

프로그래밍 기초

▶ 프로그래밍

✓ 프로그램(Program)

컴퓨터가 인식할 수 있는 명령어의 나열(집합)

✓ 프로그래밍(Programming)

프로그램을 작성하는 과정 = 코딩

✓ 프로그래머(Programmer)

프로그램을 작성하는 사람

▶ 프로그래밍 언어

프로그램을 작성하기 위한 언어체계, 사람이 컴퓨터와 소통하게 하는 요소

Sep 2018	Sep 2017	Change	Programming Language	Ratings	Change
1	1		Java	17.436%	+4.75%
2	2		C	15.447%	+8.06%
3	5	▲	Python	7.653%	+4.67%
4	3	▼	C++	7.394%	+1.83%
5	8	▲	Visual Basic .NET	5.308%	+3.33%
6	4	▼	C#	3.295%	-1.48%
7	6	▼	PHP	2.775%	+0.57%
8	7	▼	JavaScript	2.131%	+0.11%
9	-	▲	SQL	2.062%	+2.06%
10	18	▲	Objective-C	1.509%	+0.00%
11	12	▲	Delphi/Object Pascal	1.292%	-0.49%
12	10	▼	Ruby	1.291%	-0.64%
13	16	▲	MATLAB	1.276%	-0.35%
14	15	▲	Assembly language	1.232%	-0.41%
15	13	▼	Swift	1.223%	-0.54%
16	17	▲	Go	1.081%	-0.49%
17	9	▼	Perl	1.073%	-0.88%
18	11	▼	R	1.016%	-0.80%
19	19		PL/SQL	0.850%	-0.63%
20	14	▼	Visual Basic	0.682%	-1.07%

Worldwide, Sept 2018 compared to a year ago:				
Rank	Change	Language	Share	Trend
1	▲	Python	24.58 %	+5.7 %
2	▼	Java	22.14 %	-0.6 %
3	▲	Javascript	8.41 %	+0.0 %
4	▼	PHP	7.77 %	-1.4 %
5		C#	7.74 %	-0.4 %
6		C/C++	6.22 %	-0.8 %
7		R	4.04 %	-0.2 %
8		Objective-C	3.33 %	-0.9 %
9		Swift	2.65 %	-0.9 %
10		Matlab	2.1 %	-0.3 %
11		Ruby	1.62 %	-0.4 %
12	▲▲▲	TypeScript	1.57 %	+0.2 %
13	▼	VBA	1.49 %	-0.0 %
14	▼	Scala	1.23 %	-0.1 %
15	▼	Visual Basic	1.11 %	-0.2 %
16	▲▲	Kotlin	1.02 %	+0.4 %
17		Go	0.94 %	+0.3 %
18	▼▼	Perl	0.7 %	-0.1 %
19		Lua	0.38 %	-0.0 %
20	▲	Rust	0.38 %	+0.0 %
21	▲	Haskell	0.3 %	-0.0 %
22	▼▼	Delphi	0.25 %	-0.1 %
© Pierre Carbonnelle, 2018				

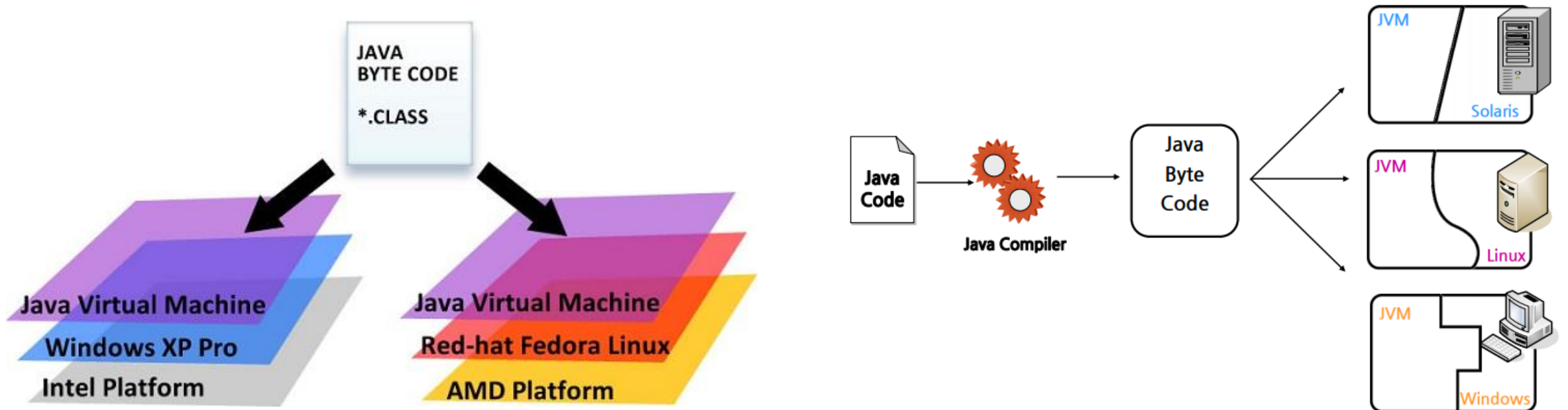
▶ 자바 언어 특징



1. 운영체제에 독립적
2. 사용하기 쉬운 언어
 - 다른 언어의 단점 보완(포인터, 메모리 관리)
 - 객체 지향 언어
 - 능률적이고 명확한 코드 작성 가능
3. 자동 메모리 관리(Garbage Collection)
4. 네트워크와 분산환경 지원
5. 멀티쓰래드 지원

▶ JVM(Java Virtual Machine)

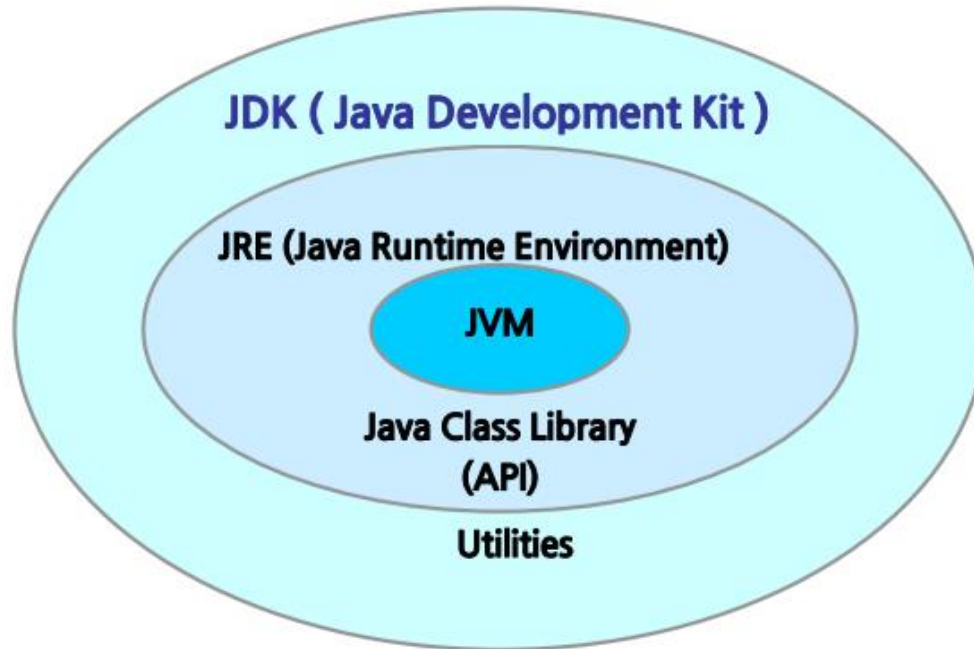
자바를 실행하기 위한 가상 기계로 플랫폼에 의존적
byte code(class파일)를 해석하고 실행하는 interpreter



▶ 자바 개발 환경

✓ 설치 범위

사용자/개발자 입장에 따라 설치하는 범위가 달라짐



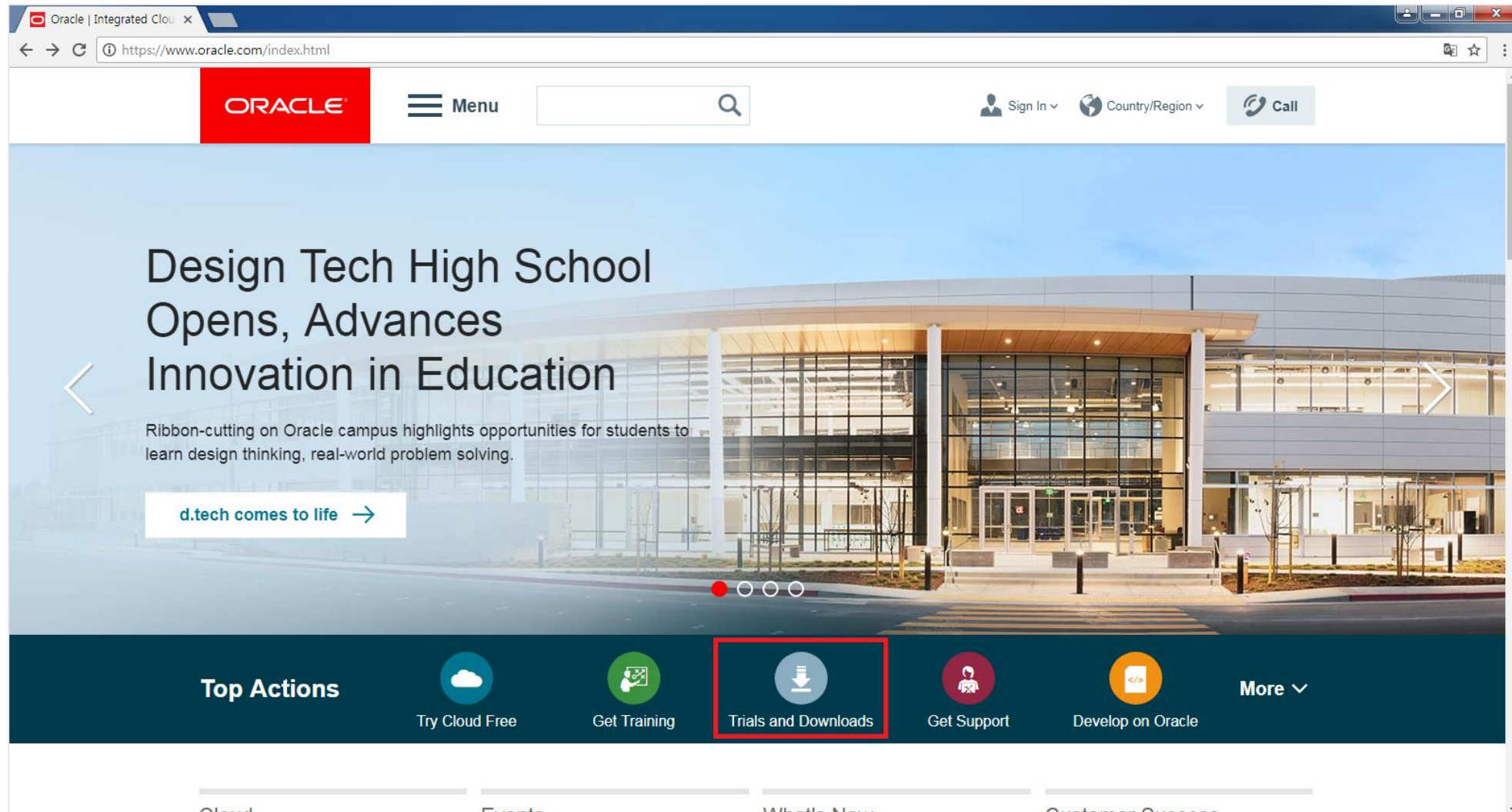
Java SE : Java Standard Edition

Java EE : Java Enterprise Edition

Java ME : Java Micro Edition

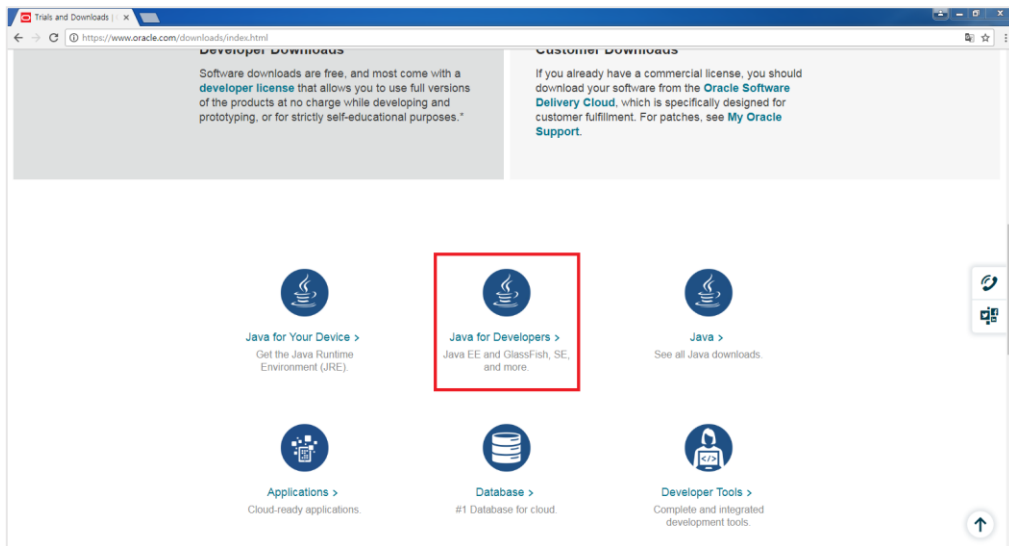
▶ 자바 설치

Oracle 홈페이지 접속(www.oracle.com) – Trials and Downloads

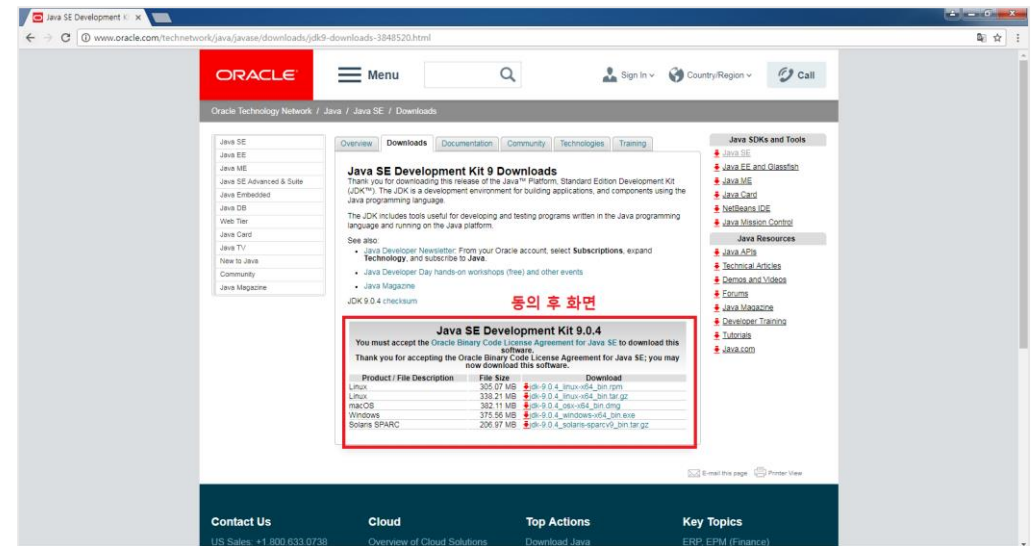


▶ 자바 설치

Java for Developers -> Java Platform (JDK) 9

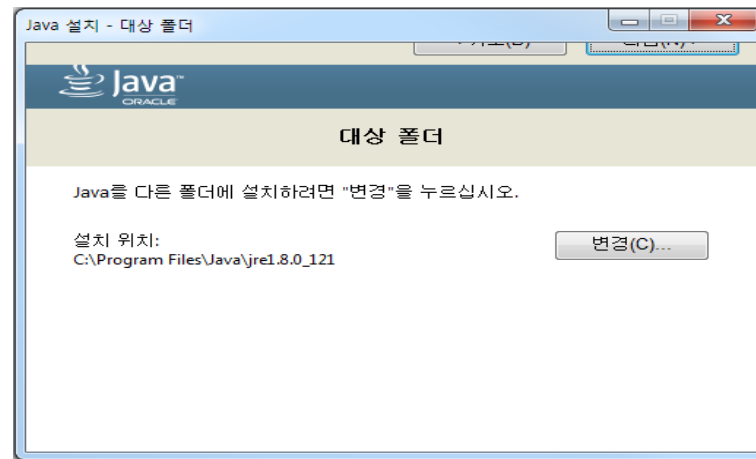
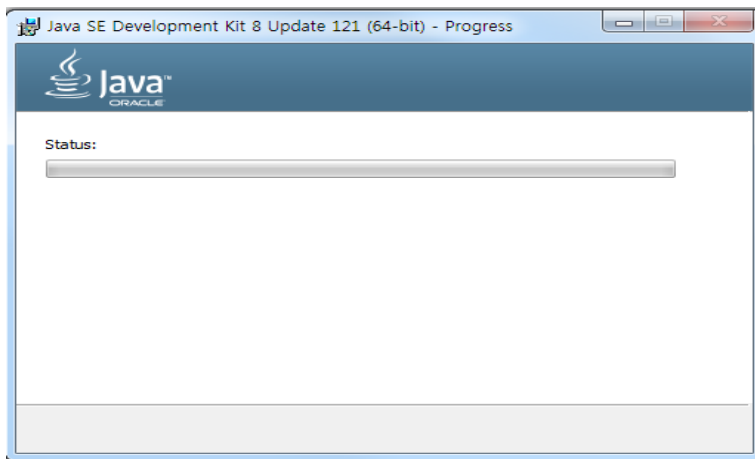
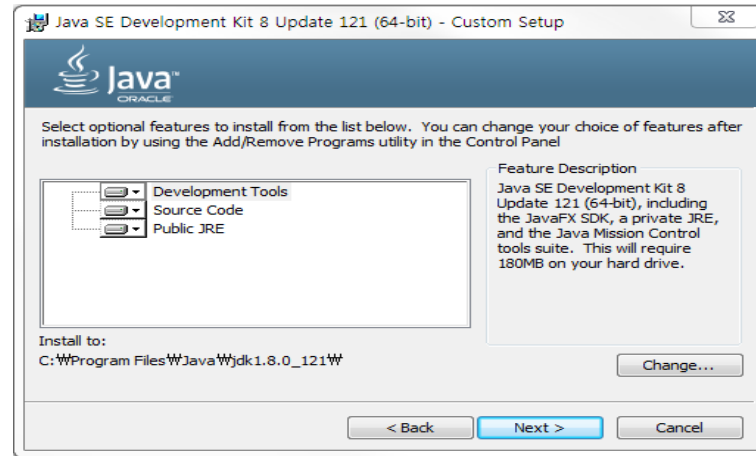
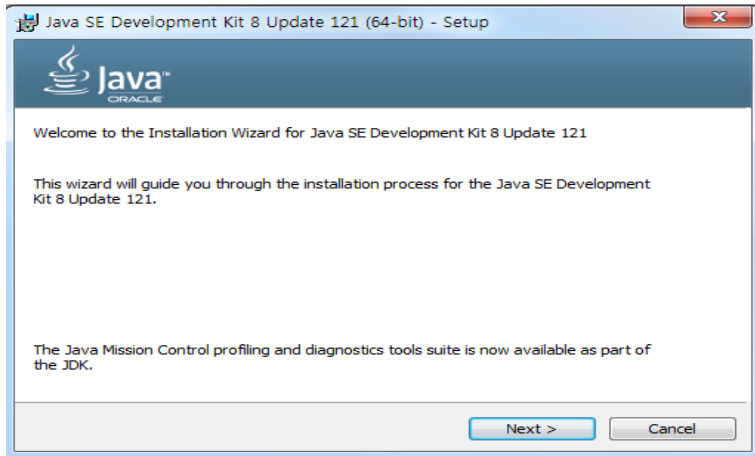


Java SE 최신 버전 동의 후 운영체제에 맞게 선택



▶ 자바 설치

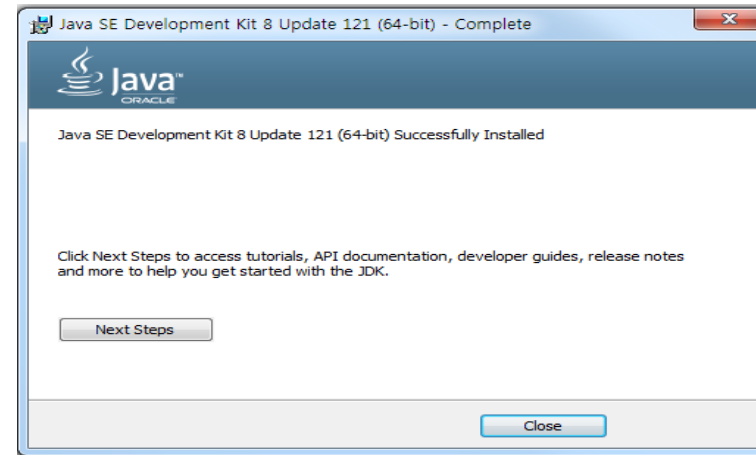
모두 Default로 놓고 next, 다음 버튼 클릭



폴더 변경 시 폴더 위치 기억할 것.

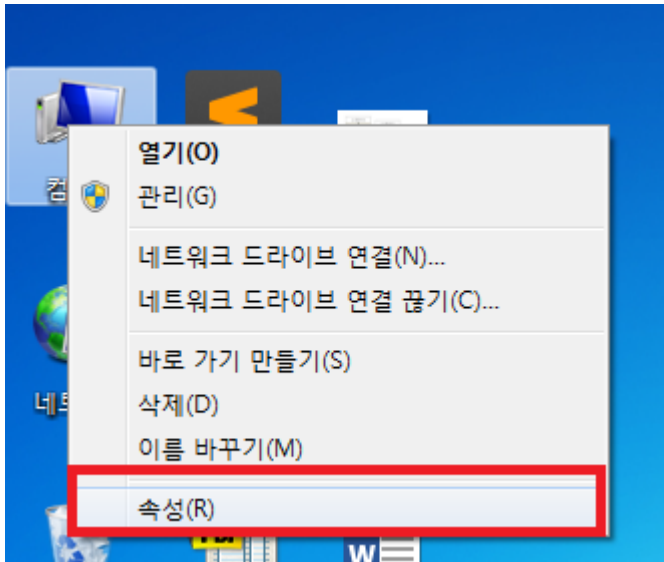
▶ 자바 설치

모두 Default로 놓고 next, 다음 버튼 클릭



▶ 자바 설정

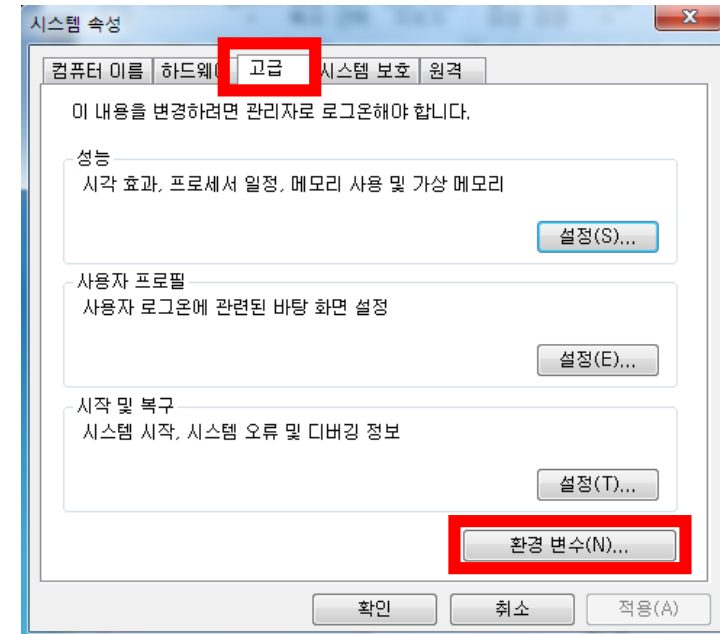
'내 컴퓨터' 오른쪽 클릭 - 속성



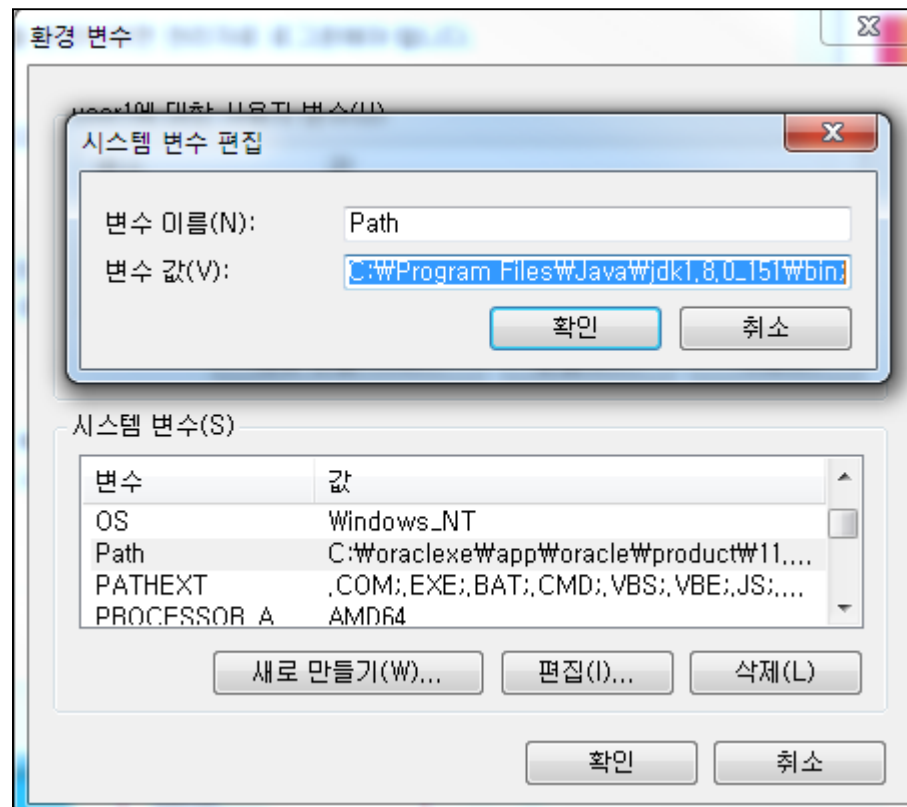
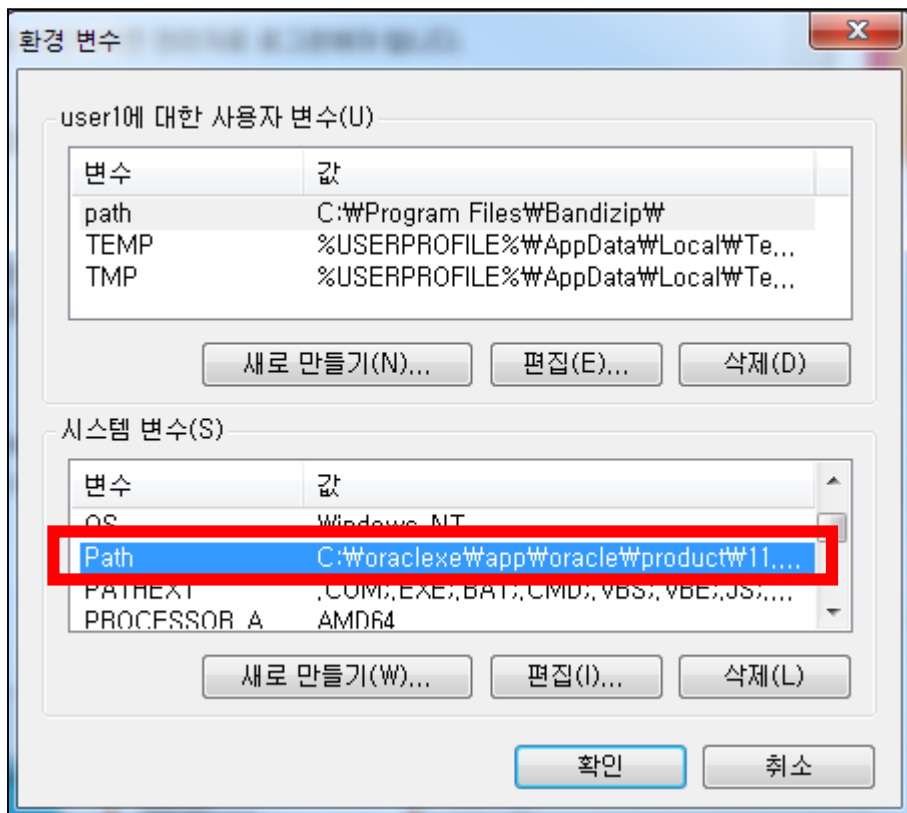
고급시스템 설정



고급 - 환경변수



▶ 자바 설정

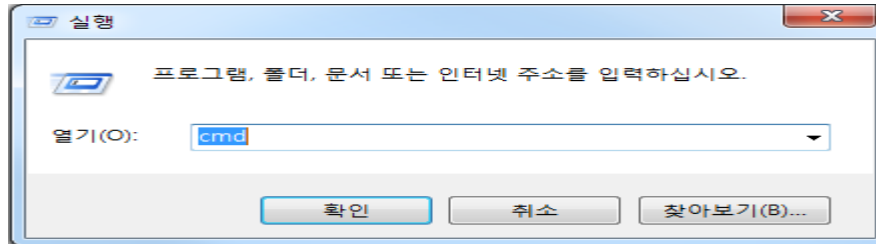


변수 값 : JDK가 설치된 폴더

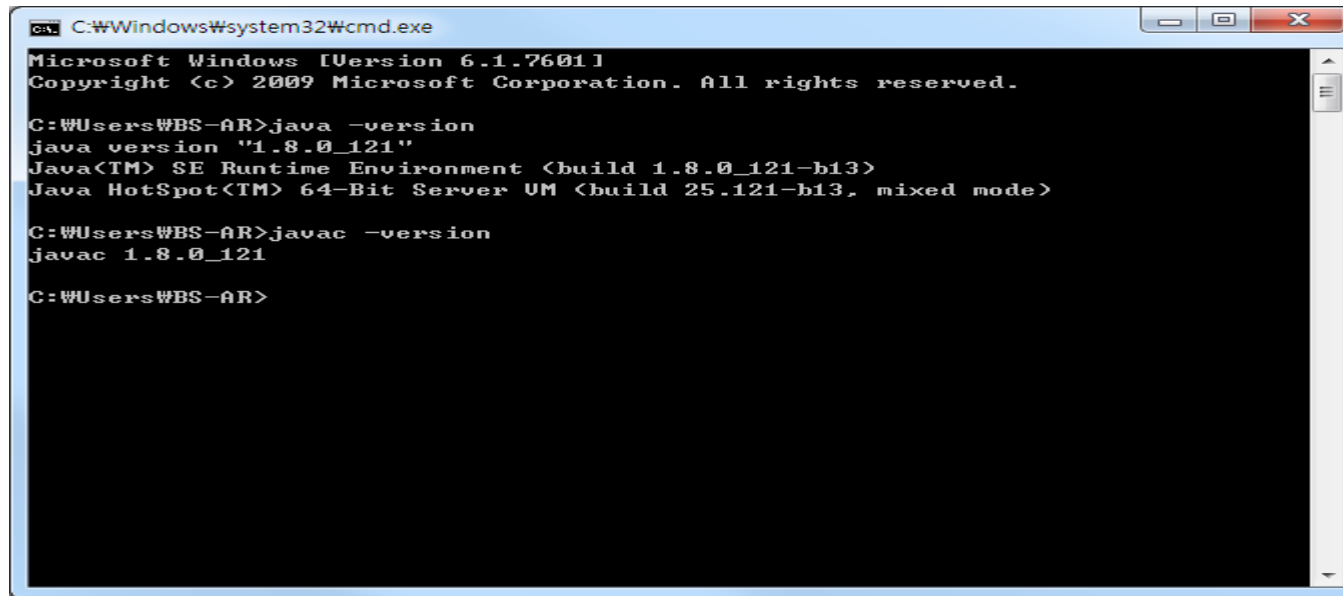
(C:\Program Files\Java\jdk1.8.0_151\bin;)

▶ 자바 설정

✓ 환경변수 테스트



윈도우키 + R 버튼 누르고 실행창에서 cmd엔터

A screenshot of a Windows Command Prompt window. The title bar shows 'C:\Windows\system32\cmd.exe'. The window contains the following text:

```
Microsoft Windows [Version 6.1.7601]  
Copyright (c) 2009 Microsoft Corporation. All rights reserved.  
  
C:\Users\WBS-AR>java -version  
java version "1.8.0_121"  
Java(TM) SE Runtime Environment (build 1.8.0_121-b13)  
Java HotSpot(TM) 64-Bit Server VM (build 25.121-b13, mixed mode)  
  
C:\Users\WBS-AR>javac -version  
javac 1.8.0_121  
  
C:\Users\WBS-AR>
```

java -version 엔터 / Javac -version 엔터
위 화면같이 버전 정보 나오면 설정완료

▶ 자바 개발 환경

✓ Compile Test

1. workspace 만들기

C:\workspace 폴더 생성

2. HelloWorld.java 파일 생성

C:\workspace\HelloWorld.java 파일 생성

3. cmd창 열기

시작 - 실행 - cmd 입력(윈도우 키 + r)

4. workspace 경로 찾아가기

C:\> cd workspace

5. Compile 하기

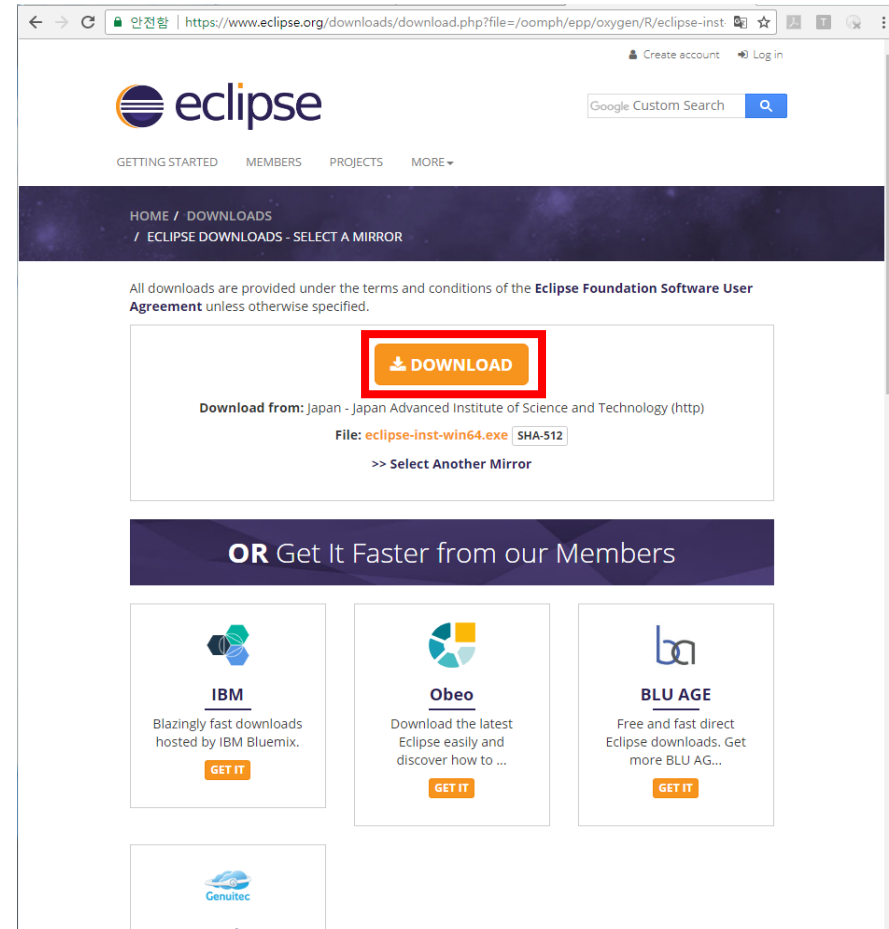
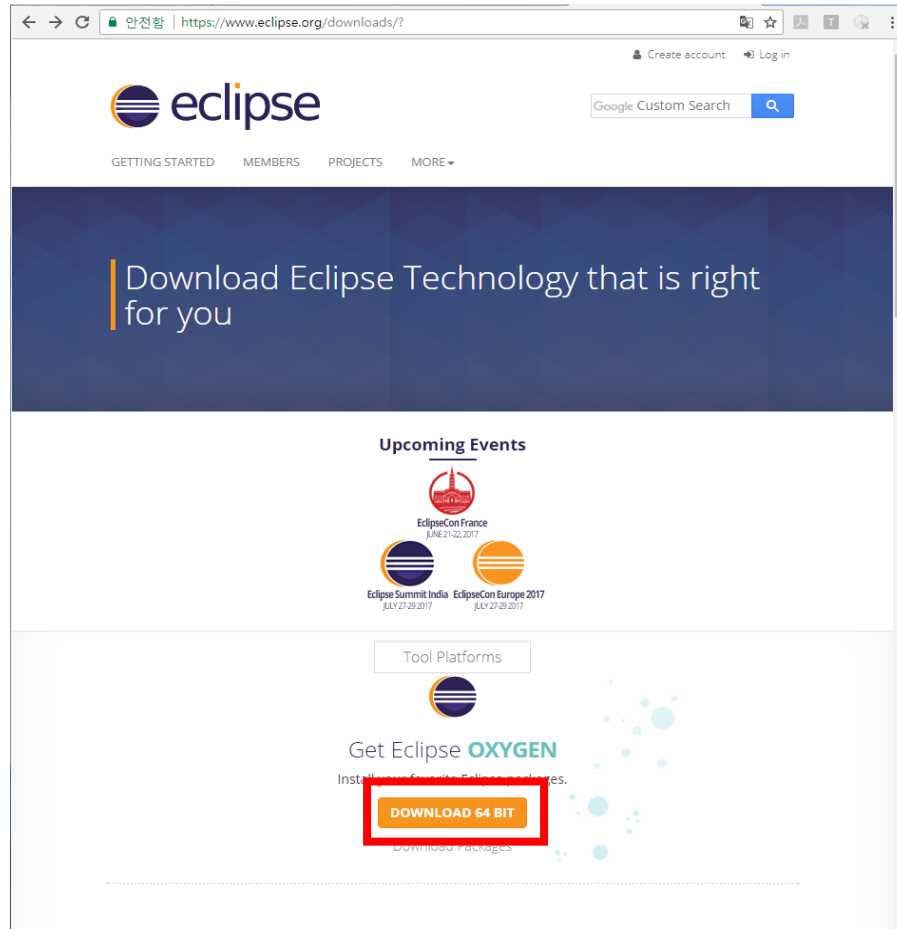
C:\workspace> javac HelloWorld.java

6. 실행하기

C:\workspace> java HelloWorld

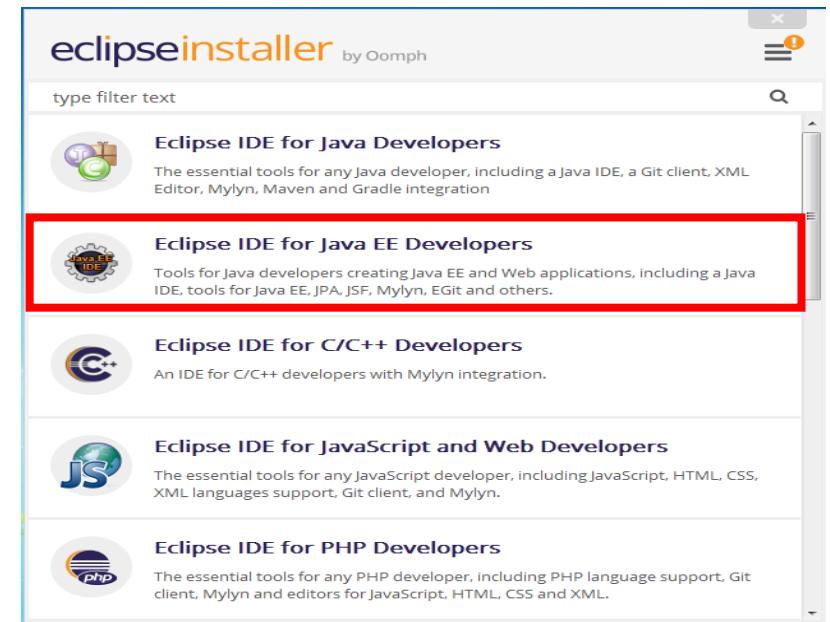
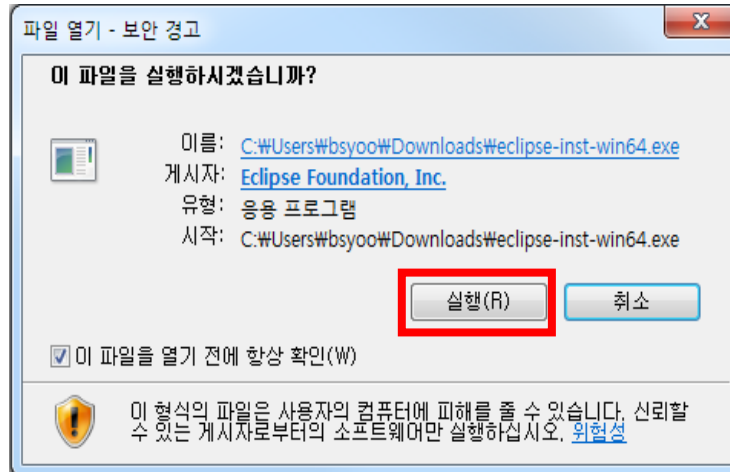
▶ 이클립스 설치

이클립스 다운로드 홈페이지(www.eclipse.org/downloads/)



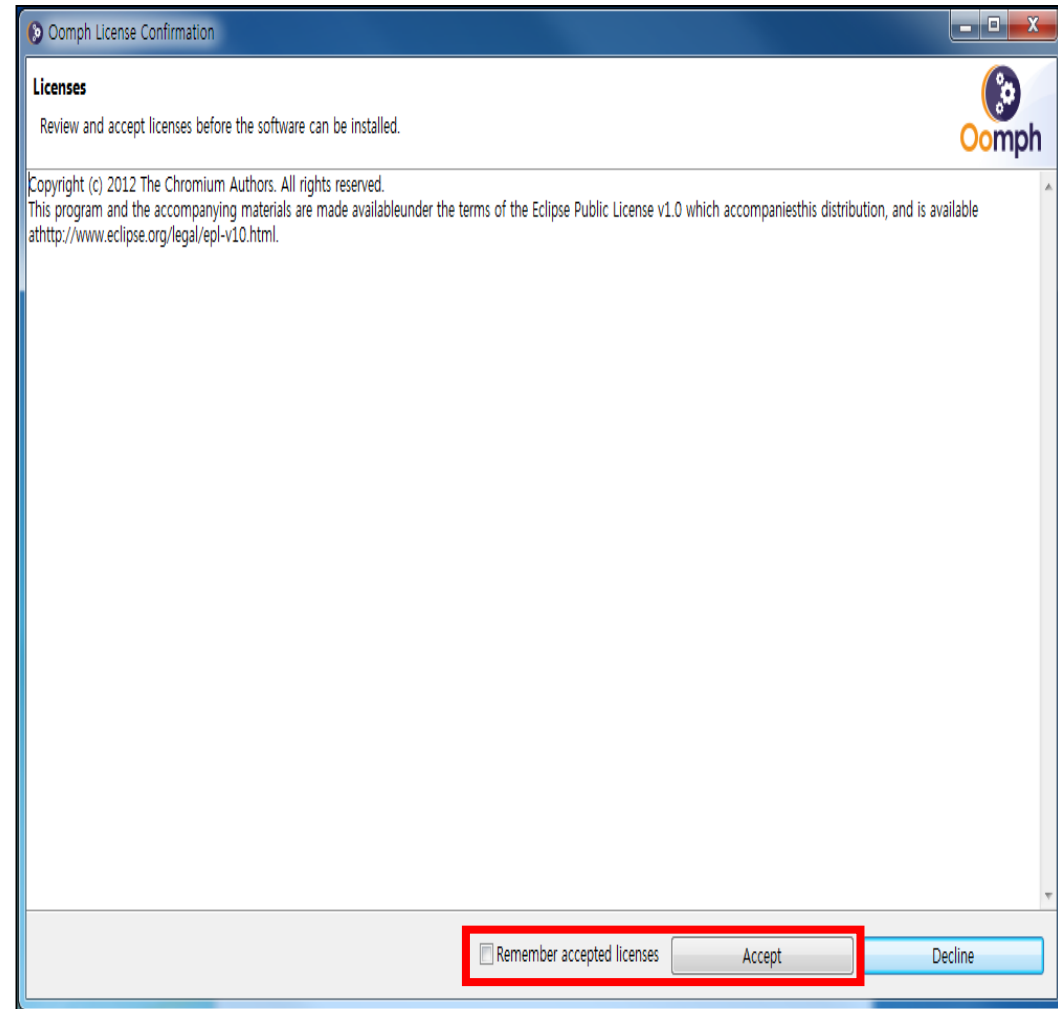
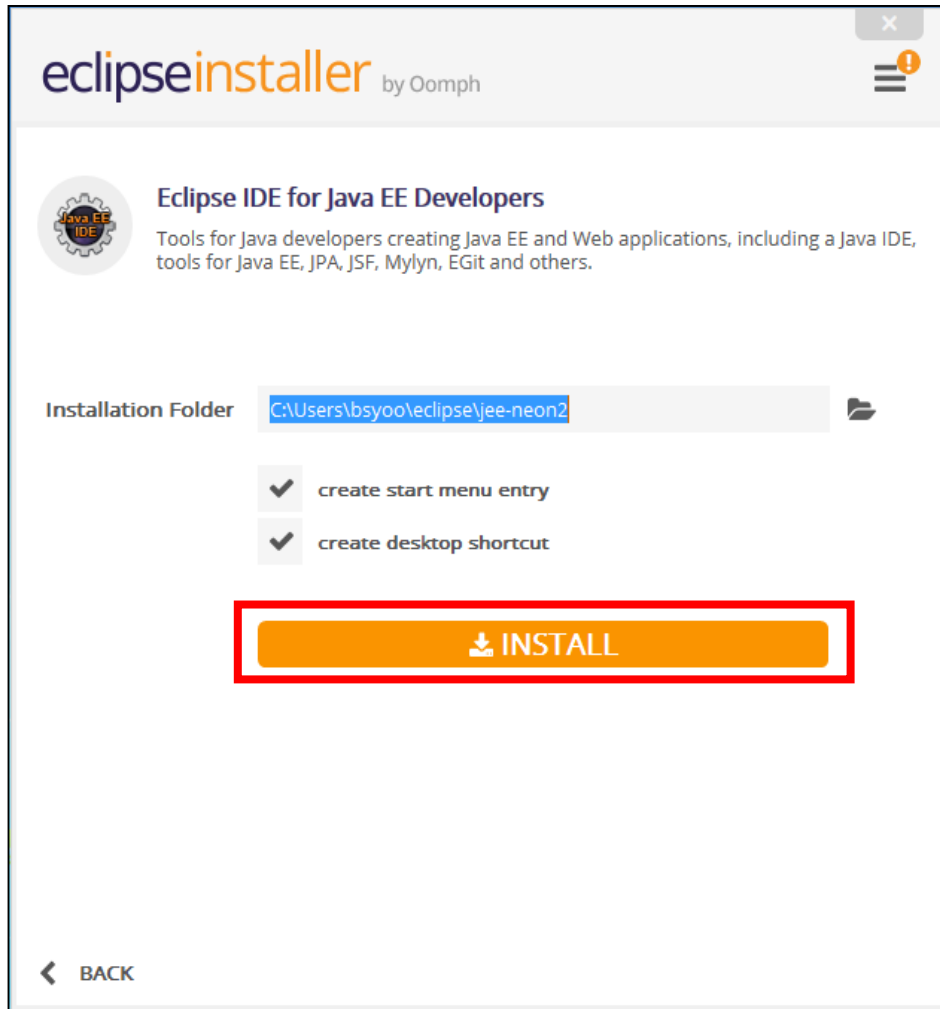
▶ 이클립스 설치

다운로드 폴더/다운로드 파일 확인



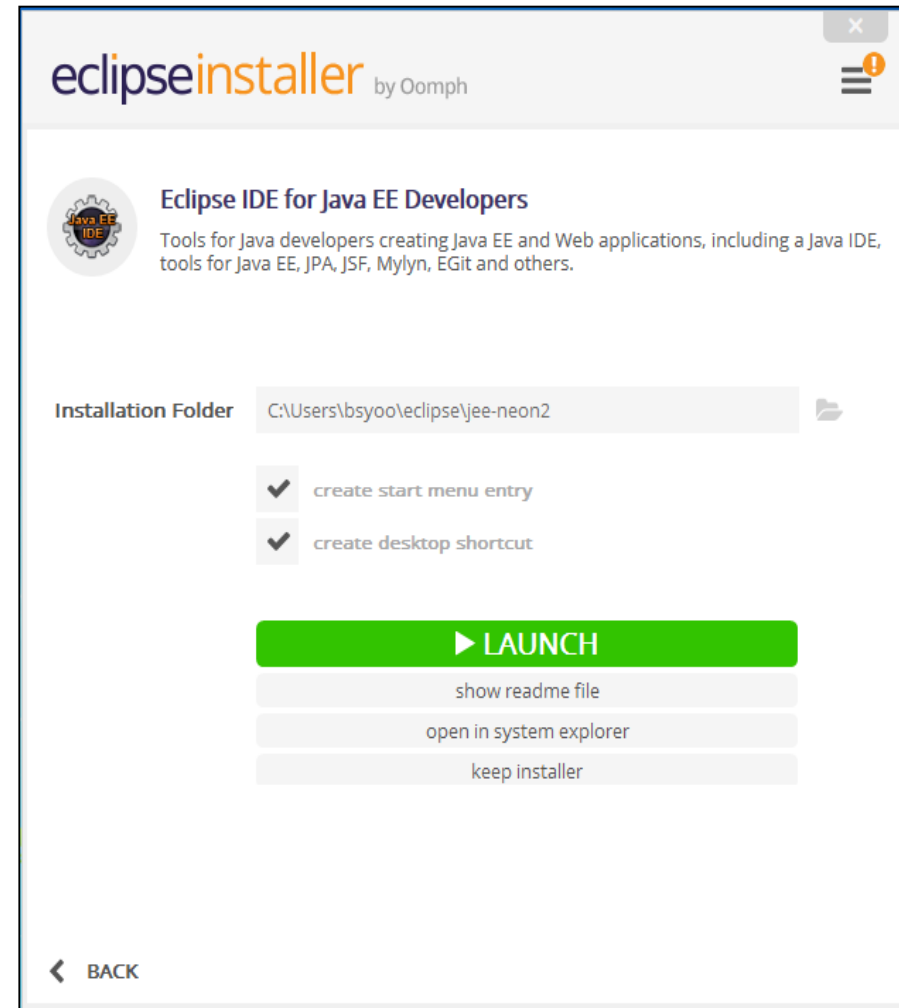
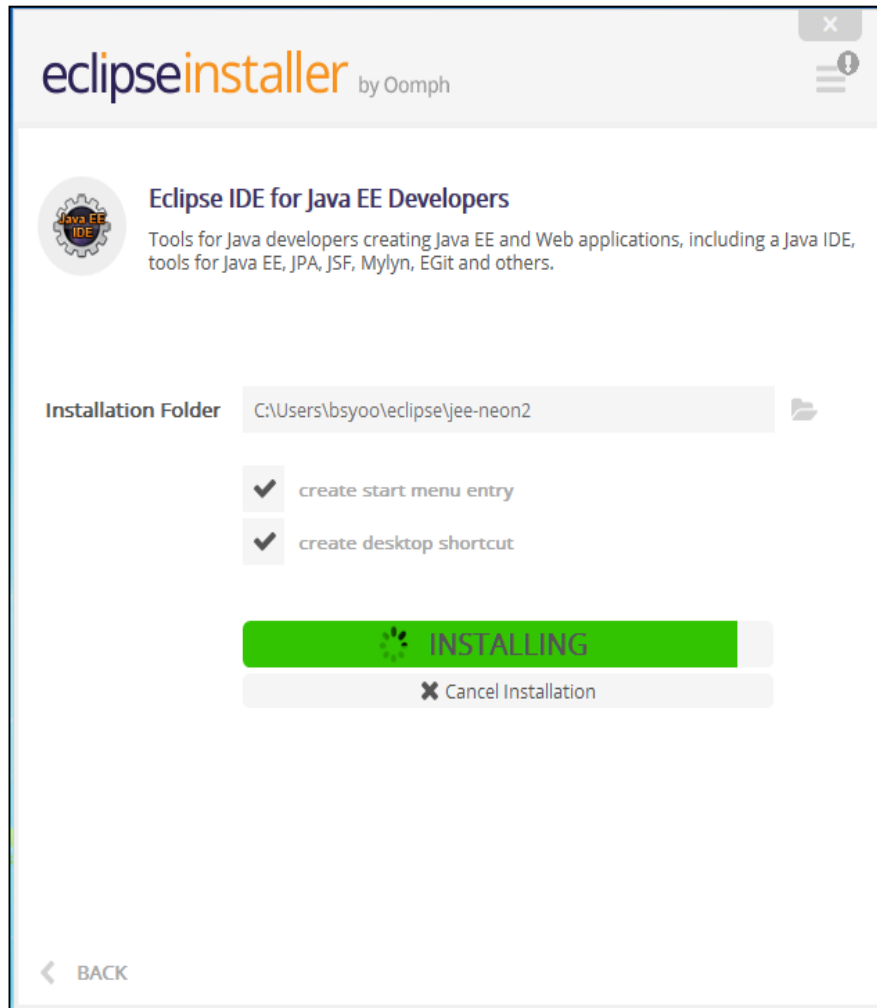
▶ 이클립스 설치

Java EE Developer 설치



▶ 이클립스 설치

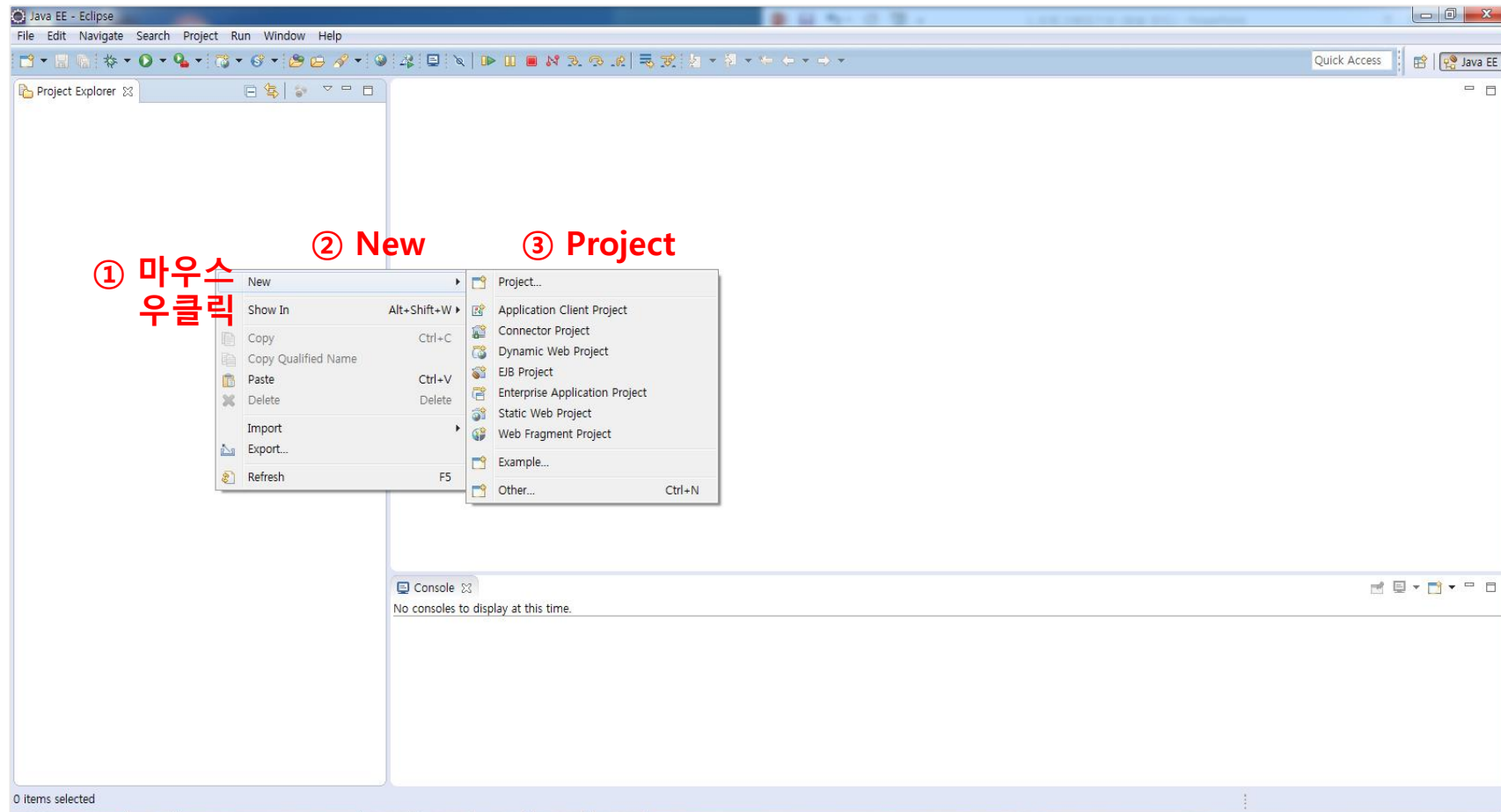
Java EE Developer 설치



▶ 자바 프로그래밍 순서

✓ Eclipse 환경

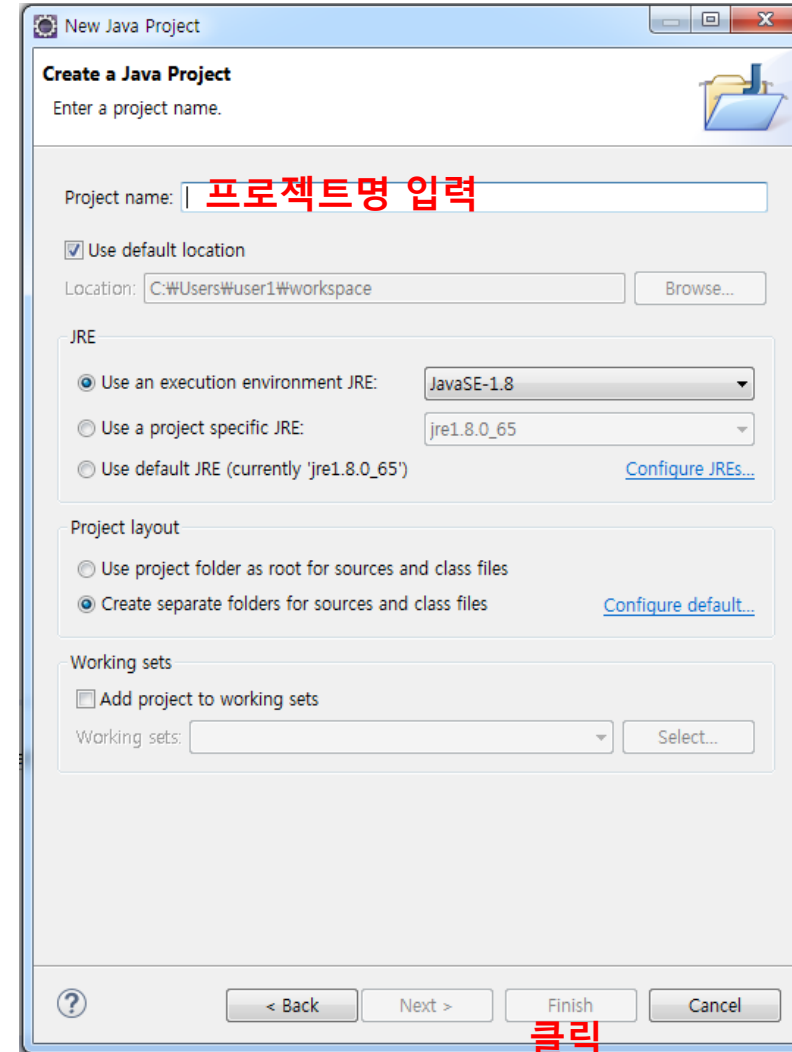
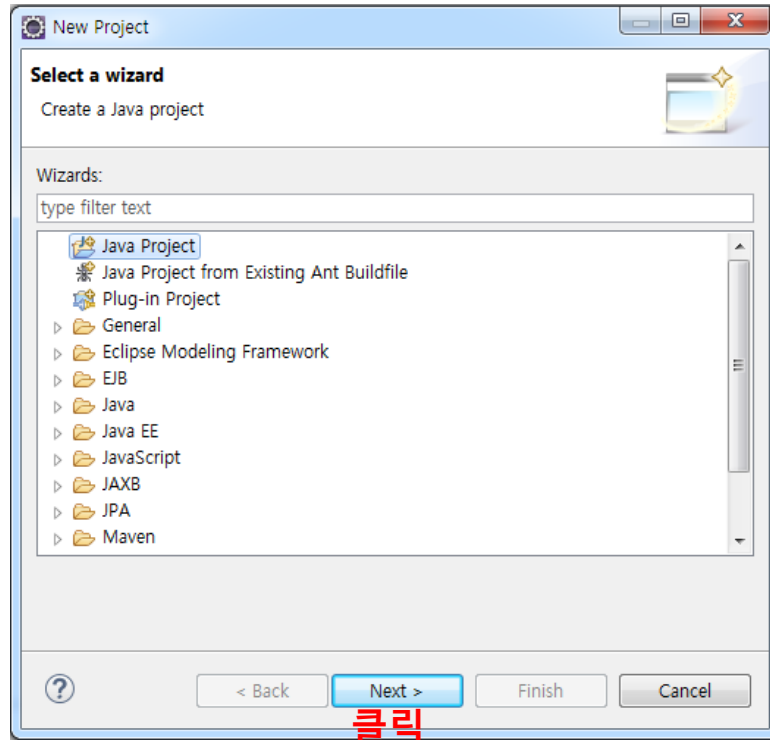
1. project 만들기



▶ 자바 프로그래밍 순서

✓ Eclipse 환경

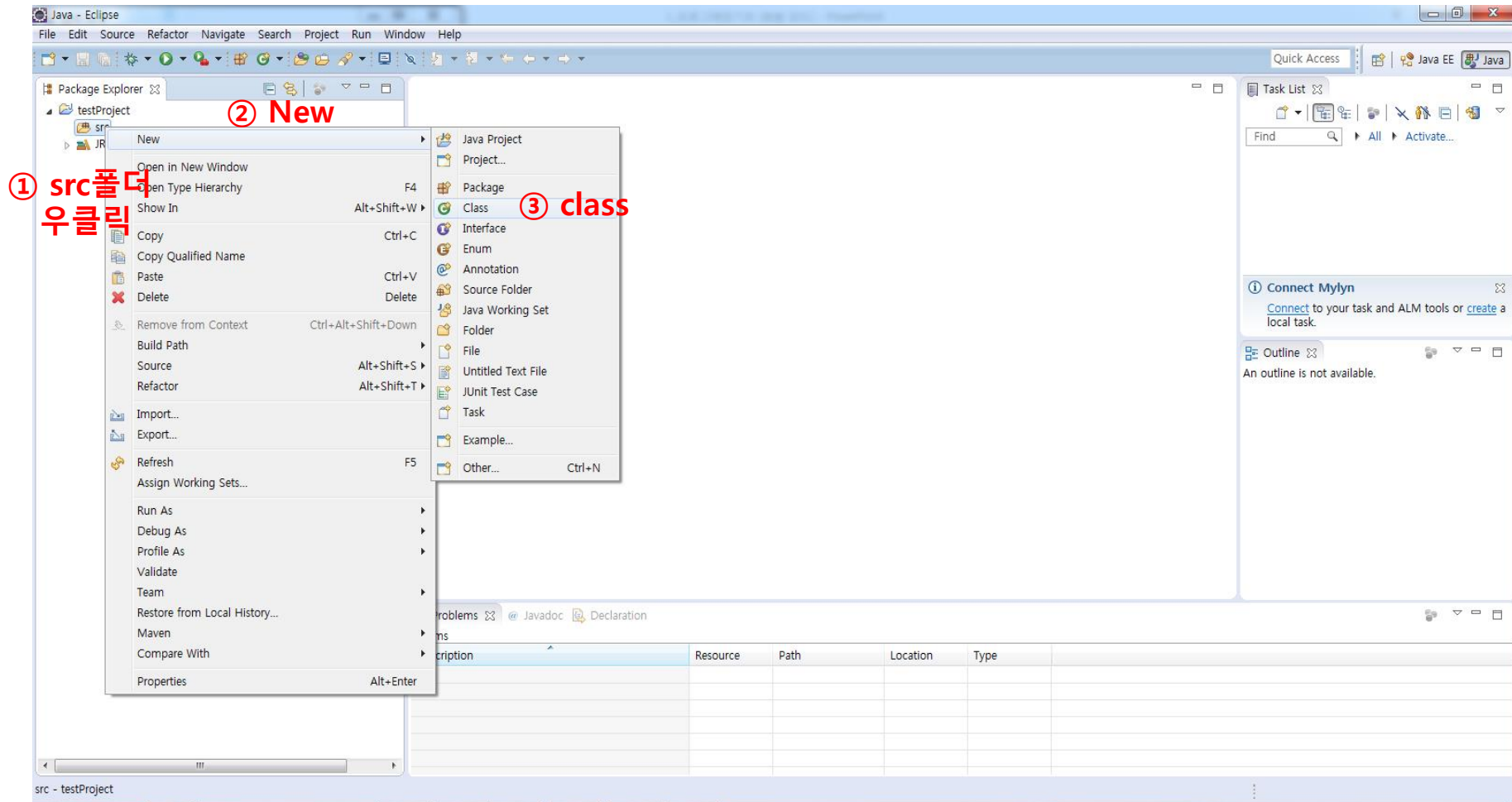
1. project 만들기



▶ 자바 프로그래밍 순서

✓ Eclipse 환경

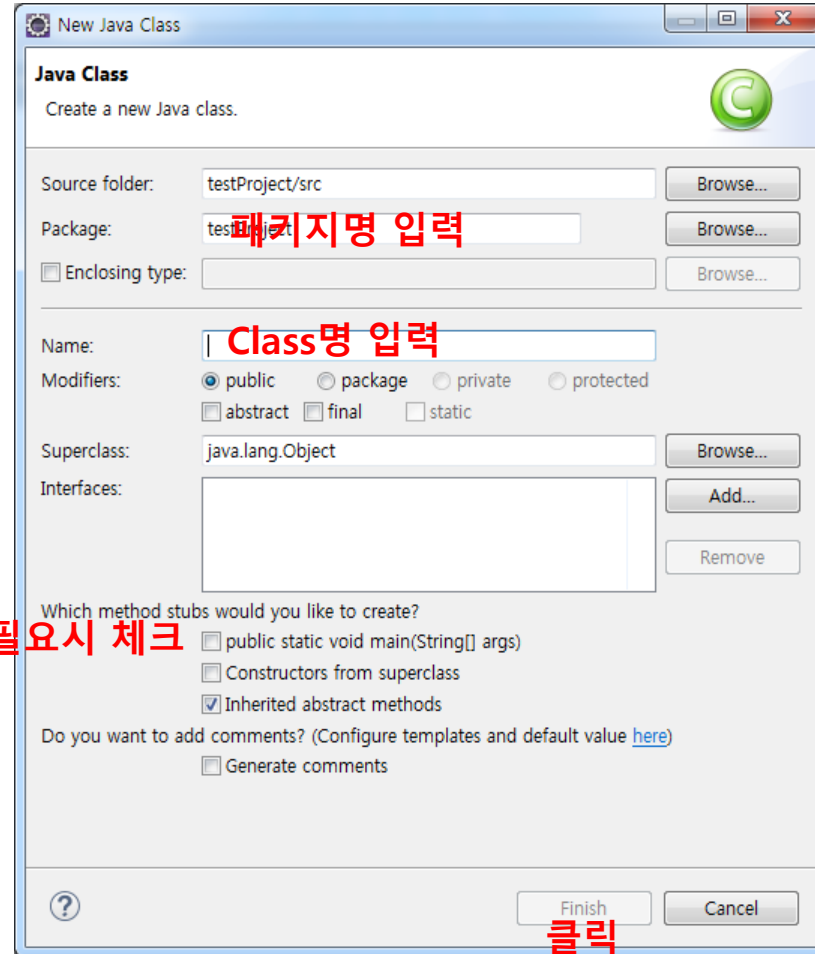
2. Class 만들기



▶ 자바 프로그래밍 순서

✓ Eclipse 환경

2. Class 만들기



▶ 자바 프로그래밍 순서

✓ Eclipse 환경

3. Class 작성 순서

① 패키지(package) 선언 `package member.model.vo;`

② импорт(import) 선언 `import java.util.Date;`

③ 클래스(class) 작성부

```
public class Member {  
    private String name;  
    private int age;  
    private Date enrollDate;  
  
    public Member() {}  
  
    public Member(String name, int age, Date enrollDate) {  
        super();  
        this.name = name;  
        this.age = age;  
        this.enrollDate = enrollDate;  
    }  
  
    public String getName() {  
        return name;  
    }  
  
    public void setName(String name) {  
        this.name = name;  
    }  
  
    ... 이하 생략...  
}
```

멤버 변수

생성자 함수

멤버 함수

▶ 자바 프로그래밍 기본

✓ class

자바에서 모든 코드는 반드시 클래스 안에 존재해야 하며
서로 관련된 코드들을 그룹으로 나누어 별도의 클래스 구성
클래스들이 모여 하나의 Java 애플리케이션 구성

```
public class 클래스 이름 {
```

```
    /*  
    * 주석을 제외한 모든 코드는 블록 { } 내에 작성  
    */
```

```
}
```


▶ 자바 프로그래밍 기본

✓ main

'public static void main(String[] args)'는 main메서드의 고정된 선언부로
프로그램 실행 시 java.exe에 의해 호출됨
모든 클래스가 main메서드를 가지고 있어야 하는 것은 아니지만 하나의 Java애플리케이션에
는 main메서드를 포함한 클래스가 반드시 하나는 있어야 함

```
public class 클래스 이름 {
```

```
    public static void main(String[] args) { //메인 메서드의 선언부  
        // 실행될 문장들을 적는다  
    }
```

```
}
```

▶ 자바 프로그래밍 기본

✓ 주석(comment)

코드에 대한 설명이나 그 외 다른 정보를 넣을 때 사용하는 것으로
컴파일 시 컴파일러가 주석 부분은 건너 뛴다

`/* */` : 범위 주석, `/*`와 `*/` 사이 내용은 주석으로 간주

`//` : 한 줄 주석, `//` 뒤의 내용은 주석으로 간주