



ProjectHandler. class

Study - project (05-C → 06-a) part

while문 안에서 선언한 경우
반복문이 끝나면 사라짐.

조건이 참일 때 반복해서 작업 수행 (잘못 자연 무한루프각...)

String owner = null;

★ ⇒ while 문 안에서 String owner = Prompt ~ 바로 선언해버리면

while문이 끝날 때 project.owner = owner;로 리턴해줘야 하는데

이 경우 추후에 묻는 "정말 변경하시겠습니까?"에서

변경하지 않음을 선택할 때 문제가 생길 수 있다.

따라서 while 문 밖에서 선언해야 함.

while (true) {

owner = Prompt.inputString(String.format(

"만들어(%s)? (추소: 빈문자열)", project.owner));

if (memberHandler.exists(owner)) {

break;

Prompt로 입력한 owner 이름이 boolean type의 exists 메소드에서

} else if (owner.length() == 0) { true로 리턴되면 반복문을 종료한다

System.out.println("프로젝트 변경을 취소합니다.");

return; 빈 문자열을 입력하여 메서드 실행 종료

}

System.out.println("등록된 회원이 아닙니다.");

}

↳ 리턴하지 않고 계속 만들어를 물어봄. "반복문 계속 실행"

(해당되는 owner를 입력하거나 빈문자열을 입력해야

반복문이 종료 or 메서드 자체를 종료)

if문이 true일 경우

실행하지 않음

if문이 false일 때

실행

그래서! add() 부분에서는

바로 while문 내에서 선언하고

project.owner 값에 바로 owner 값을 넣어버림.

(projects[])의 배열 수소가 있는 project 변수

ProjectHandler.class

Study - project (05-C → 06-a) part

```
String members = "";
while (true) {
    String member = Prompt.inputString(String.format(
        "팀원(%s)? (완료: 빈문자열)", project.members));
    if (memberHandler.exists(member)) {
        if (member.length() > 0) {
            members += ",";
        }
        members += member;
        * continue;
    } else if (member.length() == 0) {
        break;
    }
    System.out.println("등록된 회원이 아닙니다.");
}
```

기존에 입력한 값과 길이가 0보다 길때 (존재할때)

기존 members

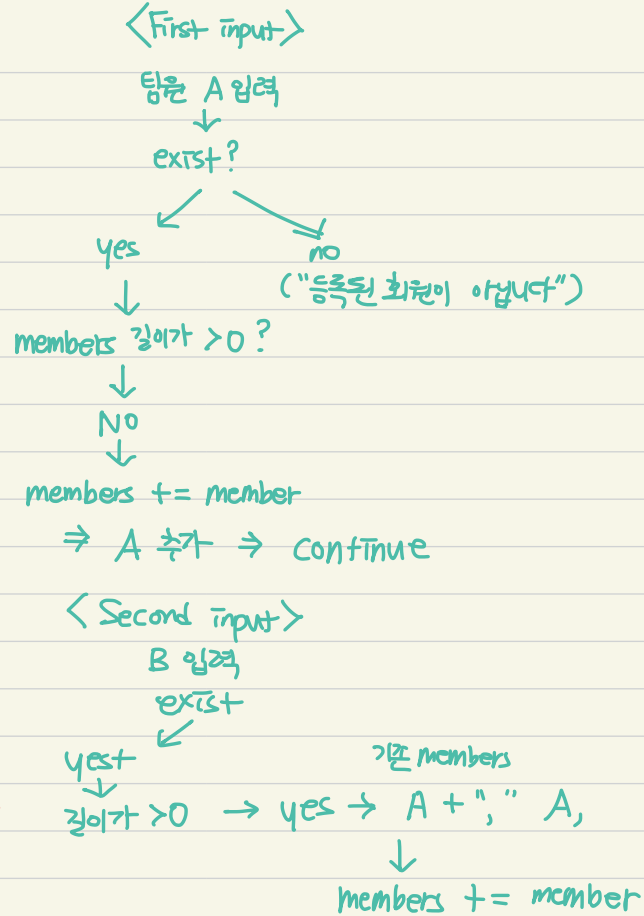
나가기 " " 빈문자열을 입력하면 (enter) 반복문 while 종료

while 처음으로

↓

없을 경우 아래의

← 이 줄이 반복 실행됨 //



MemberHandler.class

Study - project (05-C → 06-a) part

Parameter 매개변수 (어떤 입력 값이 들어올지 형태 정의)

```
boolean exist(String name) {  
    for (int i = 0; i < this.size; i++) {  
        if (this.members[i].name.equals(name)) {  
            return true;  
        }  
    }  
    return false;  
}
```

(size)
for문으로 입력한 배열 값을
다 뒤졌는데 없다?
← false

*

true/false
if (memberHandler.exist(owner)) {
 ↓ 실행 ↓ 실행 X ↓ argument 전달 인자 (인수)

members[i].name.equals(name)