

프로그래밍 기초

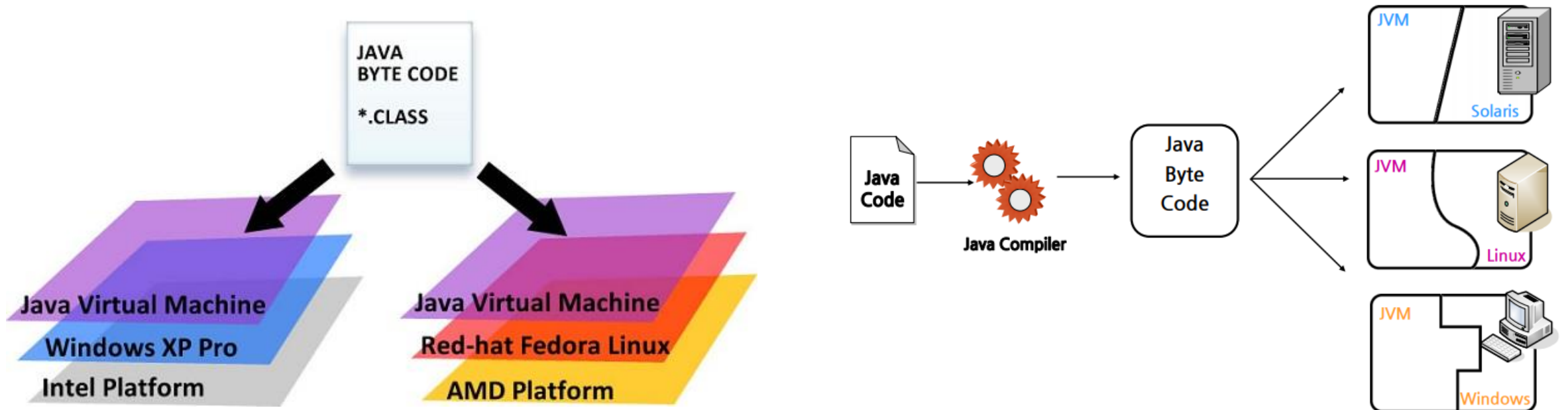
▶ 자바 언어 특징



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. 멀티쓰래드 지원

▶ JVM(Java Virtual Machine)

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



▶ 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

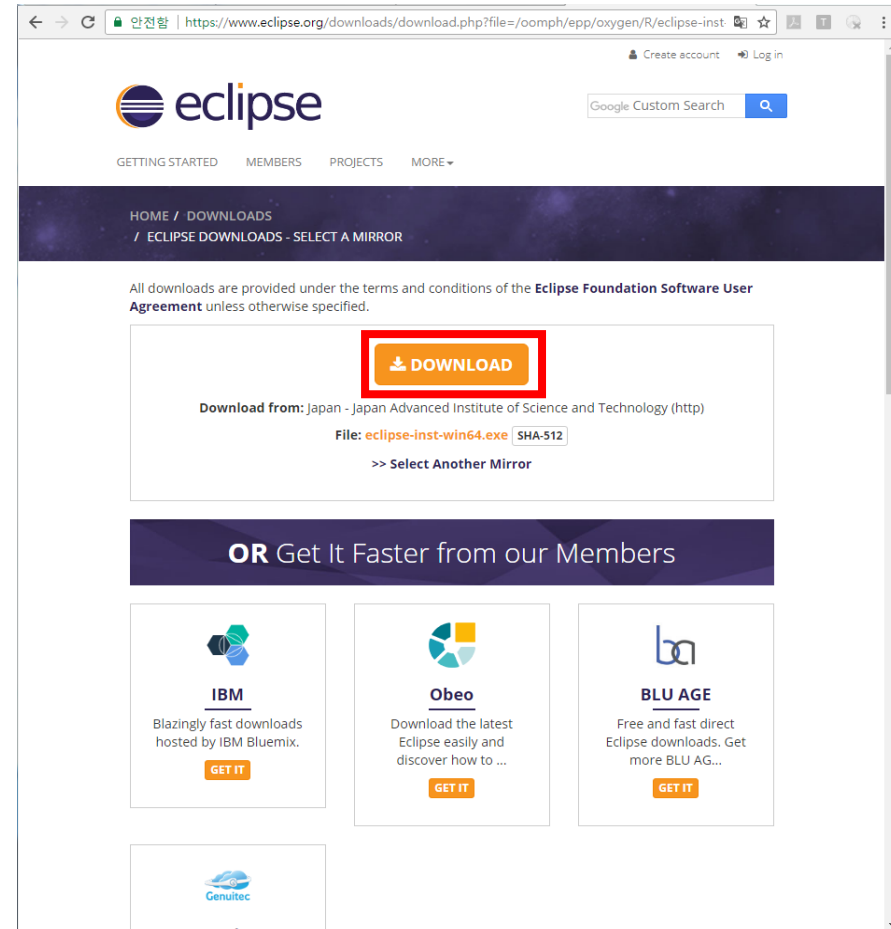
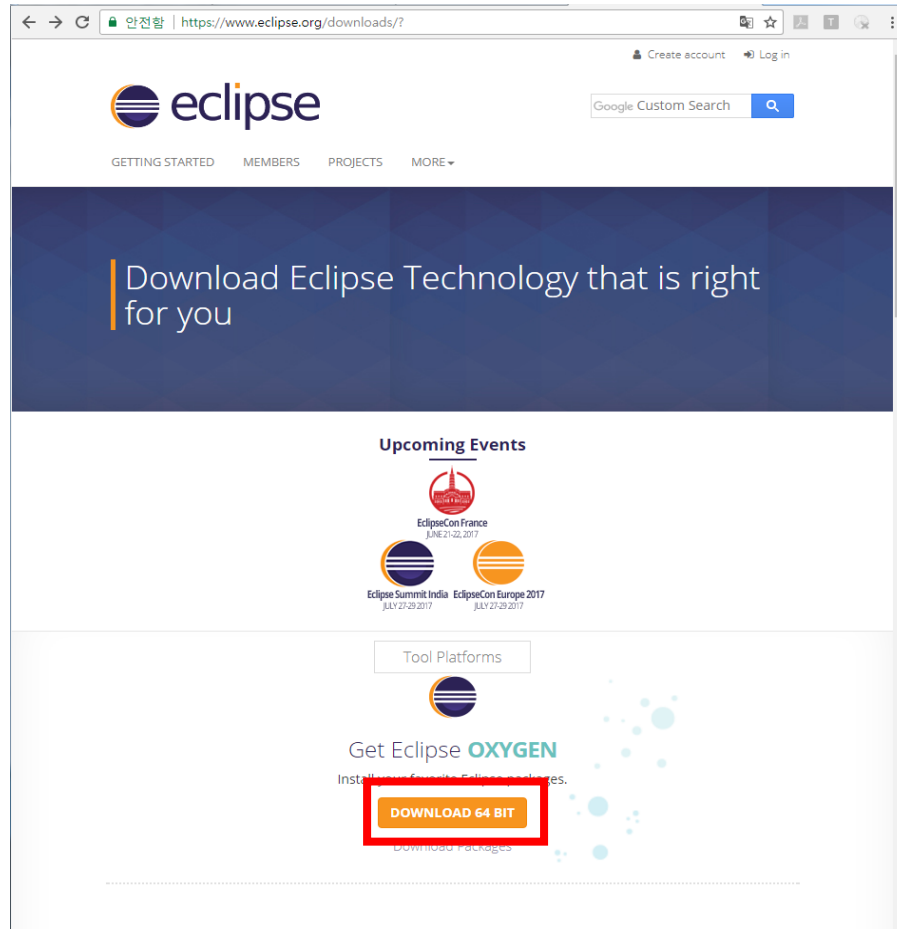
C:\workspace> javac HelloWorld.java

6. 실행하기

C:\workspace> java HelloWorld

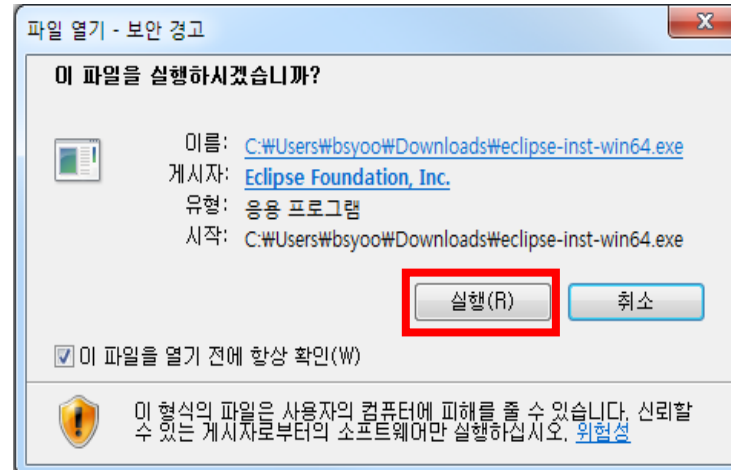
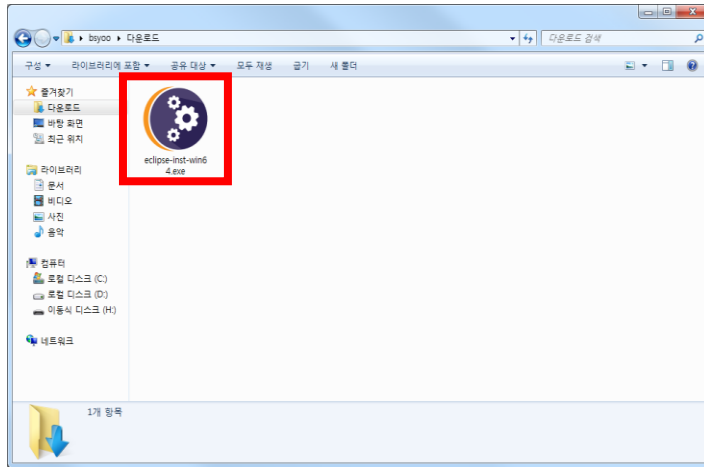
▶ 이클립스 설치

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



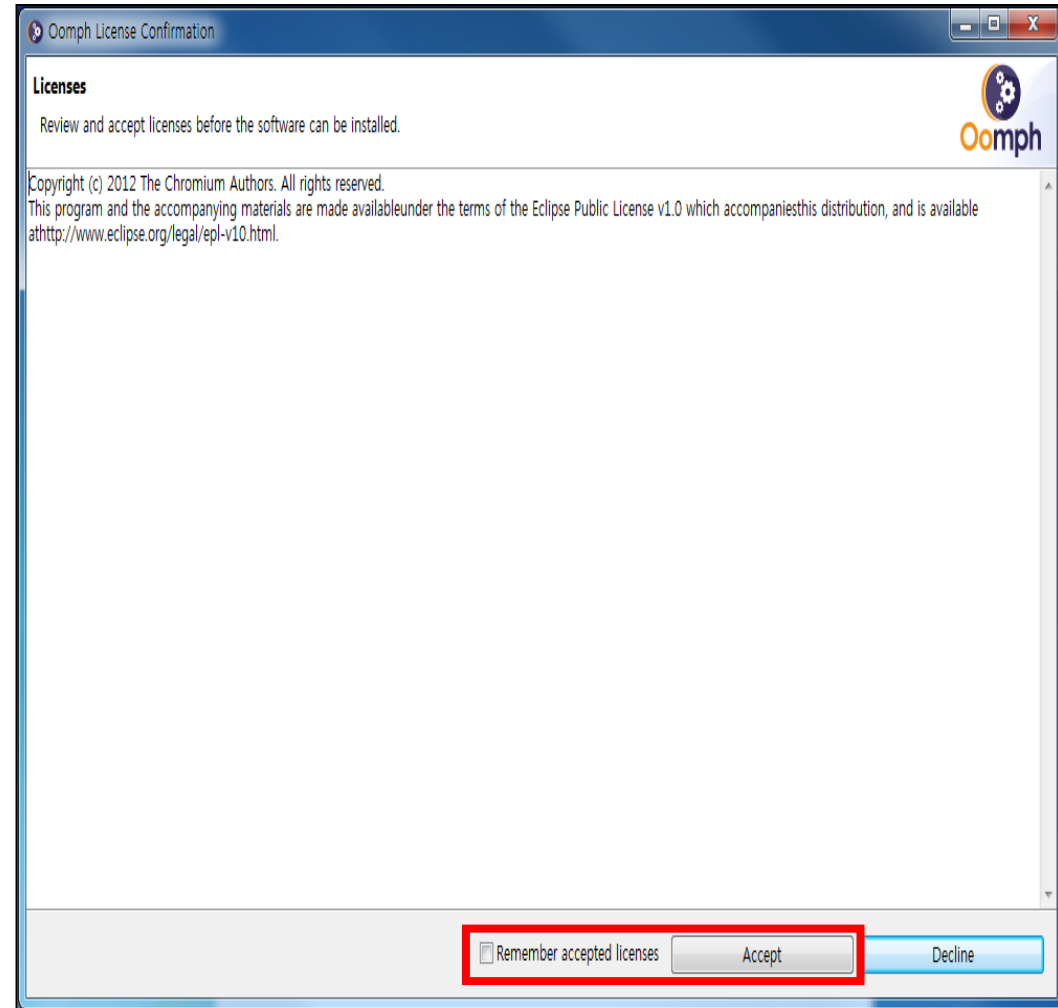
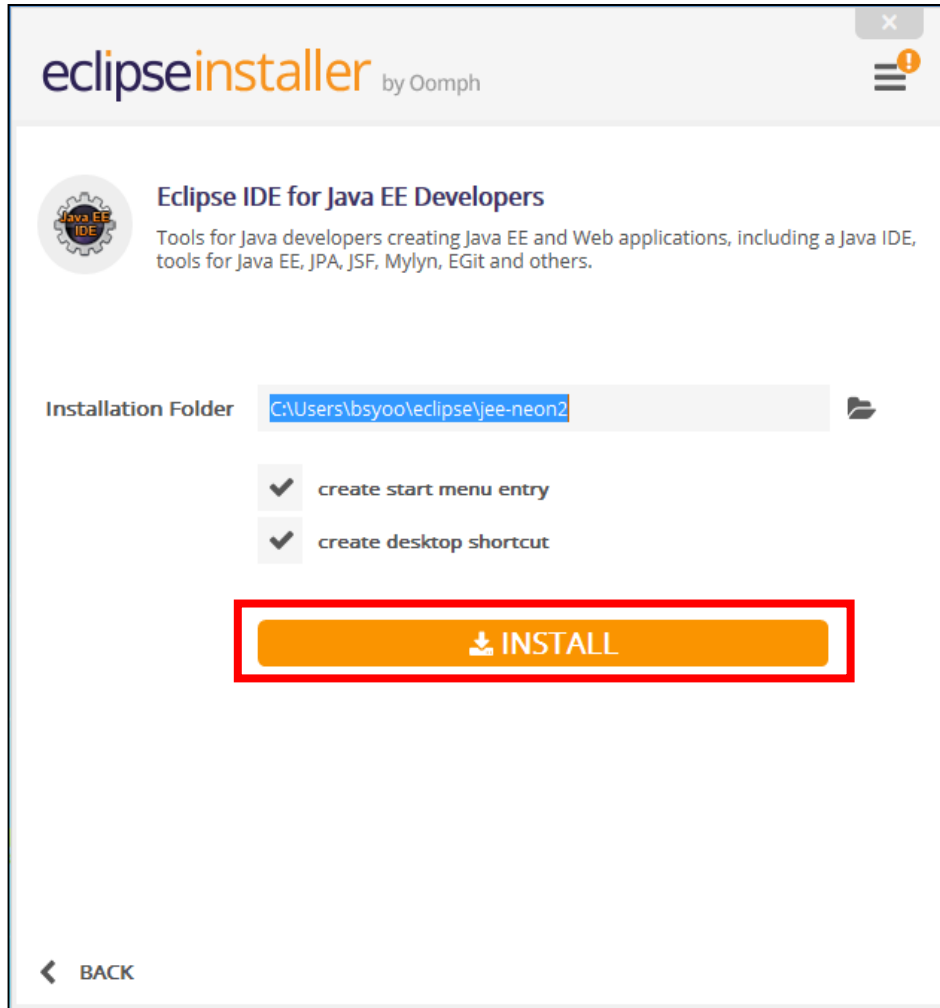
▶ 이클립스 설치

다운로드 폴더/다운로드 파일 확인



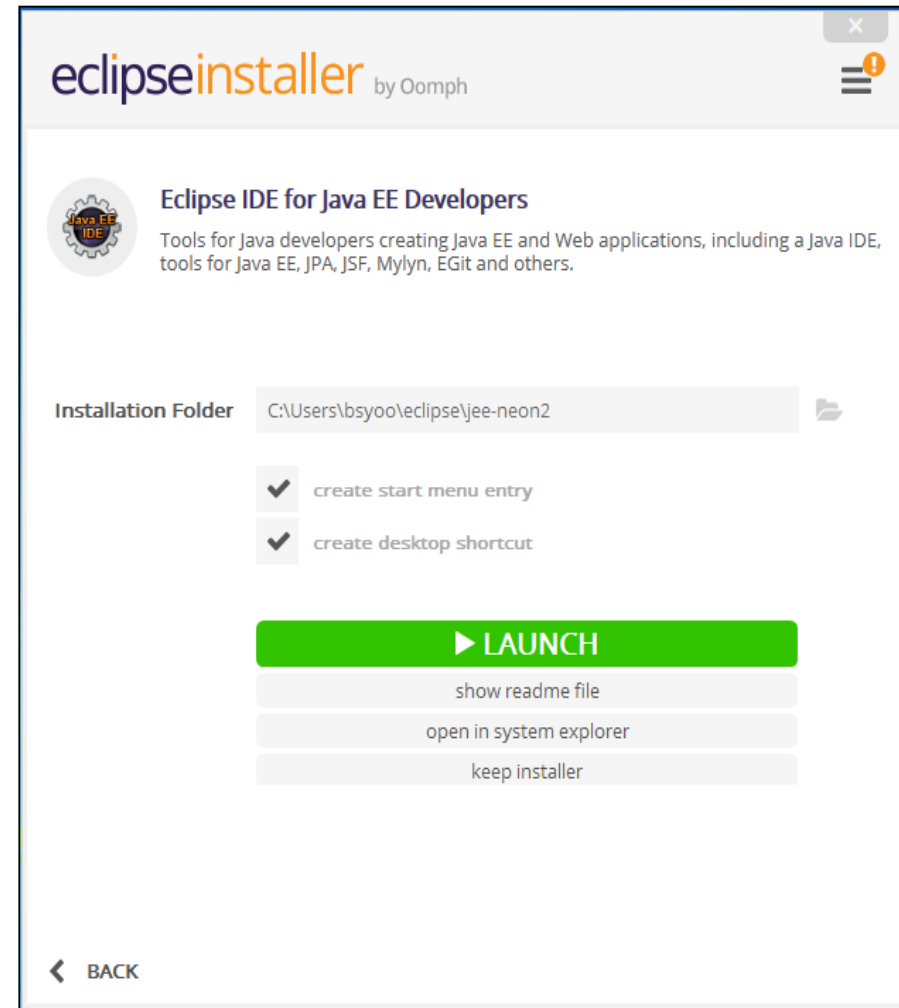
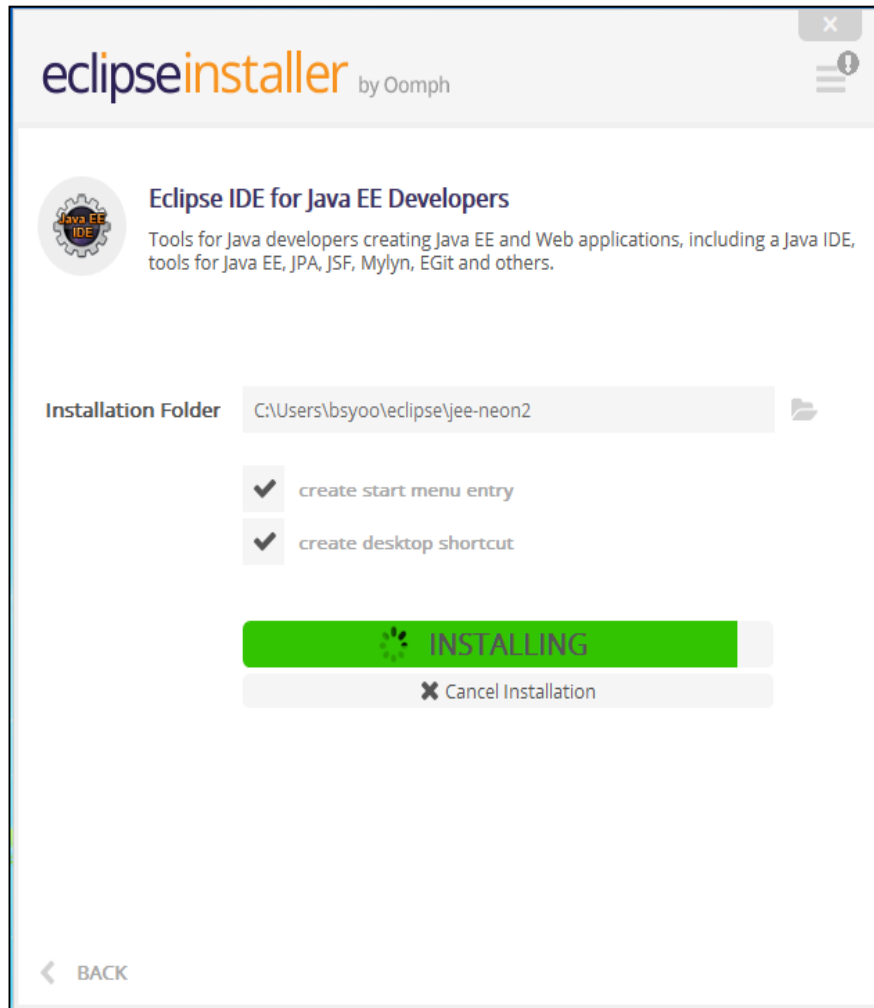
▶ 이클립스 설치

Java EE Developer 설치



▶ 이클립스 설치

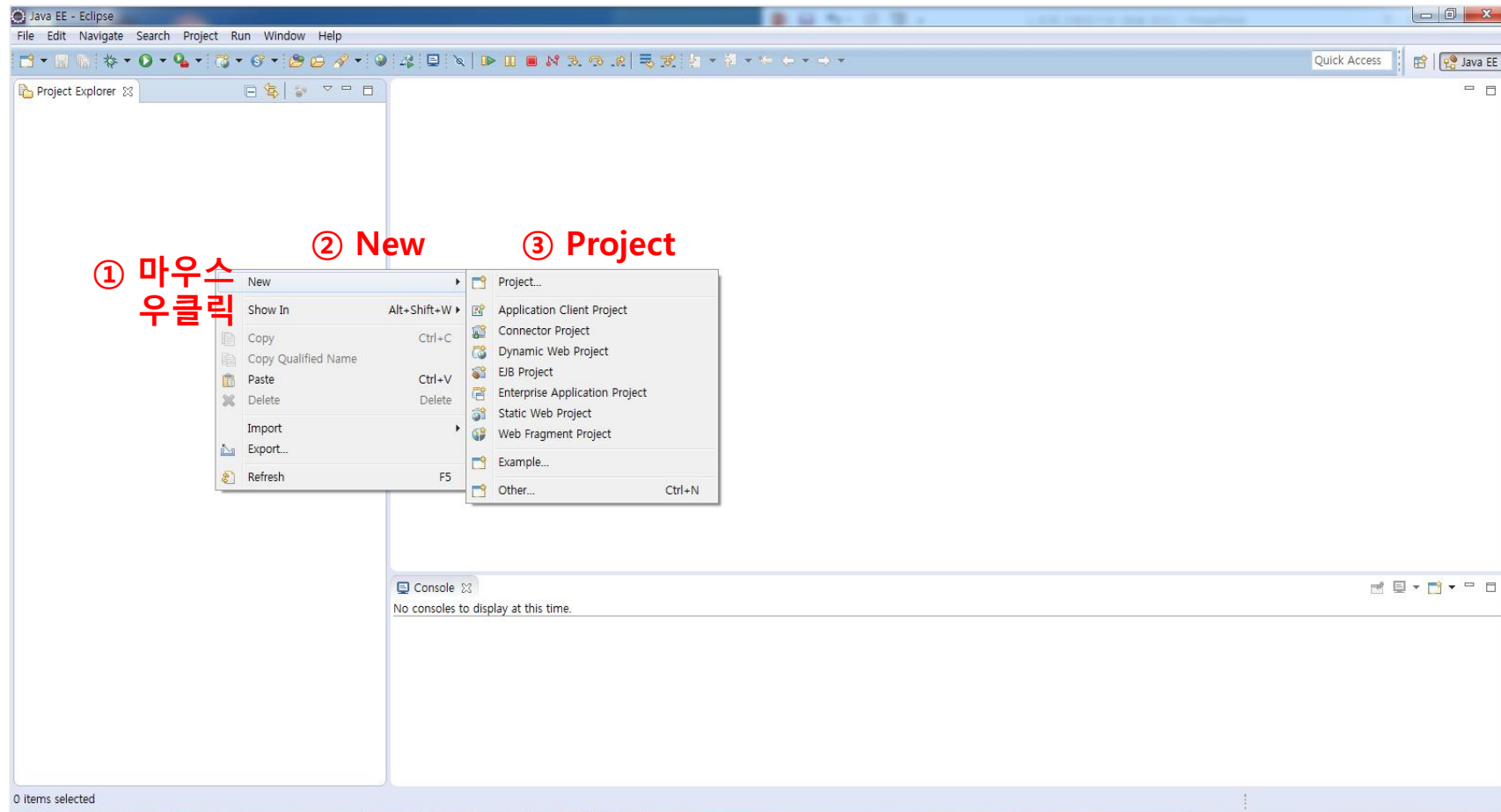
Java EE Developer 설치



▶ 자바 프로그래밍 순서

✓ Eclipse 환경

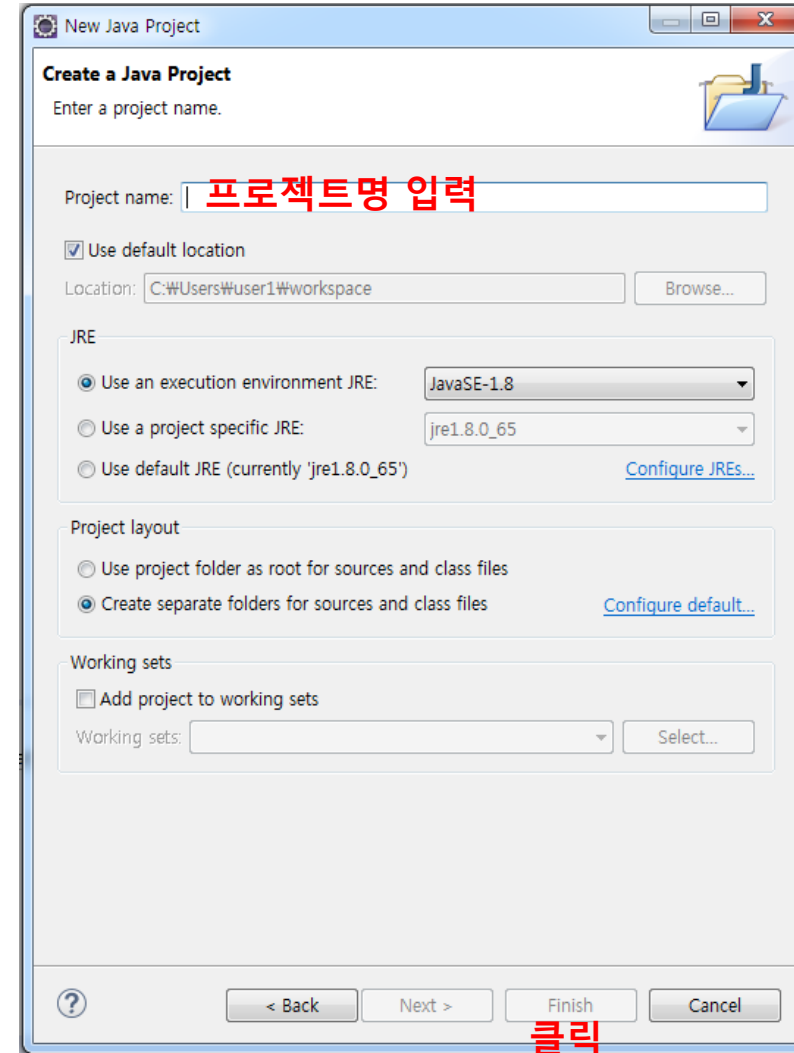
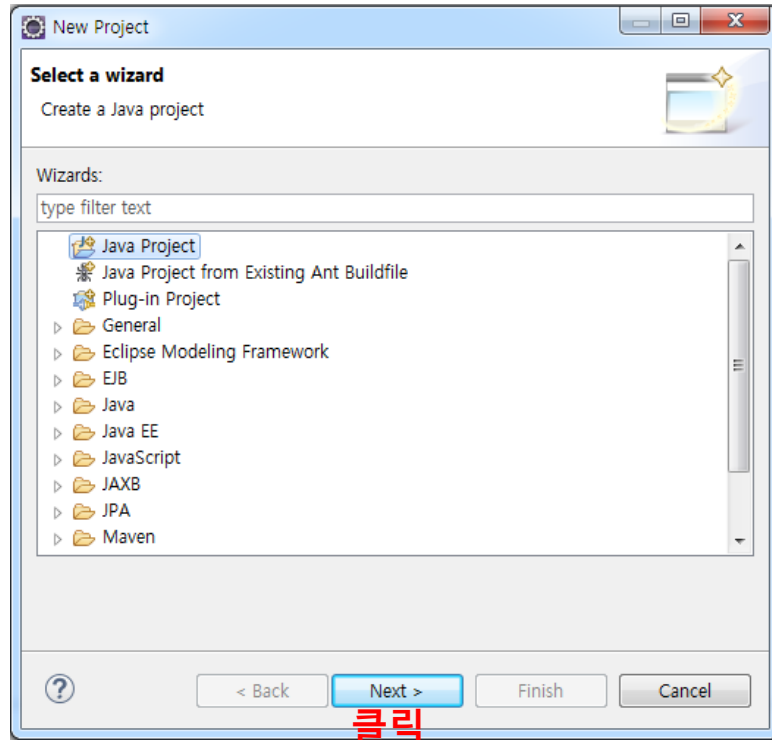
1. project 만들기



▶ 자바 프로그래밍 순서

✓ Eclipse 환경

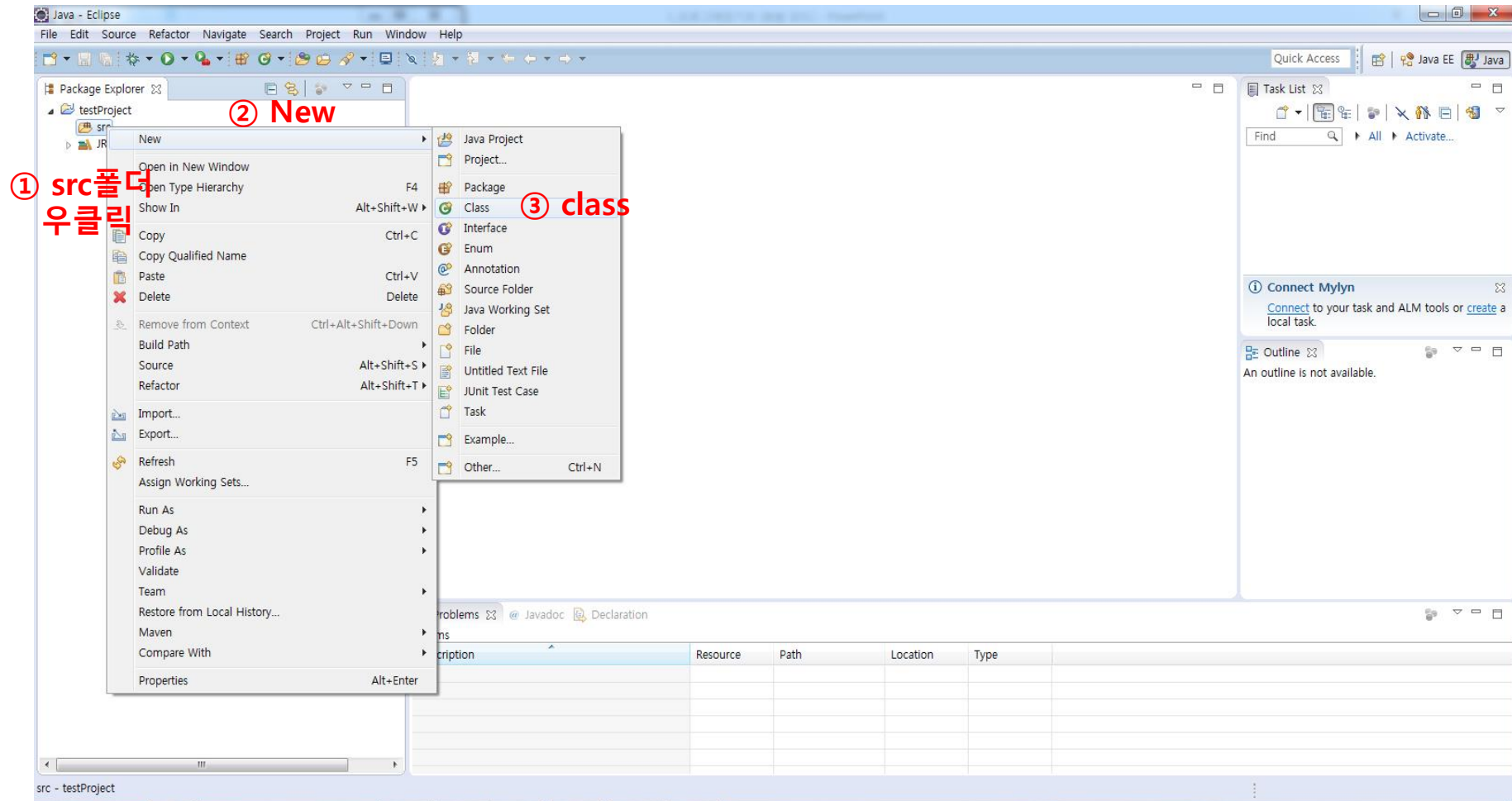
1. project 만들기



▶ 자바 프로그래밍 순서

✓ Eclipse 환경

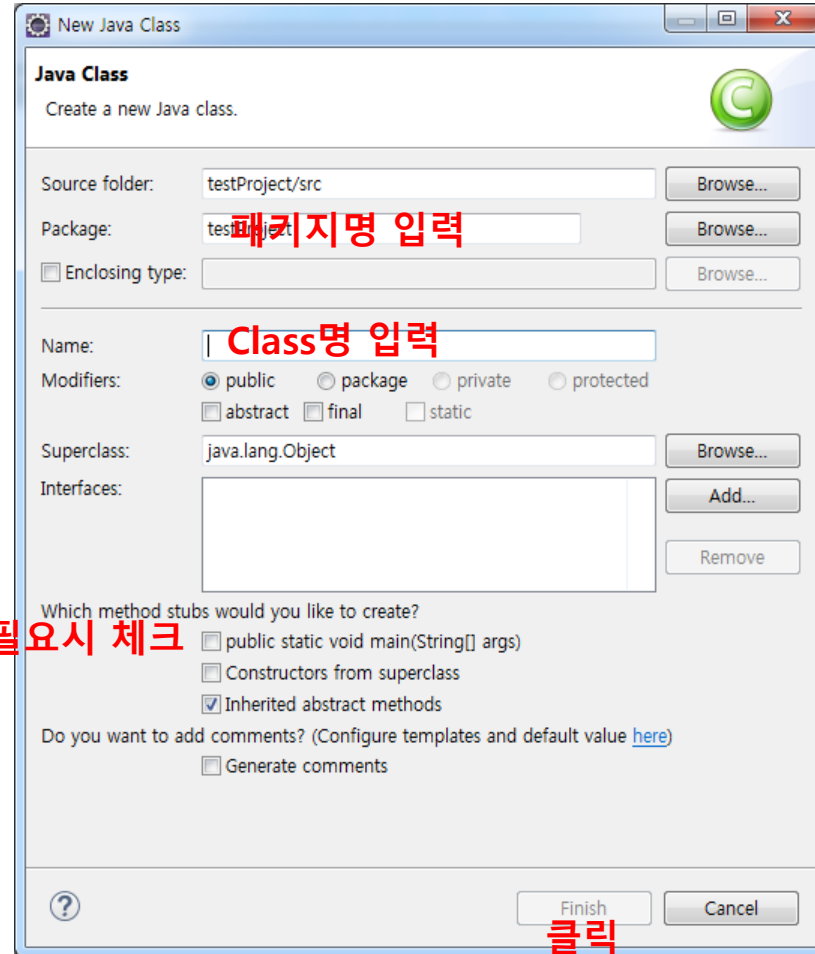
2. Class 만들기



▶ 자바 프로그래밍 순서

✓ Eclipse 환경

2. Class 만들기



▶ 자바 프로그래밍 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 Java애플리케이션에는 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