

# 프로그래밍 기초

## ▶ 프로그래밍

### ✓ 프로그램(Program)

컴퓨터가 인식할 수 있는 명령어의 나열(집합)

### ✓ 프로그래밍(Programming)

프로그램을 작성하는 과정 = 코딩

### ✓ 프로그래머(Programmer)

프로그램을 작성하는 사람

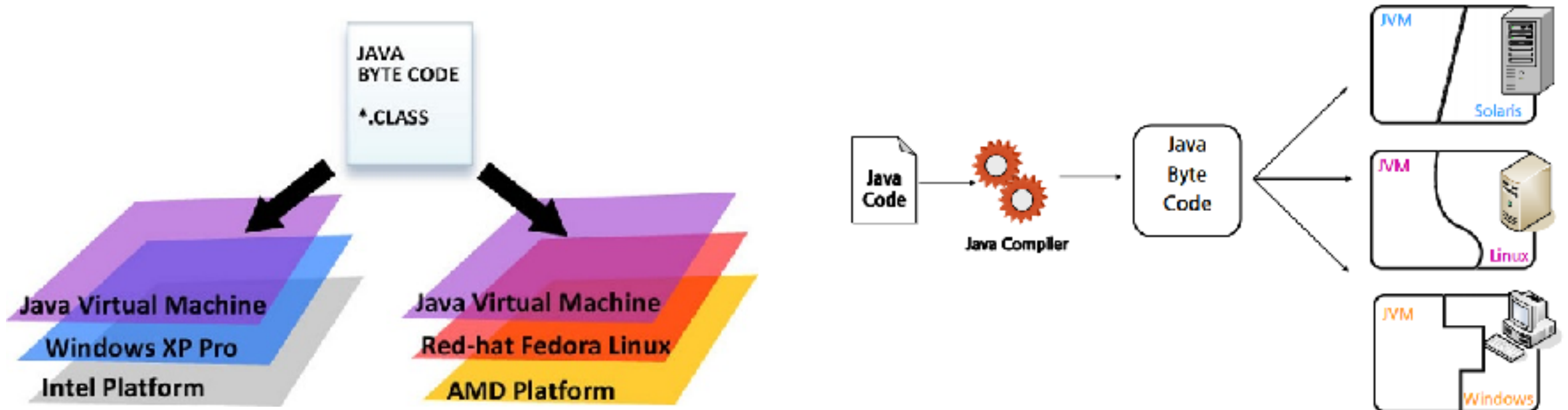
## ▶ 자바 언어 특징



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)

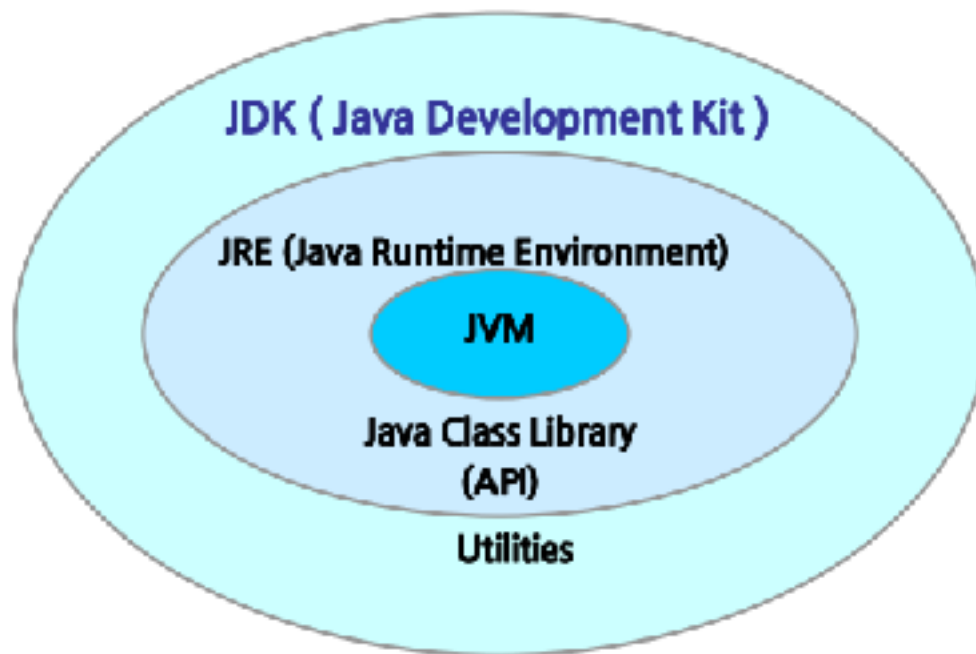
자바를 실행하기 위한 가상 기계로 플랫폼에 의존적  
byte code(class파일)를 해석하고 실행하는 interpreter



# ▶ 자바 개발 환경

## ✓ 설치 범위

사용자/개발자 입장에 따라 설치하는 범위가 달라짐



Java SE : Java Standard Edition

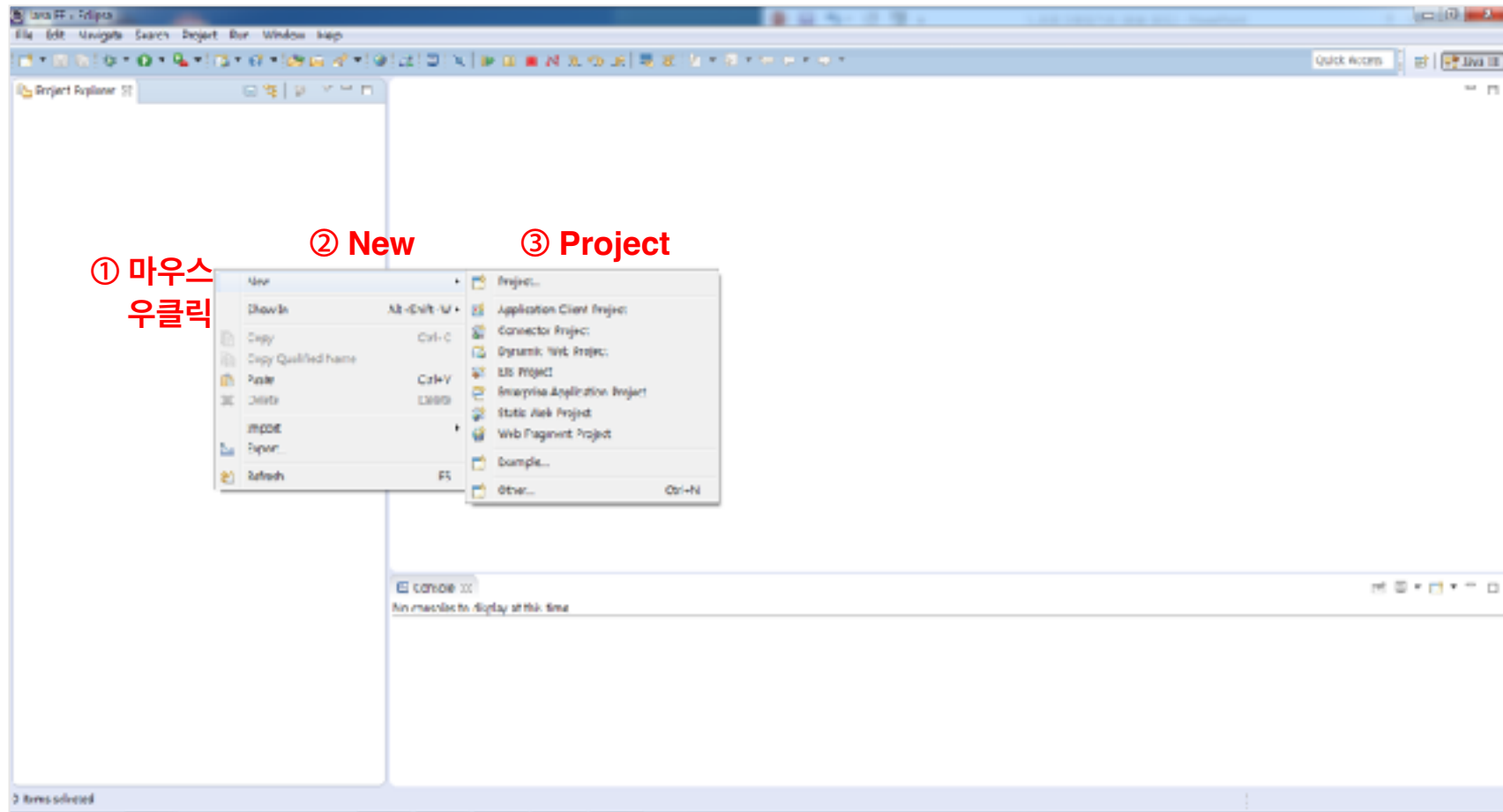
Java EE : Java Enterprise Edition

Java ME : Java Micro Edition

# ▶ 자바 프로그래밍 순서

## ✓ Eclipse 환경

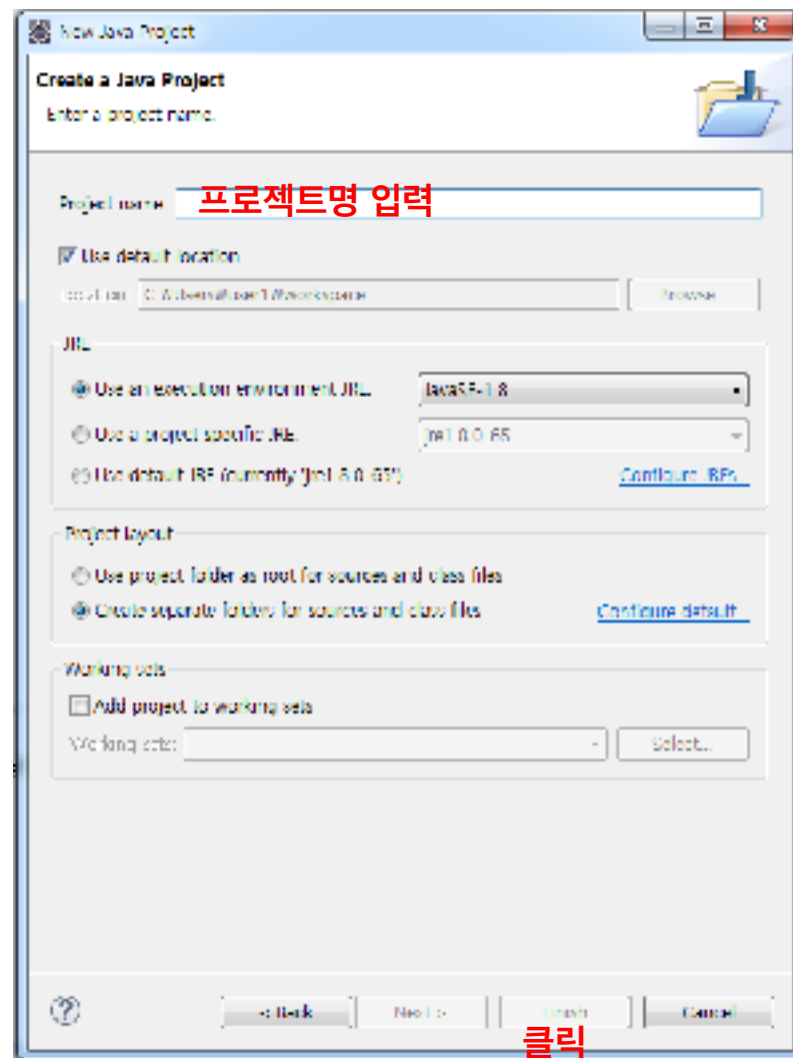
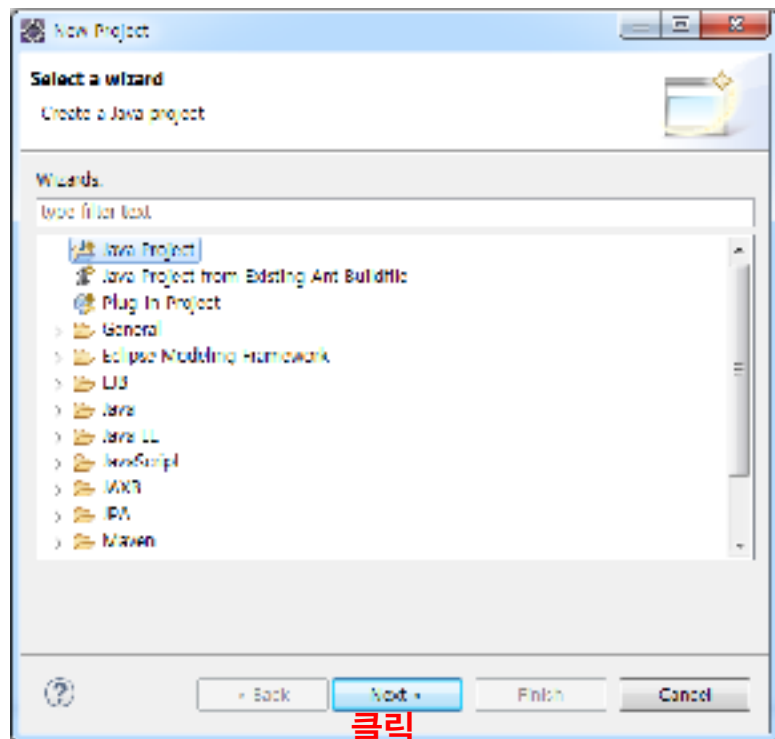
### 1. project 만들기



# ▶ 자바 프로그래밍 순서

## ✓ Eclipse 환경

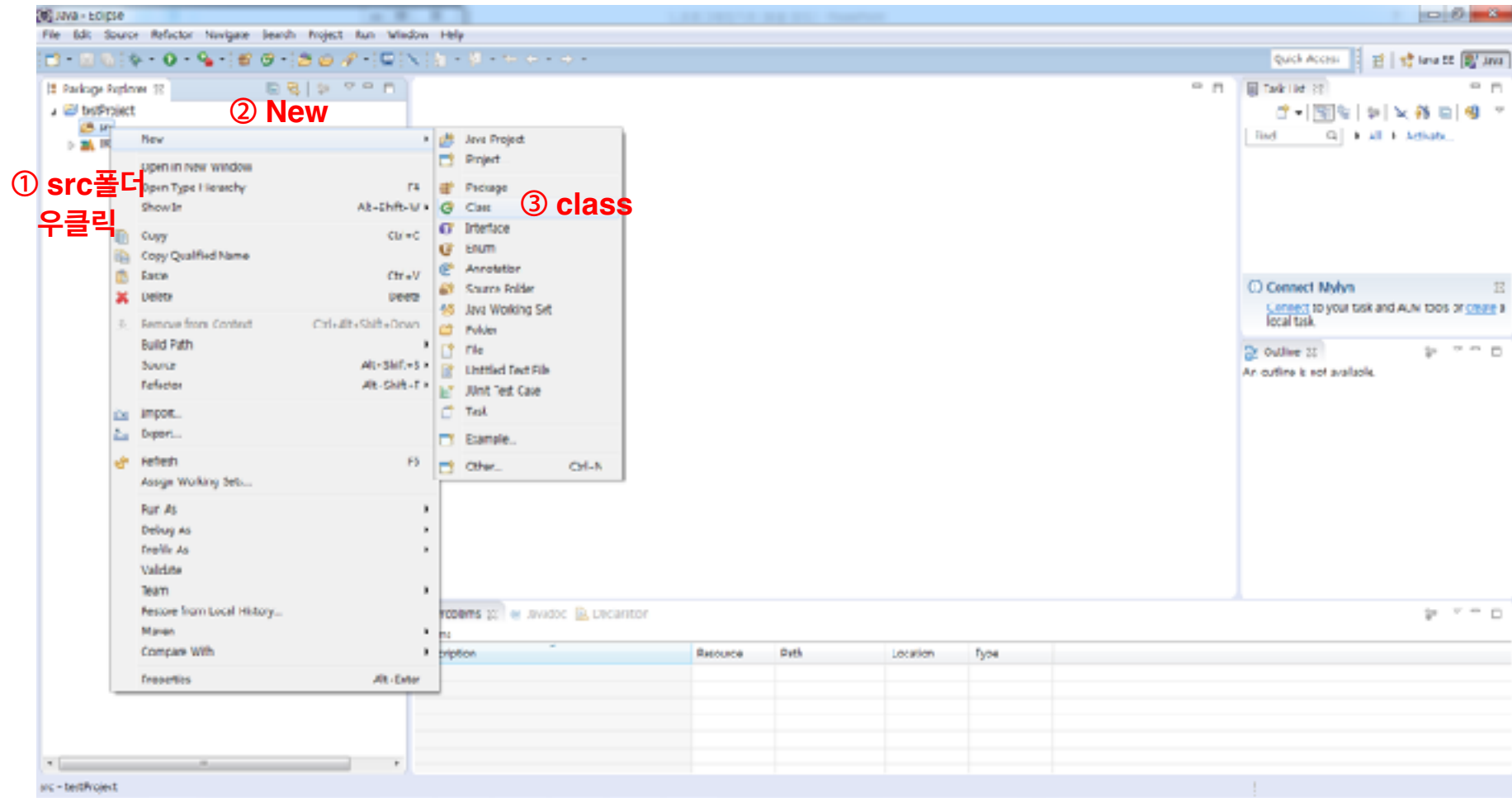
### 1. project 만들기



## ▶ 자바 프로그래밍 순서

## ✓ Eclipse 환경

## 2. Class 만들기

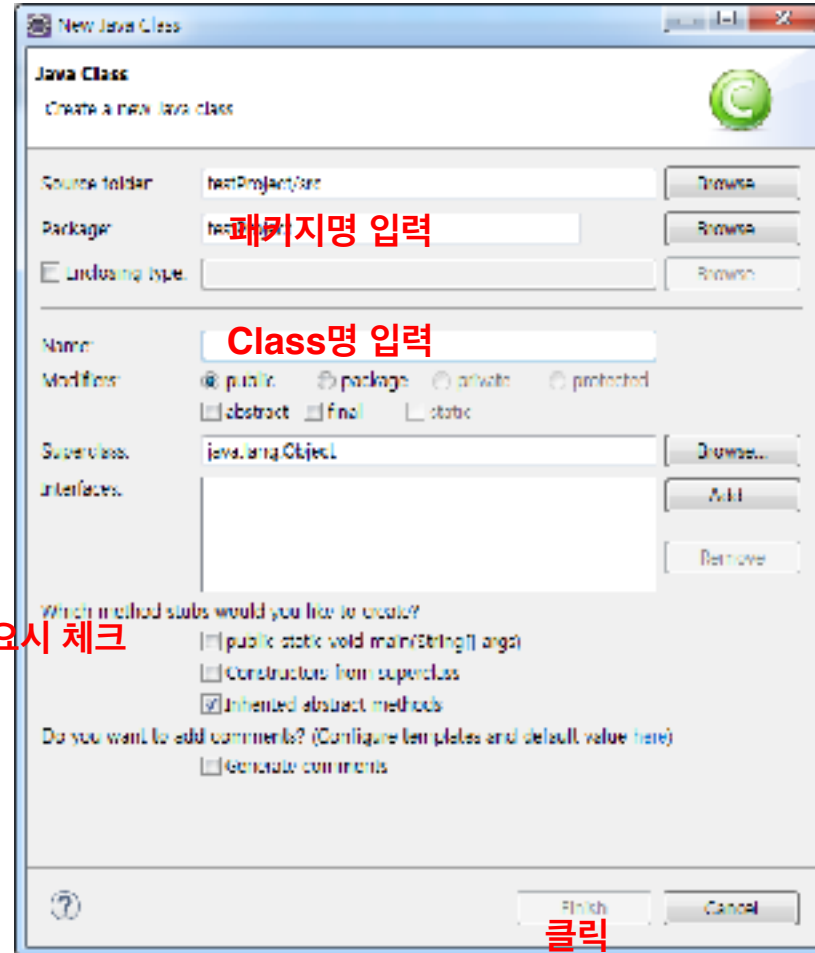




# ▶ 자바 프로그래밍 순서

## ✓ Eclipse 환경

### 2. Class 만들기



# ▶ 자바 프로그래밍 순서

## ✓ Eclipse 환경

### 3. Class 작성 순서

① 패키지(package) 선언

```
package member.model.vo;
```

② 임포트(import) 선언

```
import java.util.Date;
```

③ 클래스(class) 작성부

```
public class Member {
```

```
    private String name;
```

```
    private int age;
```

```
    private Date enrollDate;
```

멤버 변수

```
    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...
```

```
}
```

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 Java애플리케이션에는 main메서드를 포함한 클래스가 반드시 하나는 있어야 함

```
public class 클래스 이름 {
```

```
    public static void main(String[] args) { //메인 메서드의 선언  
        // 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