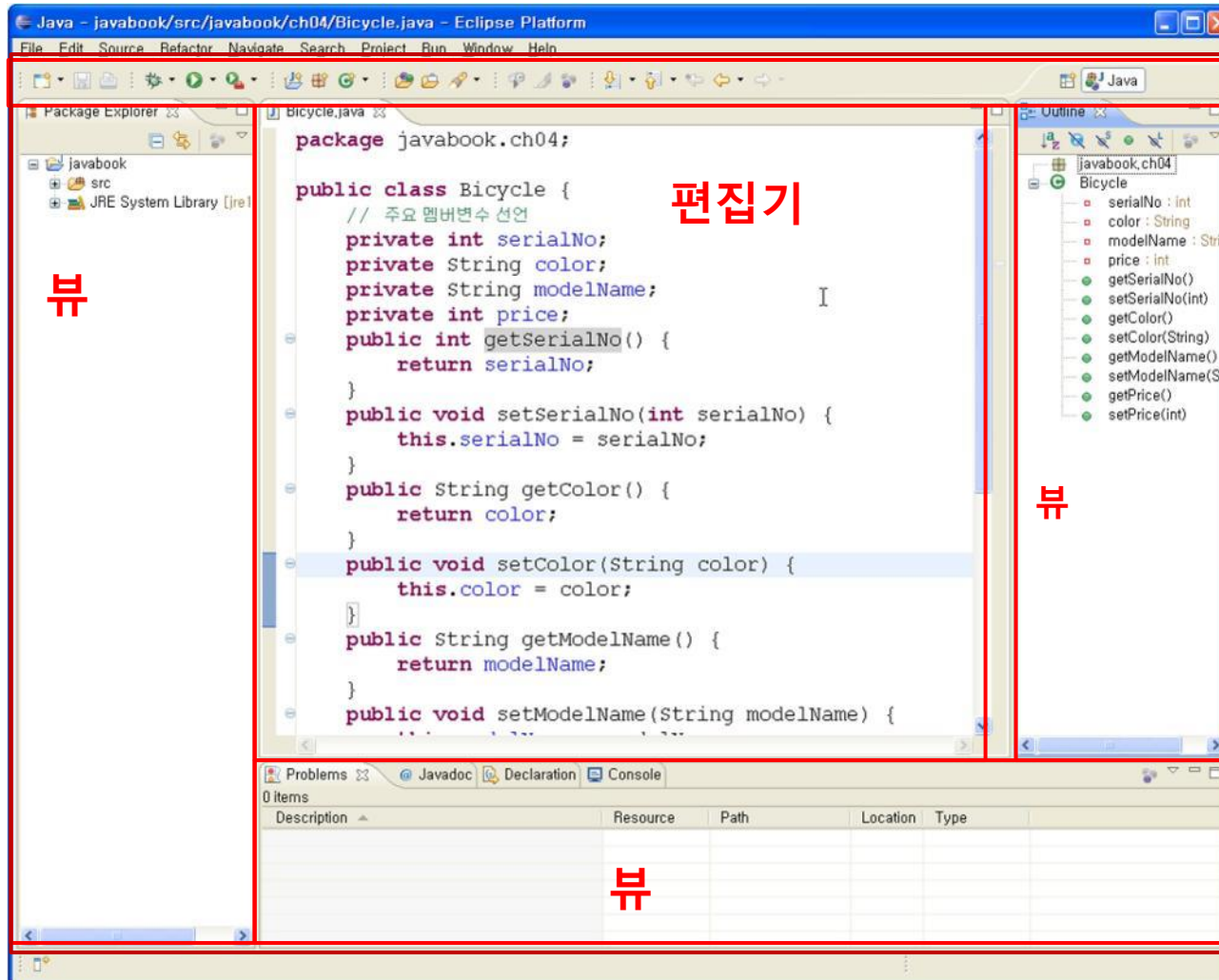
- 부분을 클릭



이클립스 기본 구조

- 이클립스 기본구조와 프로젝트 개념

워크벤치



퍼스펙티브

[이클립스 기본구조 화면]

이클립스 기본 구조

- 이클립스 기본구조와 프로젝트 개념

- Java 퍼스펙티브 주요 뷰

- 자바 개발은 기본적으로 Java 퍼스펙티브에서 작업하는 것이 유리함.

- **Package Explorer**

- 프로젝트를 중심으로 자바 패키지와 클래스를 관리하는 뷰

- **Navigator**

- Package Explorer와 유사하나 실제 폴더공간을 보여준다.

- **Outline**

- 자바 소스를 트리 구조로 쉽게 보여준다.

- **Console**

- 자바 프로그램에서의 표준 출력 메시지를 보여준다.

- **Problems**

- 소스에 문제가 있는 부분을 보여준다.

- **Javadoc**

- 자바 API 문서를 보여준다.

이클립스 살펴보기

- 이클립스 기본구조와 프로젝트 개념

- 프로젝트

- 이클립스에서 작업하는 기본적인 단위. 어느 정도 규모가 있는 작업의 기준임.

- 패키지

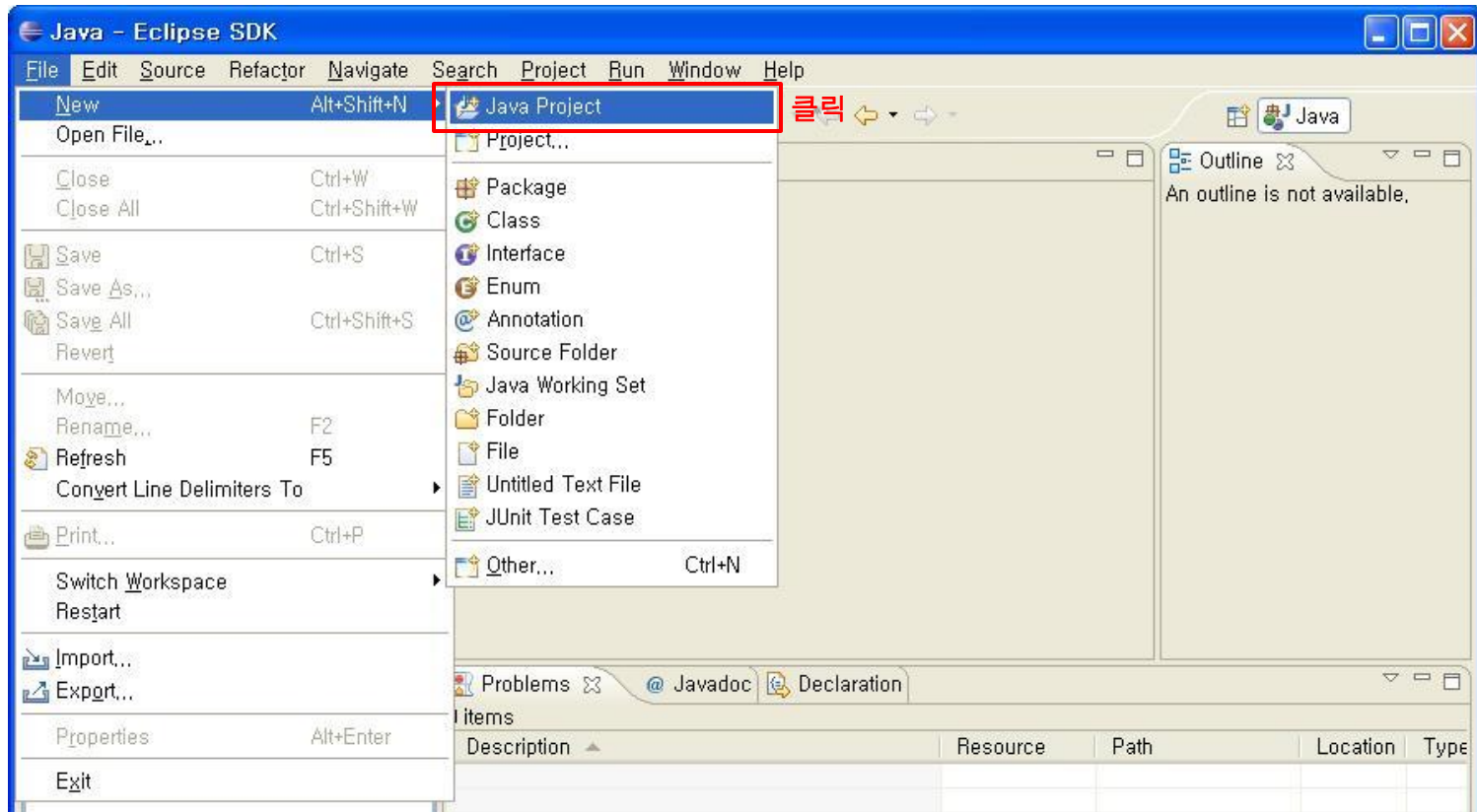
- 서로 관계가 있는 자바 클래스를 하나의 폴더에 묶어 놓은 것.
 - 예) mp3파일 앨범 별, 가수 별, 장르별 등.

- 클래스

- 자바 프로그램의 기본이 되는 단위.

[실습하기] 이클립스를 이용해 자바 프로그램 작성하고 실행하기

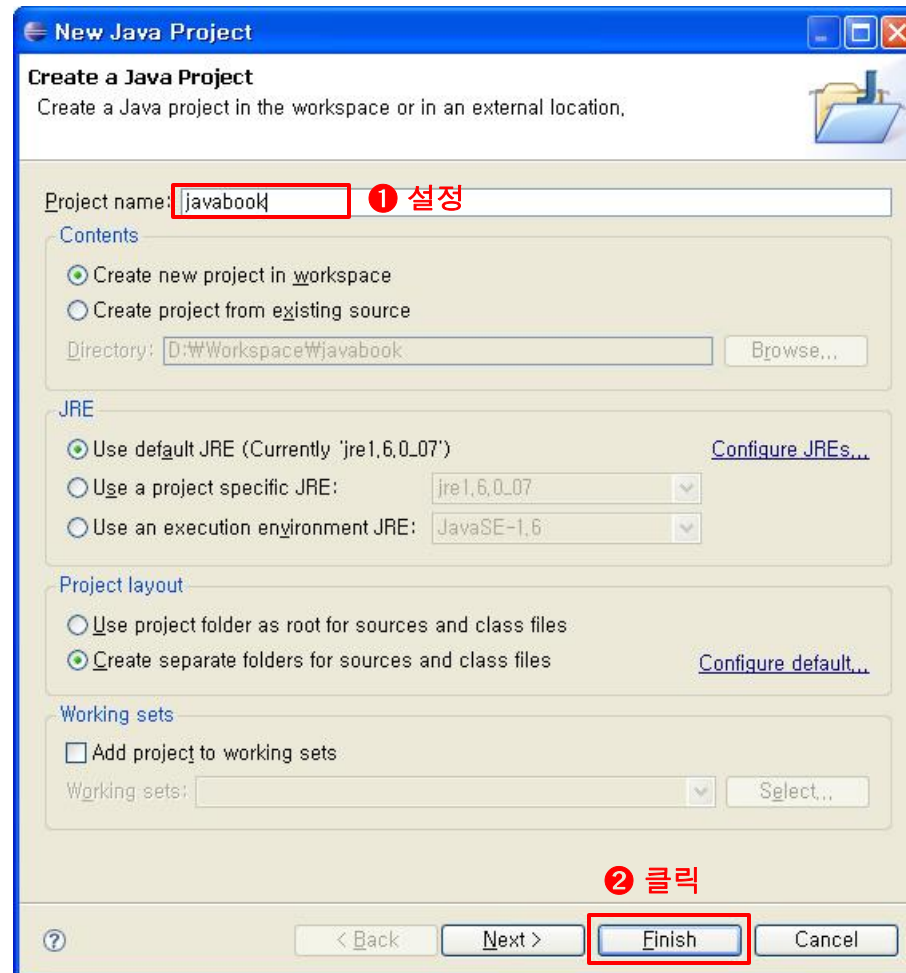
- Java Project 생성 메뉴 선택
 - 이클립스 실행 후 [File]-[New]-[Java Project] 메뉴 클릭



[실습하기] 이클립스를 이용해 자바 프로그램 작성하고 실행하기

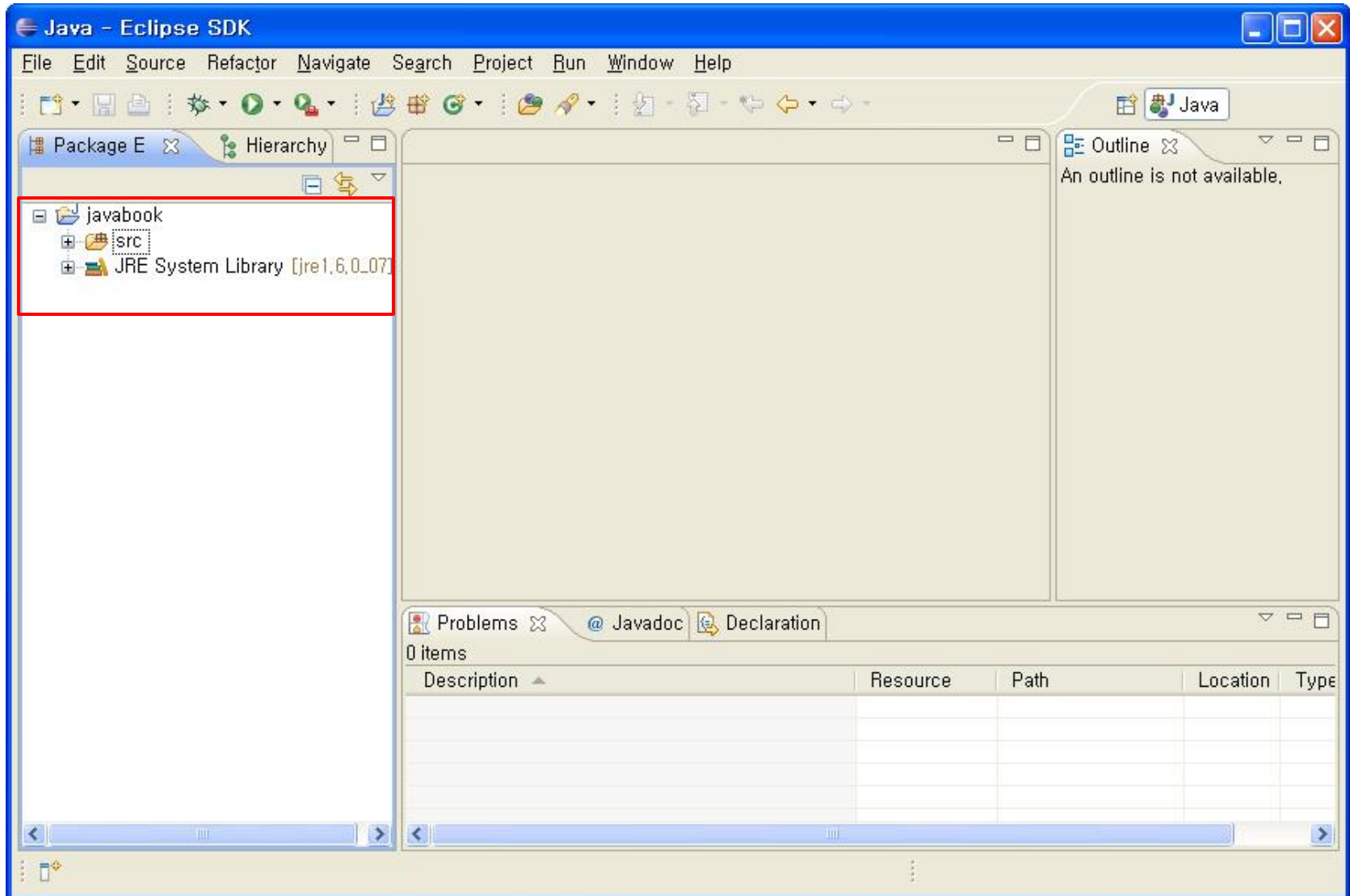
• 프로젝트 환경설정

- ① Project name란에 원하는 이름 입력(javabook)
- ② 기본 설정대로 두고 <Finish> 버튼 클릭



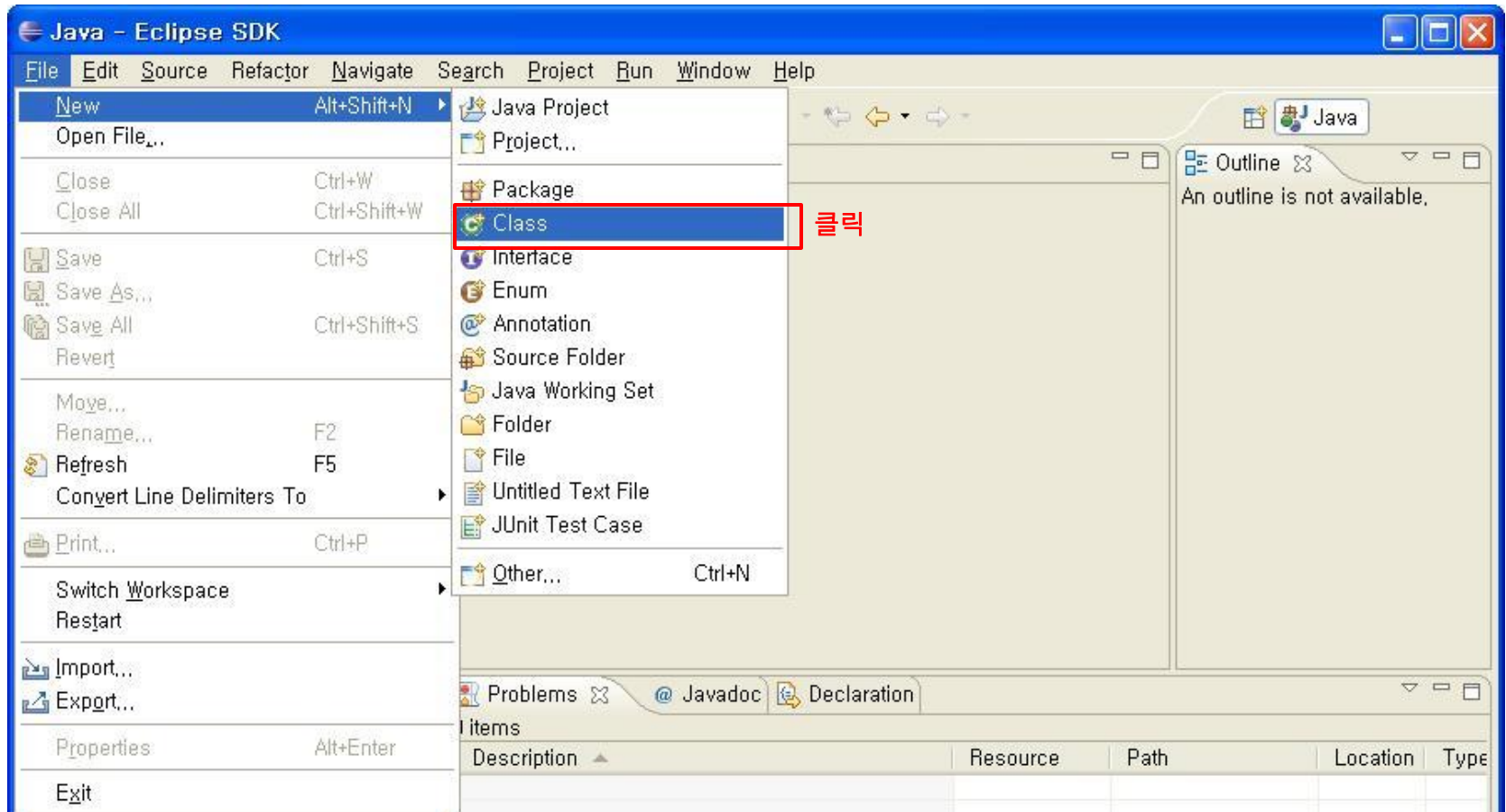
[실습하기] 이클립스를 이용해 자바 프로그램 작성하고 실행하기

- 프로젝트 추가 확인



[실습하기] 이클립스를 이용해 자바 프로그램 작성하고 실행하기

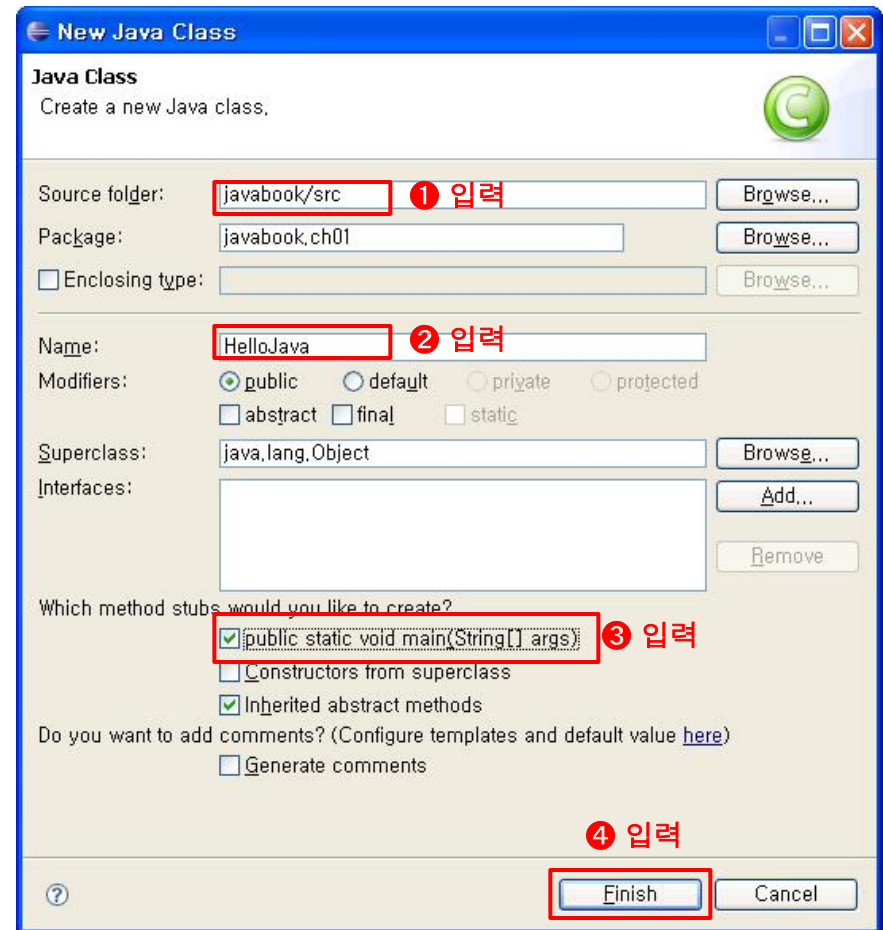
- 클래스를 추가하는 메뉴 선택
 - [File]-[New]-[Class] 메뉴 클릭



[실습하기] 이클립스를 이용해 자바 프로그램 작성하고 실행하기

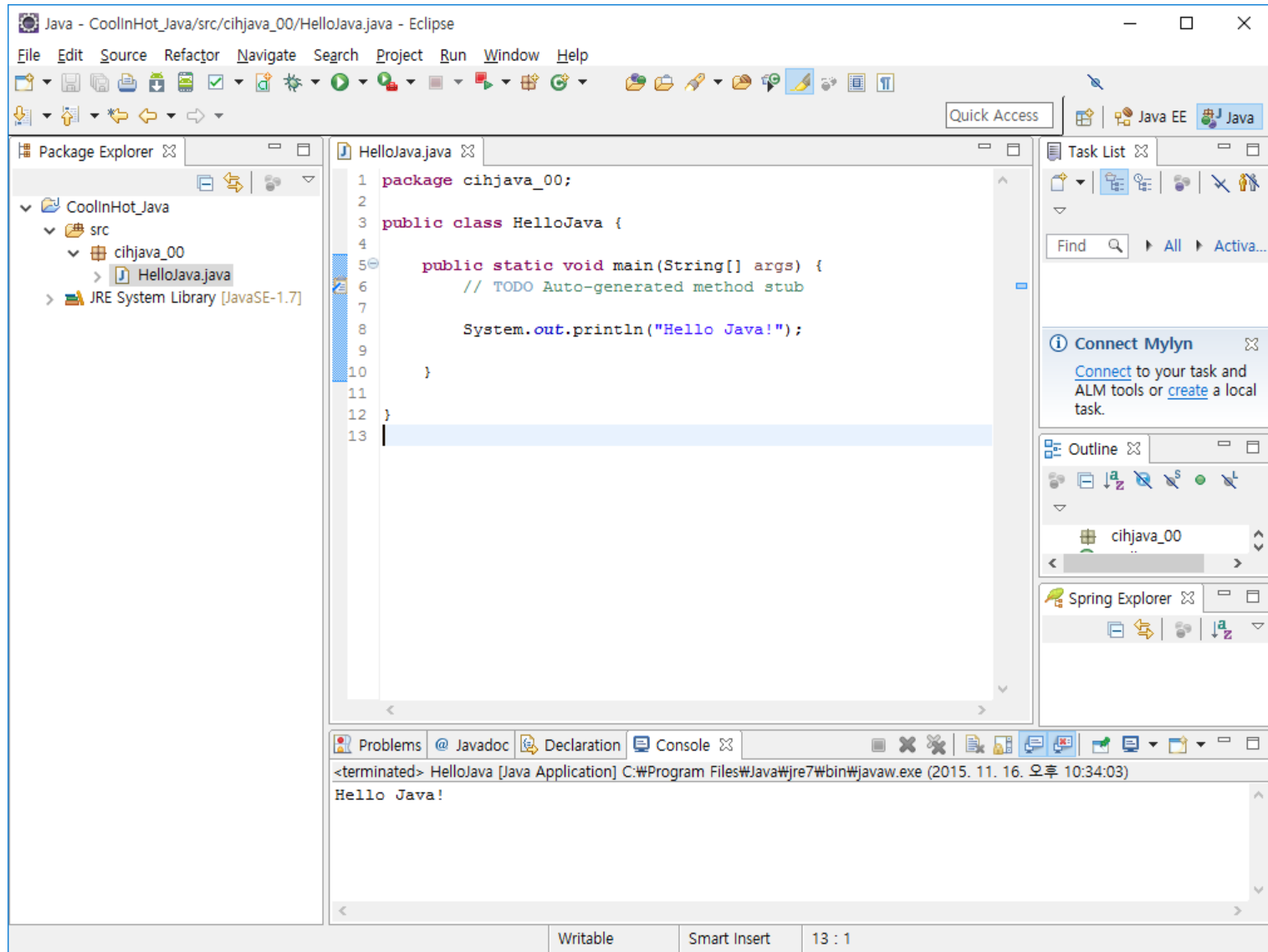
• HelloJava 클래스 생성

- ① [New Java Class] 대화상자의 Package에 **javatest** 를,
- ② Name 에는 HelloJava를 입력한 다음
- ③ public static void main(String[] args)항목에 체크하고
- ④ <Finish> 버튼 클릭



[실습하기] 이클립스를 이용해 자바 프로그램 작성하고 실행하기

- main 메소드를 갖는 클래스 생성 확인



[실습하기] 이클립스를 이용해 자바 프로그램 작성하고 실행하기

- 코드 입력

```
package javatest;

public class HelloJava {

    public static void main(String[] args) {
        // TODO Auto-generated method stub

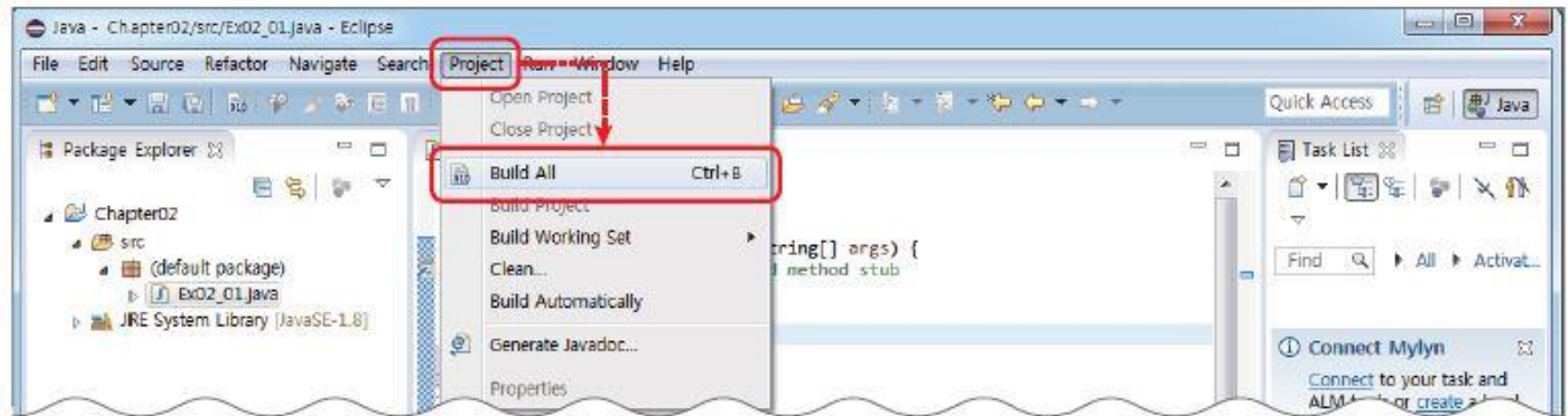
        System.out.println("Hello Java!");
    }

}
```

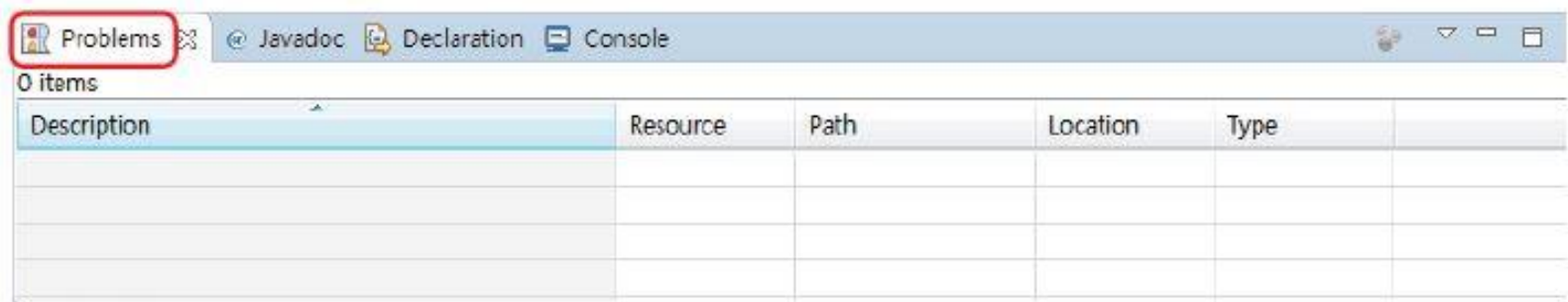
[실습하기] 이클립스를 이용해 자바 프로그램 작성하고 실행하기

- 빌드(=컴파일+링크)

① 작성한 프로그램 빌드. 메뉴에서 [Project]-[Build All] 선택



② 아래쪽의 [Problems] 탭 클릭했을 때 아무 메시지도 나오지 않으면 오류 없음

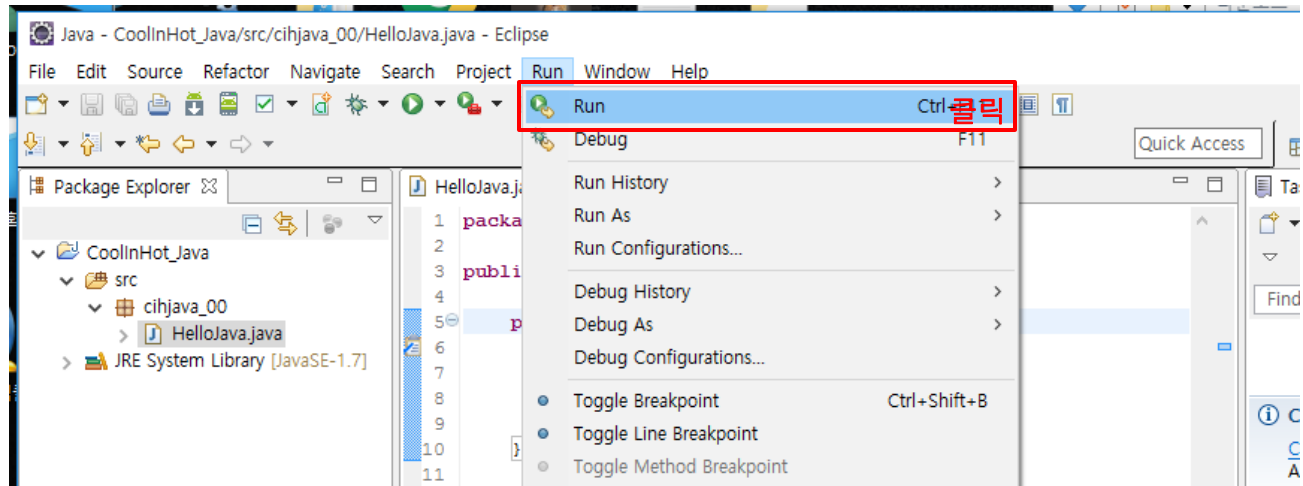


③ 만약 실패가 나오면 소스코드에서 틀린 부분을 수정하고 다시 빌드

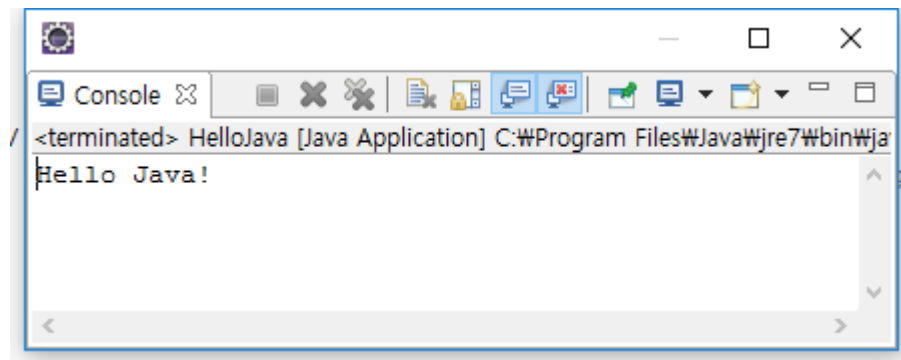
[실습하기] 이클립스를 이용해 자바 프로그램 작성하고 실행하기

- 실행 메뉴 선택

- [Run]-[Run]
메뉴 클릭
(Ctrl+F11)

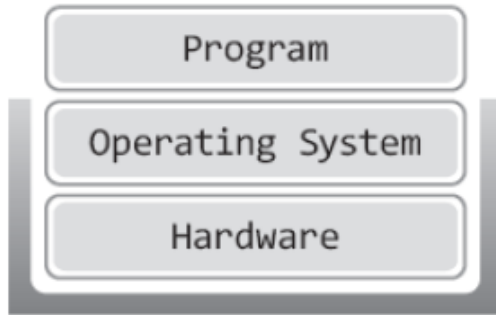


- 실행결과 확인

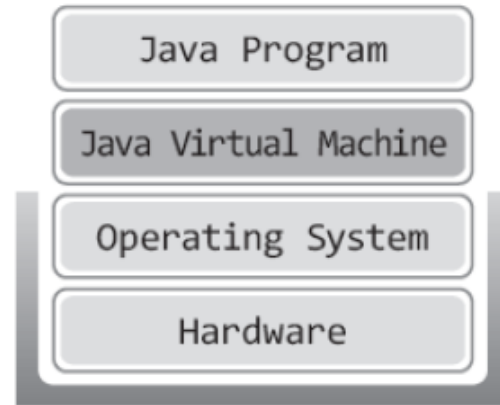


자바 프로그램??

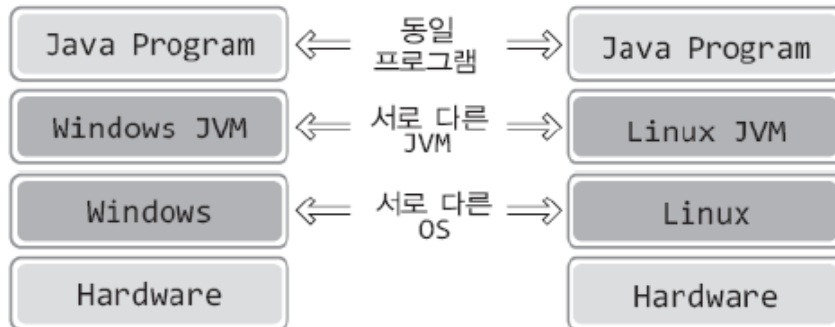
자바 프로그램의 실행의 구조와 자바 가상머신



일반적인 프로그램의 실행구조



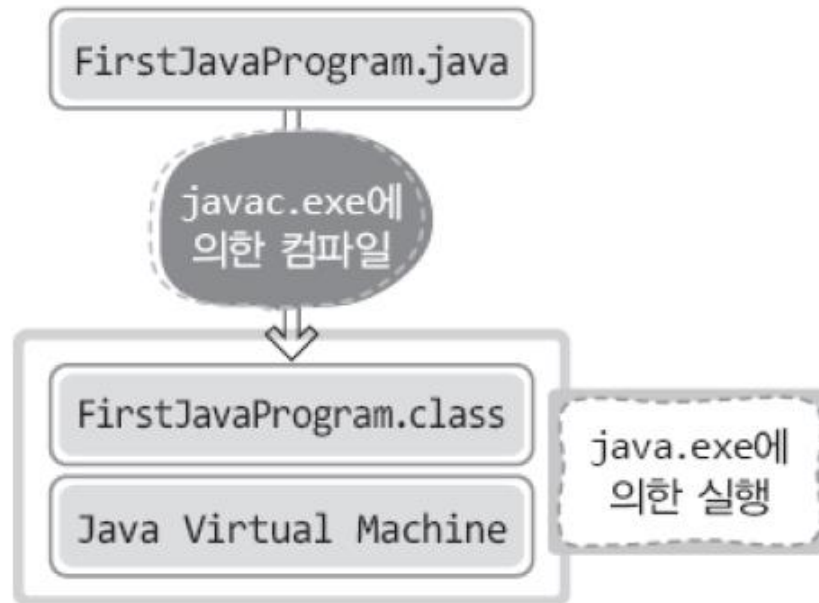
자바 프로그램의 실행구조



운영체제에 독립적인 자바 프로그램

자바 프로그램은 운영체제에 독립적!
자바 가상머신은 운영체제에 의존적!

자바 컴파일러와 자바 바이트 코드

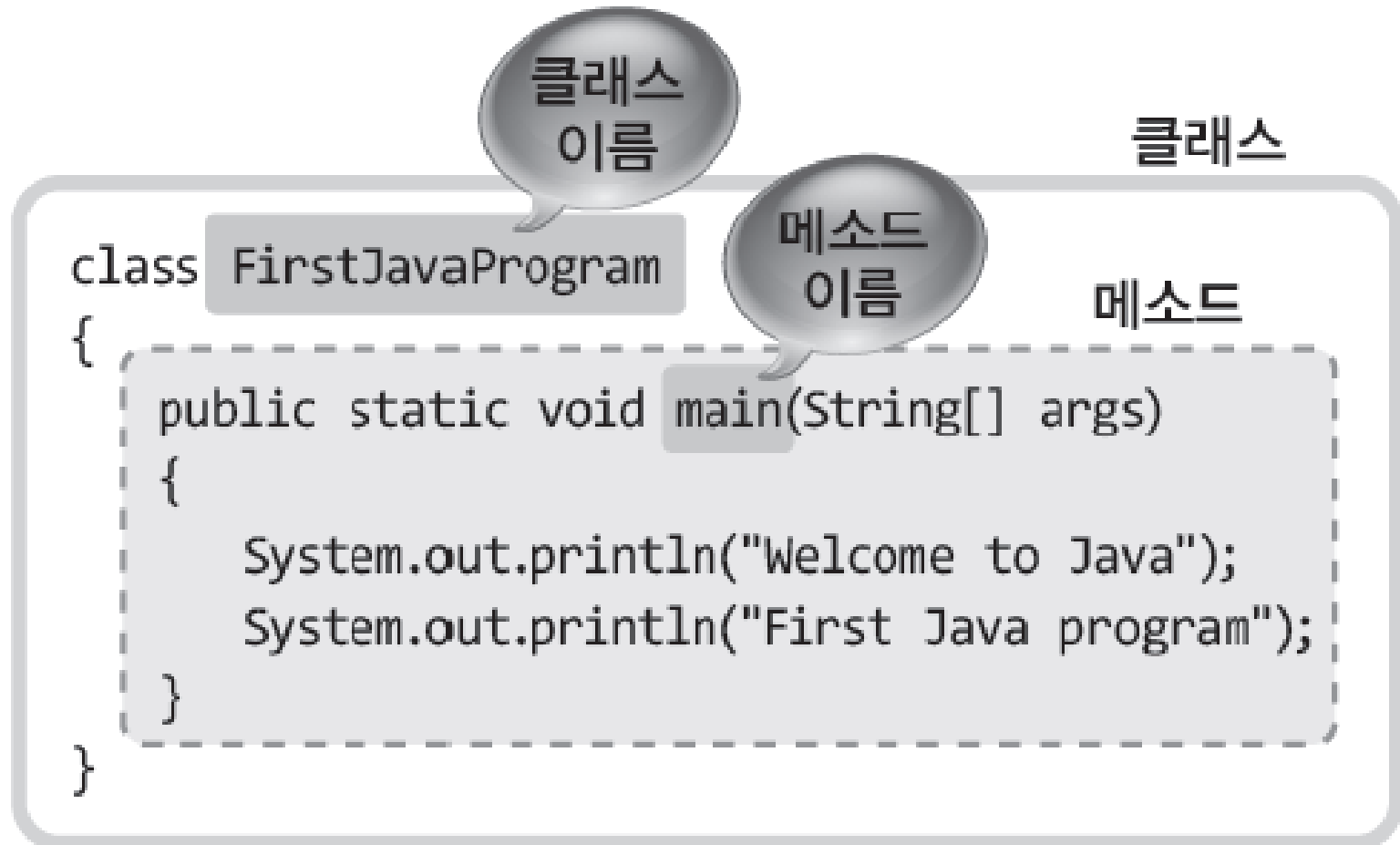


자바의 가상머신 때문에 속도의 저하가?

- 대부분의 소프트웨어는 속도보다 안전성이 우선시 된다.
- 물론 속도는 중요하다. 하지만 여기서 말하는 속도는 소프트웨어의 개발 속도이다.
- 개발중인 소프트웨어의 90% 이상은 인터넷 네트워크 기반의 소프트웨어이다.
- 인터넷, 네트워크상에서는 소프트웨어 속도보다 데이터의 전송 속도가 더 중요하다.
- 속도가 문제가 되면 대부분의 개발자들은 데이터베이스를 먼저 의심한다. 그리고 그곳에서 대부분 문제를 발견하고 해결한다.

결론! 자바의 속도를 문제시 삼지 말자!

프로그램의 골격과 구성



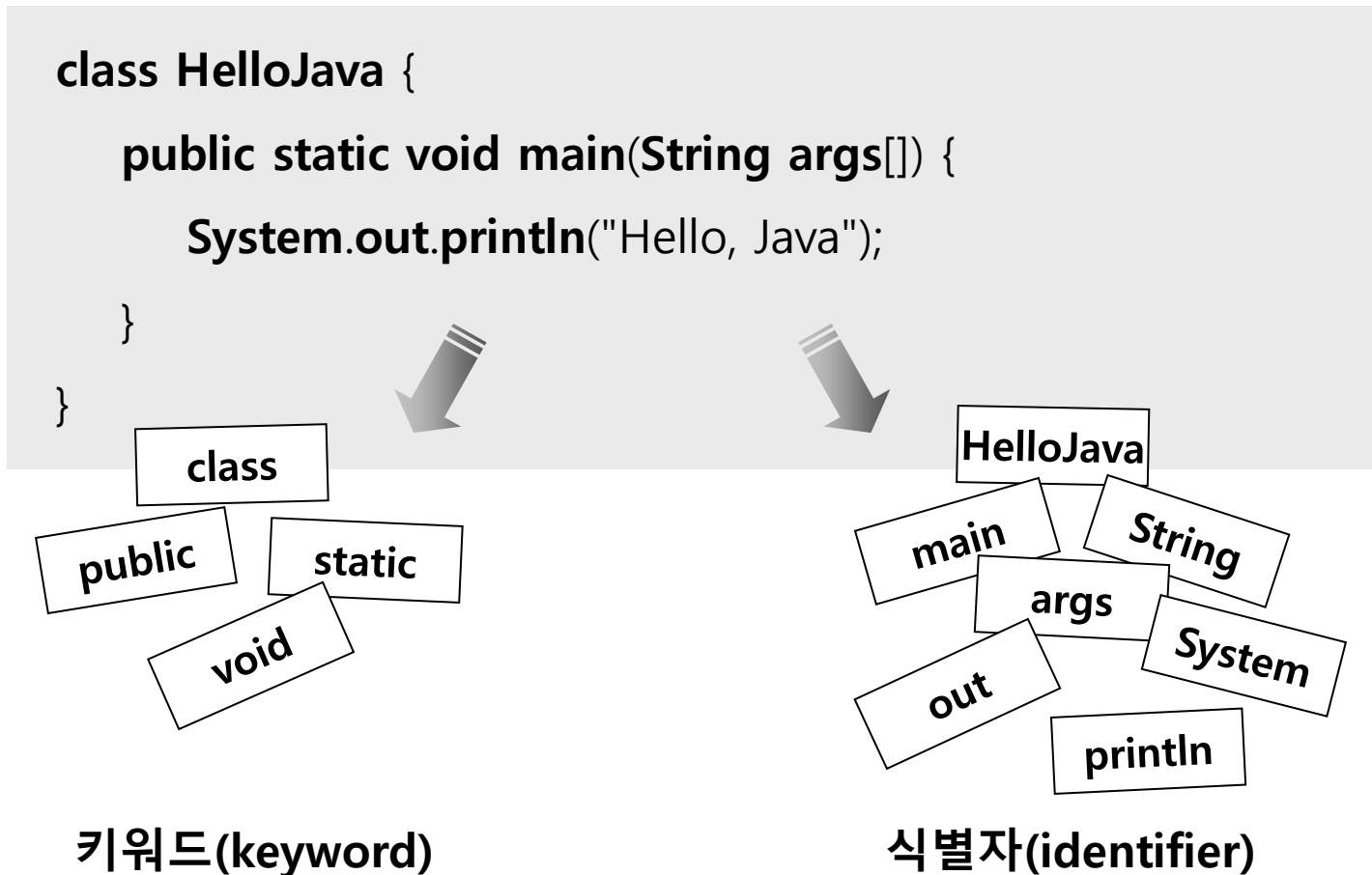
프로그램의 골격과 구성

- 프로그램을 실행시키면 main 메소드 안에 있는 문장이 순차적으로 실행된다.
- 클래스 이름이 **MyClass**이면, 컴파일 시 생성되는 파일의 이름은 **MyClass.class**이다
- `System.out.println`의 괄호 안에 출력하고자 하는 것을 큰 따옴표로 감싸서 넣으면 출력이 된다.
- `System.out.println` 은 출력을 한 다음에 행(Line)을 바꾼다.
- `System.out.println` 과 같이 컴퓨터에게 무엇인가 일을 시키는 문장을 가리켜 '명령문(statement)' 이라 한다. 그리고 이러한 명령문의 끝에는 반드시 세미콜론(;)을 붙여서 명령문의 끝을 표시해야 한다.

자바의 프로그램 작성의 기초

자바 프로그램의 구성요소

- 단어: 키워드(keyword), 식별자(identifier), 상수를 표현하는 단어



자바 프로그램 작성의 기초

자바 프로그램의 구성요소

- 자바의 키워드들

abstract	continue	for	new	switch
assert	default	if	package	synchronized
boolean	do	goto	private	this
break	double	implements	protected	throw
byte	else	import	public	throws
case	enum	instanceof	return	transient
catch	extends	int	short	try
char	final	interface	static	void
class	finally	long	strictfp	volatile
const	float	native	super	while

- 상수 값을 표현하는 단어들

true	false	null
------	-------	------

자바 프로그램 작성의 기초

자바 프로그램의 구성요소

- 여러가지 기호: 대괄호, 중괄호, 소괄호, 마침표, 세미콜론(;) 등

```
class HelloJava {  
    public static void main(String args[]) {  
        System.out.println("Hello, Java");  
    }  
}
```

자바 프로그램 작성의 기초

자바 프로그램의 구성요소

- 데이터: 문자열(string), 문자(character), 정수, 소수 등

```
class HelloJava {  
    public static void main(String args[]) {  
        System.out.println("Hello, Java");  
    }  
}
```

문자열

자바 프로그램 작성의 기초

변수의 선언문

- 변수 : 데이터를 담는 일종의 그릇
- 변수를 사용하기 위해서는 먼저 선언을 해야 함
- 변수의 선언문(declaration statement)

`int num;`

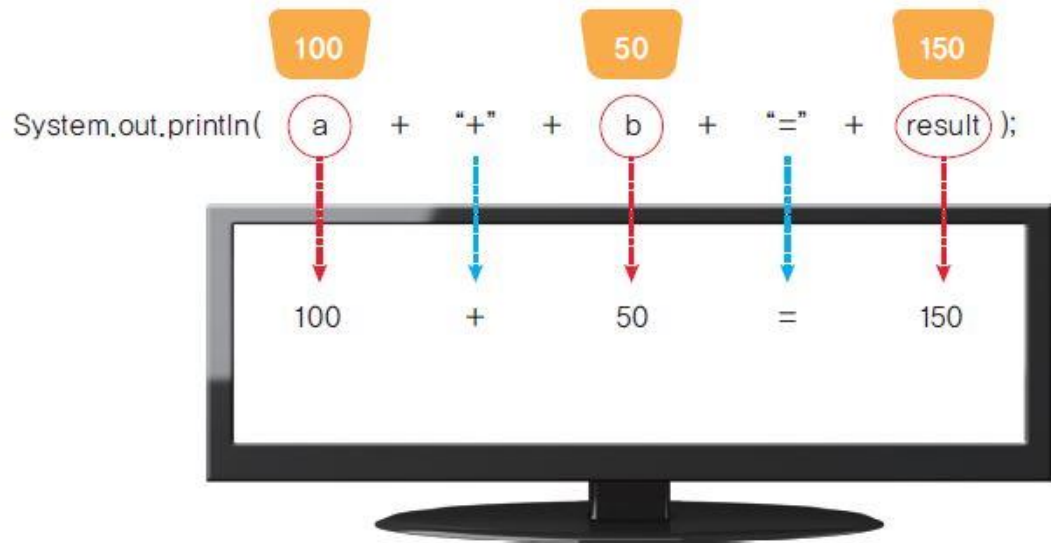
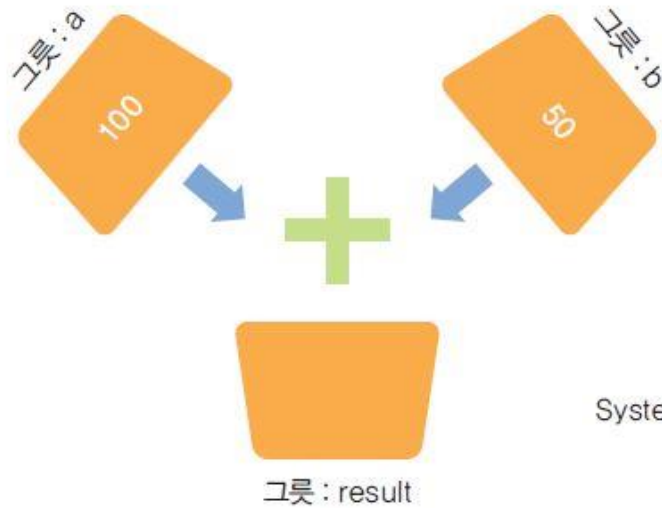
↑ ↑

정수(int) 타입의 값을 num이라는 이름의 변수를
담을 수 있는 선언하는 선언문

← 세미콜론(;)은 명령문의 끝을
표시하는 문자

자바 프로그램 작성의 기초

변수의 선언문



자바 프로그램 작성의 기초

대입문

- 대입문(assignment statement) : 변수에 데이터를 담는 명령문
- 기호 =를 이용해서 만들 수 있음
- 대입문의 예

num = 10 + 20;

↑ ↑

num이라는 10 + 20의 계산 결과를
이름의 변수에 대입하는 대입문

자바 프로그램 작성의 기초

예제

- 선언문과 대입문을 포함하는 프로그램

```
1    class SimpleAdder {  
2        public static void main(String args[]) {  
3            int num;  
4            num = 10 + 20;  
5            System.out.println(num);  
6        }  
7    }
```

자바 프로그램 작성의 기초

조건문

- 조건문(conditional statement) : 조건에 따라 주어진 일을 하는 명령문
- if 키워드를 이용해서 만들 수 있음
- 조건문의 예

```
if (num > 10) {  
    System.out.println("계산 결과가 10보다 큽니다.");  
}
```

num 변수의
값이 10보다 크면
"계산 결과가
10보다 큽니다."라고
출력합니다.

자바 프로그램 작성의 기초

예제

- 조건문을 포함하는 프로그램

```
1  class SimpleAdder2 {  
2      public static void main(String args[]) {  
3          int num;  
4          num = 10 + 20;  
5          if (num > 10)  
6              System.out.println("계산 결과가 10보다 큽니다.");  
7      }  
8  }
```

자바 프로그램 작성의 기초

용어 설명

- 부명령문(substatement) : 다른 명령문 안에 포함된 명령문

```
if (num > 10)
```

```
    System.out.println("계산 결과가 10보다 큼니다.");
```

if 조건문

부명령문
(substatement)

자바 프로그램 작성의 기초

반복문

- 반복문(iterative statement) : 주어진 일을 반복하는 명령문
- while, do, for 키워드를 이용해서 만들 수 있음
- 반복문의 예

```
while (num < 10) {  
    System.out.println("Hello, Java");  
    num = num + 1;  
}
```

num < 10 라는
조건을 만족하는 동안
이 부분을 반복 실행합니다.

자바 프로그램 작성의 기초

예제

- 반복문을 포함하는 프로그램

```
1    class HelloJava1 {  
2        public static void main(String args[]) {  
3            int num = 0;  
4            while (num < 10) {  
5                System.out.println("Hello, Java");  
6                num = num + 1;  
7            }  
8        }  
9    }
```

자바 프로그램 작성의 기초

용어 설명

- 블록(block) : 명령문들을 중괄호로 둘러싼 것

```
while (num < 10) {  
    System.out.println("Hello, Java");  
    num = num + 1;  
}
```

블록(block)

while 문

자바 프로그램 작성의 기초

```
class SystemOutPrintln
{
    public static void main(String[] args)
    {
        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 연산결과는 8입니다.");
        System.out.println(3+5);
    }
}
```

7

3.15

3+5=8

3.15는 실수입니다.

3+5의 연산결과는 8입니다.

8

실행결과

블록(block) 단위 주석 : /* ~ */

```
/*
```

파일이름: BlockComment.java

작성자: 유영진

작성일: 2014년 2월 26일

작성이유: System.out.println 메소드 기능 테스트

```
*/
```

```
class SystemOutPrintln
```

```
{
```

```
    public static void main(String[] args)
```

```
    {
```

```
        System.out.println(7);    /* 정수의 출력 */
```

```
        System.out.println(3.15);
```

```
        System.out.println("3+5=" + 8);
```

```
        System.out.println(3.15 + "는 실수입니다.");
```

```
        System.out.println("3+5" + "의 연산결과는 8입니다.");
```

```
        System.out.println(3+5);    /* 덧셈 결과 출력 */
```

```
    }
```

```
}
```

자바 프로그램 작성의 기초

문제 1.

다음 두 문장을 출력하는 프로그램을 작성해 보자.

```
System.out.println( "2+5=" + 2+5 );
```

```
System.out.println( "2+5=" + (2+5) );
```

문제2.

숫자 15를 총 5회 출력하는 프로그램을 작성해 보자.

단 총 5회에 걸쳐서 출력이 이루어 져야 하고, 이를 위해서 구성이 되는 다섯 문장 모두 차이가 있어야 한다.

회원만 가입하면 강의가 평생 무료
온라인 학습 사이트/소스코드
www.orentec.co.kr에서 이 책의 소스코드를 다운로드
할 수 있으며, 회원으로 가입만 하시면 본 도서의 강의를 평생 무료로
수강하실 수 있습니다.

탄탄한 내용으로 무장하고 체계적인 학습을 유도하는
새로운 개념의 JAVA 입문서!

나는 강박질 JAVA를 공부한 적이 없대요

윤성우 저 | 김문석 감수

이 책 한권으로 JAVA의 기본이 탄탄해집니다.

- 초보자가 알아야 할 JAVA의 기본문법을 제대로 설명한 기본서
- 객체지향 전문가의 기획으로 올바른 객체지향적 사고를 유도하는 입문서
- 이해를 돕는 풍부한 양의 그림과 해설에 있어서의 다양한 접근방식 적용
- 초보자도 쉽게 따라올 수 있는 저자 윤성우의 무료강의 제공

ORANGE MEDIA



[Http://www.orentec.co.kr](http://www.orentec.co.kr)

PPT 와 동영상 강의 참고