

# 자바프로그래밍

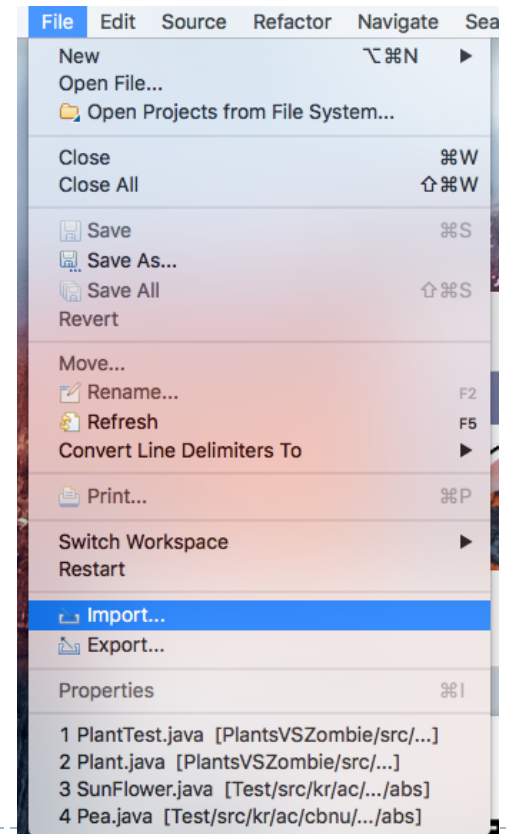
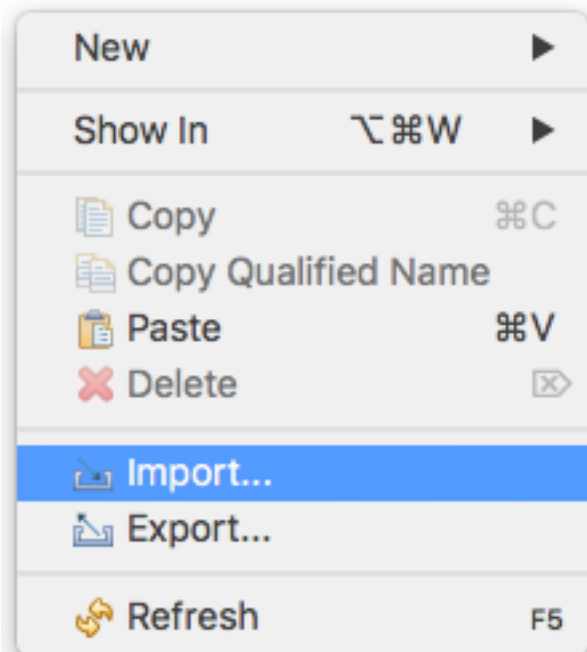
**OOM 1**



준비하기

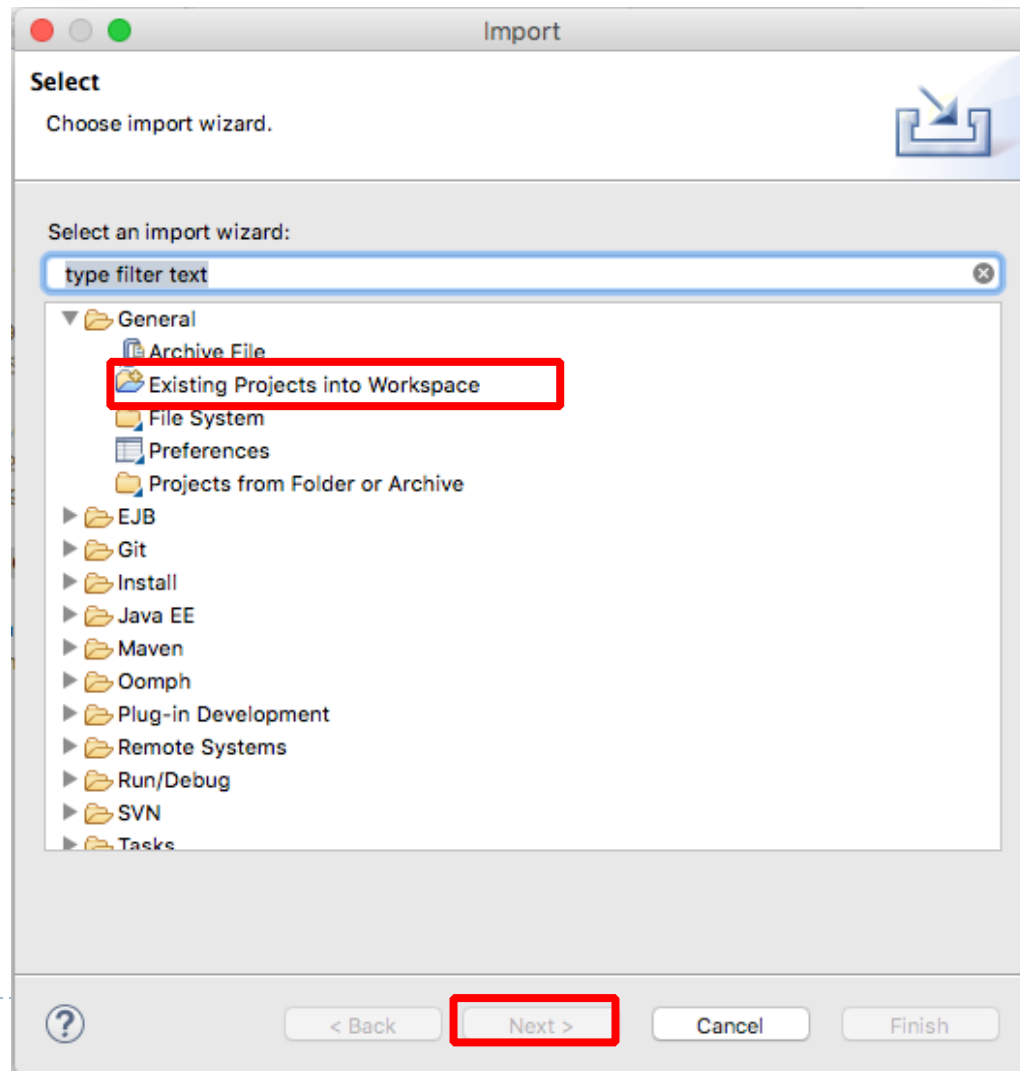
# 이클립스에서 프로젝트 불러오기

- ▶ Open 또는 Load의 개념을 “Import”로
- ▶ Package Explorer 에서 우클릭>import 또는 File>Import

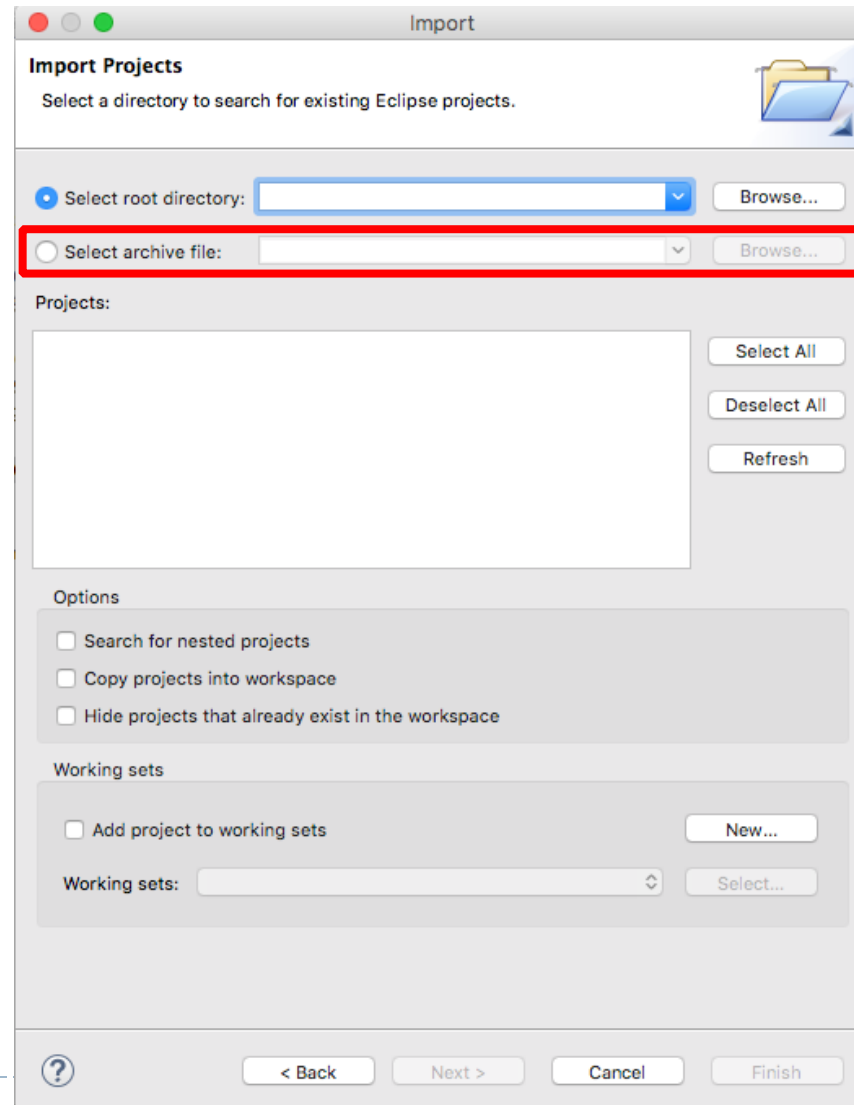


# 이클립스에서 프로젝트 불러오기

## ▶ General>Exisiting Projects into Workspace

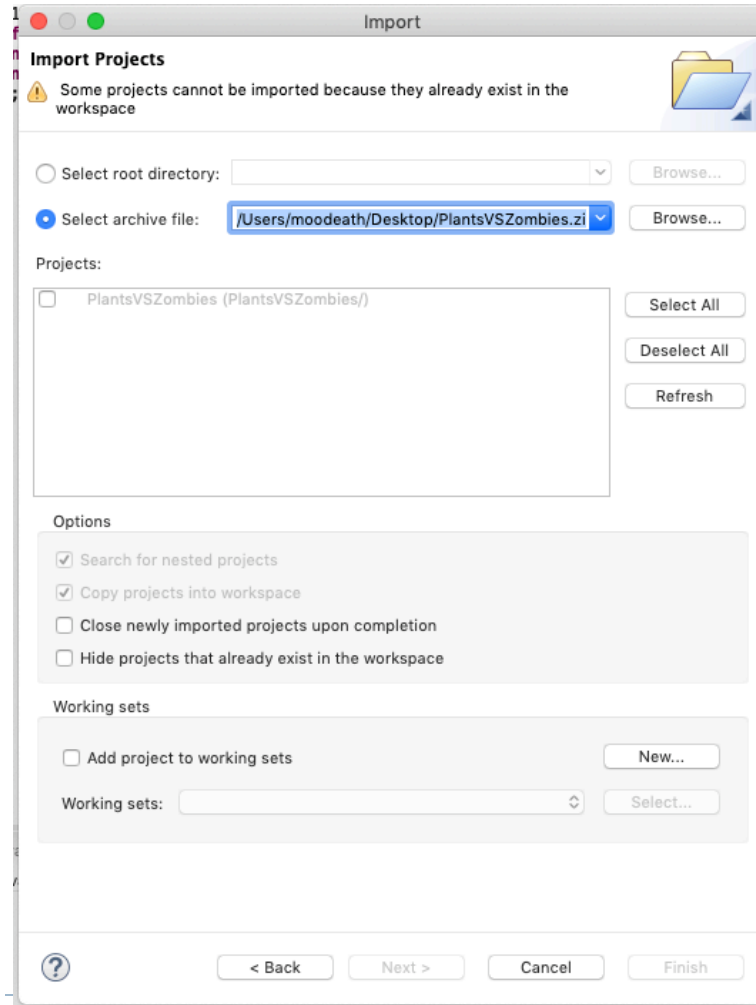


# 이클립스에서 프로젝트 불러오기



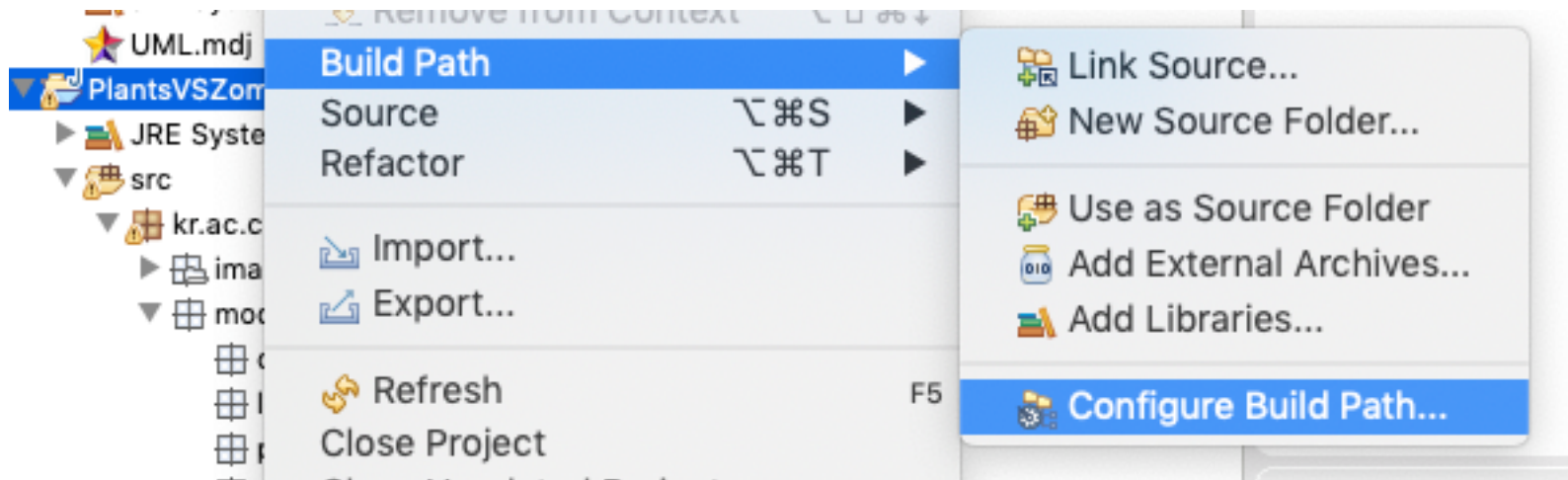
# 이클립스에서 프로젝트 불러오기

## ▶ 다운로드 된 PlantsVSZombies.zip 선택



# 이클립스에서 프로젝트 불러오기

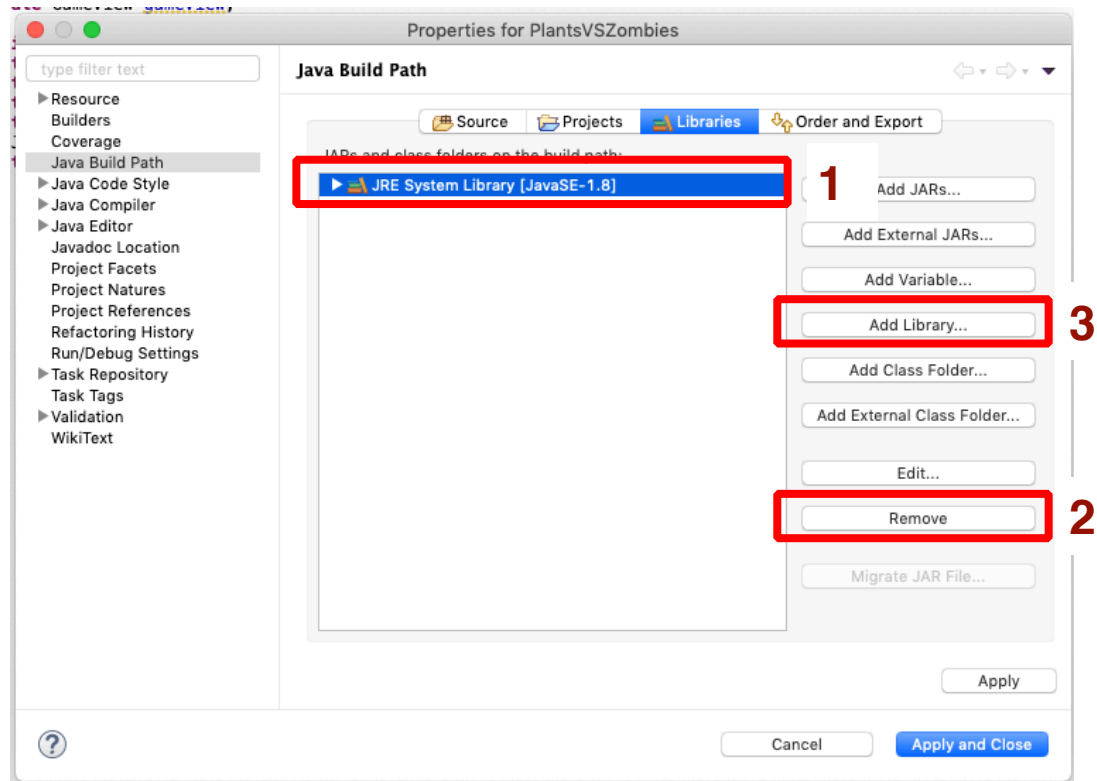
- ▶ 프로젝트 에러 발생시
  - ▶ 프로젝트 우클릭 > Build Path > Configure Build Path...



# 이클립스에서 프로젝트 불러오기

## ▶ 프로젝트 에러 발생시

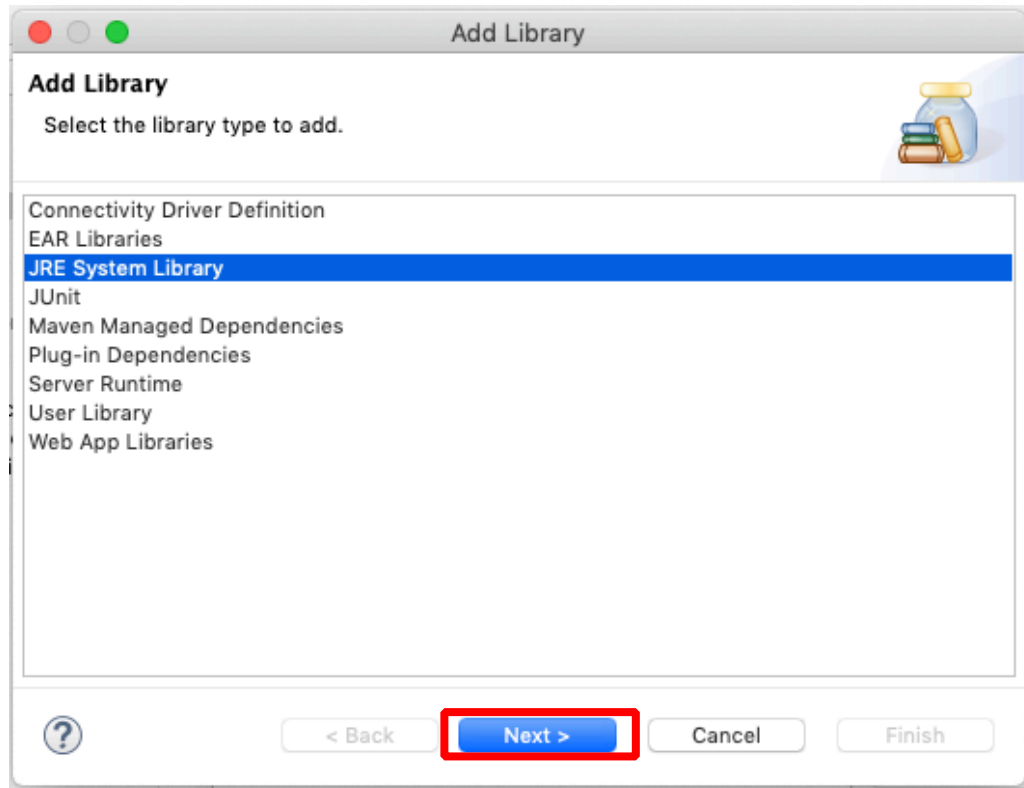
### ▶ 1 선택 > 2 remove > 3 Add Library





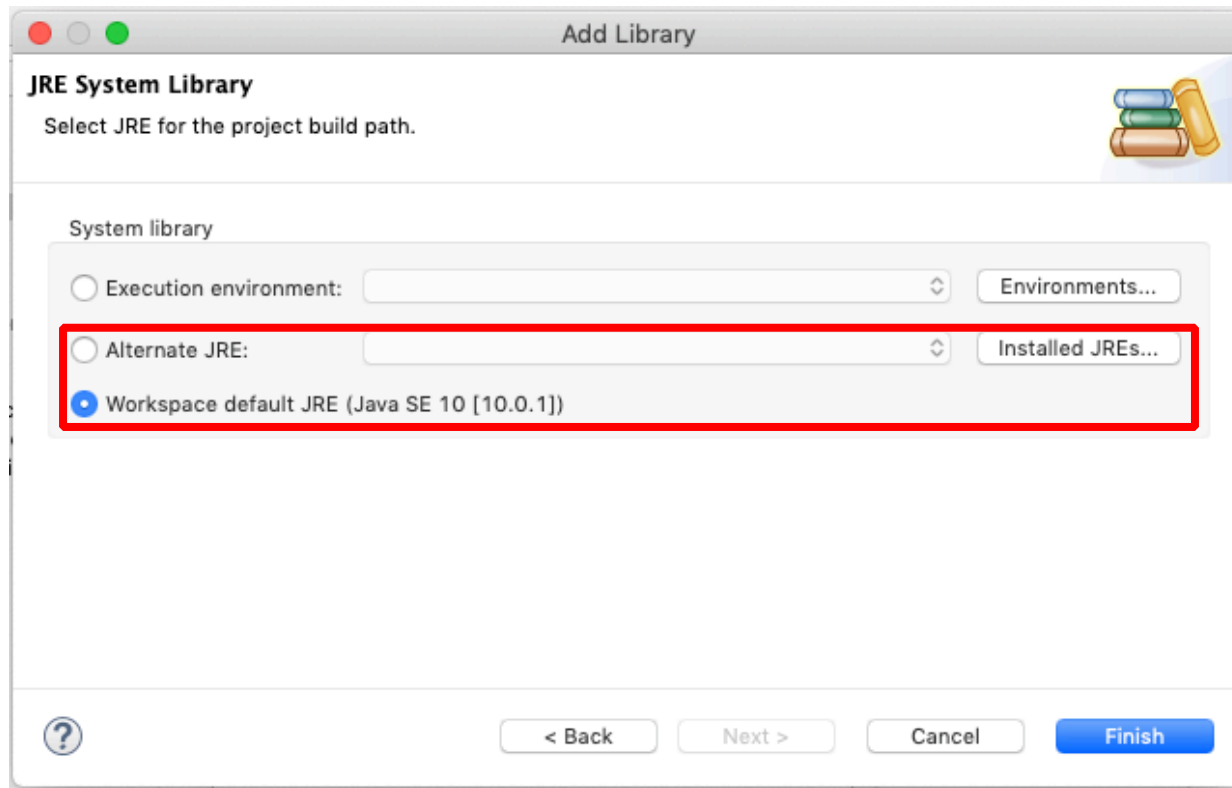
# 이클립스에서 프로젝트 불러오기

## ▶ 프로젝트 에러 발생시



# 이클립스에서 프로젝트 불러오기

- ▶ 프로젝트 에러 발생시
- ▶ 자신의 컴퓨터에 설치된 JRE 선택



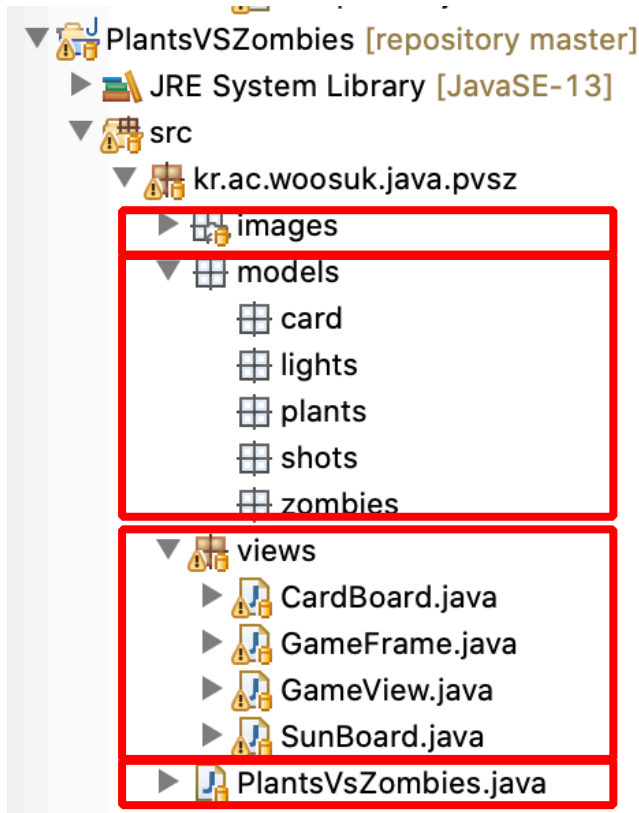
# 이클립스에서 프로젝트 불러오기

- ▶ 프로젝트 에러 발생시
  - ▶ 프로젝트 우클릭 > Run As > Java Application



# 이클립스에서 프로젝트 불러오기

## ▶ PlantsVSZombies 프로젝트 구조



이미지 파일

모델링한 클래스를 작성하는 곳

배경 화면 및  
카드 선택 영역, 햇빛 숫자 영역

main method 가 있는 클래스.  
실행만 시킴

# **OOM(Object Oriented Modeling)**

with Plants VS Zombies

# OOM with Plants VS Zombies

---

- ▶ 필요한 사고력 : 추상화
  - ▶ 필요한 부분만을 표현할 수 있고 불필요한 부분을 제거하여 간결하고 이해하기 쉽게 만드는 작업.
  - ▶ 예) 동그라미 모양, 빨간색, 식용가능, 이름은 사과



# OOM with Plants VS Zombies

---

## ▶ 객체 (Object)

### ▶ 철학

#### ▶ 일반적 대상

▶ 주체와 연관됨으로써 이 주체의 행위가 지향하는 것

▶ 경험을 통해 의식에 주어진 대상 또는 인식 주체와의 관계에서 본 실재(實在)

### ▶ 객체 지향 설계 및 객체 지향 프로그래밍

▶ 데이터(실체)와 그 데이터에 관련되는 동작(절차, 방법, 기능)을 모두 포함한 개념

▶ 공간을 차지하고 있는 물질적 사물(entity). 경계(boundary)와 의미가 확실해야 함

## ▶ 실체

▶ 상황에 따라 여러 가지로 변화할 수 있는 성질, 상황, 적용, 관계 등을 토대로 그것들을 받들고 있는 기본 존재

▶ 목적을 가지고 존재하는 것

---



# OOM with Plants VS Zombies

---

## ▶ 실체의 종류

- ▶ 정체성의 유무로 독립적 실체와 의존적 실체를 구분할 수 있다.
- ▶ **정체성**을 갖는 독립적 실체
- ▶ **정체성이 없는** 의존적 실체
- ▶ 빨간색 vs 토마토, 사과, 딸기





# OOM with Plants VS Zombies

---

## ▶ 객체와 속성

- ▶ 정체성의 유무로 독립적 실체와 의존적 실체를 구분할 수 있다.
- ▶ 객체(Object)는 **행위**와 **정체성**을 갖는 독립적 실체, 속성(attribute)은 **정체성이 없는** 의존적 실체



# OOM with Plants VS Zombies

---

## ▶ 객체

- ▶ 실체(데이터)와 동작(절차, 방법, 기능)을 모두 포함하는 것
  - ▶ 사람, 동물, 식물, 자동차, 비행기, 건물, 컴퓨터 등등
- ▶ 사람은 객체를 기준으로 사고함
- ▶ 실세계의 객체는 눈에 보이는 것과 추상적인 개념으로 나눌 수 있음
  - ▶ 추상적인 개념은 그 개념을 익히고 설계해야 함 -> 설계의 어려움을 가져옴
  - ▶ 무생물은 스스로 동작하지 않음 -> 설계의 어려움을 가져옴
- ▶ 모든 객체는 행위(behavior/method/function)를 갖는다.
- ▶ 객체는 속성(attribute)[값(value)/특성(property)]을 갖을수도 그렇지 않을수도 있다.



# OOM with Plants VS Zombies

---

- ▶ 객체와 속성의 구분과 범위는 \*도메인에 따라 달라짐

- ▶ 도메인

- ▶ 집합 (수학)

- ▶ 인터넷상의 컴퓨터 주소를 알기 쉬운 영문으로 표현한 것.

(네트워크 영역)

- ▶ 개발하고자 하는 범위와 그에 따른 이해관계자 또는 조직의 역할 등을 정하는 것

(Software 영역)

- ▶ 인식의 내용

- ▶ 우리가 관심을 가져야 하는 것은 개발하고자 하는 범위

- ▶ 보통 요구사항 문서에 모두 정의되어 있음.



# OOM with Plants VS Zombies

---

- ▶ 예제 1)

- ▶ 도메인이 실생활일 때, 강아지의 객체 모델링



# OOM with Plants VS Zombies

---

- ▶ 예제 2)

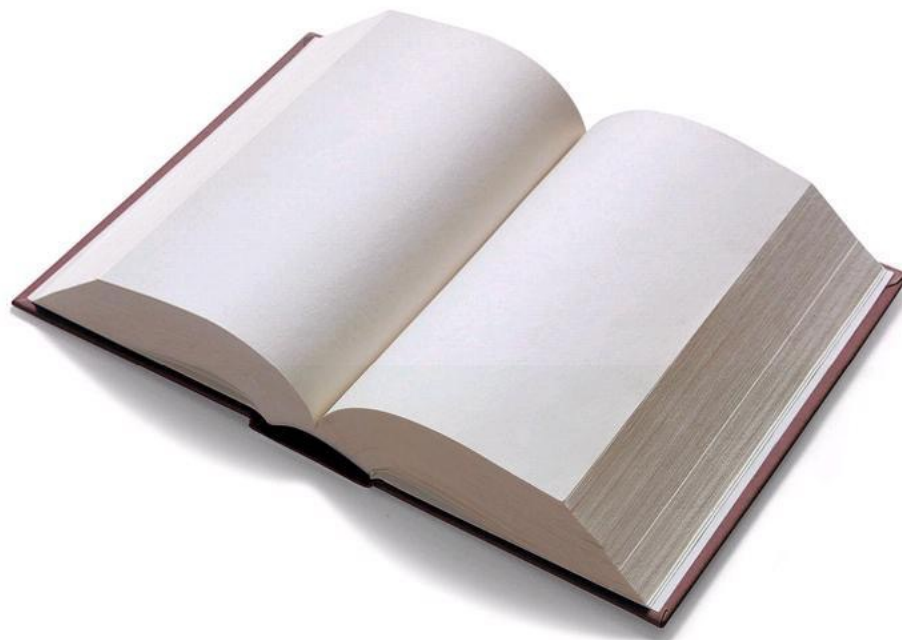
- ▶ 도메인이 실생활일 때, 나무의 객체 모델링



# OOM with Plants VS Zombies

---

- ▶ 예제 3)
  - ▶ 도메인이 실생활일 때, 책의 객체 모델링

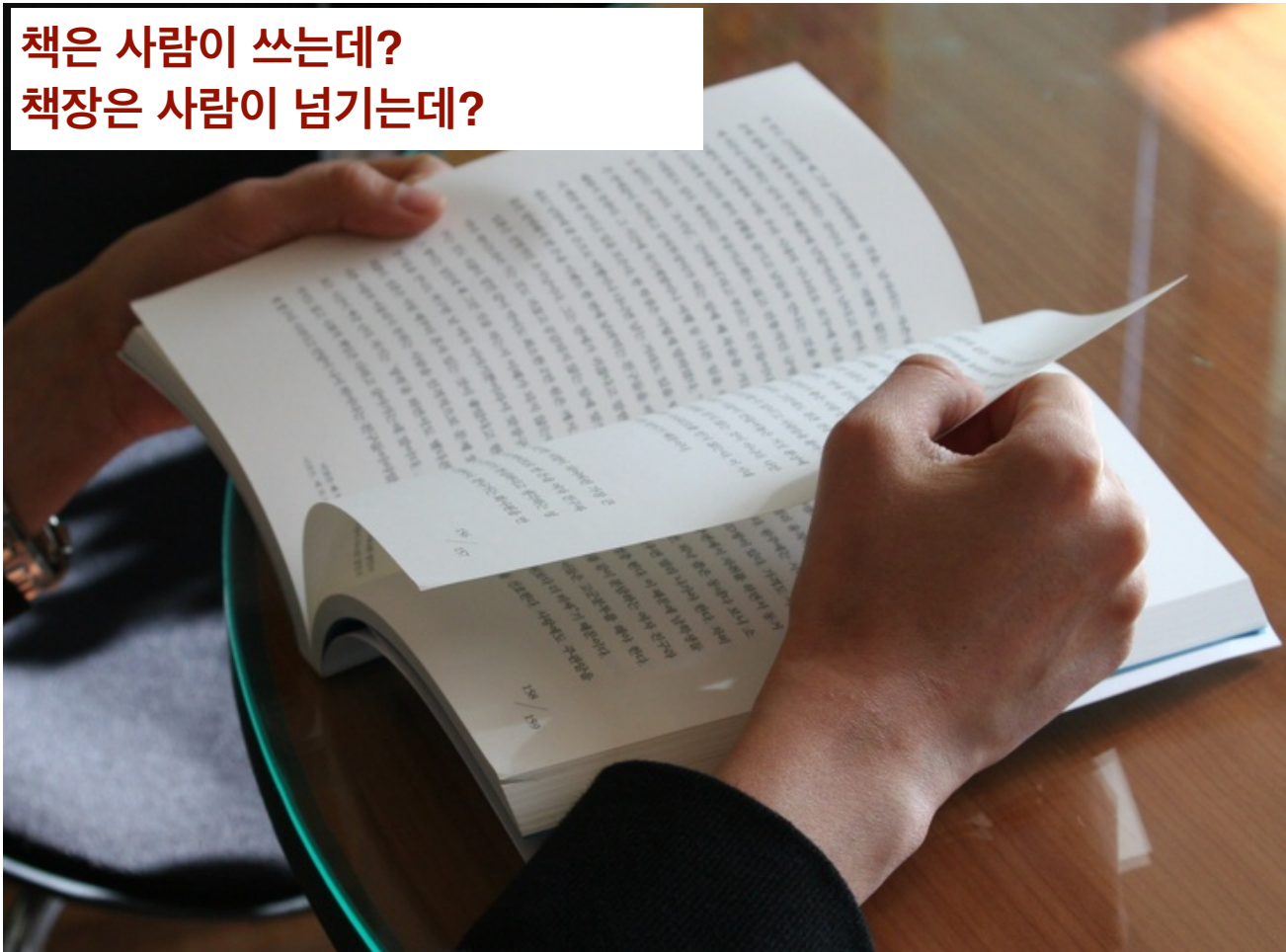


# OOM with Plants VS Zombies

---

## ▶ 인간의 관점

책은 사람이 쓰는데?  
책장은 사람이 넘기는데?



# OOM with Plants VS Zombies

---

- ▶ 객체 세상에서의 책



- ▶ `writtenByAuthor(String contents, Human author);`
- ▶ `String nextPage();`





# OOM with Plants VS Zombies

---

- ▶ 행위를 어떤 객체에 넣을 것인가를 결정짓는 기준
  - ▶ 책이 없다면 인간이 책장을 넘길 수 있는가?
  - ▶ 객체의 세상에서 행위는 객체에 종속적일 수도 있고, 그 자체가 객체로 될 수 있다.
  - ▶ 만약 어떤 객체(A)가 객체(B)가 없을 때 어떤 행위를 할 수 없다면, 그 행위는 객체(B)에 종속적이다.



# OOM with Plants VS Zombies

---

- ▶ 클래스 다이어그램
  - ▶ 사각형으로 작성
  - ▶ 이름 부분, 속성 부분, 행위 부분을 갖고, 각 부분은 구분 선으로 나뉨



# OOM with Plants VS Zombies

---

- ▶ 클래스 다이어그램의 클래스와 객체지향 언어의 클래스는 1:1 매핑 가능
  - ▶ 객체 이름 : 클래스명
  - ▶ 속성 : 멤버변수 (member variable)
  - ▶ 행위 : 멤버메소드 (member method)



# OOM with Plants VS Zombies

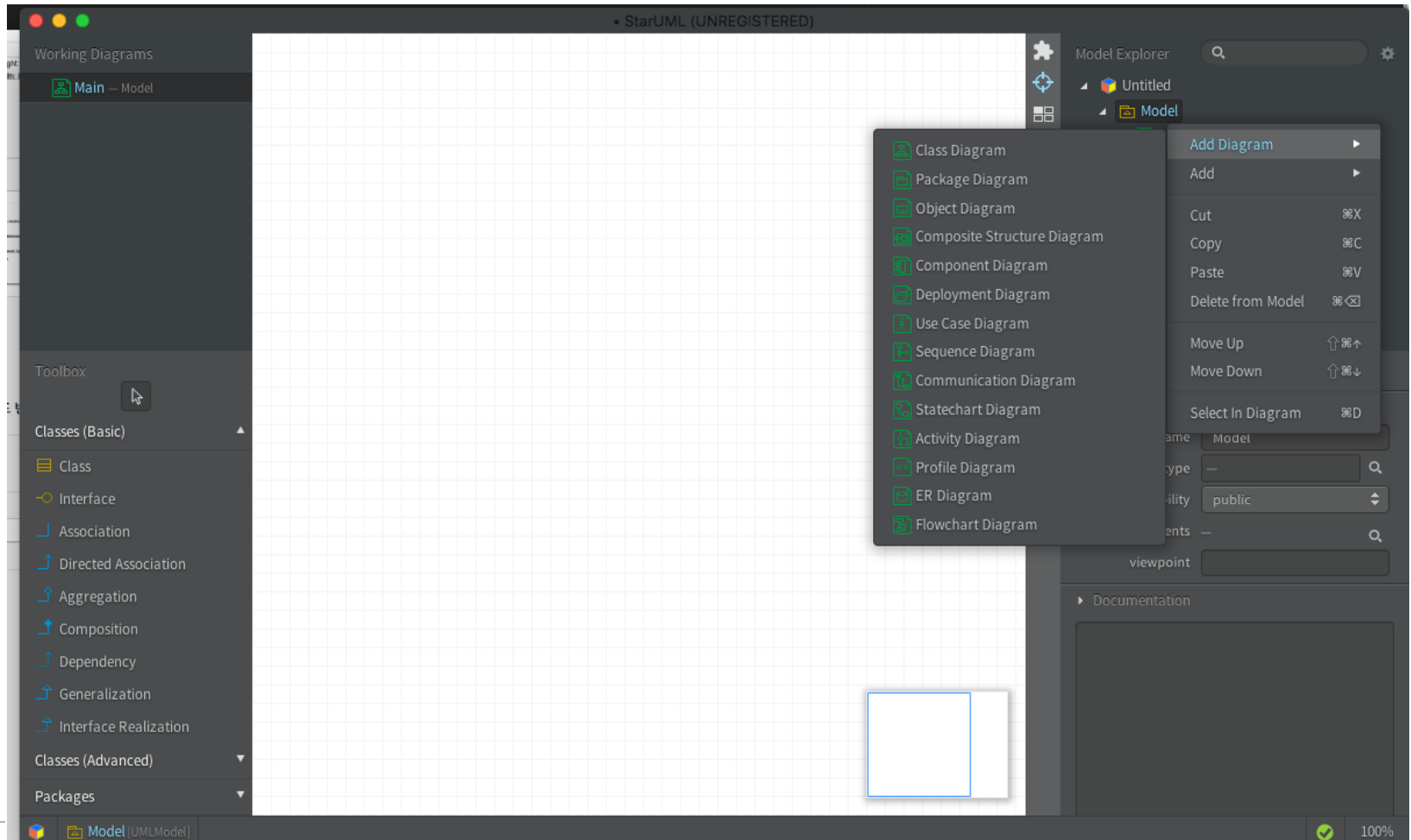
---

- ▶ UML 작성 도구
  - ▶ StarUML (<http://staruml.io>)
  - ▶ PPT
  - ▶ 자유롭게 선택



# OOM with Plants VS Zombies

## ► StarUML



## ▶ 찾은 객체를 Class로 생성



# OOM with Plants VS Zombies

## ▶ 객체 모델링 예제)



# OOM with Plants VS Zombies

---

- ▶ 객체 모델링 실습)
  - ▶ 보여지는 것들을 명사화 시킨다.
    - ▶ 햇빛, 일반좀비, 깃발든 좀비, 해바라기, 완두콩, 메뉴, 스테이지, 남은 좀비 상태, 식물 선택 창, 햇빛 수, 총알, 잔디깎는 기계, 집, 게임 배경, 식물 카드, 라인
  - ▶ 독립적 실체와 의존적 실체를 구분한다.
    - ▶ 스스로 무언가를 하는 행위(변화)가 있는 것들을 따로 분류한다.
- ▶ \*처음부터 완벽한 설계를 한다는 것은 불가능에 가까움





# OOM with Plants VS Zombies

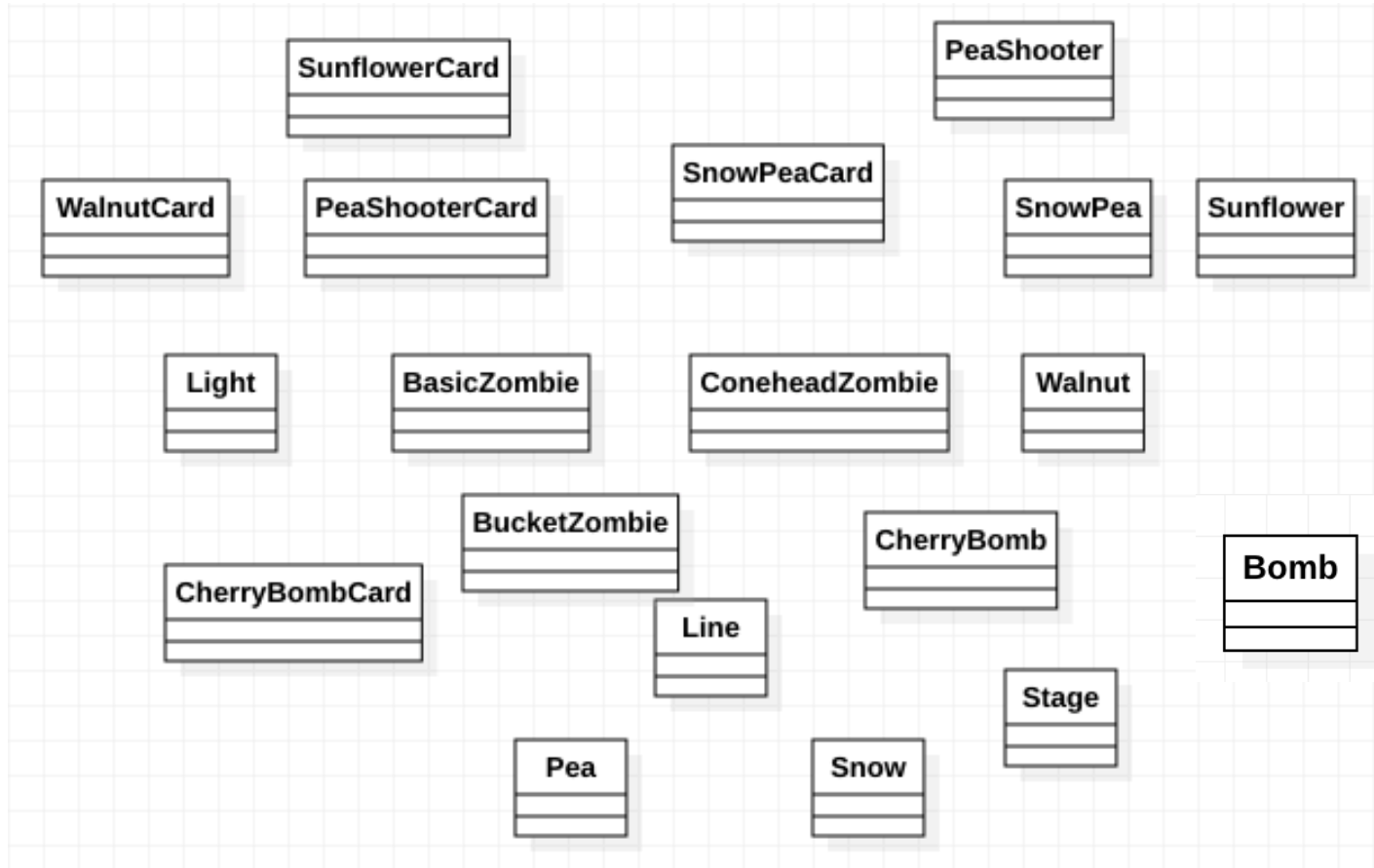
---

- ▶ 객체 모델링 실습) - 남은 시간상 전부를 구현 할 수 없기 때문에 일부만 구현
  - ▶ 햇빛, 일반좀비, 고깔좀비, 양동이좀비, 해바라기, 완두, 눈완두, 호두, 체리폭탄, 메뉴, 스테이지, 남은 좀비 상태, 식물 선택 창, 햇빛 수, 총알, 잔디깎는 기계, 집, 게임 배경, 식물 카드, 라인
- ▶ 클래스만 생성
- ▶ 모델링 결과를 토대로 자바 클래스 생성



# OOM with Plants VS Zombies

---



# OOM with Plants VS Zombies

## ▶ 모델링 결과를 토대로 자바 클래스 생성

