



자바 설치 및 시작

# 자바 공부

- 대상1. 전공자 및 유경험자

- ✓ 클래스, 메소드 만드는 연습을 많이 하세요.
- ✓ 코딩 테스트 대비해서 문제 많이 풀어보세요.(백준, 프로그래머스 등)  
(전공자를 선호하는 기업은 대부분 코딩 테스트가 있음.)

- 대상2. 비전공자 및 무경험자

- ✓ 제어문을 많이 연습하세요.
- ✓ 클래스, 메소드의 코드 동작 과정에 대해서 이해하세요.
- ✓ 새로운 문제보다 수업에서 다루는 문제 위주로 복습하세요.  
(새로운 문제를 푸는 실력은 하루 아침에 만들어지지 않음.)

- 대상3. 전체

- ✓ 불필요한 부분은 배제하고 중요한 부분 위주로 수업을 진행합니다.  
(수업 내용이 많으니 줄여달라는 요청은 받아들이기 어렵습니다.)
- ✓ 비전공자나 무경험자도 이해할 수 있는 수준으로 수업을 진행합니다.  
(유사 과정을 수강했었다면 수업이 쉬울 수 있습니다.)
- ✓ 다른 사람과 내 실력을 비교하지 마세요. **내 실력이 느는 것이 중요합니다.**

# 자바 개요

- 소개

- ✓ 개발자 : 제임스 고슬링(대표 개발자) 외.
- ✓ 제임스 고슬링이 자바 섬의 커피를 무척 좋아하여 언어 이름을 자바로 하고, 아이콘을 커피로 하였음
- ✓ 1995년 썬마이크로시스템즈(Sun Microsystems)에서 최초 발표.
- ✓ 2010년 오라클(Oracle)에서 썬마이크로시스템즈 인수.  
현재 오라클에서 유지, 보수, 배포

- 용도

- ✓ 웹 애플리케이션(Web Application)
- ✓ 모바일 안드로이드 애플리케이션(Mobile Android Application)
- ✓ SI(System Integration) Project

- 특징

- ✓ C++ 기반의 객체지향언어
- ✓ 플랫폼에 독립적(Write Once, Run Anywhere)
- ✓ 멀티스레드 지원
- ✓ 네트워크 기능 지원

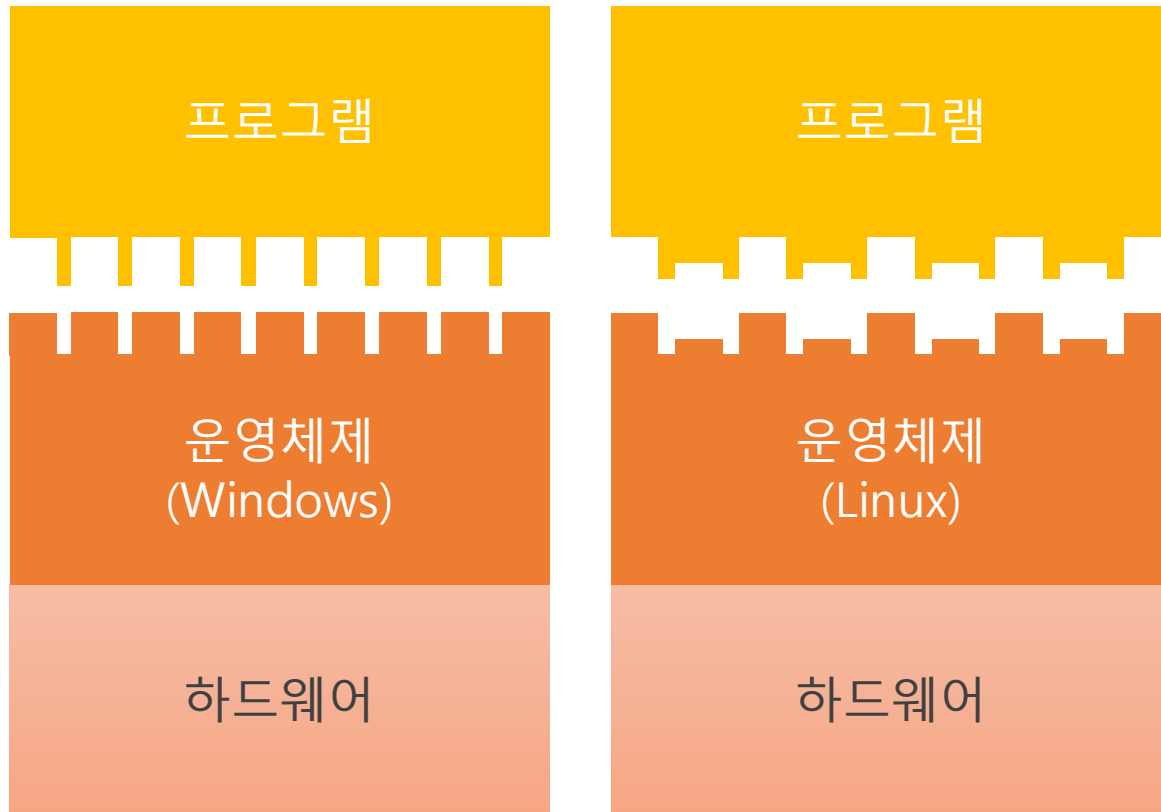


제임스 고슬링  
(유명한 개발자들은 왜 다들 머리가 없지?)

# 객체지향언어

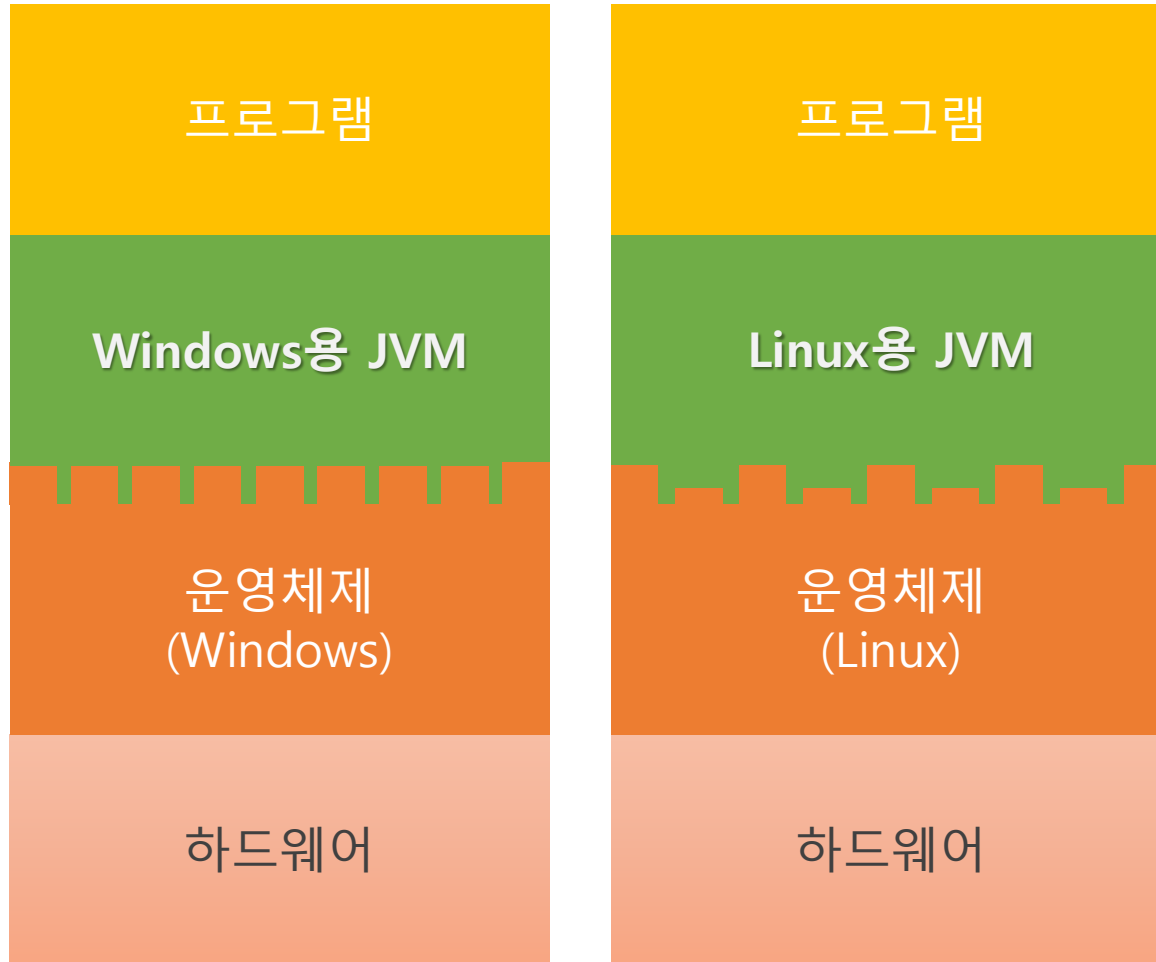
- 절차지향언어
  - ✓ 프로그램의 흐름에 따른 코드 작성
  - ✓ 변수와 함수로 구성
  - ✓ COBOL, C 등 초기에 사용된 프로그래밍 언어
- 객체지향언어
  - ✓ 프로그램의 기능에 따른 코드 작성
  - ✓ 어떤 기능을 처리하기 위한 변수와 메소드(함수)를 하나의 클래스로 구성
  - ✓ 구성된 클래스를 이용해 객체(Object)를 생성
  - ✓ C++, Java, Python 등 현재 주로 사용되는 프로그래밍 언어
- 객체지향언어 특징
  - ✓ 캡슐화
  - ✓ 정보은닉
  - ✓ 상속
  - ✓ 다형성

# 플랫폼에 종속적



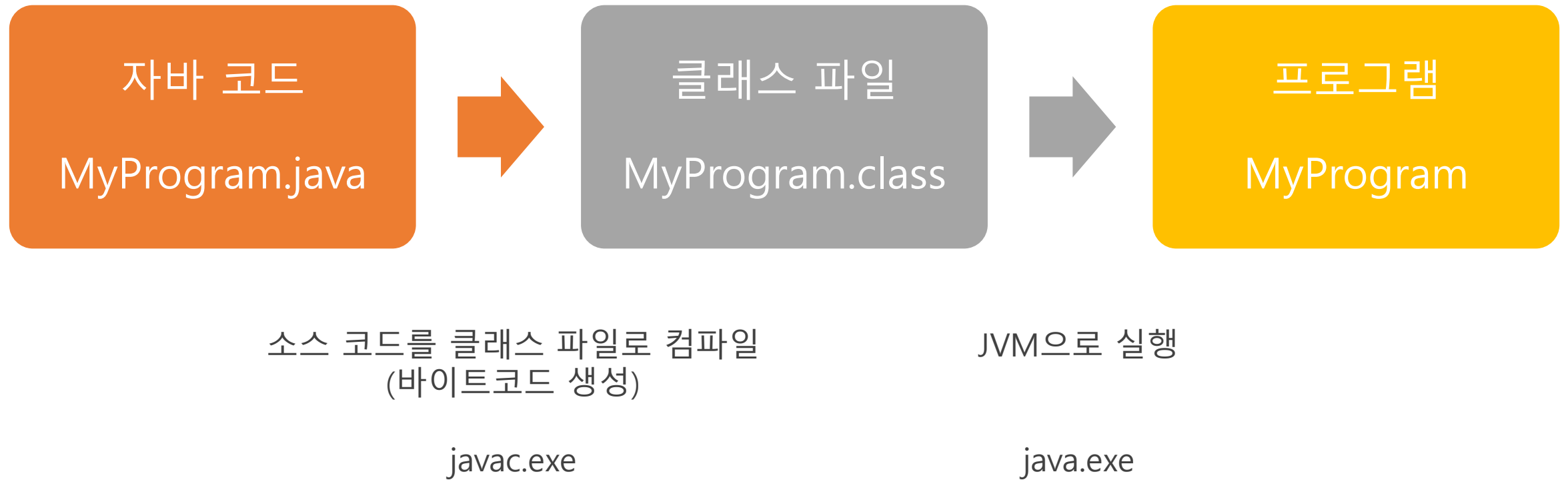
- 동일한 기능의 프로그램을 운영체제(OS)마다 다르게 만들어야 함
- 많은 임베디드 제어 소프트웨어는 플랫폼에 종속적인 C언어로 제작되었는데, 이로 인해 종속 문제가 발생함
- Windows 98 환경에서 제작한 소프트웨어는 Windows 98에서만 동작하기 때문에 운영체제를 업그레이드 할 수 없음.  
(구 버전의 운영체제를 아직도 회사에서 사용하고있는 이유임)

# 플랫폼에 독립적



- 자바는 운영체제에 맞는 JVM(Java Virtual Machine)을 제공함
- 자바를 이용해서 프로그램을 개발해 놓으면 운영체제가 변하더라도 계속해서 동일한 프로그램을 사용할 수 있음

# 자바 실행 과정



# 자바 개발 환경 구축

- 자바 개발 도구(JDK) 설치

- ✓ Java Standard Edition(Java SE) 설치
- ✓ 다운로드 (JDK1.8기준)  
<https://www.oracle.com/kr/java/technologies/javase/javase8-archive-downloads.html>
- ✓ JDK와 JRE로 구성
- ✓ JDK : Java Development Kit (자바 프로그램을 개발)
- ✓ JRE : Java Runtime Environment (자바 프로그램을 실행)
- ✓ JDK를 설치하면 JRE는 함께 설치됨

- API 도큐먼트

- ✓ JDK에서 제공하는 표준 클래스 라이브러리의 매뉴얼 제공
- ✓ JDK1.8기준  
<https://docs.oracle.com/javase/8/docs/api/index.html>



# 자바 다운로드 및 설치

- 다운로드 링크 : <https://www.oracle.com/technetwork/java/javase/downloads>

Java SE Development Kit 8u202

→JDK Standard Edition 1.8 버전 202번째 업데이트














설치된 Windows 종류에 따라 선택

- 32비트 운영체제 : Windows x86
- 64비트 운영체제 : Windows x64

**반드시 기본 설정을 이용하여 설치할 것**

## Java SE Development Kit 8u202

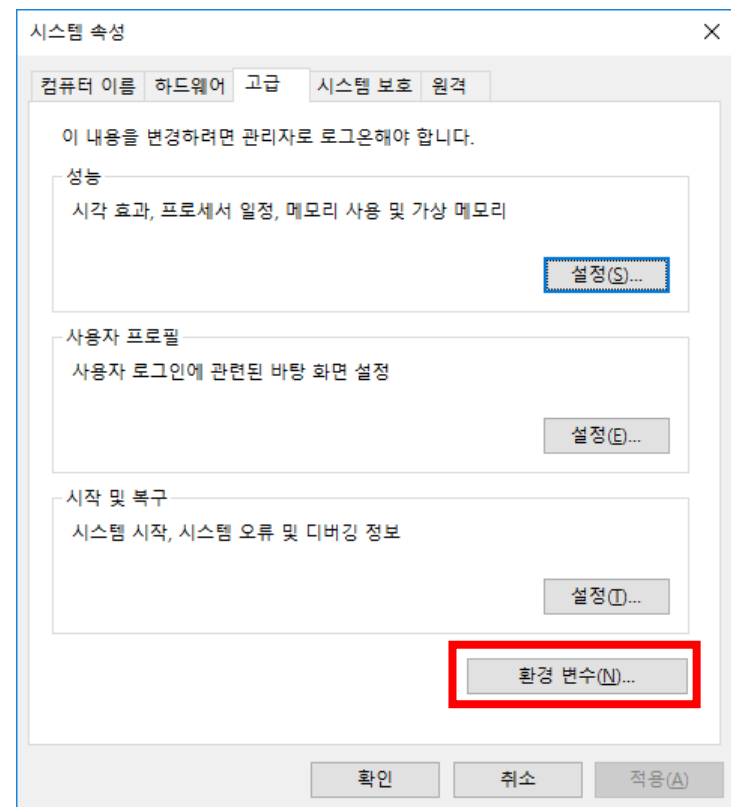
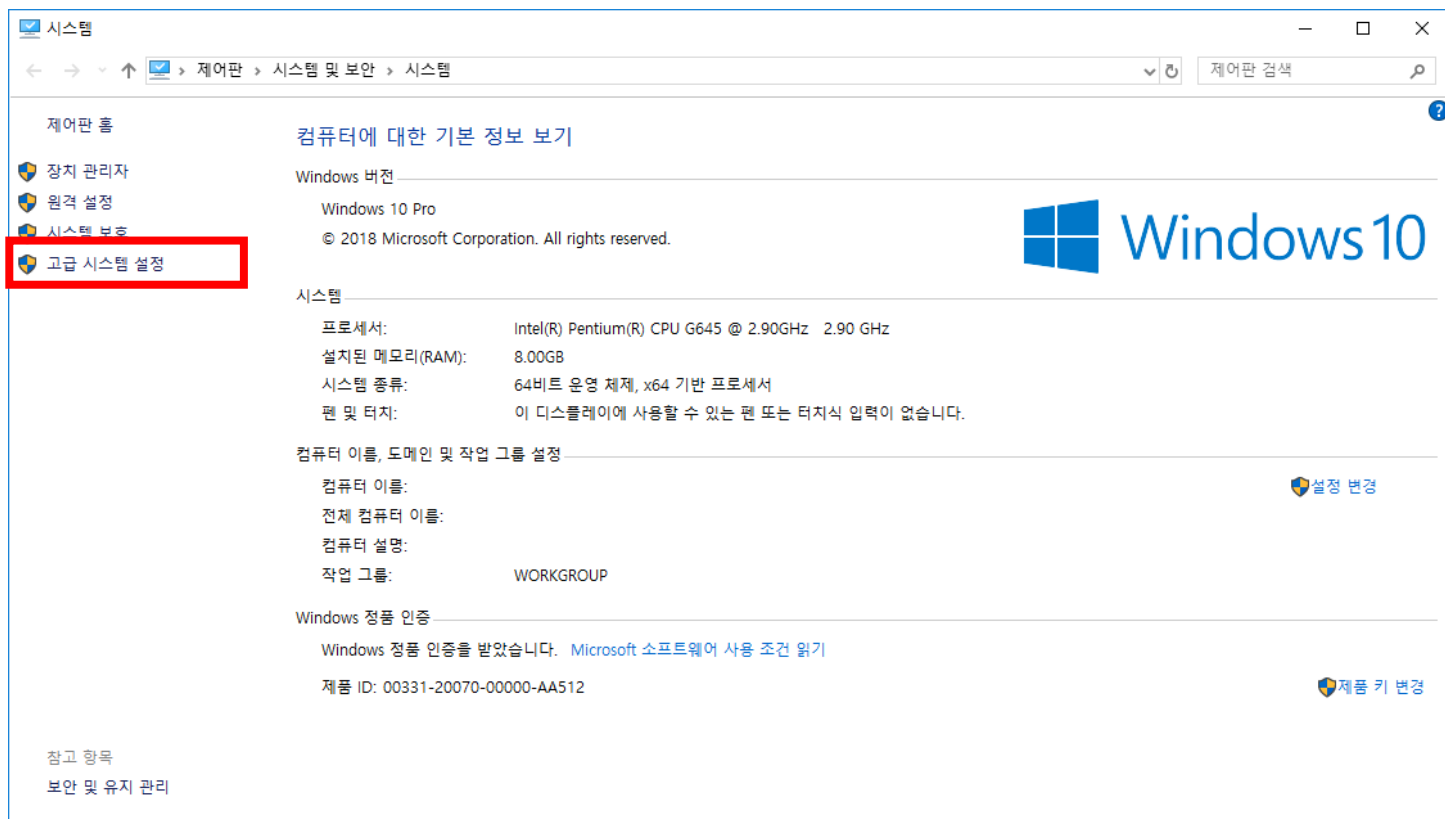
This software is licensed under the Oracle Binary Code License Agreement for Java SE Platform Products

Product / File Description	File Size	Download
Linux ARM v6/v7 Soft Float ABI	72.86 MB	 <a href="#">jdk-8u202-linux-arm32-vfp-hflt.tar.gz</a>
Linux ARM v6/v7 Soft Float ABI	69.75 MB	 <a href="#">jdk-8u202-linux-arm64-vfp-hflt.tar.gz</a>
Linux x86	173.08 MB	 <a href="#">jdk-8u202-linux-i586.rpm</a>
Linux x86	187.9 MB	 <a href="#">jdk-8u202-linux-i586.tar.gz</a>
Linux x64	170.15 MB	 <a href="#">jdk-8u202-linux-x64.rpm</a>
Linux x64	185.05 MB	 <a href="#">jdk-8u202-linux-x64.tar.gz</a>
Mac OS X x64	249.15 MB	 <a href="#">jdk-8u202-macosx-x64.dmg</a>
Solaris SPARC 64-bit (SVR4 package)	125.09 MB	 <a href="#">jdk-8u202-solaris-sparcv9.tar.Z</a>
Solaris SPARC 64-bit	88.1 MB	 <a href="#">jdk-8u202-solaris-sparcv9.tar.gz</a>
Solaris x64 (SVR4 package)	124.37 MB	 <a href="#">jdk-8u202-solaris-x64.tar.Z</a>
Solaris x64	85.38 MB	 <a href="#">jdk-8u202-solaris-x64.tar.gz</a>
Windows x86	201.64 MB	 <a href="#">jdk-8u202-windows-i586.exe</a>
Windows x64	211.58 MB	 <a href="#">jdk-8u202-windows-x64.exe</a>

# 환경 변수 설정

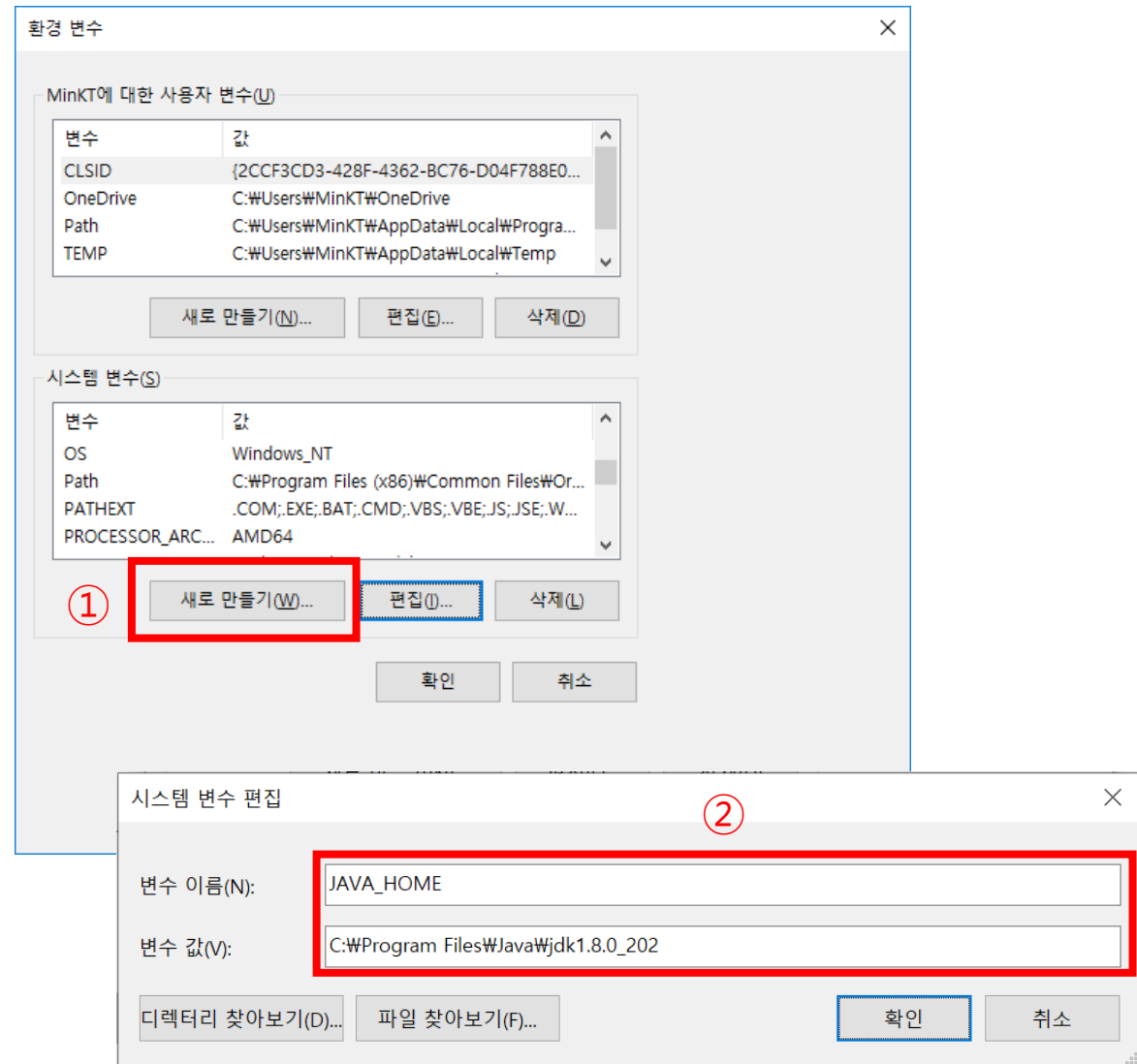
- 환경 변수 설정하기

✓ 제어판 > 시스템 및 보안 > 시스템 > 고급 시스템 설정



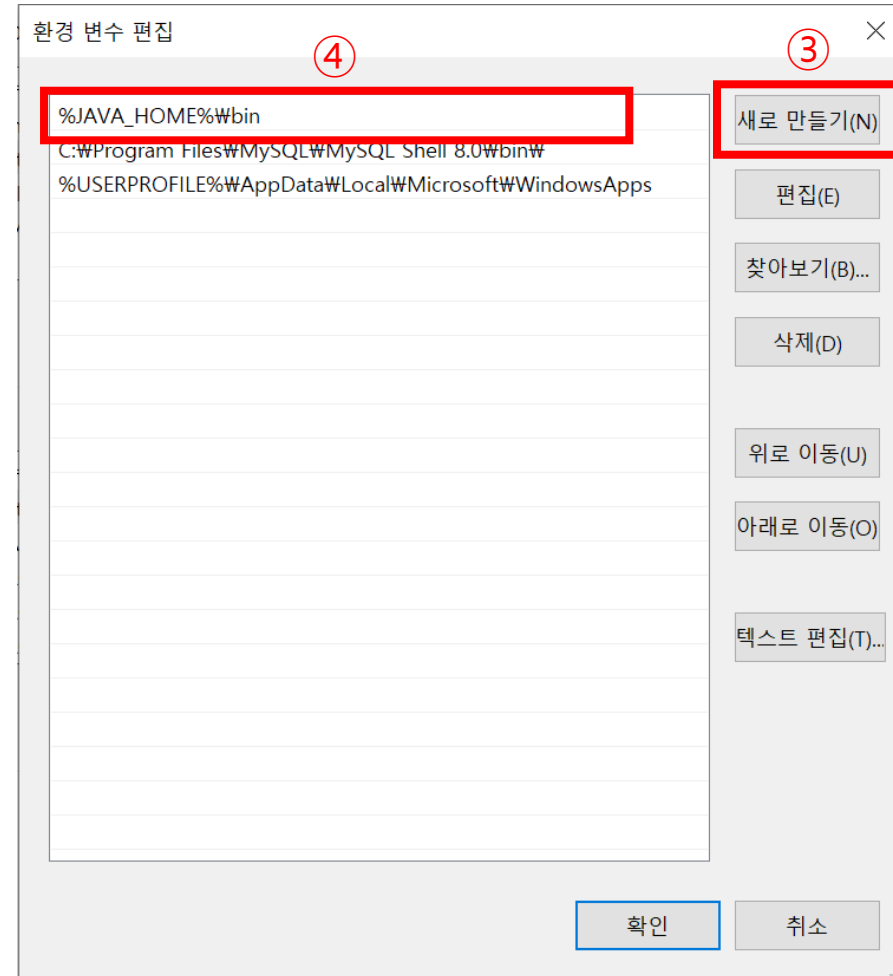
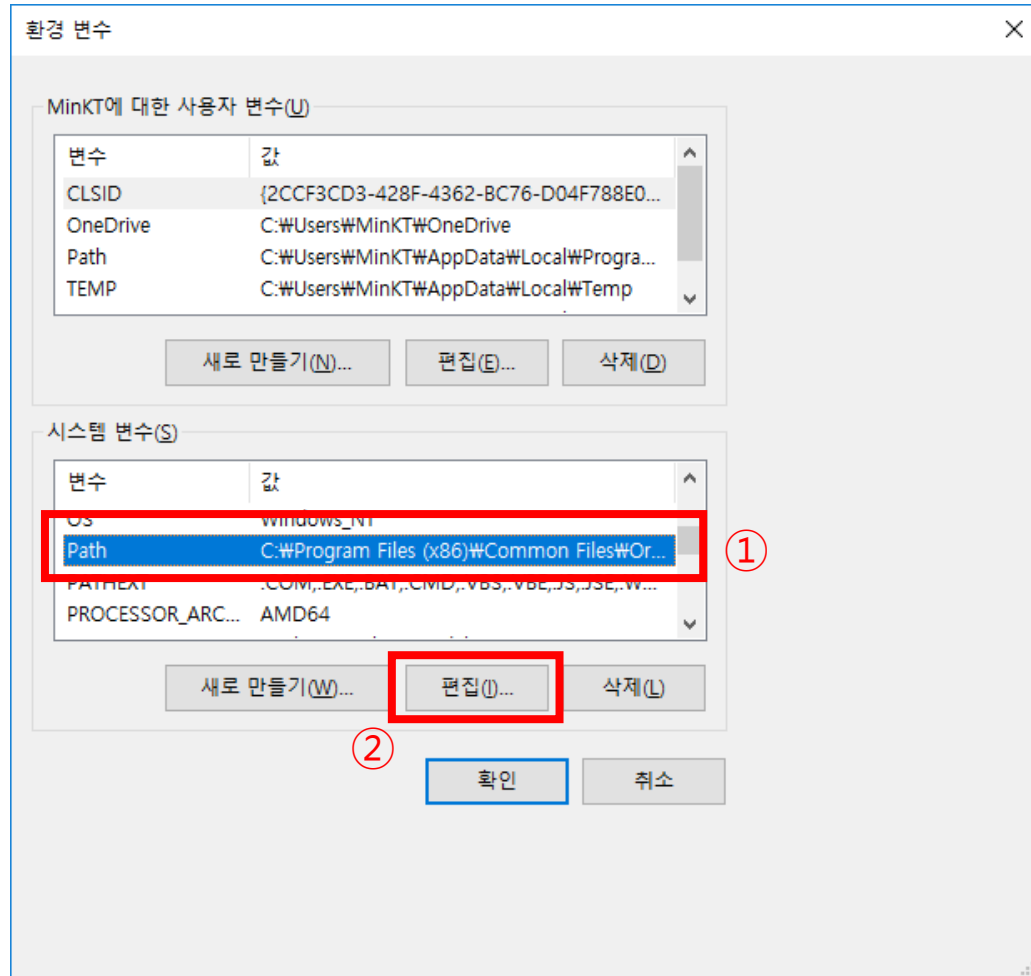
# 환경 변수 설정

- 시스템 변수 추가
  - ✓ 시스템 변수(s)
  - ✓ 새로 만들기
- 변수 이름
  - ✓ JAVA\_HOME
- 변수 값
  - ✓ C:\Program Files\Java\jdk1.8.0\_191



# 환경 변수 설정

- Path 설정



# 이클립스


- 이클립스
  - ✓ 통합 개발 환경(IDE)을 제공하는 개발 툴(Tool)  
(IDE : Integrated Development Environment, 소스 코드 작성부터 실행까지 모든 기능을 제공하는 툴)
  - ✓ 무료로 사용 가능
  - ✓ 자바의 프로젝트, 패키지, 클래스 등을 손쉽게 생성할 수 있음
  - ✓ 자동 코드 완성 기능이 있음
  - ✓ 디버깅(오류를 찾아서 수정하는 것)이 쉬움
- 다양한 플러그 인(Plug-In) 지원
  - ✓ 안드로이드 개발 환경
  - ✓ 스프링 개발 환경



# 이클립스 설치

- 다운로드 링크 : <https://www.eclipse.org/downloads/>

The Eclipse Installer 2022-06 R now includes a JRE for macOS, Windows and Linux.



## Get **Eclipse IDE 2022-06**

Install your favorite desktop IDE packages.

[Download x86\\_64](#)

[Download Packages](#) [Need Help?](#)

Download Packages를 이용해서  
다운로드(installer 아님)

## Eclipse IDE 2022-06 R Packages



### Eclipse IDE for Java Developers

302 MB 222,544 DOWNLOADS

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



Windows x86\_64  
macOS x86\_64 | AArch64  
Linux x86\_64 | AArch64



### Eclipse IDE for Enterprise Java and Web Developers

495 MB 128,505 DOWNLOADS

Tools for developers working with Java and Web applications, including a Java IDE, tools for JavaScript, TypeScript, JavaServer Pages and Faces, Yaml, Markdown, Web Services, JPA and Data Tools, Maven and Gradle, Git, and more.

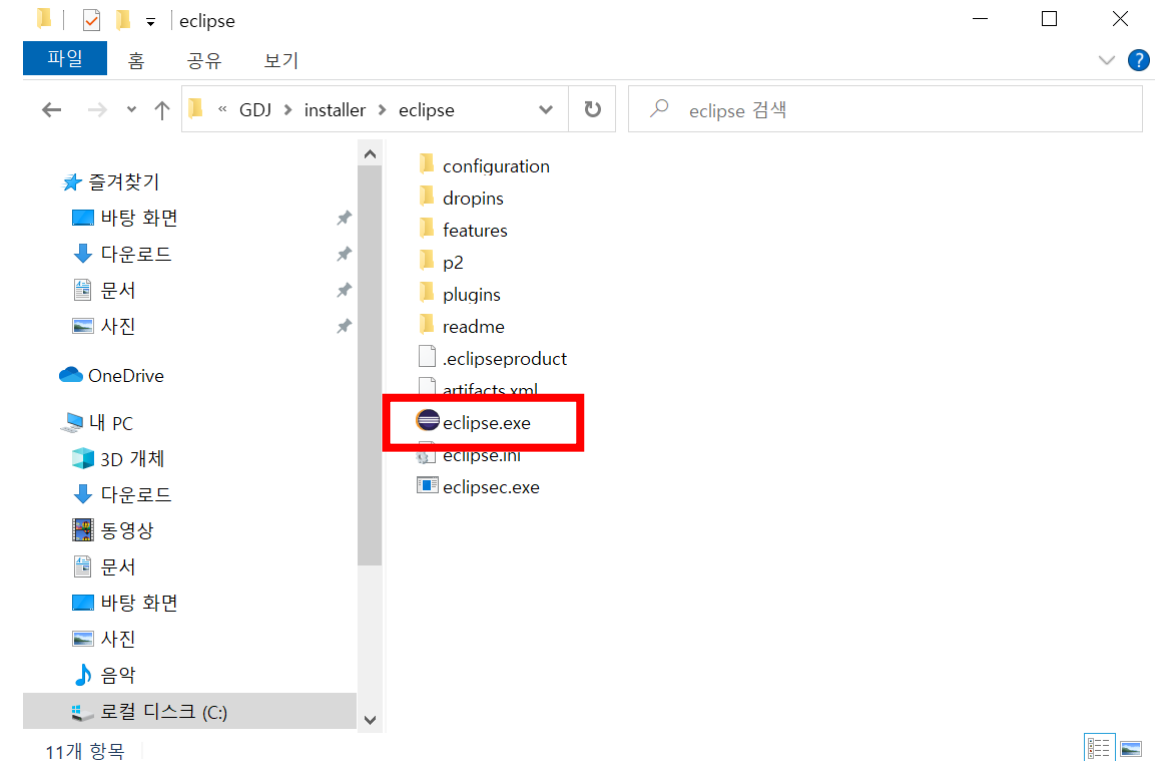


Windows x86\_64  
macOS x86\_64 | AArch64  
Linux x86\_64 | AArch64

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.  
Click [here](#) to report an issue against Eclipse Wild Web Developer (incubating).

# 이클립스 실행

1. 다운로드 받은 이클립스 패키지의 압축을 풀면 eclipse 디렉터리가 생김
2. 이 eclipse 디렉터리를 C:\WGDJ\installer에 둠
3. eclipse.exe 파일 실행  
(.exe 확장자가 안 보이면 [보기] - [파일 확장명] 체크)



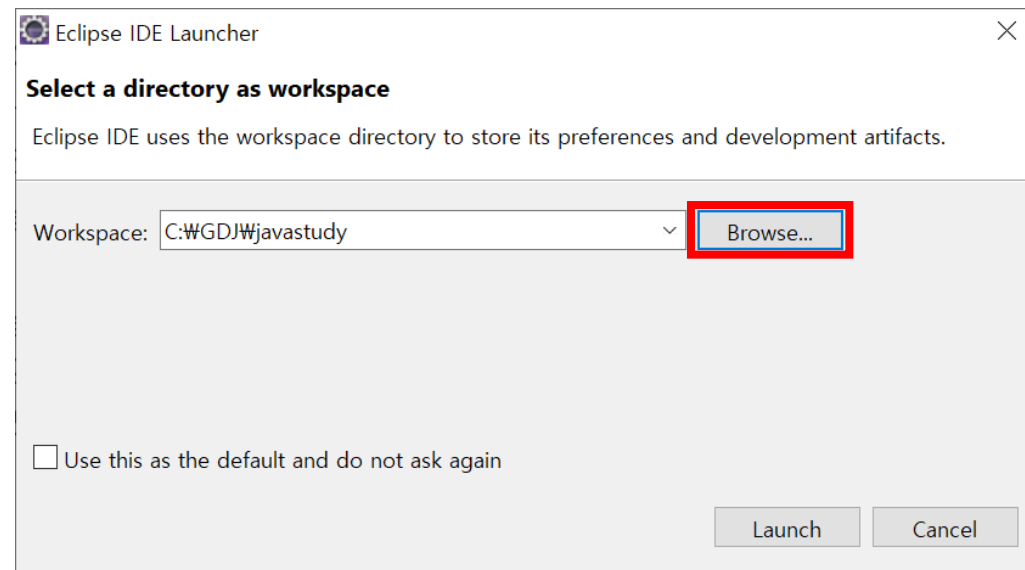
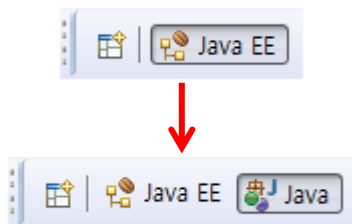
# 이클립스 실행

- 워크스페이스(workspace)

- ✓ 프로젝트가 저장되는 디렉토리
- ✓ 이클립스 최초 실행 시 Eclipse IDE Launcher에서 설정 가능
- ✓ 이클립스 실행 이후 [file] – [Switch Workspace]에서 변경 가능
- ✓ .metadata 디렉터리
  - 이클립스 실행 시 자동으로 생성되는 디렉터리
  - 실행에 필요한 메타데이터를 저장함
  - 이 디렉터리를 삭제하면 이클립스에 해 둔 설정이 모두 초기화 됨

- 퍼스펙티브(Perspective)

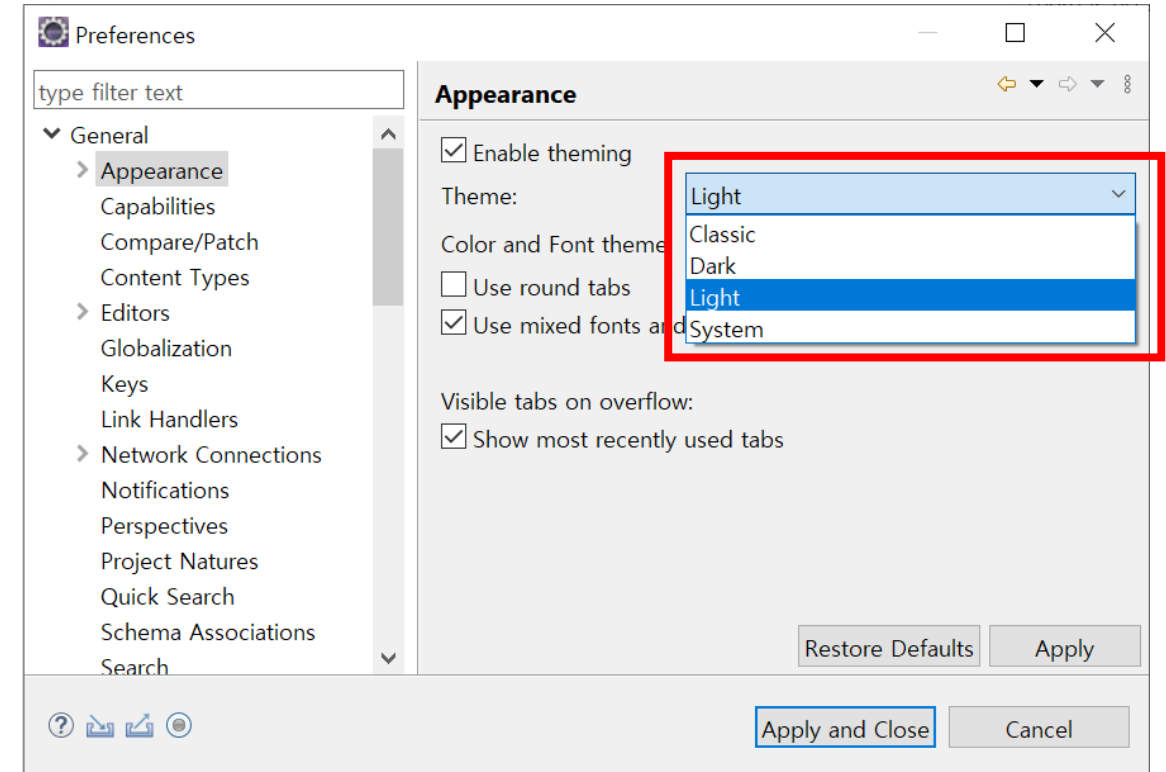
- ✓ 개발 프로젝트 종류별로 유용한 View들을 묶어 놓은 것
- ✓ Eclipse IDE for Java EE Developers를 사용하면 기본적으로 Java EE 퍼스펙티브가 사용됨
- ✓ [Window] – [Perspective] – [Open Perspective] 메뉴 이용해 Java 퍼스펙티브로 변경





# Preferences 설정

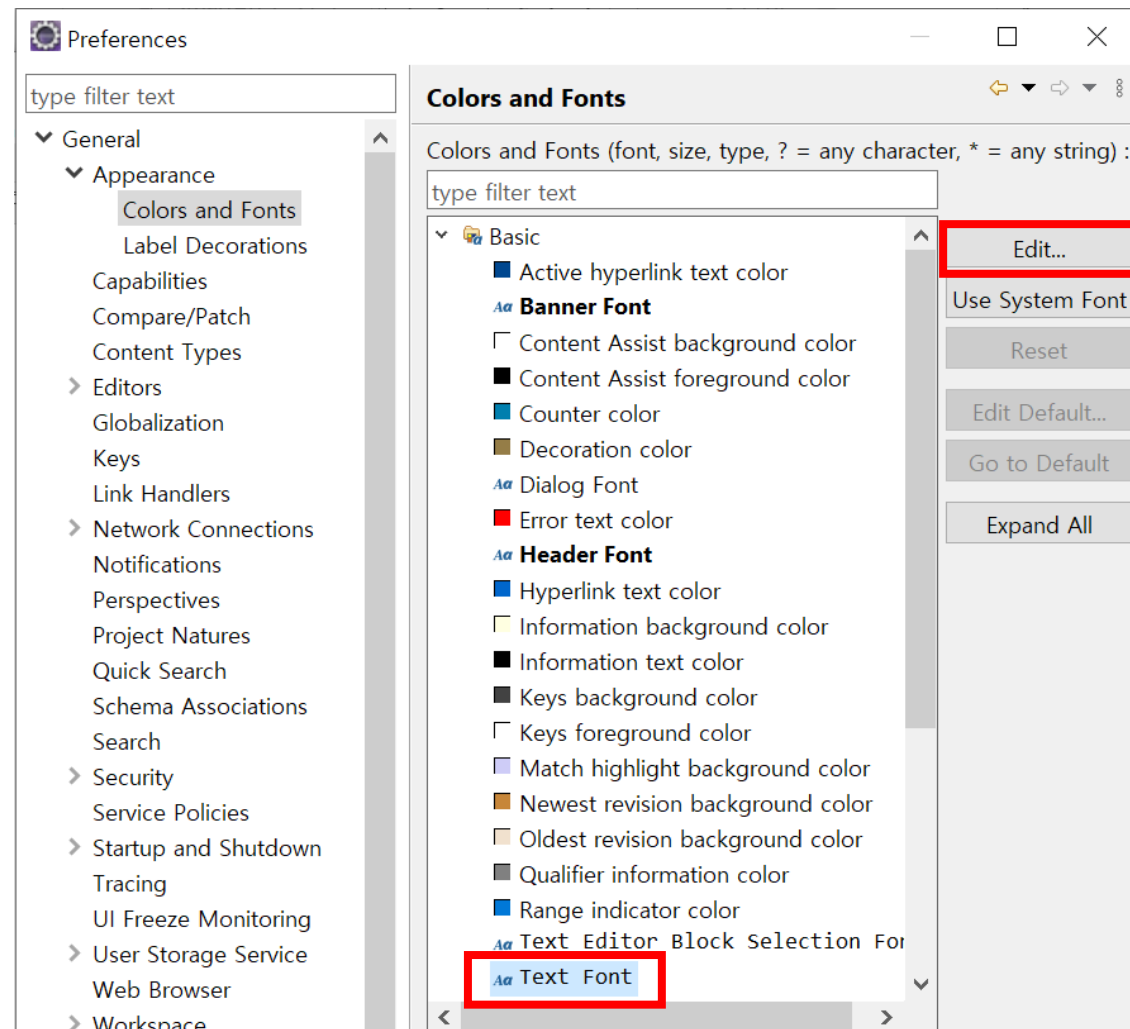
- 테마 변경
  - ✓ General – Appearance
  - ✓ Classic, Dark, Light, System 중 하나를 사용



# Preferences 설정

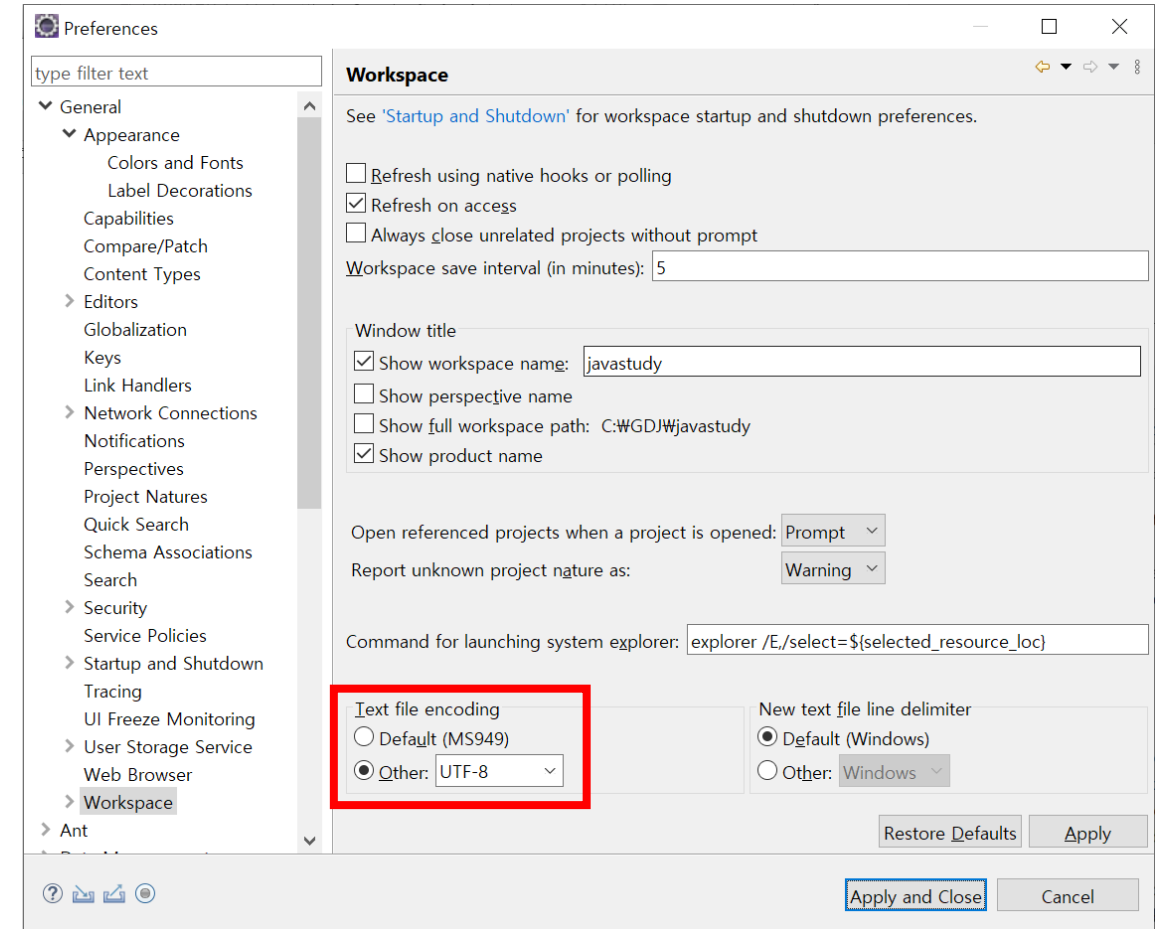
- 글꼴 변경

- ✓ General – Appearance – Colors and Fonts – Basic – Text Font – Edit...
- ✓ 글꼴은 반드시 코딩 전용 폰트 사용할 것 (D2Coding, 나눔코딩고딕 등)
- ✓ 코딩 전용 폰트가 없는 경우에는 모든 글자의 폭이 동일한 Monospaced Font 중 하나를 선택



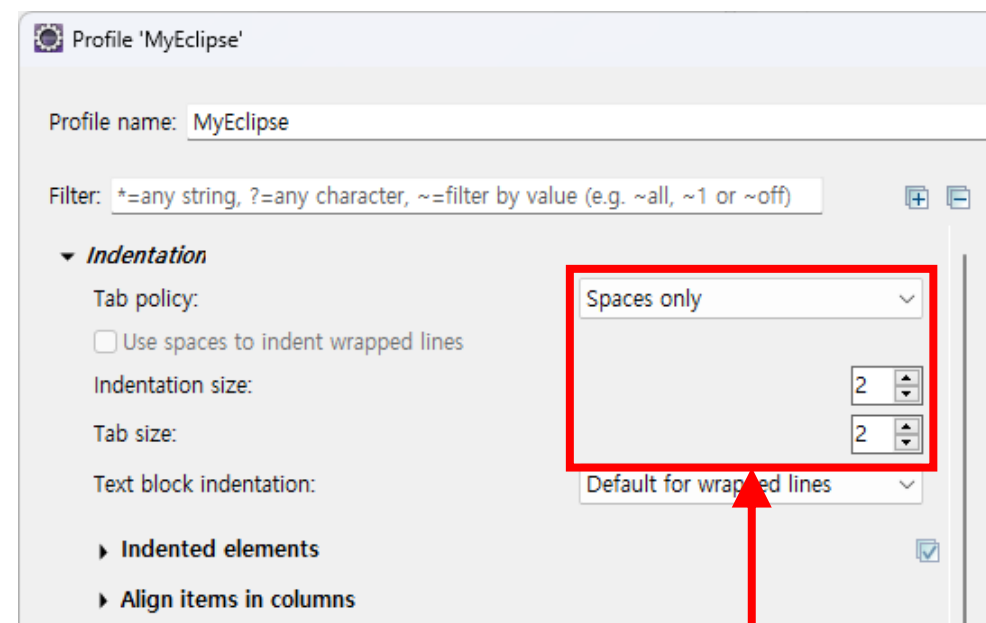
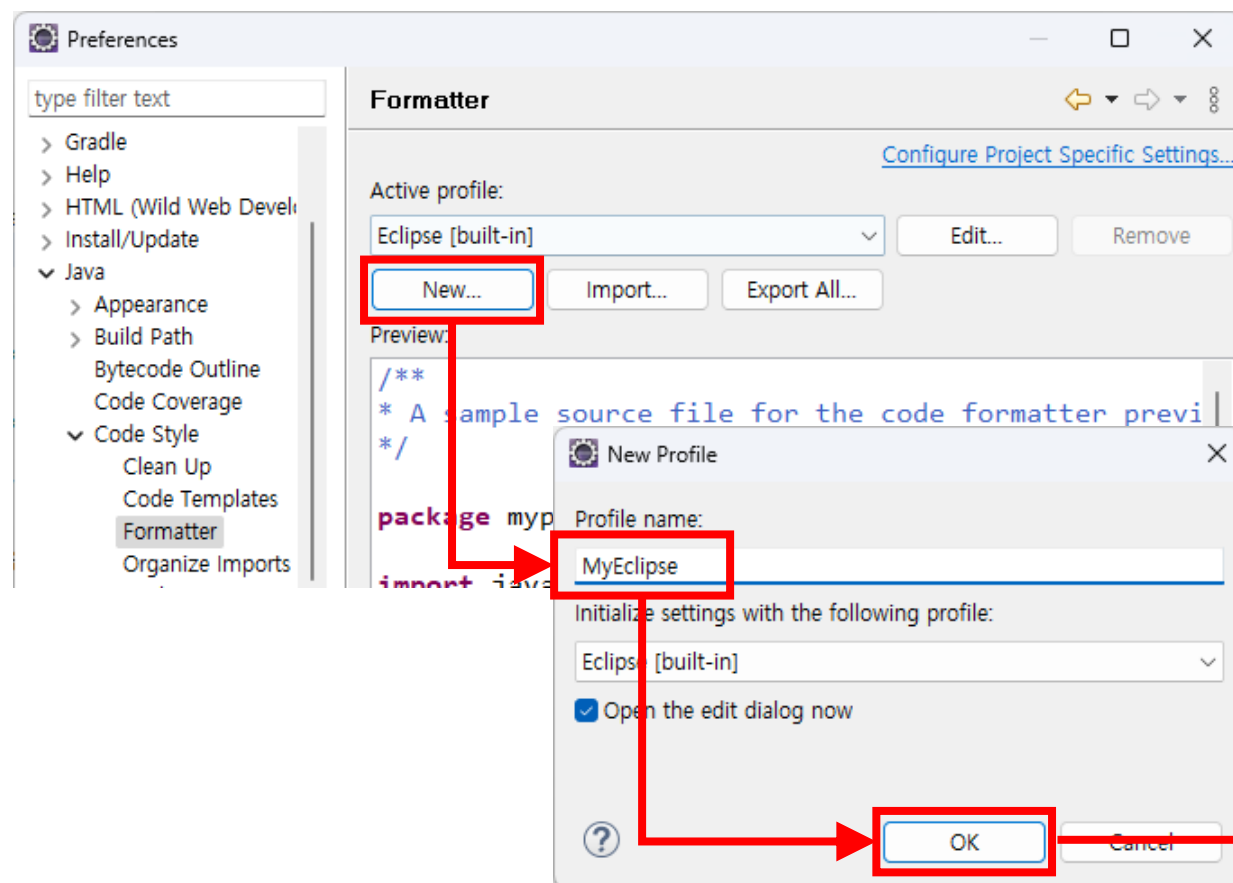
# Preferences 설정

- 인코딩 변경
  - ✓ General – Workspace – Text file encoding
  - ✓ Default (MS949)를 UTF-8로 변경



# Preferences 설정

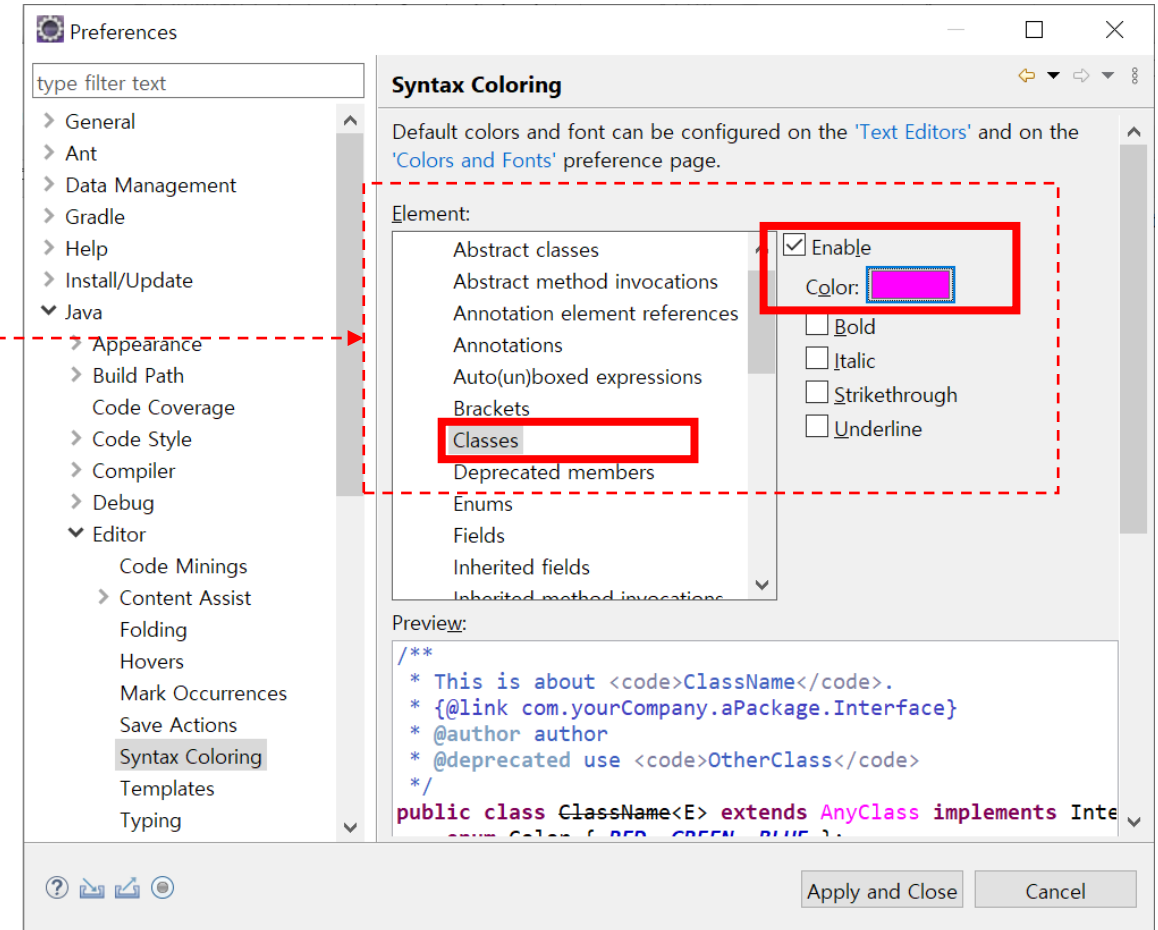
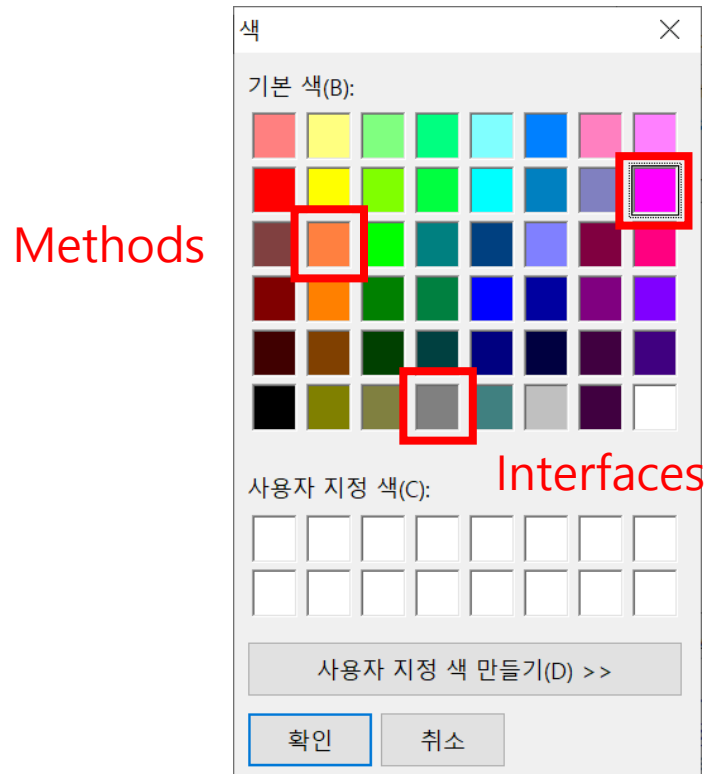
- 들여쓰기 설정
  - ✓ Java – Code Style – Formatter
  - ✓ 새로운 profile 생성 후 들여쓰기 수준을 4에서 2로 변경



# Preferences 설정

- Syntax Coloring

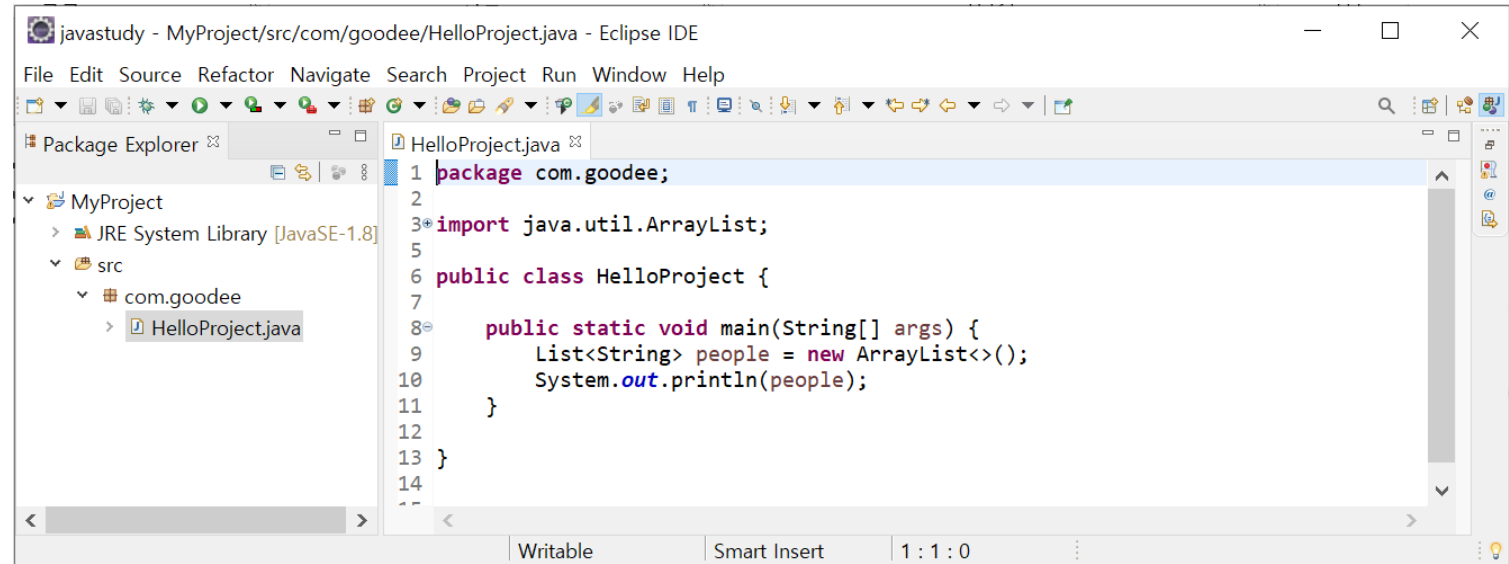
- ✓ Java – Editor – Syntax Coloring
- ✓ 3가지 색상 변경 : Classes, Interfaces, Methods



# Syntax Coloring이 필요한 이유

- Syntax Coloring 이전

- ✓ 클래스, 메소드, 인터페이스가 모두 검정색으로 구분이 안 됨



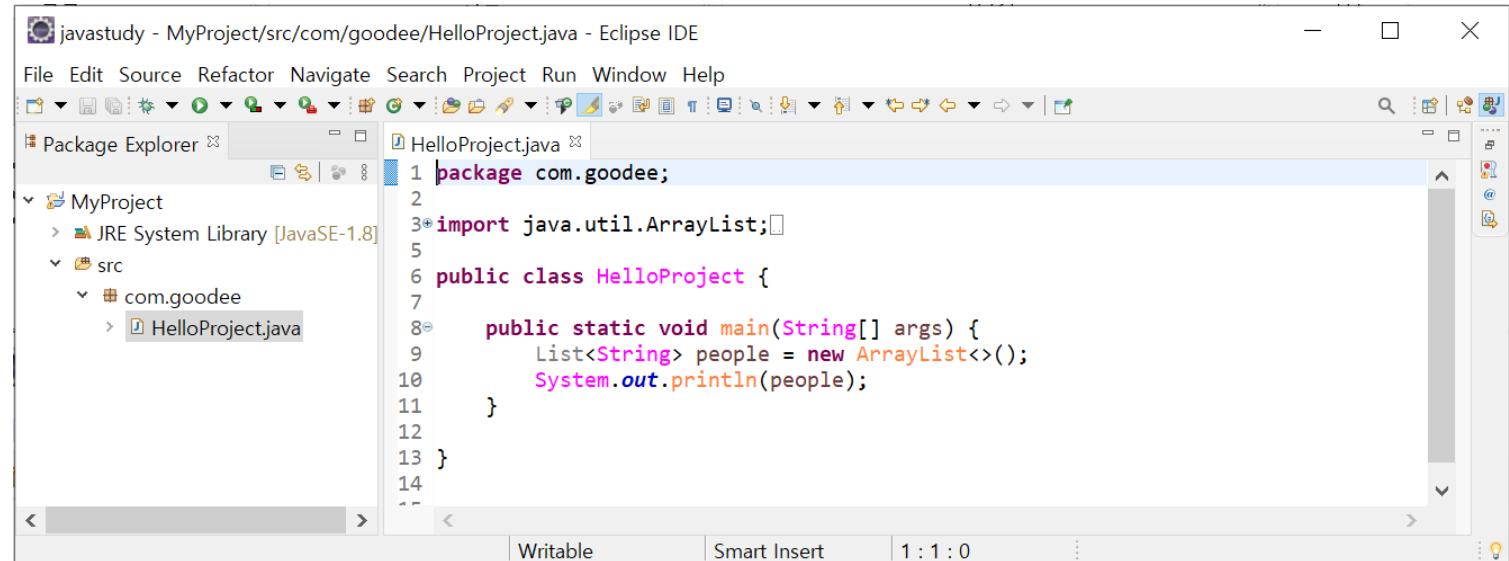
The screenshot shows the Eclipse IDE interface with a Java file named 'HelloProject.java'. The code is as follows:

```
1 package com.goodee;  
2  
3 import java.util.ArrayList;  
4  
5  
6 public class HelloProject {  
7  
8     public static void main(String[] args) {  
9         List<String> people = new ArrayList<>();  
10        System.out.println(people);  
11    }  
12  
13 }  
14
```

In this version, all text is black, making it difficult to distinguish between different syntactic elements like keywords, identifiers, and literals.

- Syntax Coloring 이후

- ✓ 클래스, 메소드, 인터페이스를 모두 구분할 수 있음



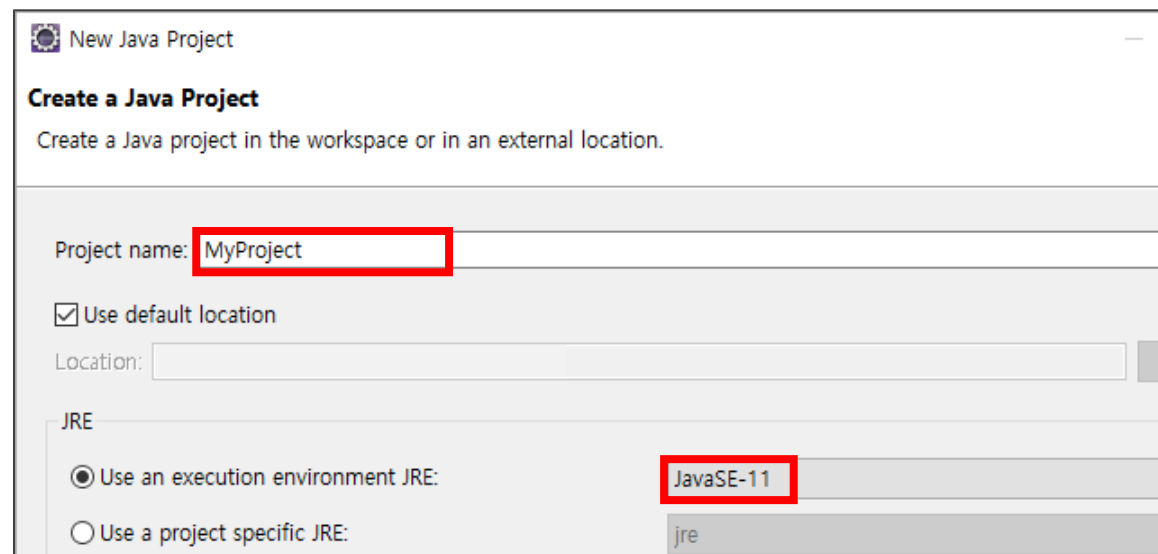
The screenshot shows the same Eclipse IDE interface with the same Java file, but now with syntax coloring applied. The code is as follows:

```
1 package com.goodee;  
2  
3 import java.util.ArrayList;  
4  
5  
6 public class HelloProject {  
7  
8     public static void main(String[] args) {  
9         List<String> people = new ArrayList<>();  
10        System.out.println(people);  
11    }  
12  
13 }  
14
```

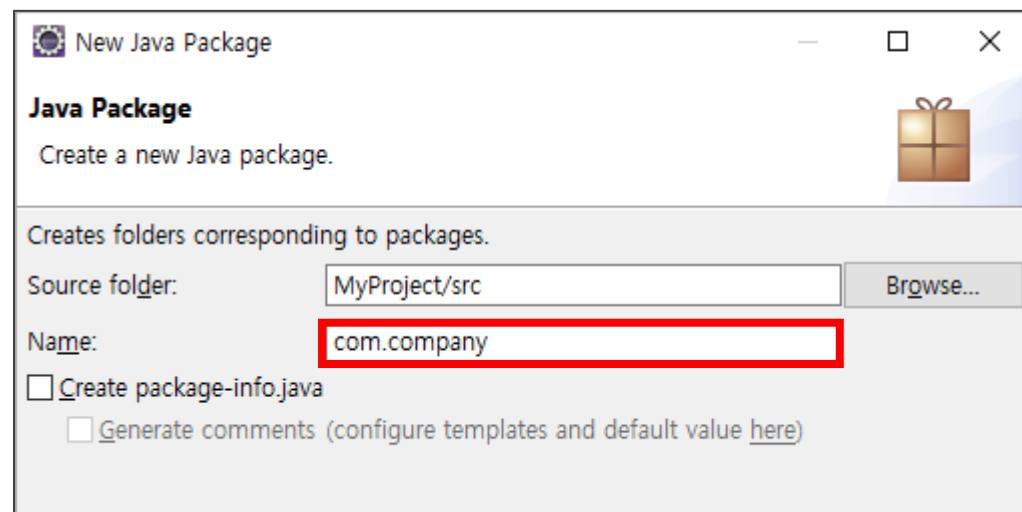
With syntax coloring, keywords like 'package', 'import', 'public', 'class', 'static', 'void', 'main', 'String', 'List', 'new', 'ArrayList', 'System.out', and 'println' are highlighted in different colors (purple, blue, green, etc.), making the code structure much clearer and easier to read.

# 이클립스를 이용한 첫 프로그램 작성

## 1. Java Project 생성 (MyProject)



## 2. 소스 폴더 src에 패키지 생성 (com.company)



# 이클립스를 이용한 첫 프로그램 작성

## 1. 자바 클래스 생성 (Hello)

## 2. 소스 코드 작성

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

**New Java Class**

Create a new Java class.

Source folder: MyProject/src Browse...

Package: com.company Browse...

☐ Enclosing type: Browse...

Name: **Hello**

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

Superclass: java.lang.Object Browse...

Interfaces: Add... Remove

Which method stubs would you like to create?

☒ **public static void main(String[] args)**  
☐ Constructors from superclass  
☒ Inherited abstract methods

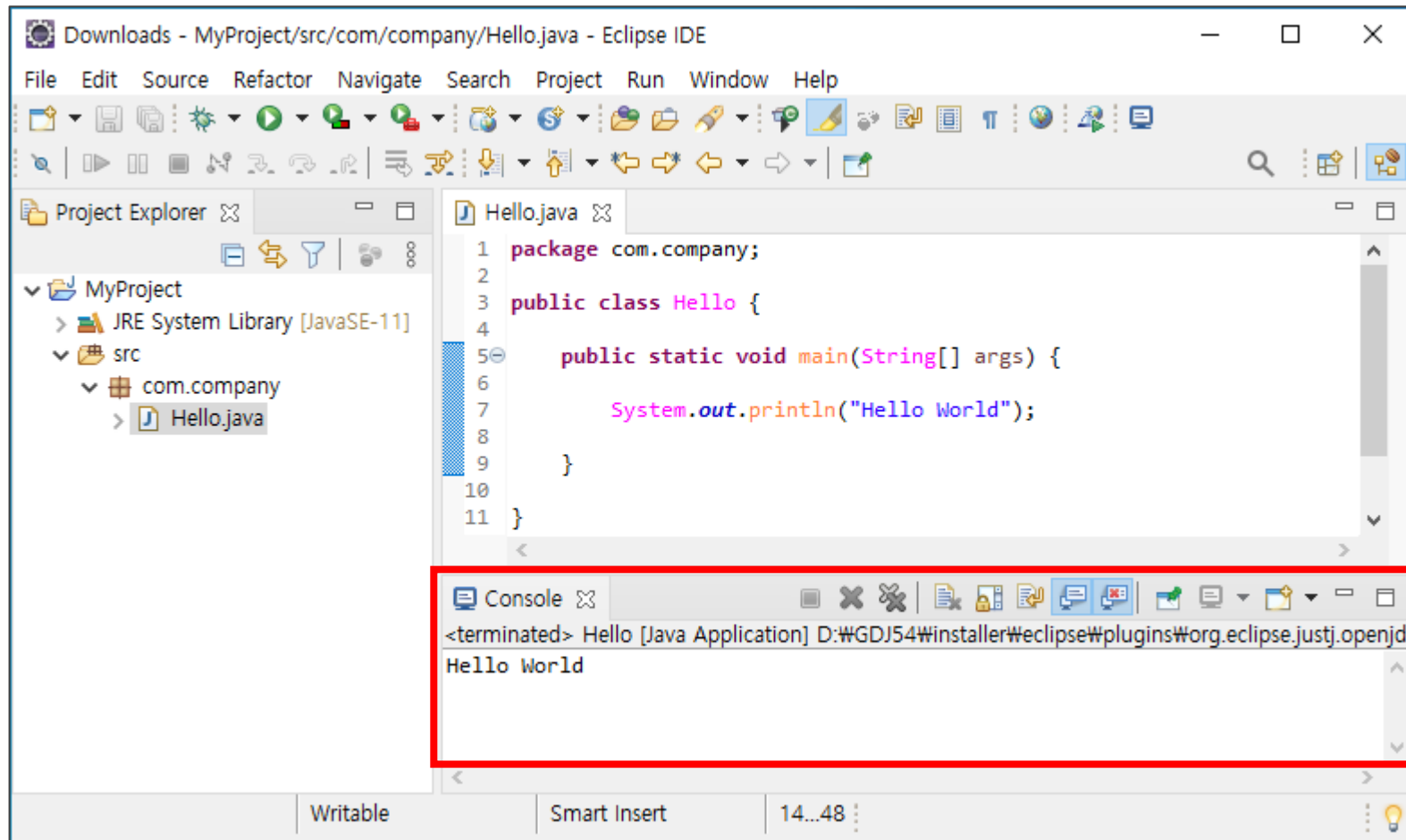
Do you want to add comments? (Configure templates and default value [here](#))  
☐ Generate comments

? Finish Cancel



# 이클립스를 이용한 첫 프로그램 작성

## 3. 실행 Ctrl + F11



# 이클립스 주요 단축키

- 주석

- ✓ Ctrl + / : 한 줄 주석(//) 설정 및 해제
- ✓ Ctrl + Shift + / : 블록 주석(\* \*/) 설정
- ✓ Ctrl + Shift + ⌘ : 블록 주석(\* \*/) 해제

- 찾기/바꾸기

- ✓ Ctrl + F : 찾기/바꾸기(해당 소스코드 내부)
- ✓ Ctrl + H : 찾기(해당 프로젝트 내부)
- ✓ Ctrl + K : 찾기(블록 지정한 단어)

- 이동

- ✓ Ctrl + 클릭 : 클릭한 클래스 및 메소드로 이동
- ✓ Alt + ← : 이전 작업 파일로 이동
- ✓ Alt + → : 다음 작업 파일로 이동

- 블록 지정

- ✓ Shift + 방향키 : 블록 지정
- ✓ Shift + Home : 커서 위치부터 처음까지 블록 지정
- ✓ Shift + End : 커서 위치부터 끝까지 블록 지정
- ✓ Ctrl + A : 전체 블록 지정

- 실행

- ✓ F11 : 실행 (에러가 나면 디버깅 함)
- ✓ Ctrl + F11 : 실행(에러가 나도 디버깅 안 함)

- 소스코드 편집

- ✓ Ctrl + Z : 작업 취소
- ✓ Ctrl + Y : 작업 취소의 취소(다시 작업)
- ✓ Ctrl + D : 한 줄 삭제
- ✓ Ctrl + S : 현재 파일 저장
- ✓ Ctrl + Shift + S : 모든 파일 저장
- ✓ Ctrl + Shift + X : 대문자로 변환
- ✓ Ctrl + Shift + Y : 소문자로 변환
- ✓ Ctrl + Shift + F : 소스코드 자동 정렬
- ✓ Ctrl + Shift + O : import 자동 정리
- ✓ Alt + ↑ ↓ : 소스코드 위아래 이동
- ✓ Ctrl + Alt + ↑ ↓ : 소스코드 위아래 복사

- 자동 완성

- ✓ Ctrl + Space : 자동완성
- ✓ sysout : System.out.println()
- ✓ try : try { } catch(Exception e) { }