

# 객체지향 프로그래밍



## ❖ 목차

- 교과목 개요
- 학습 내용
- 평가 방법



## ■ 강의명 : 객체지향 프로그래밍

## ■ 개 요

- 객체지향 프로그래밍이란 프로그램을 개발하는 기법으로 부품에 해당하는 객체를 먼저 만들고 이것들을 하나씩 조립 및 연결해서 전체 프로그램을 완성하는 기법이다
- 자바는 객체지향 언어가 가져야 할 캡슐화, 상속, 다형성 기능을 완벽하게 지원한다
- 본 과정을 통하여 자바 언어에 대한 기초적인 문법을 익히고 객체지향 프로그래밍의 개념을 학습한다

## ■ 교 재

‘이것이 자바다 – 신용권의 Java 프로그래밍 정복’, 한빛미디어, 2018년



## ■ 주차별 학습 내용

주차	학습 내용	주차	학습 내용
1	강의 소개 및 프로그램 개발 환경 구축 (비대면 수업으로 과제물 기반 원격수업)	9	클래스2
2	자바 기초 문법 익히기 – 변수와 타입 (비대면 수업으로 과제물 기반 원격수업)	10	상속
3	연산자와 연산식	11	인터페이스
4	조건문	12	중첩 클래스
5	반복문	13	예외 처리
6	참조타입(배열)	14	스레드 및 기본 API 클래스
7	객체지향 개념 익히기 – 클래스1	15	기말고사
8	중간고사		



## ■ 평가비율

- 중간고사 : 40%
- 기말고사 : 40%
- 과제 : 10%
- 출석 : 10%

## ■ 평가기준

- 개념과 원리의 이해도 및 실기 결과의 완성도



## ❖ 목차

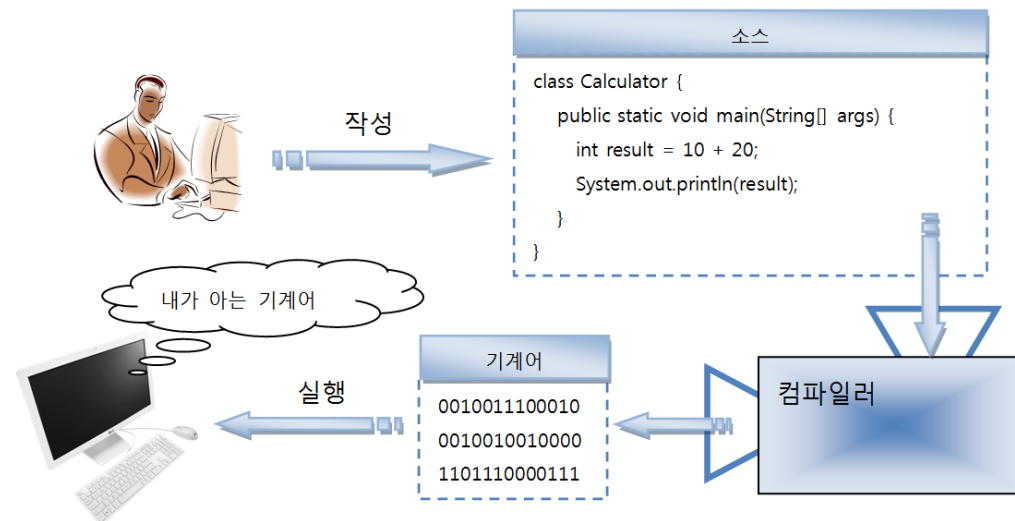
- 1절. 프로그래밍 언어란?
- 2절. 자바란?
- 3절. 자바 개발 환경 구축
- 4절. 자바 프로그램 개발 순서
- 5절. 이클립스 설치



# 1절. 프로그래밍 언어

## ❖ 프로그래밍 언어의 역할

- 사람과 컴퓨터 간의 커뮤니케이션(대화)
- 사람의 언어와 기계어 사이에서 다리와 같은 역할
- 컴퓨터의 처리를 인간이 원하는 의도대로 하기 위해서 일련의 명령어로 작업을 지시하는 것
- 컴퓨터가 이해할 수 있는 커뮤니케이션 방식이 필요
- 고급 언어(C, C++, Java)와 저급 언어(Assembly)로 구분



## 2절. 자바란?

### ❖ 자바 언어의 역사

- 1995년 썬마이크로시스템즈(Sun Microsystems)에서 최초 발표한 언어
- 1991년 가전 제품에서 사용할 목적인 오크(Oak) 언어에서부터 시작
- 1995년 Netscape 사의 브라우저가 자바 기술을 채택
- 인터넷 프로그래밍 언어로 발전하면서 자바라는 이름으로 변경
- Java라는 이름은 인도네시아산 커피 원료 이름에서 유래  
(잠들지 않는 인터넷과 의미가 상통)
- 2010년 오라클에서 썬을 인수하여 Java 개발,관리,배포 주관

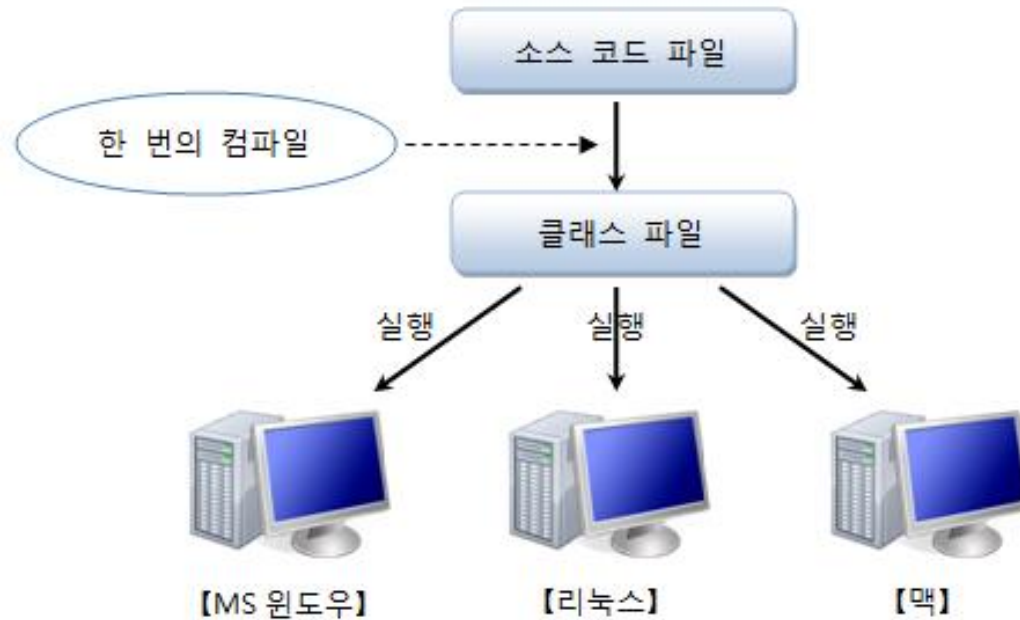




## 2절. 자바란?

### ❖ 자바의 특징

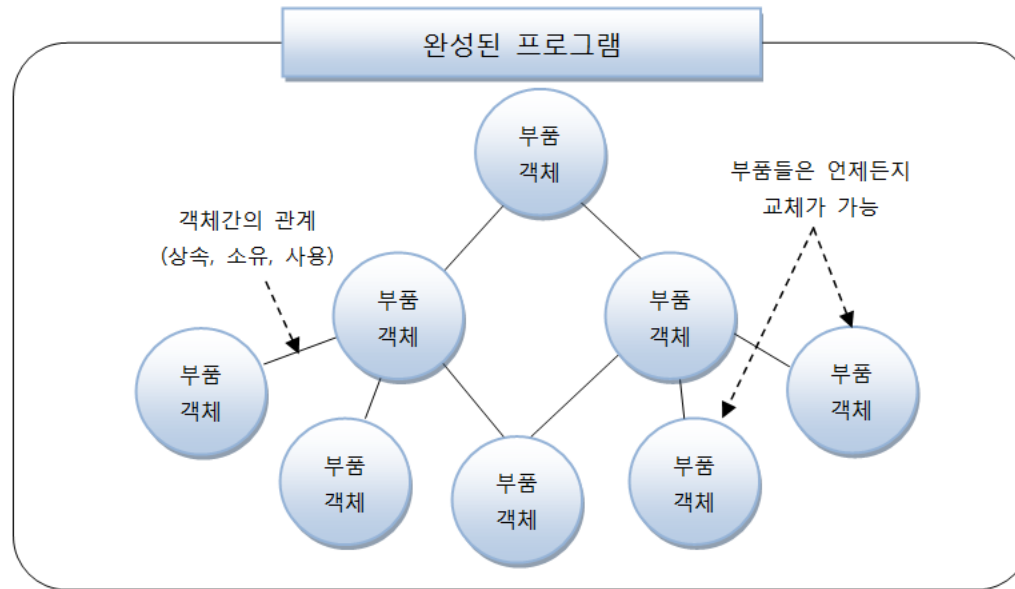
- 이식성이 높은 언어



## 2절. 자바란?

### ❖ 자바의 특징

- 객체 지향 언어 -> OOP(Object Oriented Programming)란?
  - 부품 객체를 먼저 만들고, 이것들을 조합해 전체 프로그램을 완성하는 기법
- 자바는 처음부터 OOP 개발용 언어로 설계
  - 캡슐화, 상속, 다형성 기능 완벽하게 지원



## 2절. 자바란?

### ❖ 자바의 특징

- 함수적 스타일 코딩 지원
  - 함수적 스타일 코딩 방식인 람다식(Lambda Expressions) 지원
    - 코드 간결하게 작성 가능
    - 컬렉션 요소를 필터링, 매핑, 그룹핑, 집계 처리시 주로 사용
- 메모리를 자동으로 관리
  - 사용하지 않는 객체를 자동으로 메모리에서 제거
  - 핵심 기능 코드에 집중할 수 있도록 하는 기능
- 다양한 애플리케이션 개발 가능



## 2절. 자바란?

### ❖ 자바의 특징

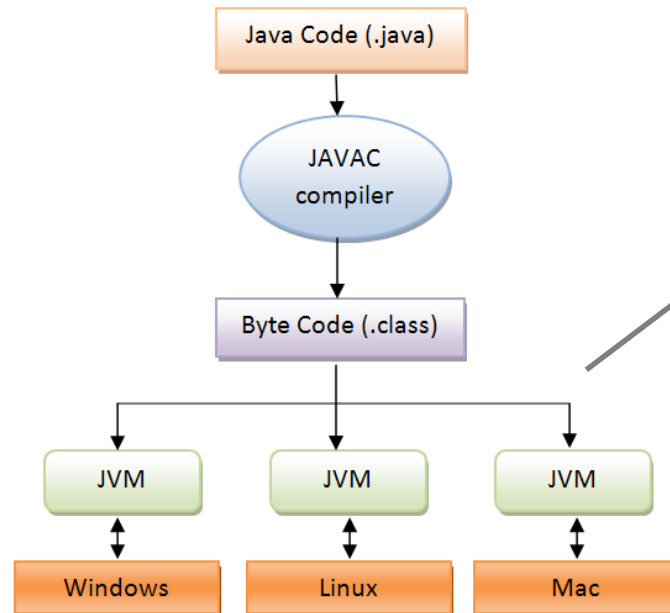
- 멀티 스레드(Multi-Thread) 쉽게 구현 가능
  - 동시에 여러 가지 작업을 할 경우
  - 대용량 작업을 빨리 처리할 경우
- 동적 로딩(Dynamic Loading) 지원
  - 미리 객체를 만들어 놓지 않고 필요한 시점에 동적 로딩해 객체 생성
  - 유지 보수 시 특정 객체만 쉽게 수정 및 교체해 사용
- 막강한 오픈 소스 라이브러리 풍부



### 3절. 자바 개발 환경 구축

#### ❖ 자바 가상 기계(JVM:Java Virtual Machine)

- 실제 운영체제를 대신해서 자바 프로그램을 실행하는 가상의 운영체제
  - 운영체제는 자바 프로그램을 바로 실행할 수 없다
  - 자바 프로그램은 완전한 기계어가 아닌 중간 단계의 바이트 코드이다
  - 운영체제와 자바 프로그램을 중계하는 JVM을 두어 자바 프로그램이 여러 운영체제에서 동일한 실행 결과가 나오도록 설계한 것이다



- 바이트 코드(.class)는 하나이다  
- JVM에 의해서 번역되는 기계어는 운영체제에 따라서 달라진다  
(JVM은 운영체제에 종속적)



### 3절. 자바 개발 환경 구축

#### ❖ 자바 개발 도구(JDK) 설치

##### ■ Java Standard Edition 구현체의 종류

- JRE(Java Runtime Environment) = JVM + 표준 클래스 라이브러리(API)
  - 자바 프로그램을 실행만 할 경우 설치

- **JDK(Java Development Kit)** = JRE + 개발에 필요한 도구
  - 자바 프로그램을 개발하고 실행하기 위해 반드시 설치

##### ■ JDK(Java Development Kit) 설치

- 설치 파일 다운로드 사이트: <http://www.oracle.com>

##### ■ API 도큐먼트

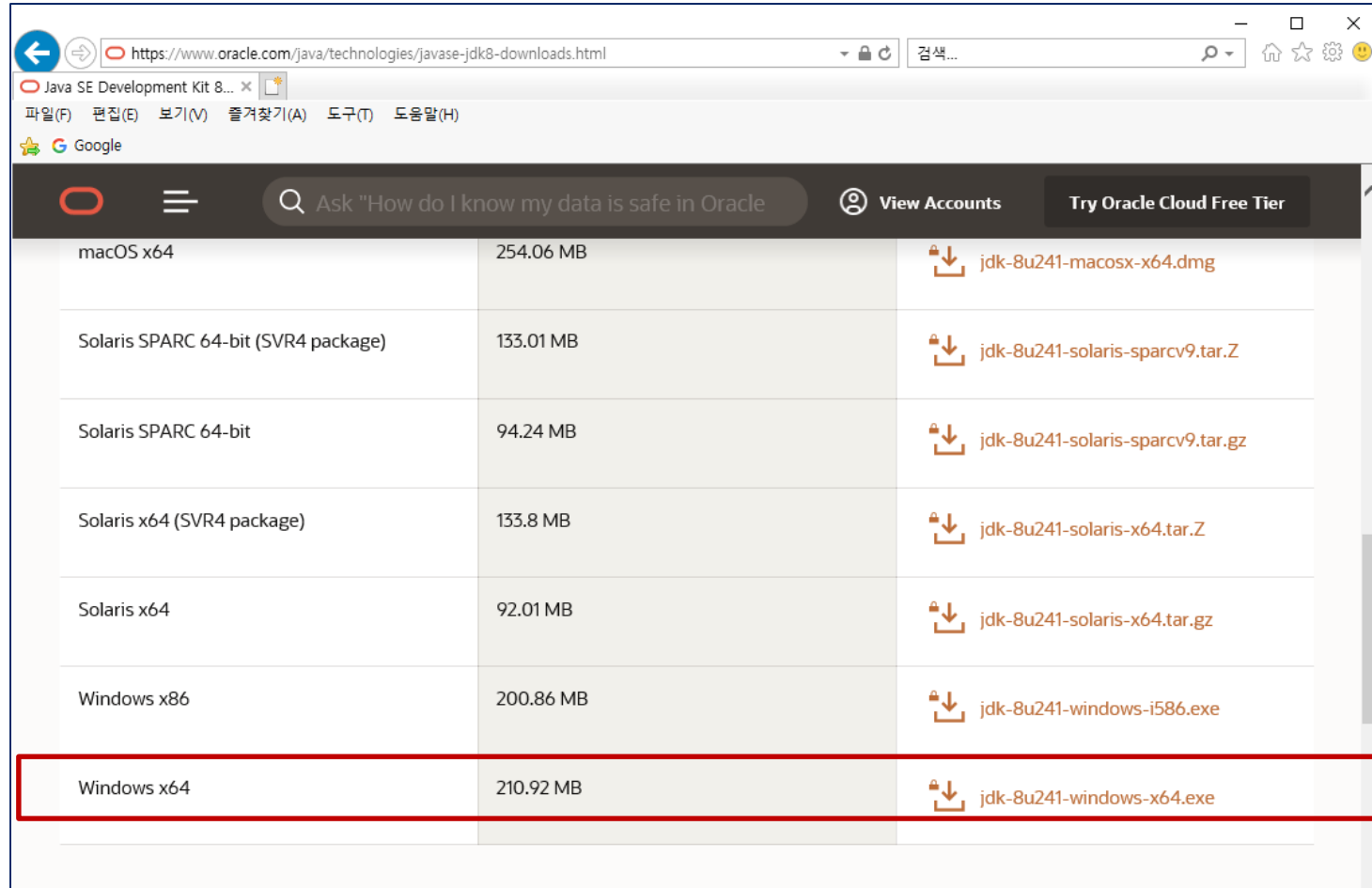
- JDK에서 제공하는 표준 클래스 라이브러리를 설명해 놓은 HTML 페이지들
- <http://docs.oracle.com/javase/버전/docs/api/>










# 3절. 자바 개발 환경 구축

## ❖ 자바 개발 도구(JDK) 설치

- [https:// www.oracle.com/java/technologies/javase-jdk8-downloads.html](https://www.oracle.com/java/technologies/javase-jdk8-downloads.html)



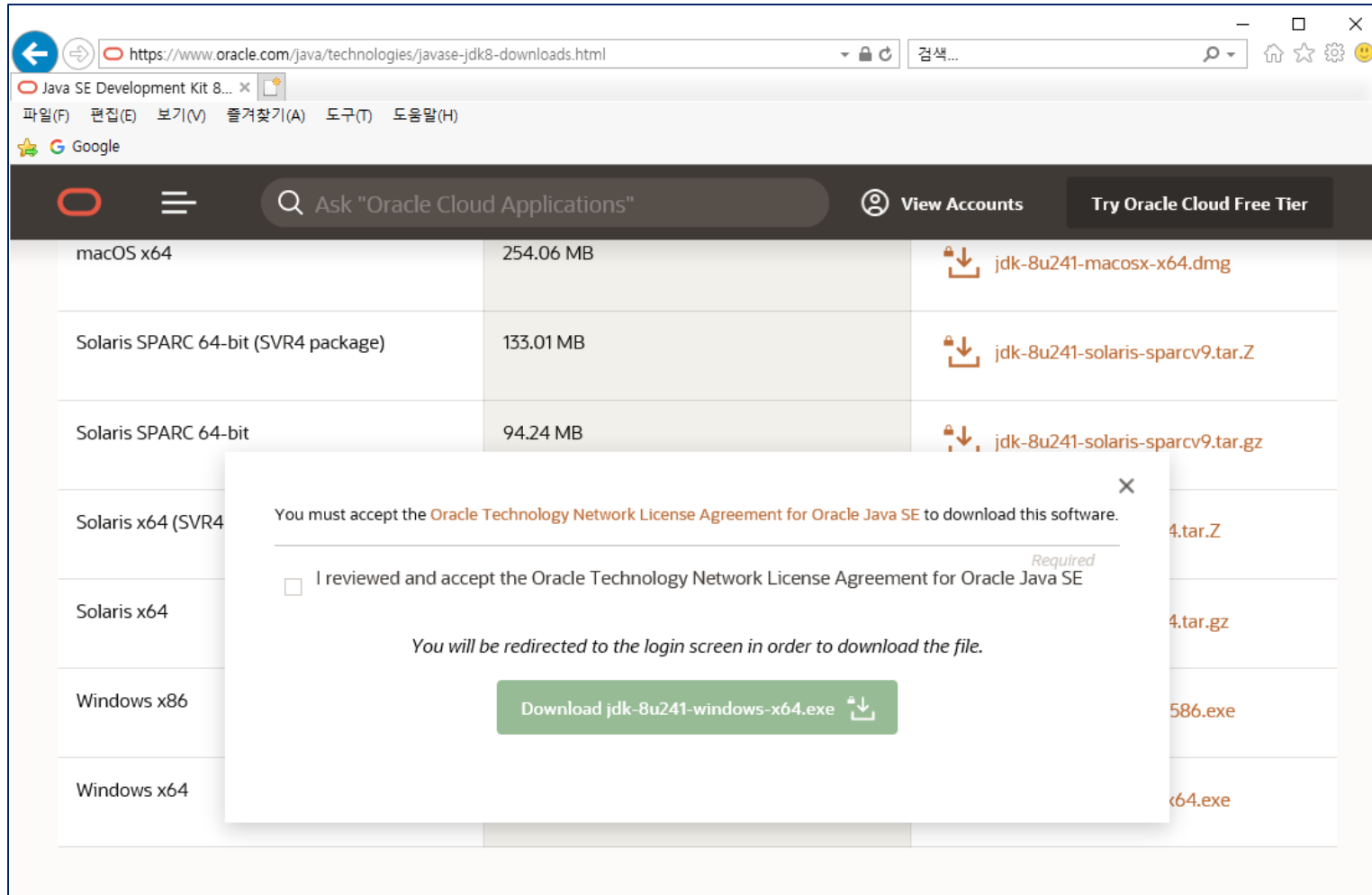
macOS x64	254.06 MB	 <a href="#">jdk-8u241-macosx-x64.dmg</a>
Solaris SPARC 64-bit (SVR4 package)	133.01 MB	 <a href="#">jdk-8u241-solaris-sparcv9.tar.Z</a>
Solaris SPARC 64-bit	94.24 MB	 <a href="#">jdk-8u241-solaris-sparcv9.tar.gz</a>
Solaris x64 (SVR4 package)	133.8 MB	 <a href="#">jdk-8u241-solaris-x64.tar.Z</a>
Solaris x64	92.01 MB	 <a href="#">jdk-8u241-solaris-x64.tar.gz</a>
Windows x86	200.86 MB	 <a href="#">jdk-8u241-windows-i586.exe</a>
Windows x64	210.92 MB	 <a href="#">jdk-8u241-windows-x64.exe</a>



# 3절. 자바 개발 환경 구축

## ❖ 자바 개발 도구(JDK) 설치

- [https:// www.oracle.com/java/technologies/javase-jdk8-downloads.html](https://www.oracle.com/java/technologies/javase-jdk8-downloads.html)

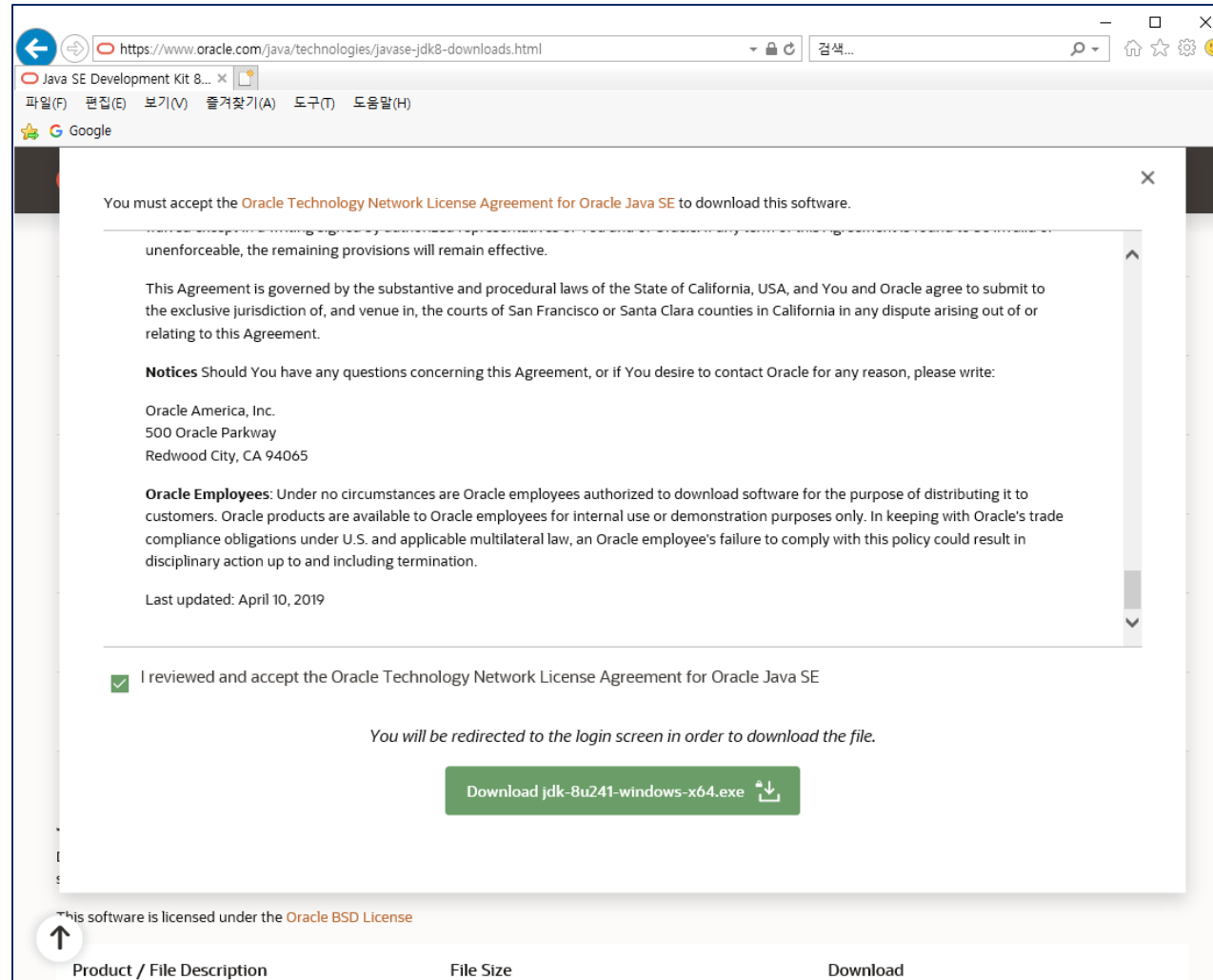




# 3절. 자바 개발 환경 구축

## ❖ 자바 개발 도구(JDK) 설치

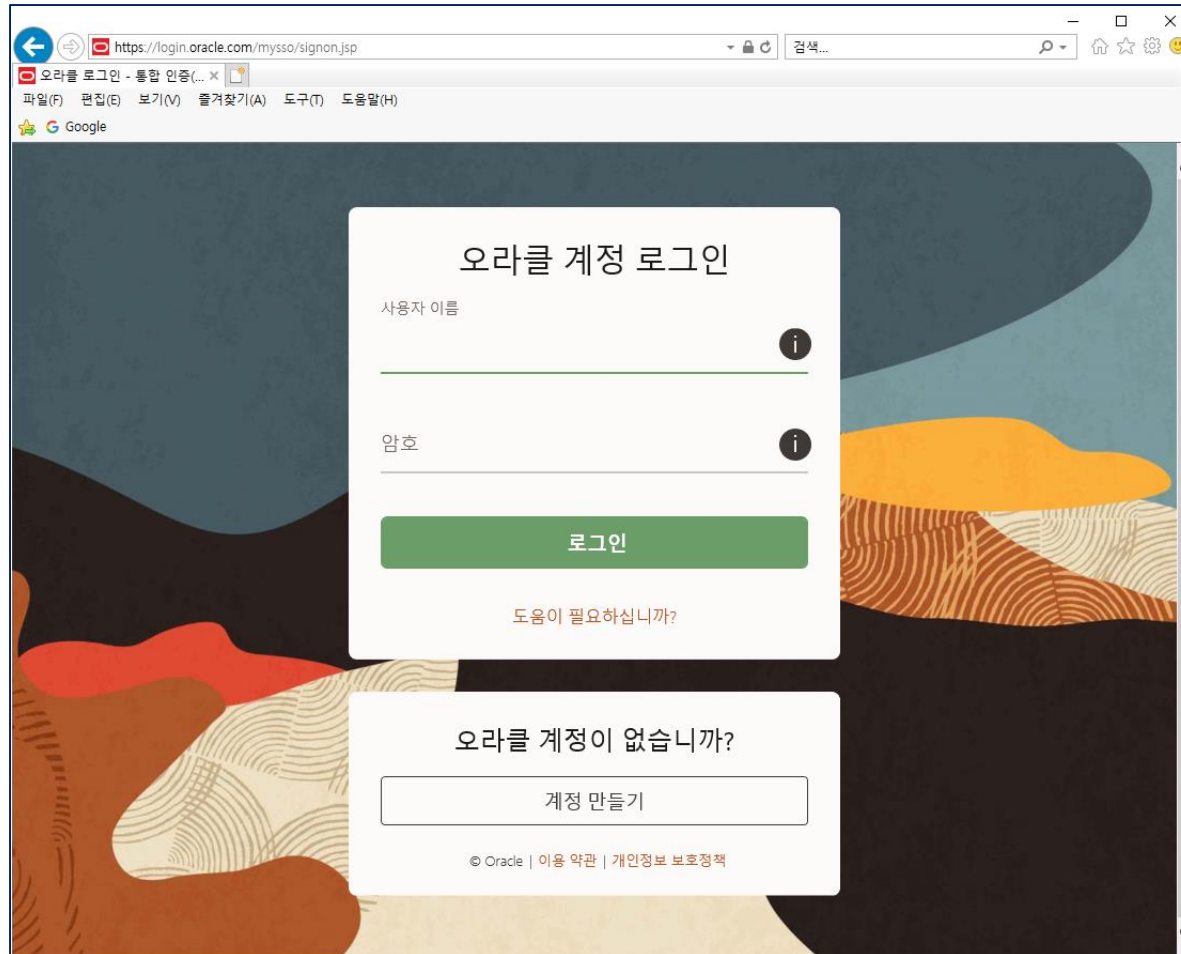
- [https:// www.oracle.com/java/technologies/javase-jdk8-downloads.html](https://www.oracle.com/java/technologies/javase-jdk8-downloads.html)



# 3절. 자바 개발 환경 구축

## ❖ 자바 개발 도구(JDK) 설치

- [https:// www.oracle.com/java/technologies/javase-jdk8-downloads.html](https://www.oracle.com/java/technologies/javase-jdk8-downloads.html)



# 3절. 자바 개발 환경 구축

## ❖ 자바 개발 도구(JDK) 설치

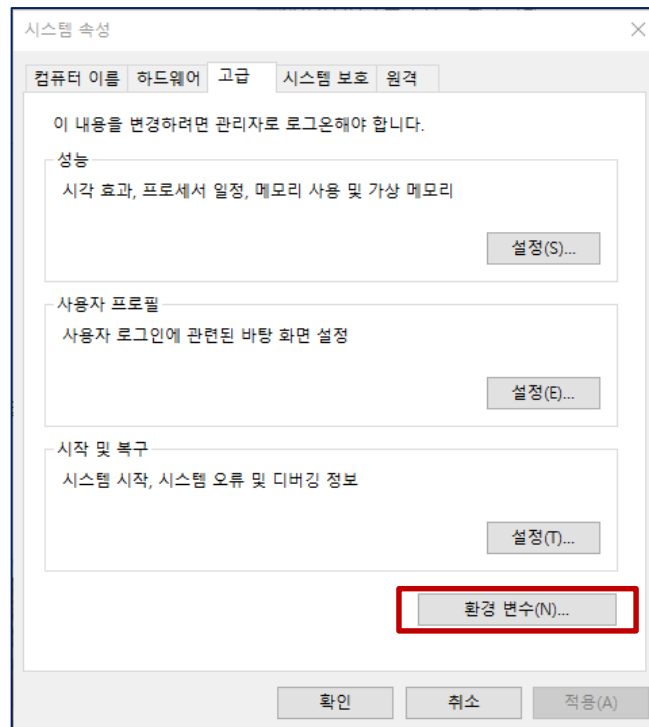
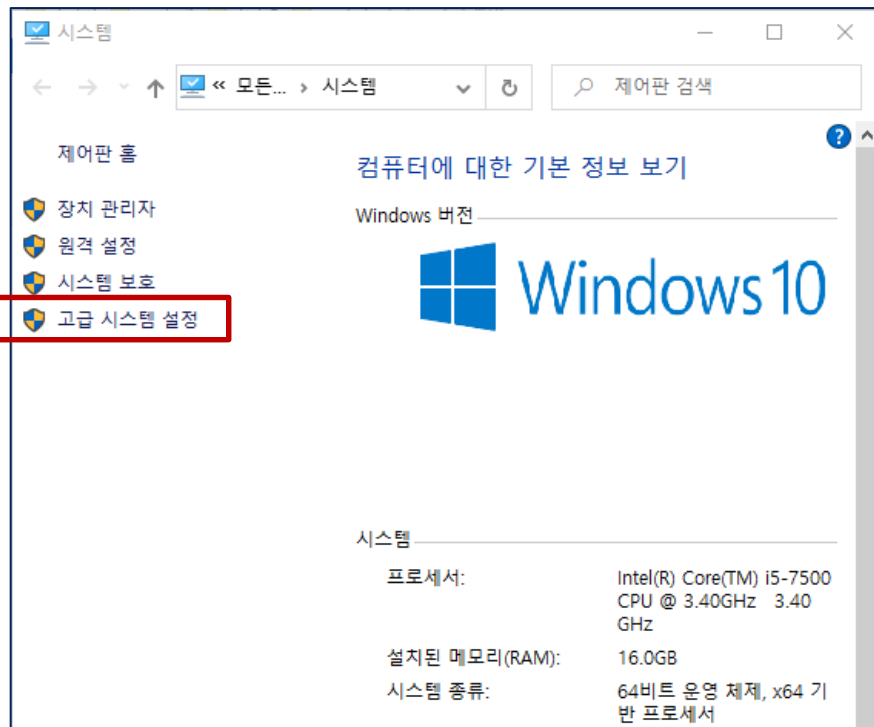
- [https:// www.oracle.com/java/technologies/javase-jdk8-downloads.html](https://www.oracle.com/java/technologies/javase-jdk8-downloads.html)
- 파일명 : jdk-8u241-windows-x64.exe
- 설 치
  - 다운로드가 끝나면 받은 파일을 클릭해서 설치한다
  - 설치 과정은 [Next] 버튼만 클릭하면 된다
  - 설치 경로를 변경하고자 할 경우에는 반드시 경로를 기억한다



# 3절. 자바 개발 환경 구축

## ❖ 환경변수 설정

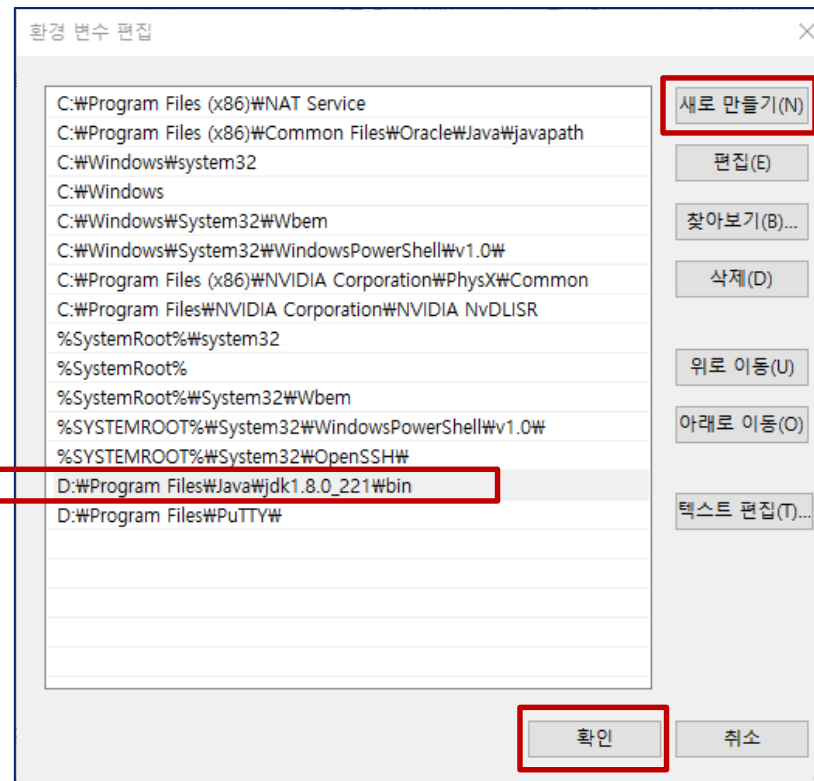
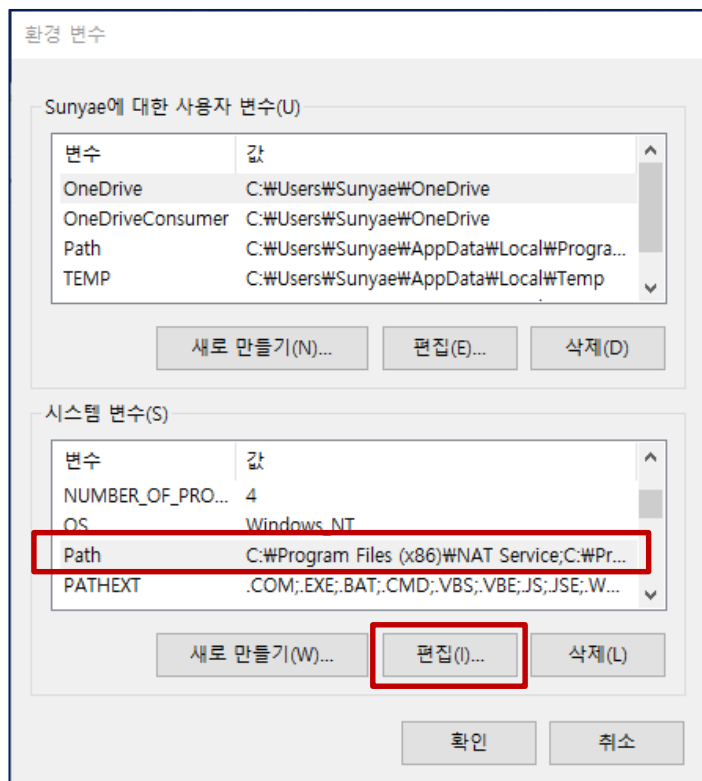
- ✓ 내컴퓨터 > 마우스 우클릭 > 속성 > 고급 시스템 설정 > 고급 탭 > 환경변수 버튼 선택
- ✓ 새로 만들기 클릭 > Java 가 설치된 디렉터리 안의 JDK 아래의 bin 디렉토리 경로를 기입한다.  
(C:\Program Files\Java\jdk1.8.0\_201\bin)



### 3절. 자바 개발 환경 구축

#### ❖ 환경변수 설정

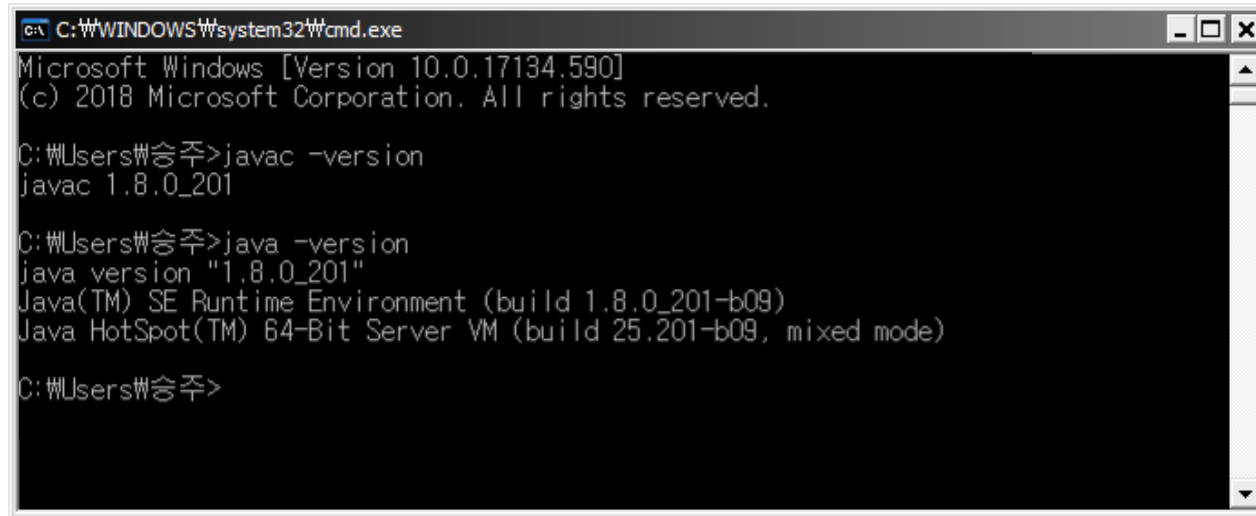
- ✓ 내컴퓨터 > 마우스 우클릭 > 속성 > 고급 시스템 설정 > 고급 탭 > 환경변수 버튼 선택
- ✓ 새로 만들기 클릭 > Java 가 설치된 디렉터리 안의 JDK 아래의 bin 디렉토리 경로를 기입한다.  
(C:\Program Files\Java\jdk1.8.0\_201\bin)
- ✓ '확인' 을 눌러서 설정된 환경변수를 저장한다



# 3절. 자바 개발 환경 구축

## ❖ JDK 설치 확인

- ✓ Window Key+R > cmd를 실행하여 명령 프롬프트를 켜준다
- ✓ Javac -version 명령어와 java -version 명령어를 실행한다
- ✓ java와 javac에 대한 버전 정보가 화면에 뜨면 성공적으로 설치된 것이다



```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Version 10.0.17134.590]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\승주>javac -version
javac 1.8.0_201

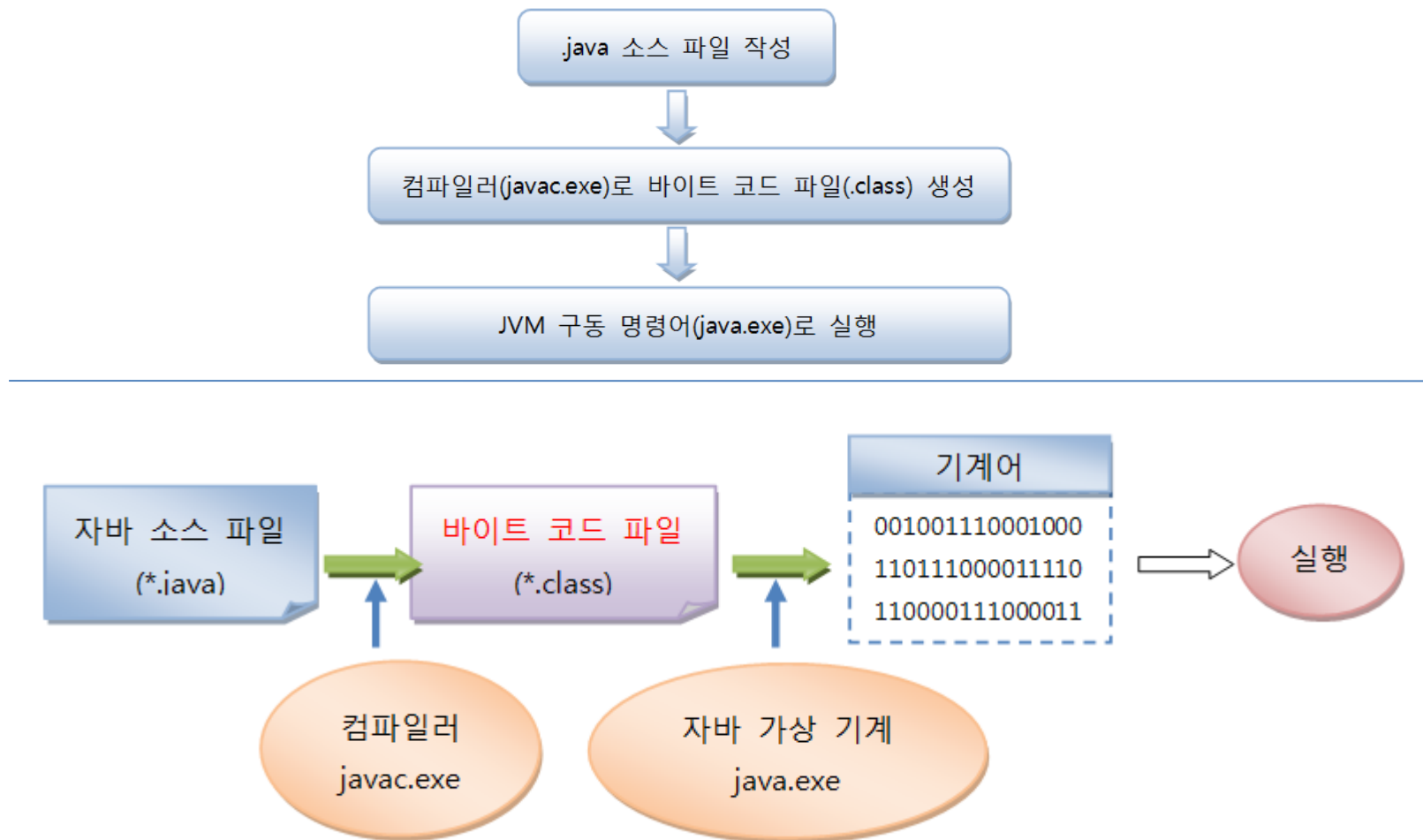
C:\Users\승주>java -version
java version "1.8.0_201"
Java(TM) SE Runtime Environment (build 1.8.0_201-b09)
Java HotSpot(TM) 64-Bit Server VM (build 25.201-b09, mixed mode)

C:\Users\승주>
```



## 4절. 자바 프로그램 개발 순서

### ❖ 소스 작성에서부터 실행까지



## 4절. Hello Java

- ❖ 메모장을 이용하여 작성 및 실행
- ❖ C:/java/src에 Hello.java란 파일이름으로 저장

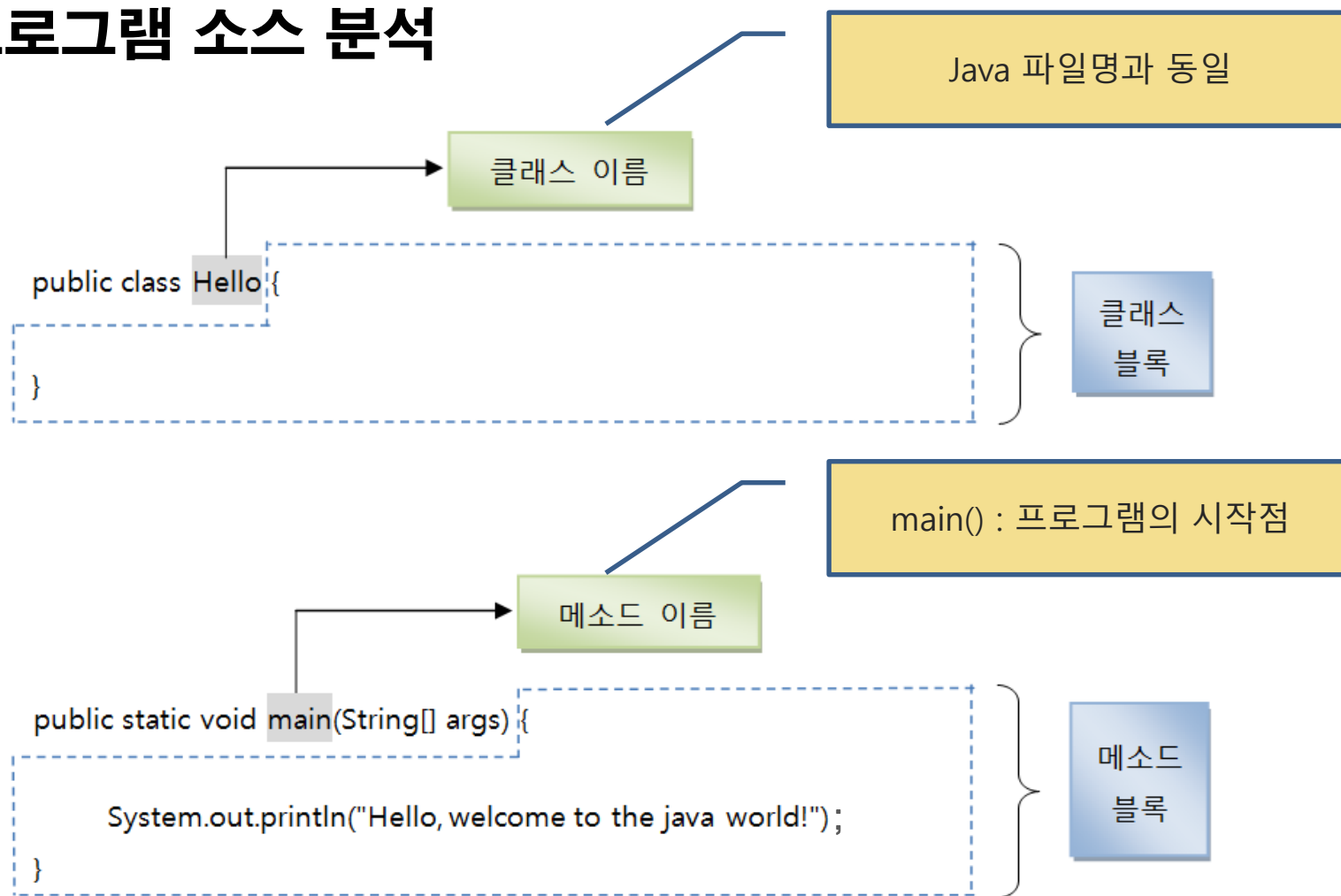
```
// 프로그램 : Hello.java
public class Hello {
    public static void main (String[] args) {
        System.out.println( "Hello, welcome to the java world!" );
    }
}
```





# 4절. Hello Java

## ❖ 프로그램 소스 분석



## 4절. Hello Java

### ❖ Class 정의

- 자바 프로그램은 확장자가 .java 인 소스 파일을 만들어서 자바 문법에 맞는 내용을 기술해야 한다
- Class를 하나의 단위로 프로그램을 작성하므로 자바 소스파일 안에 class를 정의해야 한다
- 반드시 자바 소스 파일명과 class 명은 동일해야 한다

### ❖ Main 메소드 정의

- 프로그램의 시작지점이다
- 자바 프로그램을 실행시키면 main함수 내부에 기술된 내용들을 순차적으로 수행한다



## 4절. Hello Java

### ❖ Main 메소드 정의

■ public static void main(String[] args)

①

②

③

④

⑤

- ① **public** : 누구나 접근 가능하도록 하기 위한 예약어로 접근 지정자의 일종
- ② **static** : **static**으로 선언된 메소드는 클래스만 존재하면 수행할 수 있도록 한다
- ③ **void** : 값을 갖지 않는다는 의미를 갖는 자료 형태이다
- ④ **main**
  - 자바는 JVM에 의해서 실행되는데 자바 어플리케이션을 실행시키면 JVM은 **main** 메소드를 찾아 내부에 기술된 내용들을 순차적으로 실행한다
  - 그래서 **main** 메소드를 프로그램의 시작점(혹은 진입점)이라고 한다
- ⑤ **String[] args**
  - 메소드를 실행시키기 위해 데이터를 전달받아야 할 경우 사용되는 파라미터
  - ( ) 안에 기술하며 (데이터타입 데이터저장변수) 형태로 기술



## 4절. Hello Java

### ❖ 실행문

- main 메소드까지 정의했다면 이 메소드 안에 수행할 내용을 기술
- 문장 단위로 프로그램을 작성
- 메소드 내부에 기술할 문장으로는 변수의 선언문이나 다른 메소드를 호출하는 문장들이 있다
- 반드시 문장의 끝은 세미콜론(;)으로 마무리해야 한다
- 아래 문장은 화면에 “Hello, welcome to the java world!” 를 출력하라는 실행문이다

```
System.out.println( “Hello, welcome to the java world!” );
```



## 4절. Hello Java

### ❖ 컴파일(Compile)

- 컴파일러에 의해서 소프 프로그램을 기계어(0과 1로 구성된 이진파일) 형태로 번역한 후, 컴퓨터가 기계어 파일을 실행하는 방식

#### 컴파일(compile)이란?

- 컴파일러에 의해서 소프 프로그램을 기계어(0과 1로 구성된 이진 파일)로 번역한 후, 컴퓨터가 기계어 파일을 실행하는 방식



- 명령 프롬프트(cmd창) 화면에서 java\src 디렉토리로 이동 후,  
C:\java\src> **javac Hello.java** 명령문 실행



## 4절. Hello Java

### ❖ 바이트코드(class 파일) 확인

- 자바 소스파일을 컴파일하면 생성됨

```
C:\Windows\System32\cmd.exe

C:\wjava\src>javac Hello.java

C:\wjava\src>dir
C 드라이브의 볼륨에는 이름이 없습니다.
볼륨 일련 번호: 4C04-BCE5

C:\wjava\src 디렉터리

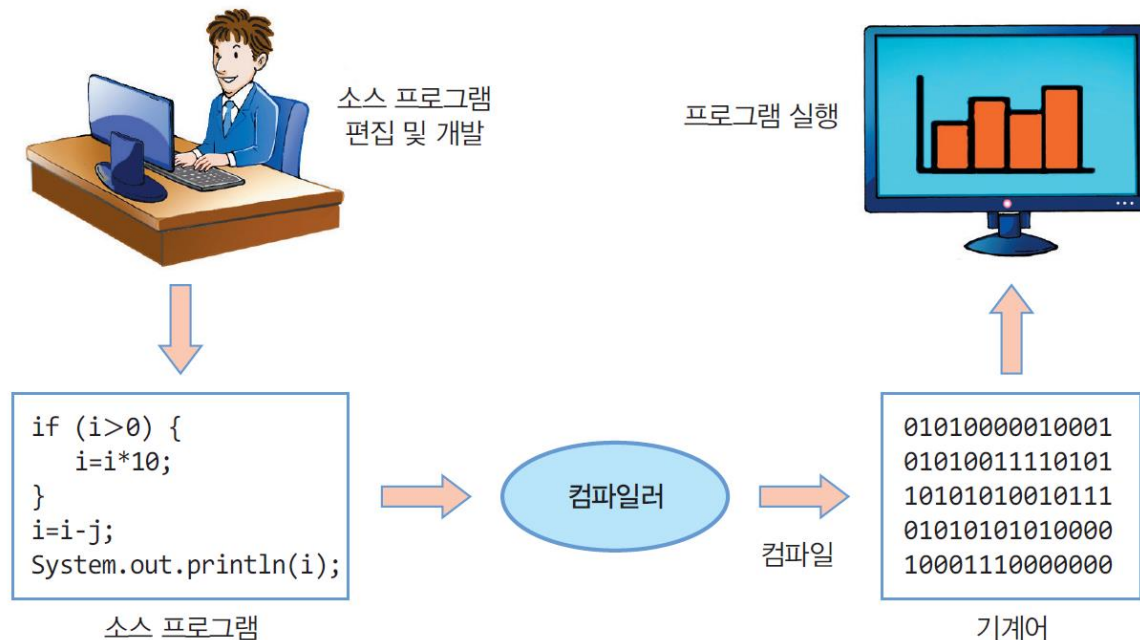
2020-03-14 오전 11:25 <DIR> .
2020-03-14 오전 11:25 <DIR> ..
2020-03-14 오전 11:25      437 Hello.class
2020-03-14 오전 11:25     145 Hello.java
2020-03-13 오후 10:14     423 HelloJava.class
2020-03-13 오후 09:54     127 HelloJava.java
2020-03-13 오후 10:37     874 JavaWorld.class
2020-03-13 오후 10:19     633 JavaWorld.java
                6개 파일             2,639 바이트
                2개 디렉터리  21,740,081,152 바이트 남음

C:\wjava\src>
```



# 컴파일

- ❖ 소스 : 프로그래밍 언어로 작성된 텍스트 파일
- ❖ 컴파일 : 소스 파일을 컴퓨터가 이해할 수 있는 기계어로 만드는 과정
  - 소스 파일 확장자와 컴파일 된 파일의 확장자
    - 자바 : .java -> .class
    - C : .c -> .obj -> .exe
    - C++ : .cpp -> .obj -> .exe



## 4절. Hello Java

### ❖ 바이트코드란?

- 플랫폼에 독립적인 자바 코드 (class 파일)

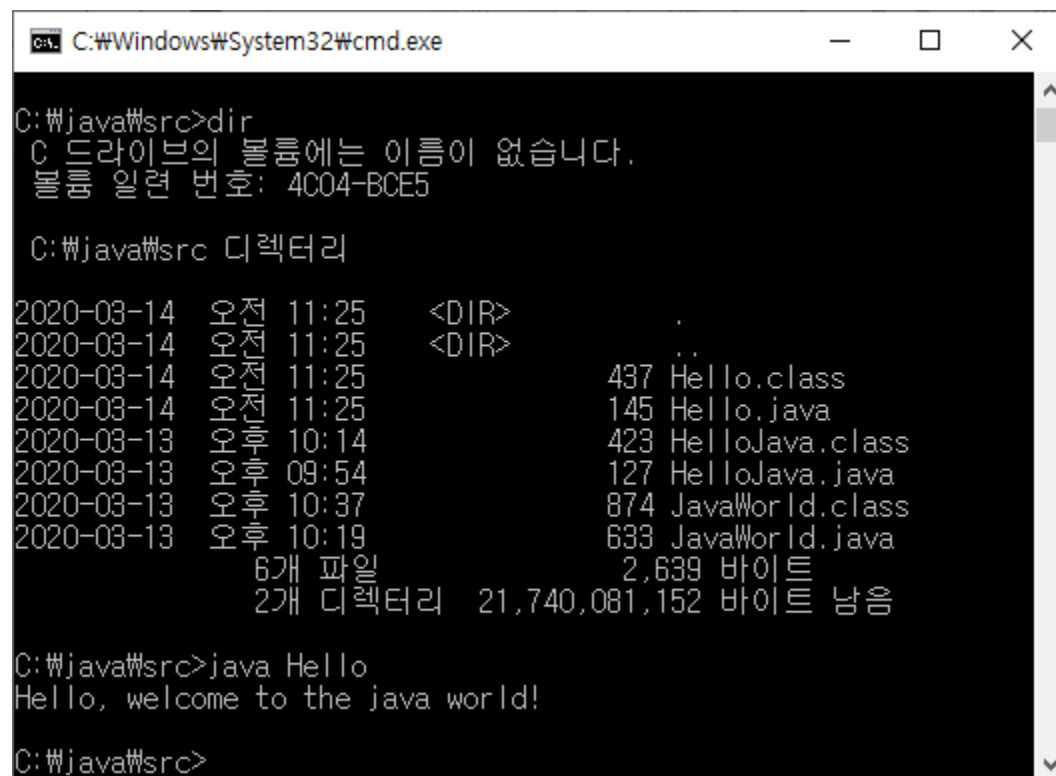




## 4절. Hello Java

### ❖ 바이트코드(class 파일) 실행

- 명령 수행 : java Hello



```
C:\Windows\System32\cmd.exe

C:\wjava\src>dir
C 드라이브의 볼륨에는 이름이 없습니다.
볼륨 일련 번호: 4C04-BCE5

C:\wjava\src 디렉터리

2020-03-14 오전 11:25 <DIR> .
2020-03-14 오전 11:25 <DIR> ..
2020-03-14 오전 11:25      437 Hello.class
2020-03-14 오전 11:25     145 Hello.java
2020-03-13 오후 10:14     423 HelloJava.class
2020-03-13 오후 09:54     127 HelloJava.java
2020-03-13 오후 10:37     874 JavaWorld.class
2020-03-13 오후 10:19     633 JavaWorld.java
                6개 파일             2,639 바이트
                2개 디렉터리  21,740,081,152 바이트 남음

C:\wjava\src>java Hello
Hello, welcome to the java world!

C:\wjava\src>
```



## 6절. 이클립스 설치

### ❖ 이클립스(Eclipse) 소개

- 2003년 IBM에서 개발
- 자바 통합 개발 환경(IDE: Integrated Development Environments) 제공
  - 프로젝트 생성 기능 제공
  - 자동 코드 완성 기능 제공
  - 디버깅 기능 제공
- 이클립스 연합(Eclipse Foundation) 설립 - 지속적 버전업과 배포
- 다양한 개발 환경을 구축할 수 있도록 플러그인(Plug-In) 설치 가능
  - 안드로이드 개발 환경
  - 스프링(Spring) 개발 환경
  - C, C++ 개발 환경



## 6절. 이클립스 설치

### ❖ 이클립스 다운로드

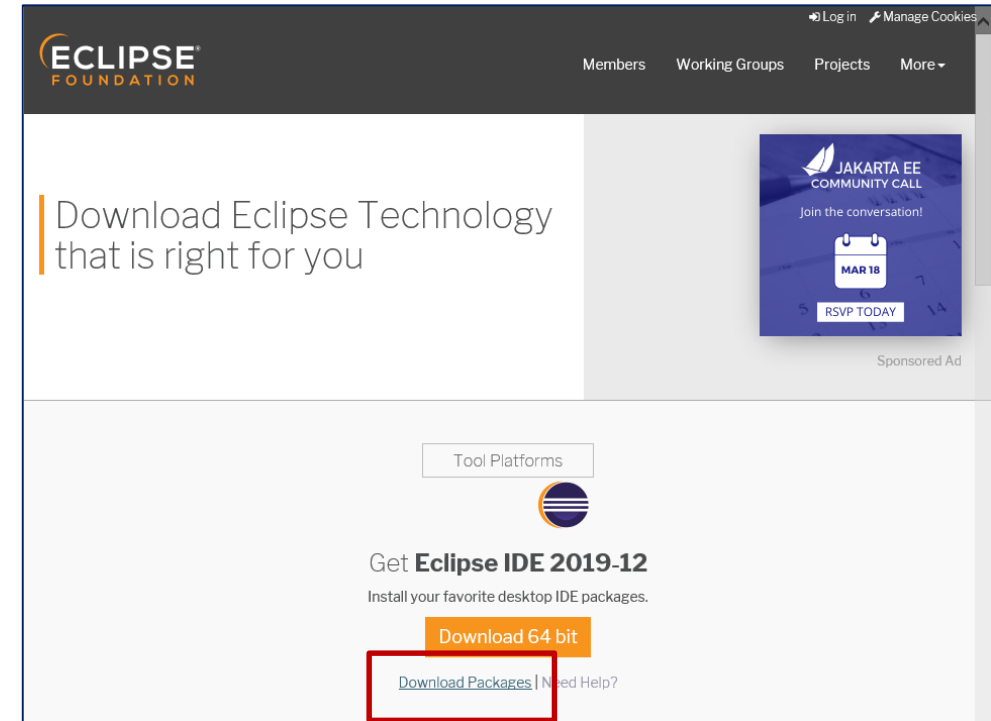
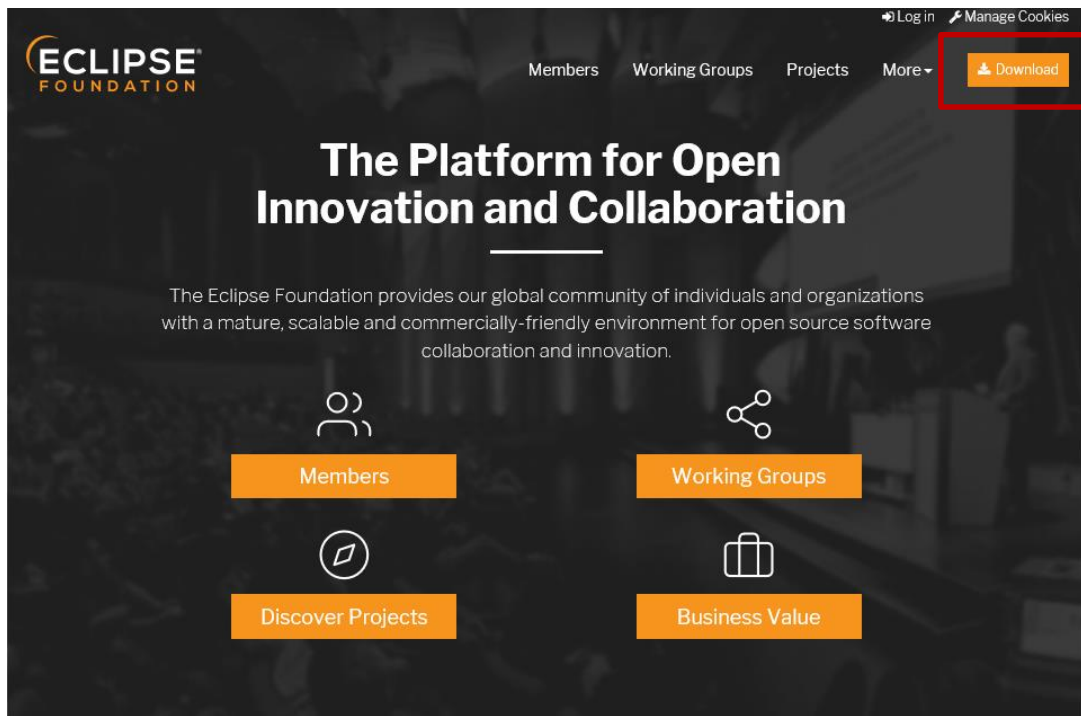
- 이클립스는 자바 언어로 개발된 툴 - JDK 필요
- 다운로드 사이트: <http://www.eclipse.org>
  - Eclipse IDE for Java Developers 버전
    - 순수 자바 학습용
  - Eclipse IDE for Java EE Developers 버전
    - 웹 애플리케이션 등의 Enterprise (Network) 환경에서 실행
  - CPU 사양에 맞게 다운로드



## 6절. 이클립스 설치

### ❖ 이클립스 다운로드

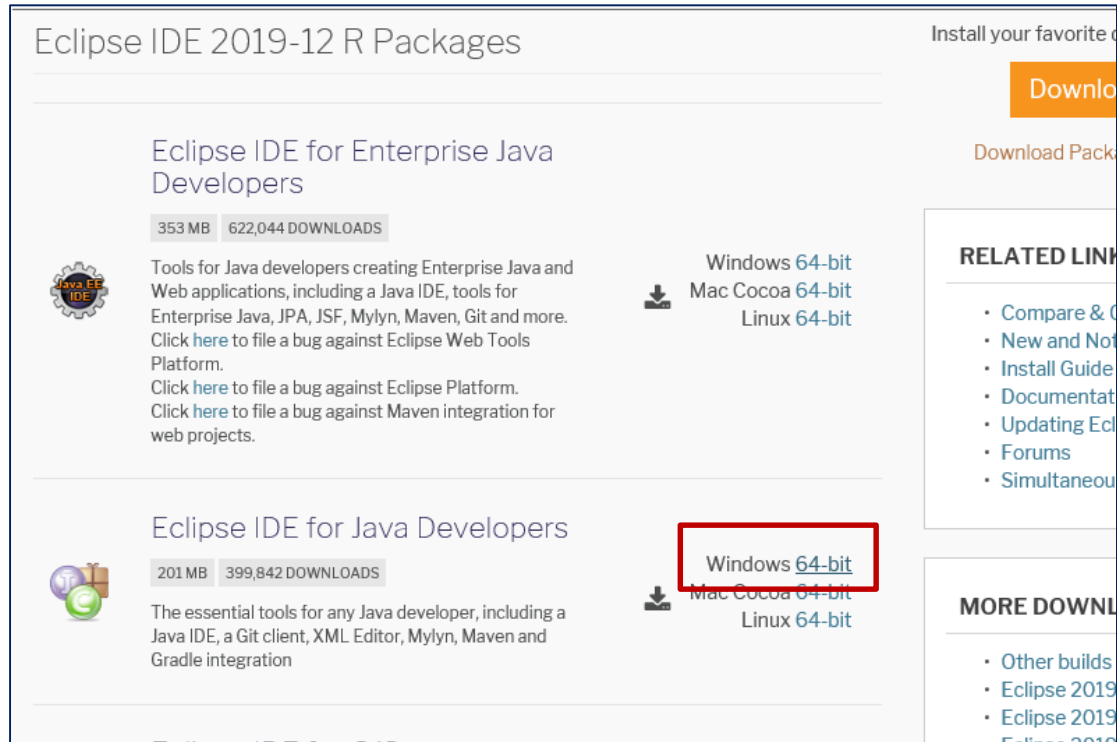
- 다운로드 사이트: <http://www.eclipse.org>



# 6절. 이클립스 설치

## ❖ 이클립스 다운로드

■ 다운로드 사이트: <http://www.eclipse.org>



Eclipse IDE 2019-12 R Packages

Install your favorite

**Eclipse IDE for Enterprise Java Developers**

353 MB 622,044 DOWNLOADS

Tools for Java developers creating Enterprise Java and Web applications, including a Java IDE, tools for Enterprise Java, JPA, JSF, Mylyn, Maven, Git and more. Click [here](#) to file a bug against Eclipse Web Tools Platform. Click [here](#) to file a bug against Eclipse Platform. Click [here](#) to file a bug against Maven integration for web projects.

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

**Eclipse IDE for Java Developers**

201 MB 399,842 DOWNLOADS

The essential tools for any Java developer, including a Java IDE, a Git client, XML Editor, Mylyn, Maven and Gradle integration

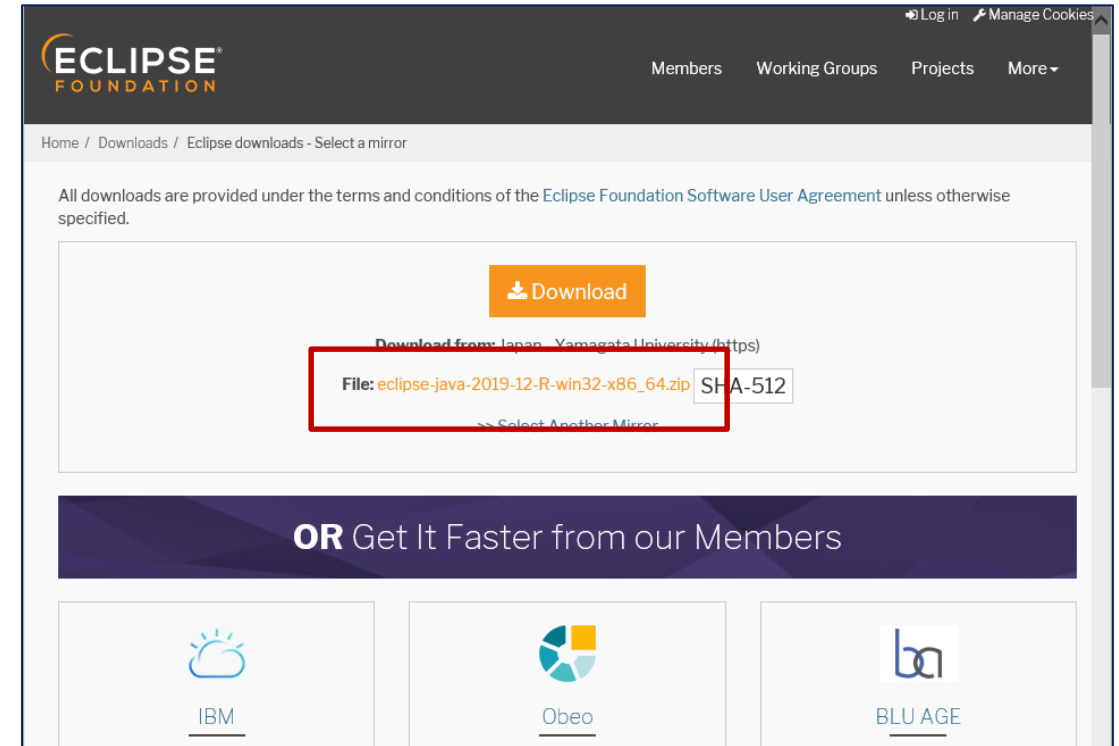
Windows 64-bit  
Mac Cocoa 64-bit  
Linux 64-bit

RELATED LINKS

- Compare & Contrast
- New and Noteworthy
- Install Guide
- Documentation
- Updating Eclipse
- Forums
- Simultaneous

MORE DOWNLOADS

- Other builds
- Eclipse 2019
- Eclipse 2019



ECLIPSE FOUNDATION

Members Working Groups Projects More

Home / Downloads / Eclipse downloads - Select a mirror

All downloads are provided under the terms and conditions of the Eclipse Foundation Software User Agreement unless otherwise specified.

Download

Download from Japan - Yamagata University (https)

File: [eclipse-java-2019-12-R-win32-x86\\_64.zip](#) SHA-512

Select Another Mirror

OR Get It Faster from our Members

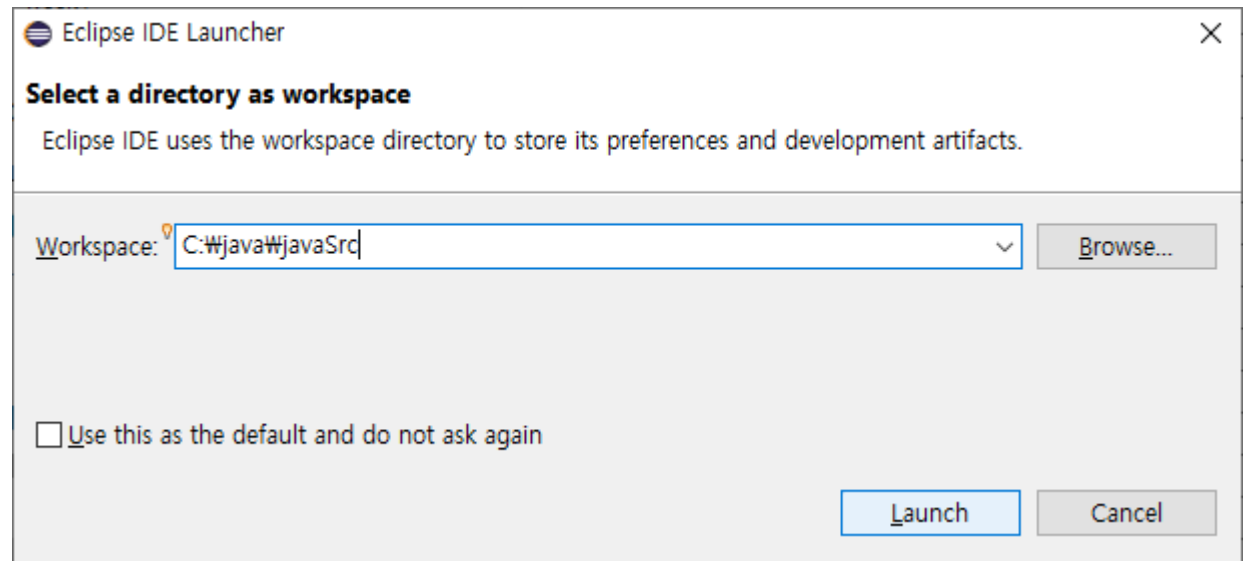
IBM Obeo BLU AGE



## 6절. 이클립스 설치

### ❖ 이클립스 실행

- 다운로드 한 이클립스 파일은 압축파일(zip) 형식
- 압축파일을 풀면 eclipse 폴더 생성
- eclipse 폴더 내 eclipse.exe 실행



## 6절. 이클립스 설치

### ❖ 워크스페이스(Workspace)

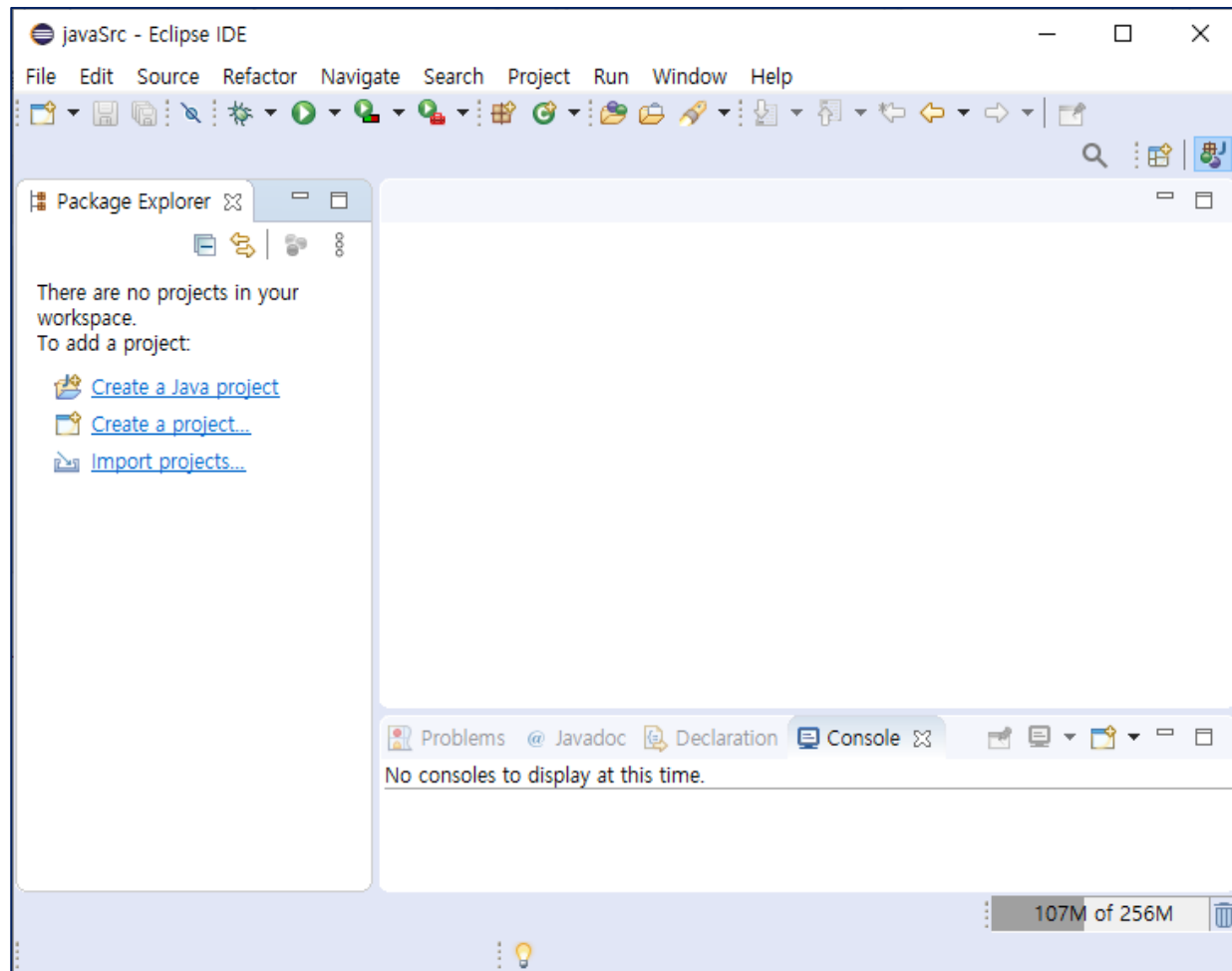
- 이클립스에서 생성한 프로젝트가 기본적으로 저장되는 디렉토리
- 최초 실행 시 워크스페이스 런처(Workspace Launcher)에서 설정
- .metadata 디렉토리
  - 자동 생성되며 이클립스 실행 시 필요한 메타데이터 저장
  - 이 디렉토리 삭제하고 이클립스 실행 - 초기 상태로 다시 실행



## 6절. 이클립스 설치

### ❖ 뷰(View)

- 퍼스펙티브를 구성하는 작은 창으로 여러가지 목적에 맞게 내용 보여줌
- 자유롭게 제거하거나 추가 가능
- 유용한 뷰들
  - Package Explorer
  - Console





## 6절. 이클립스 설치

### ❖ 이클립스 실행 후 다음과 같이 에러 발생 시

- 이클립스에서 Java를 확인하지 못한 경우 발생

#### ■ 이클립스 폴더 안에 있는 eclipse.ini 파일을 수정

```
eclipse.ini - Windows 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말
-startup
plugins/org.eclipse.equinox.launcher_1.5.600.v20191014-2022.jar
--launcher.library
plugins/org.eclipse.equinox.launcher.win32.win32.x86_64_1.1.1100.v20190907-0426
-product
org.eclipse.epp.package.java.product
-showsplash
org.eclipse.epp.package.common
--launcher.defaultAction
openFile
--launcher.defaultAction
openFile
--launcher.appendVmargs
-vm
D:\Program Files\Java\jdk1.8.0_221\bin\javaw.exe
-vmargs
-Dosgi.requiredJavaVersion=1.8
-Dosgi.instance.area.default=@user.home/eclipse-workspace
-XX:+UseG1GC
-XX:+UseStringDeduplication
--add-modules=ALL-SYSTEM
-Dosgi.requiredJavaVersion=1.8
```

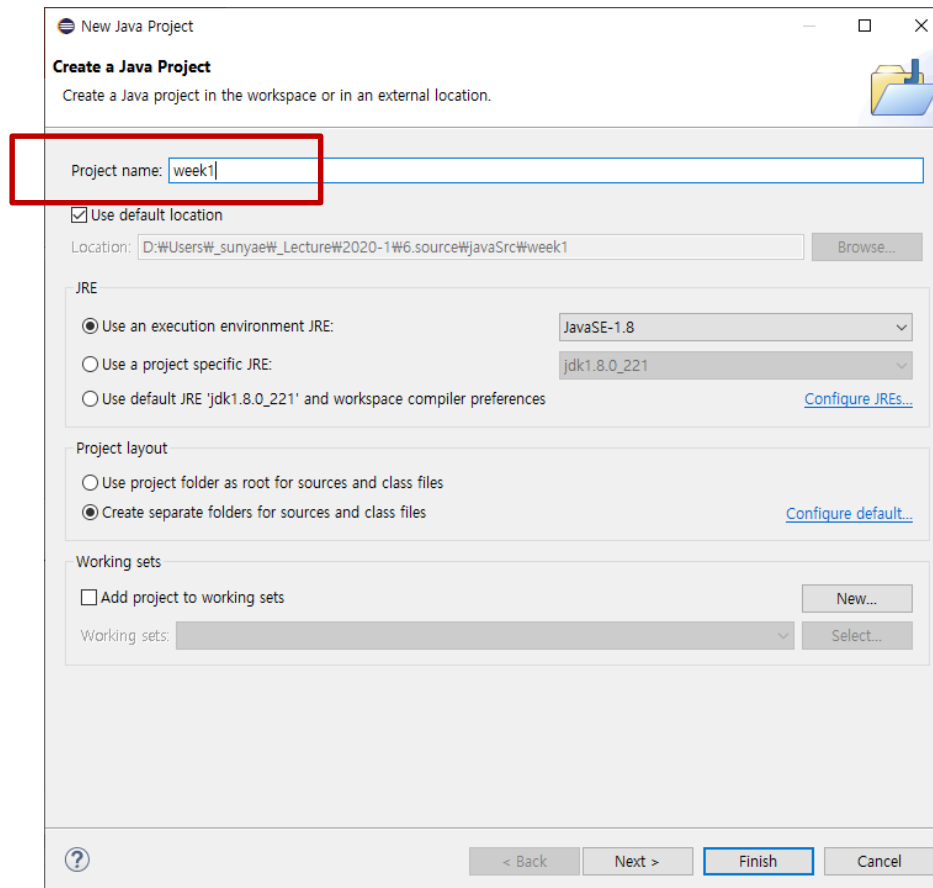
```
Eclipse
Java was started but returned exit code=13
C:\Program Files (x86)\Common Files\Oracle\Java\javapath\javaw.exe
-Dosgi.requiredJavaVersion=1.8
-Dosgi.instance.area.default=@user.home/eclipse-workspace
-XX:+UseG1GC
-XX:+UseStringDeduplication
-Dosgi.requiredJavaVersion=1.8
-Dosgi.dataAreaRequiresExplicitInit=true
-Xms256m
-Xmx1024m
-jar D:\Users\sunayae\Lecture\2019-2\2.알고리즘
Weclipse-java-2019-06-R-win32-x86_64Weclipse\plugins\org.eclipse.e
uinox.launcher_1.5.400.v20190515-0925.jar
-os win32
-ws win32
-arch x86_64
-showsplash D:\Users\sunayae\Lecture\2019-2\2.알고리즘
Weclipse-java-2019-06-R-win32-x86_64Weclipse\plugins\org.eclipse.e
pp.package.common_4.12.0.20190614-1200\plash.bmp
-launcher D:\Users\sunayae\Lecture\2019-2\2.알고리즘
Weclipse-java-2019-06-R-win32-x86_64WeclipseWeclipse.exe
-name Eclipse
--launcher.library D:\Users\sunayae\Lecture\2019-2\2.알고리즘
Weclipse-java-2019-06-R-win32-x86_64Weclipse\plugins\org.eclipse.e
uinox.launcher.win32.win32.x86_64_1.1.1000.v20190125-2016Weclipse_1801
.dll
-startup D:\Users\sunayae\Lecture\2019-2\2.알고리즘
Weclipse-java-2019-06-R-win32-x86_64Weclipse\plugins\org.eclipse.e
uinox.launcher_1.5.400.v20190515-0925.jar
--launcher.appendVmargs
-exitdata 4f24_13c
-product org.eclipse.epp.package.java.product
-vm C:\Program Files (x86)\Common
Files\Oracle\Java\javapath\javaw.exe
-vmargs
-Dosgi.requiredJavaVersion=1.8
-Dosgi.instance.area.default=@user.home/eclipse-workspace
-XX:+UseG1GC
-XX:+UseStringDeduplication
-Dosgi.requiredJavaVersion=1.8
-Dosgi.dataAreaRequiresExplicitInit=true
-Xms256m
-Xmx1024m
-jar D:\Users\sunayae\Lecture\2019-2\2.알고리즘
Weclipse-java-2019-06-R-win32-x86_64Weclipse\plugins\org.eclipse.e
uinox.launcher_1.5.400.v20190515-0925.jar
확인
```



## 6절. 이클립스 설치

### ❖ 프로젝트 생성

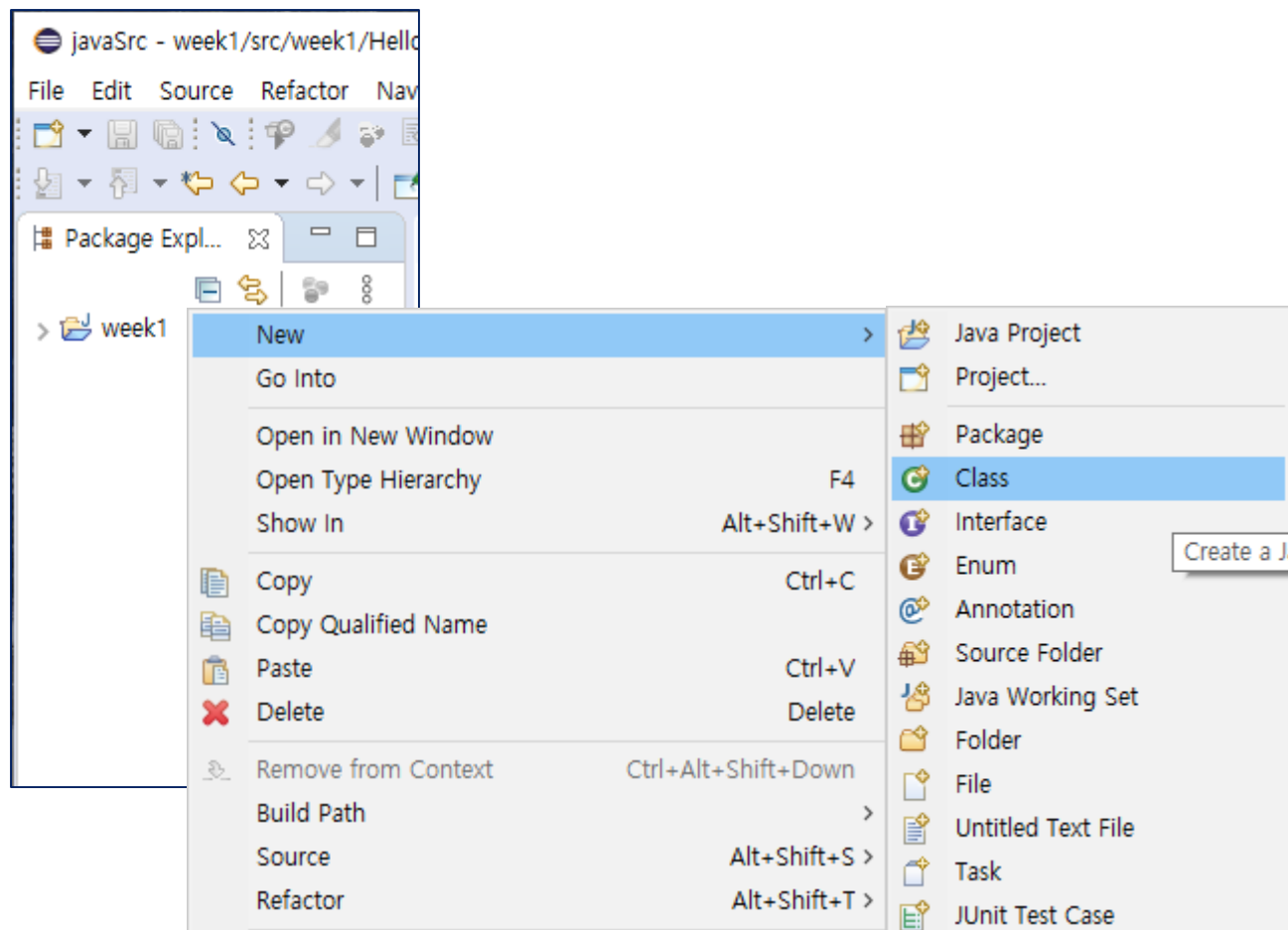
- 먼저 Java Project 생성
- File > New > Java Project > Project name : week1



## 6절. 이클립스 설치

### ❖ 소스 파일 생성

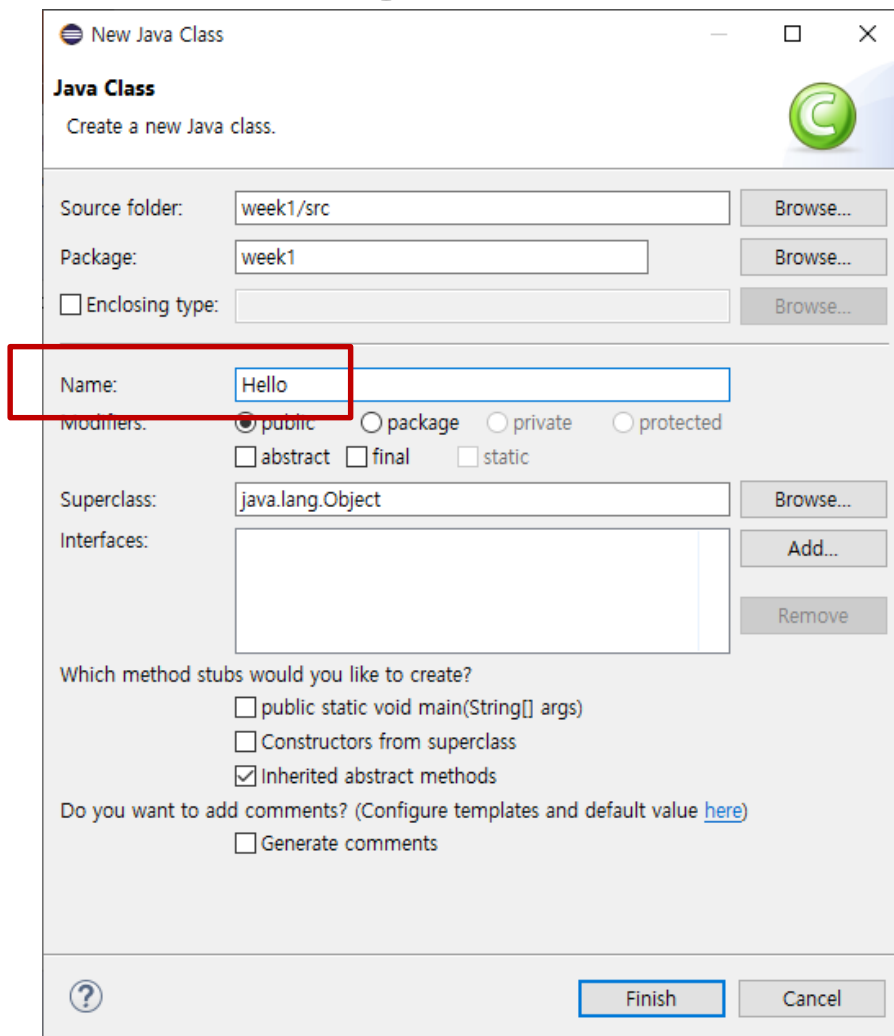
- 생성된 week1 프로젝트 위에 마우스 올려 놓고 오른쪽 마우스 클릭
- **New > Class** 선택



## 6절. 이클립스 설치

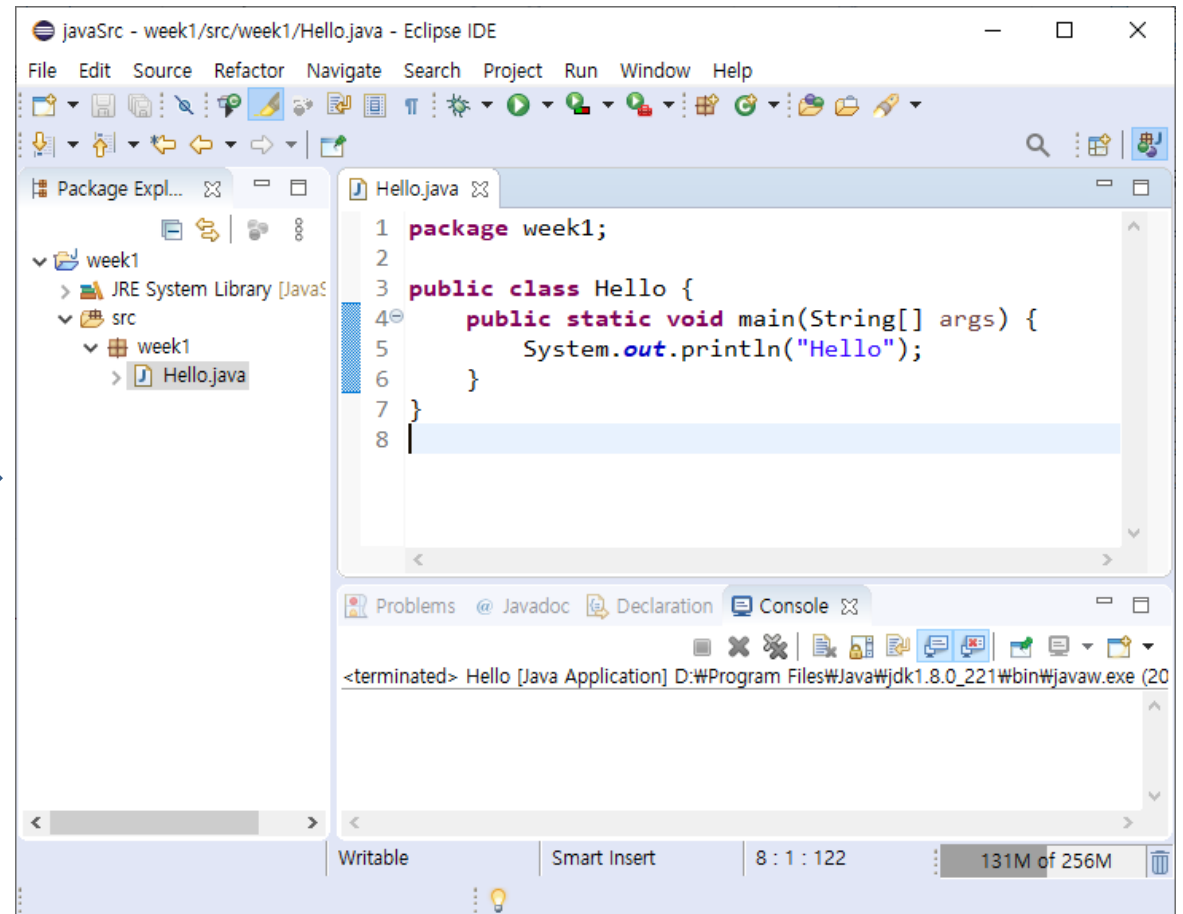
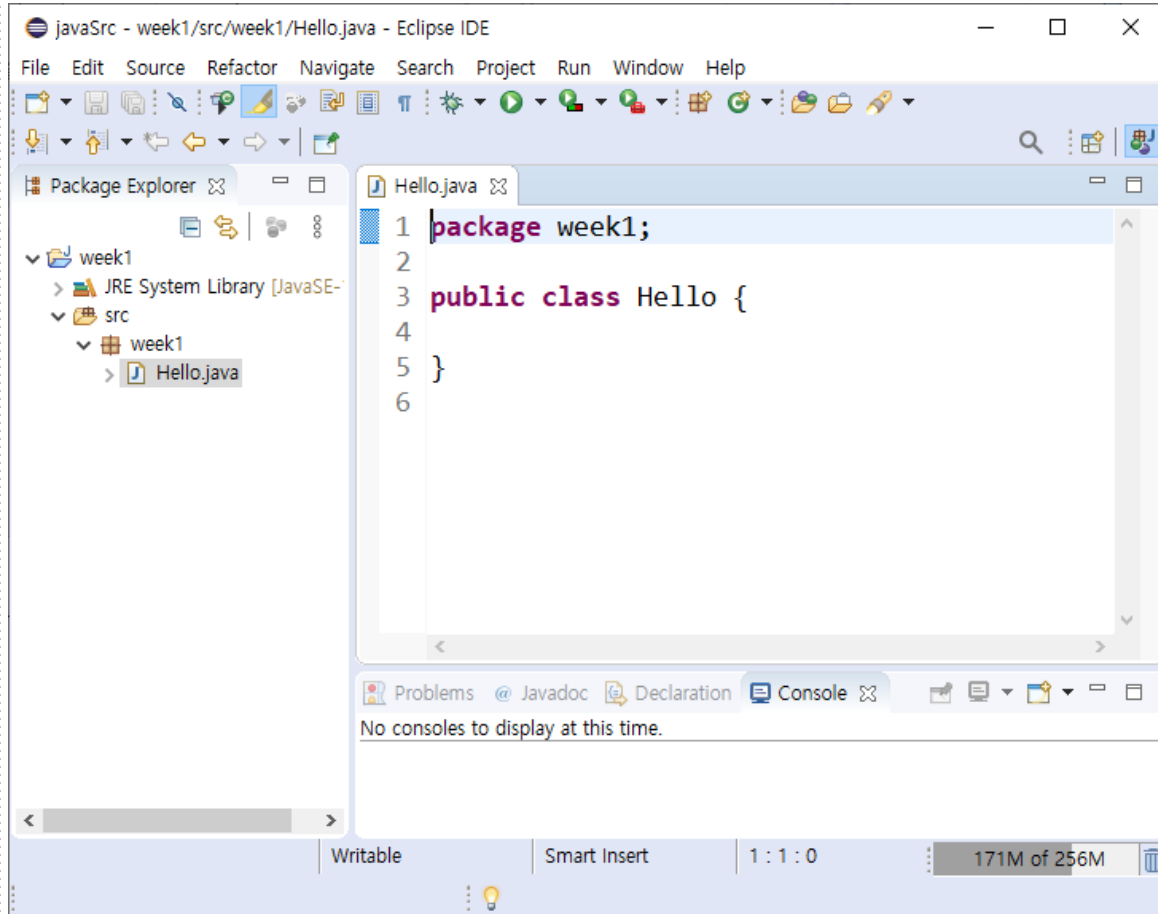
### ❖ 소스 파일 생성

- 생성된 week1 프로젝트 위에 마우스 올려 놓고 오른쪽 마우스 클릭
- New > Class 선택
- Name : Hello 입력



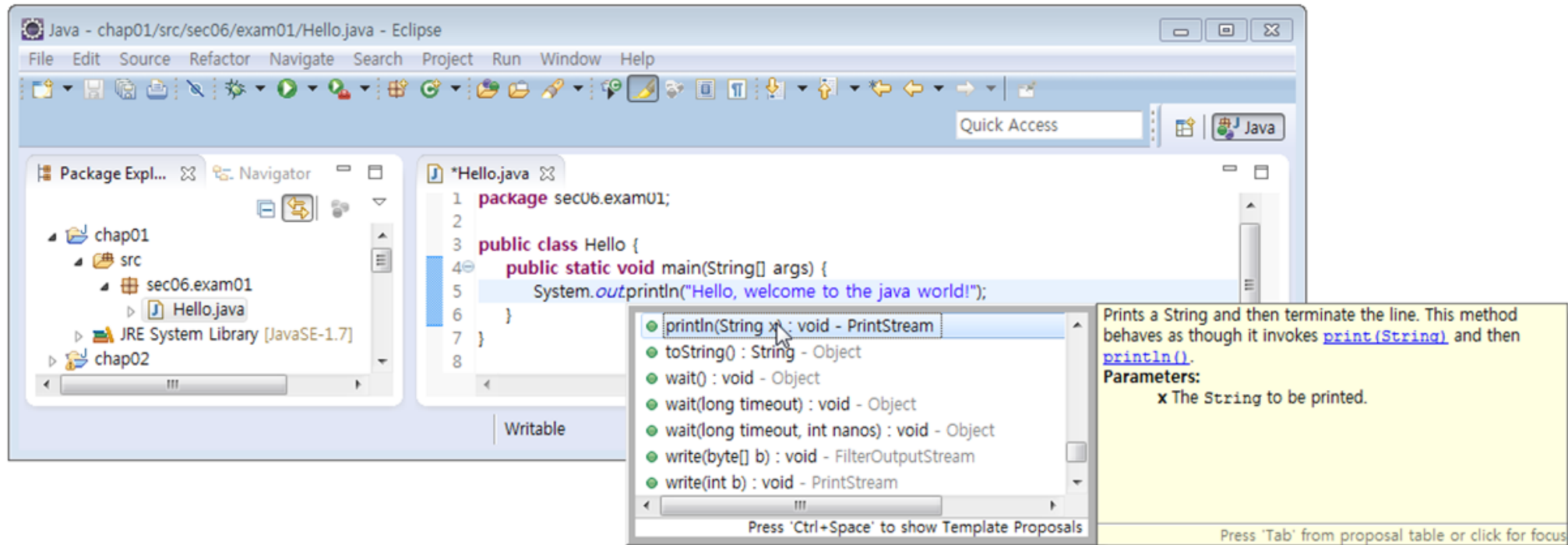
## 6절. 이클립스 설치

### ❖ 소스 파일 생성




## 6절. 이클립스 설치

### ❖ 소스 코드 작성 – 저장과 동시에 컴파일



## 6절. 이클립스 설치

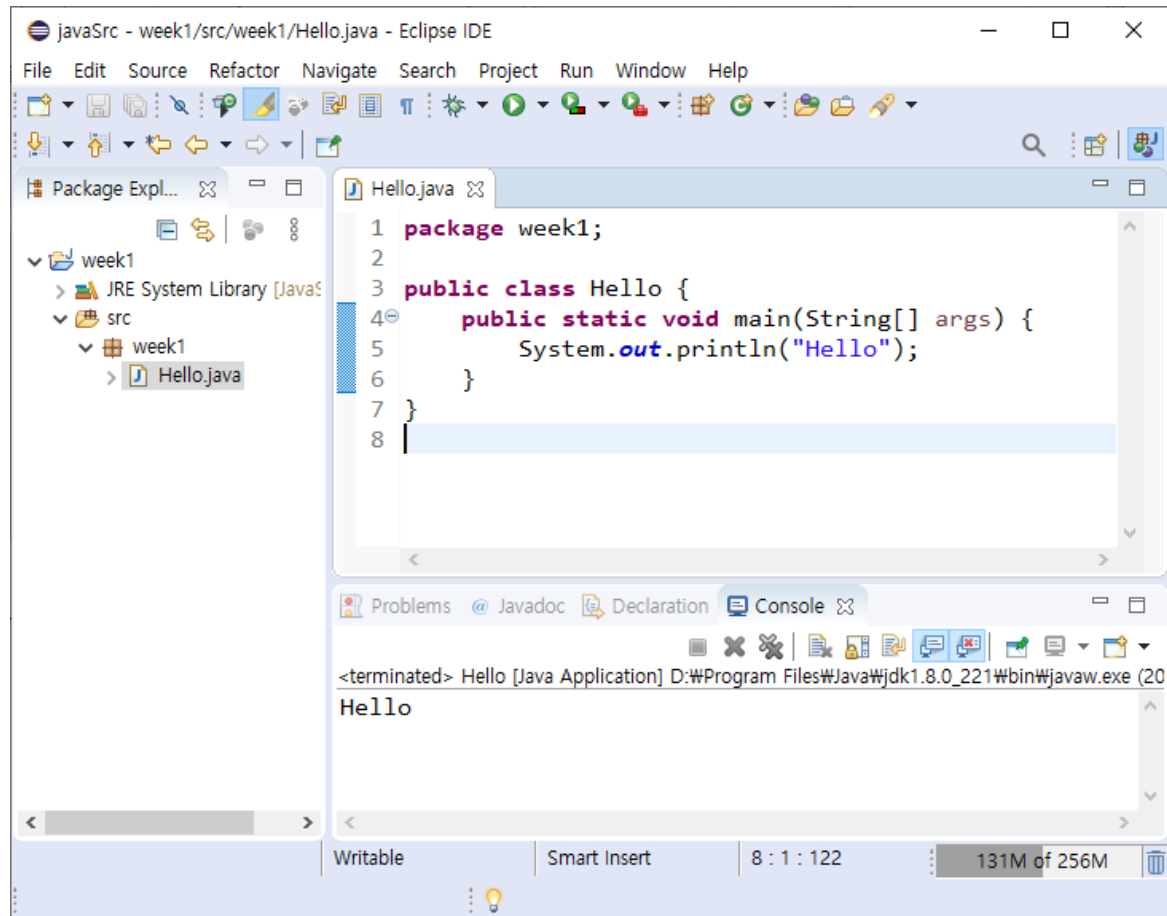
### ❖ 바이트 코드 실행

- 실행 방법 - 툴바에서  클릭
- Package Explorer뷰에서 소스 파일 선택
  - 마우스 오른쪽 버튼 눌러 > [Run As Java Application] 클릭



## 6절. 이클립스 설치

### ❖ 소스 파일 실행 (Run > Run)





# 이클립스 실습

- ❖ 클래스명 : HelloWorld.java
- ❖ 다음 소스를 입력한 후 실행
- ❖ 실행 결과를 포함하여 이클립스 전체 화면을 프린트 스크린하여 이미지 파일로 저장한 후 과제물로 제출 (아래 이미지 참조)

The screenshot shows the Eclipse IDE interface. On the left, the Package Explorer displays a project named 'week1' with a source folder 'src' containing 'Hello.java' and 'HelloWorld.java'. The main editor window shows the code for 'HelloWorld.java':

```
1 package week1;  
2  
3 public class HelloWorld {  
4     public static void main(String[] args) {  
5         System.out.println("학과 : 컴퓨터정보과");  
6         System.out.println("학번 : xxxxxxxx");  
7         System.out.println("이름 : 홍길동");  
8     }  
9 }  
10  
11
```

At the bottom, the Console window shows the output of the program:

```
<terminated> HelloWorld [Java Application] D:\Program Files\Java\jdk1.8.0_221\bin\javaw.e  
학과 : 컴퓨터정보과  
학번 : xxxxxxxx  
이름 : 홍길동
```

The output text is highlighted with a red rectangle.

본인의 학번과 이름 기술

