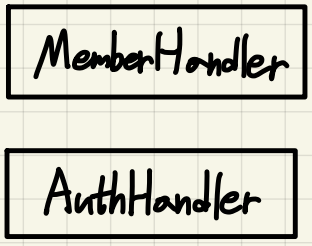
2021.08.30 (월 45일차)

study project eomcs.pms.App * 로그인

13-2



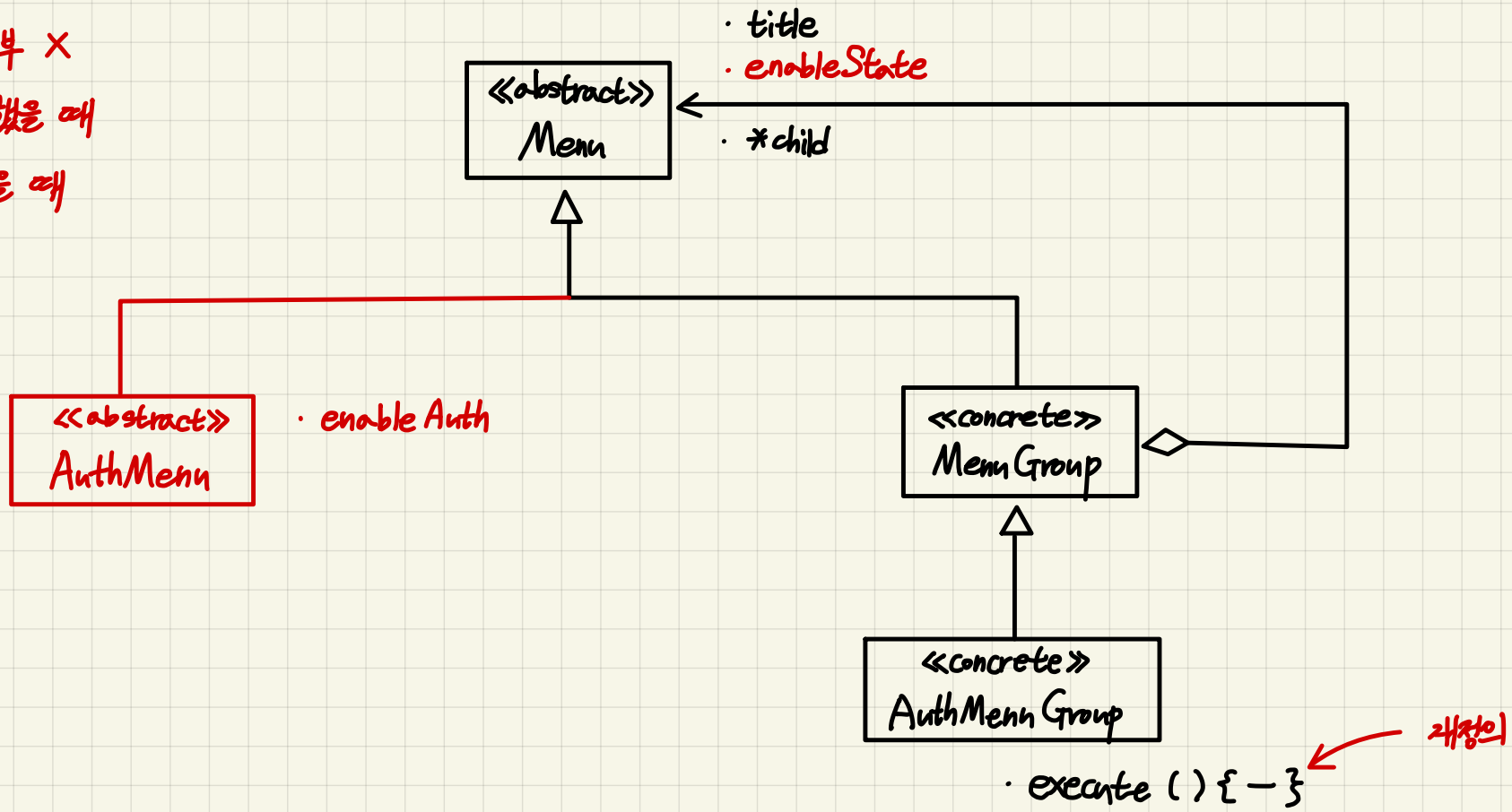
2021.08.30 (월 45일차)

* 로그인 여부에 따라 메뉴 출력 제어

0 : 로그인 여부 X

1 : 로그인 안했을 때

2 : 로그인 했을 때

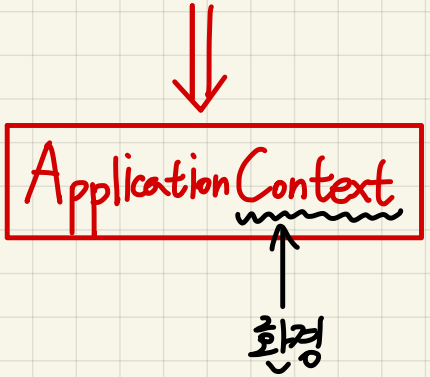


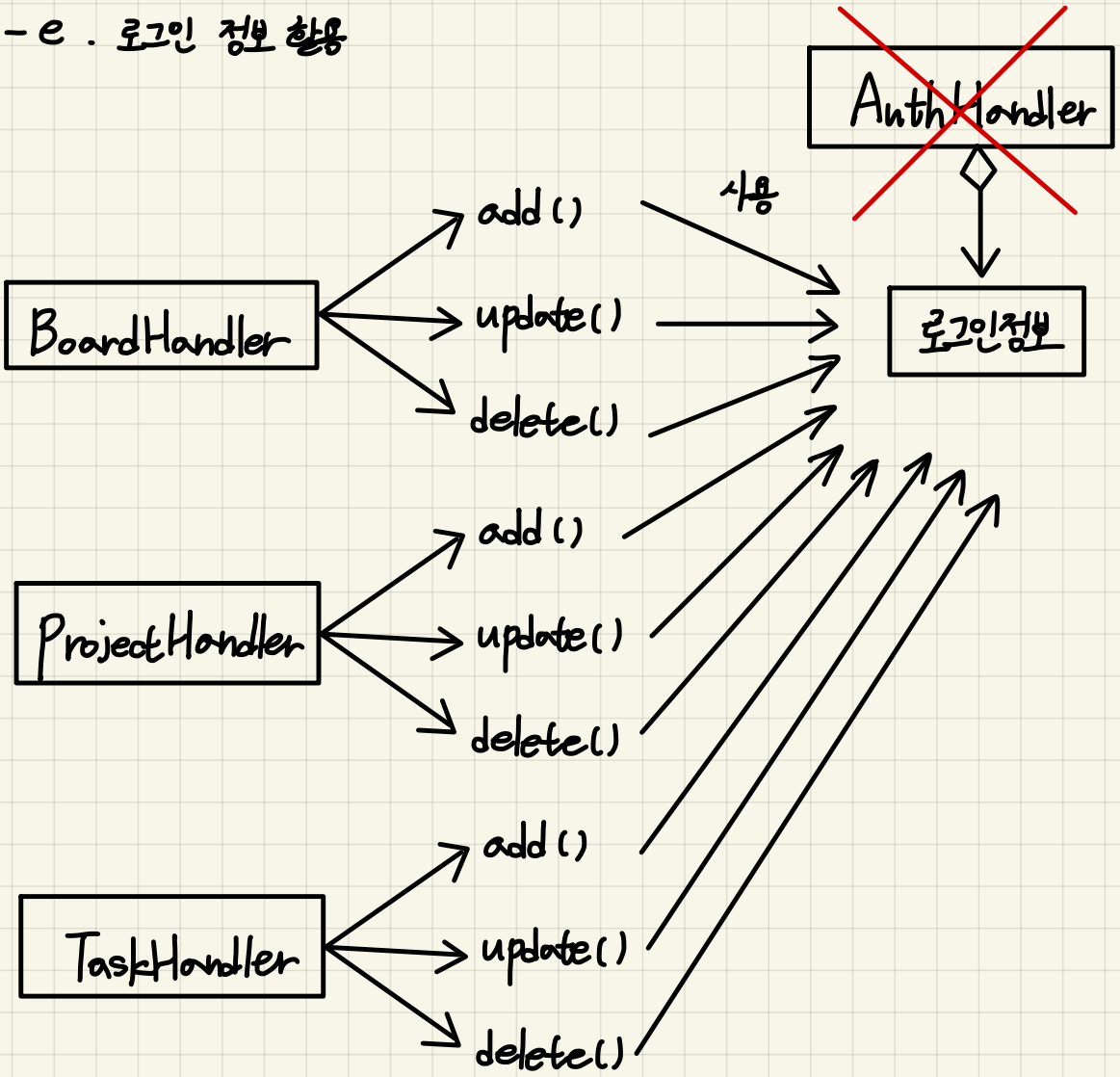
2021.08.31 (46일차) 11:30

13-e

* 13-e . 로그인 정보 활용

*로그인 정보 및 작업들 간에
공유할 정보를 별도의 전문 클래스로 분리한다.



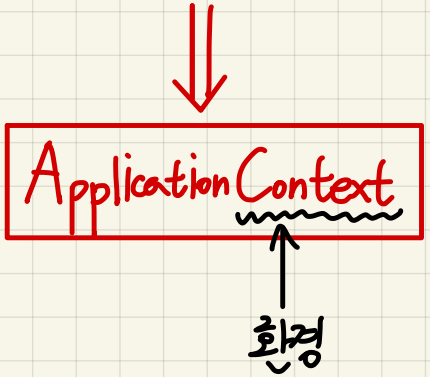


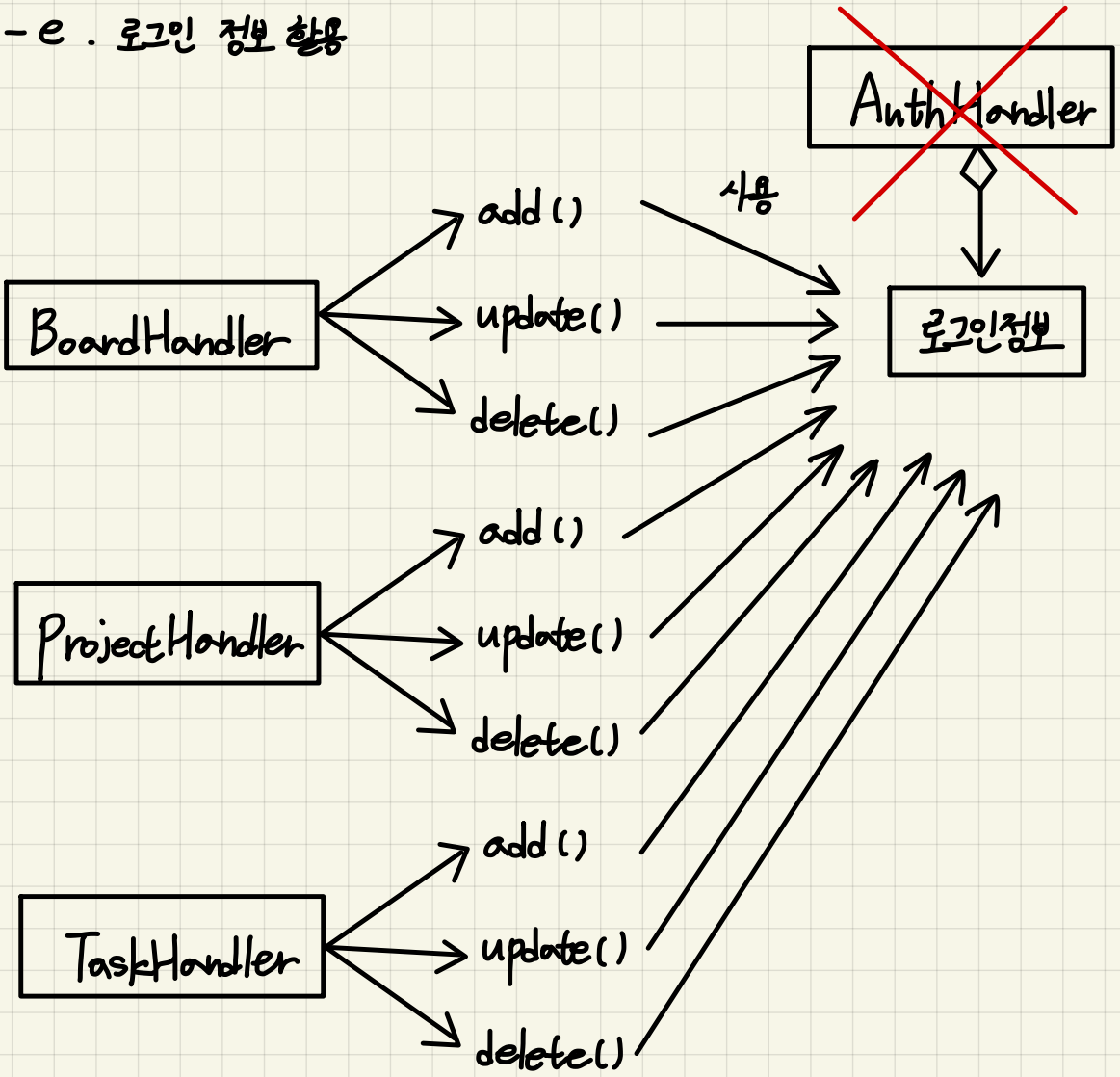
2021.08.31 (46일차) 11:30

13-e

* 13-e . 로그인 정보 활용

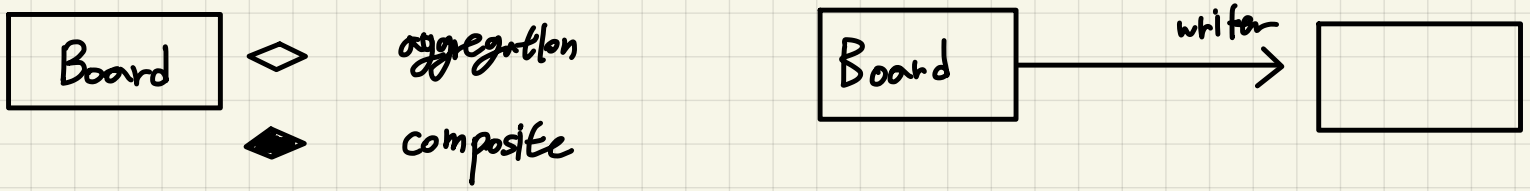
*로그인 정보 및 작업들 간에
공유할 정보를 별도의 전문 클래스로 분리한다.





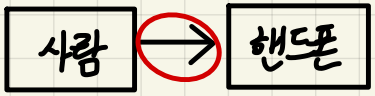
2021.08.31 (46일차)

* 게시판 로그인 정보 활용



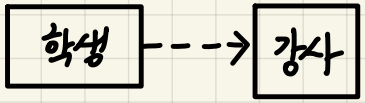
① Inheritance 상속

② Association 연관



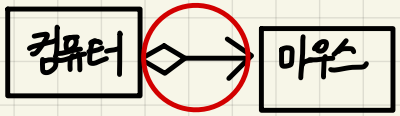
연결 강

③ Dependency (의존)



연결 느슨

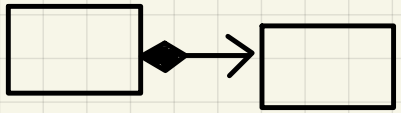
④ Aggregation (집합)



메서드 있음

Lifecycle ≠

⑤ Composite



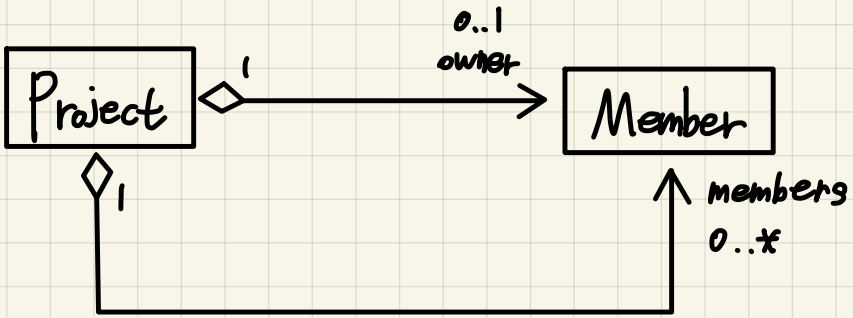
메서드 수 없음

Lifecycle =

2021.08.31 (46일차) 1:20

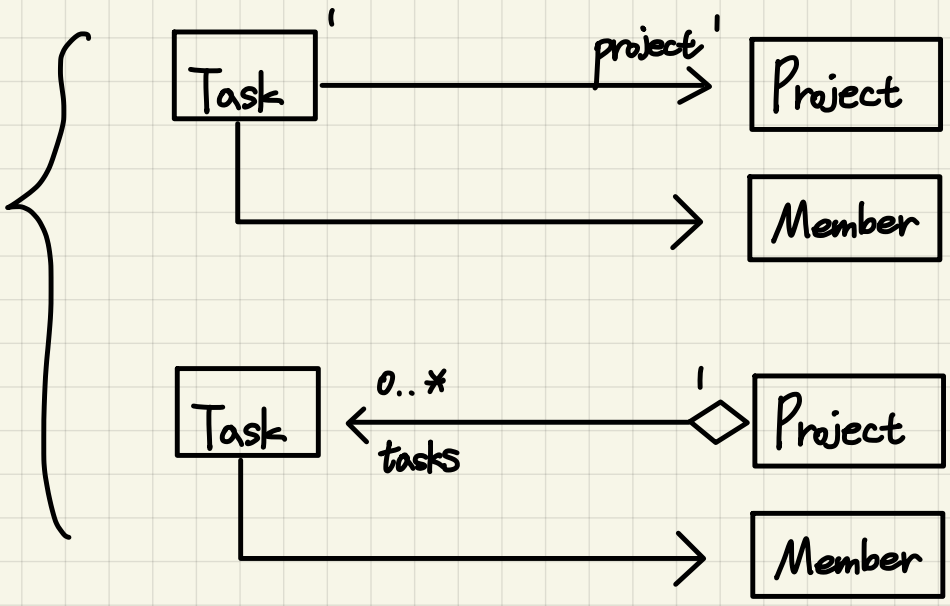
* 13-e. 프로젝트 관리 로그인 정보 활용

class diagram



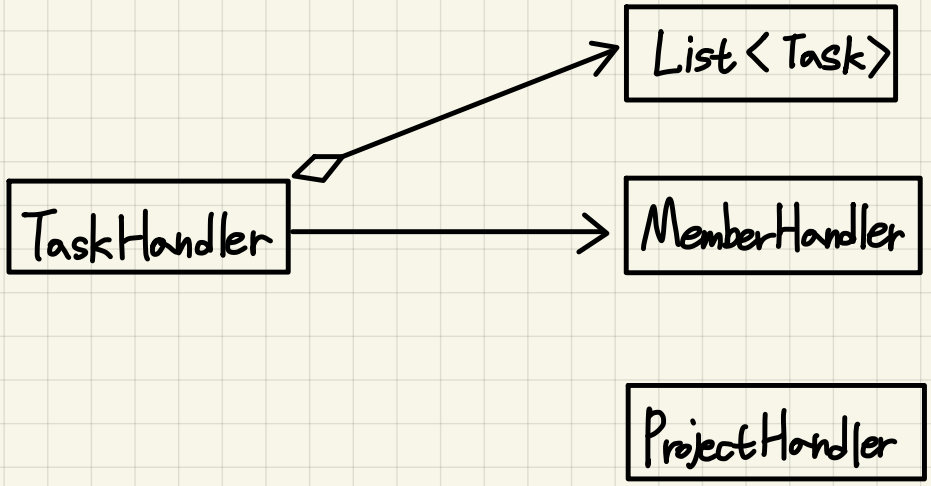
2021.08.31 (46일차)

* 13-e. 작업관리에 로그인 정보 활용하기



2021.08.31 (46일차) 14:40

* 13-e. 작업관리에 로그인 정보 활용하기



2021.09.01 (47일차)

* 14 - X Command 디자인 패턴 적용

기존방식 → 개선

- 게시글 등록 메뉴 — add()
- 게시글 목록 메뉴 — list()
- 게시글 상세조회 메뉴 — detail()
- 게시글 변경 메뉴 — update()
- 게시글 삭제 메뉴 — delete()
- 새 메뉴 추가 — call

BoardHandler

"새 메서드 추가"

* 문제점
- 새 기능을 추가할 때 기존 코드를 손댄다.

  ||

  버그가 생길 가능성을 높힌다.

만약,

하나의 명령처리를 하나의 메서드가 담당하고 있는 상황에서

⇓

메서드를 객체화 시키기 좋은조건

유지보수 하기가 더 쉬운 구조가 된다.

- BoardAddHandler
- BoardListHandler
- BoardDetailHandler
- BoardUpdateHandler
- BoardDeleteHandler
- 새 메뉴를 처리할 클래스

* 메서드를 객체화?

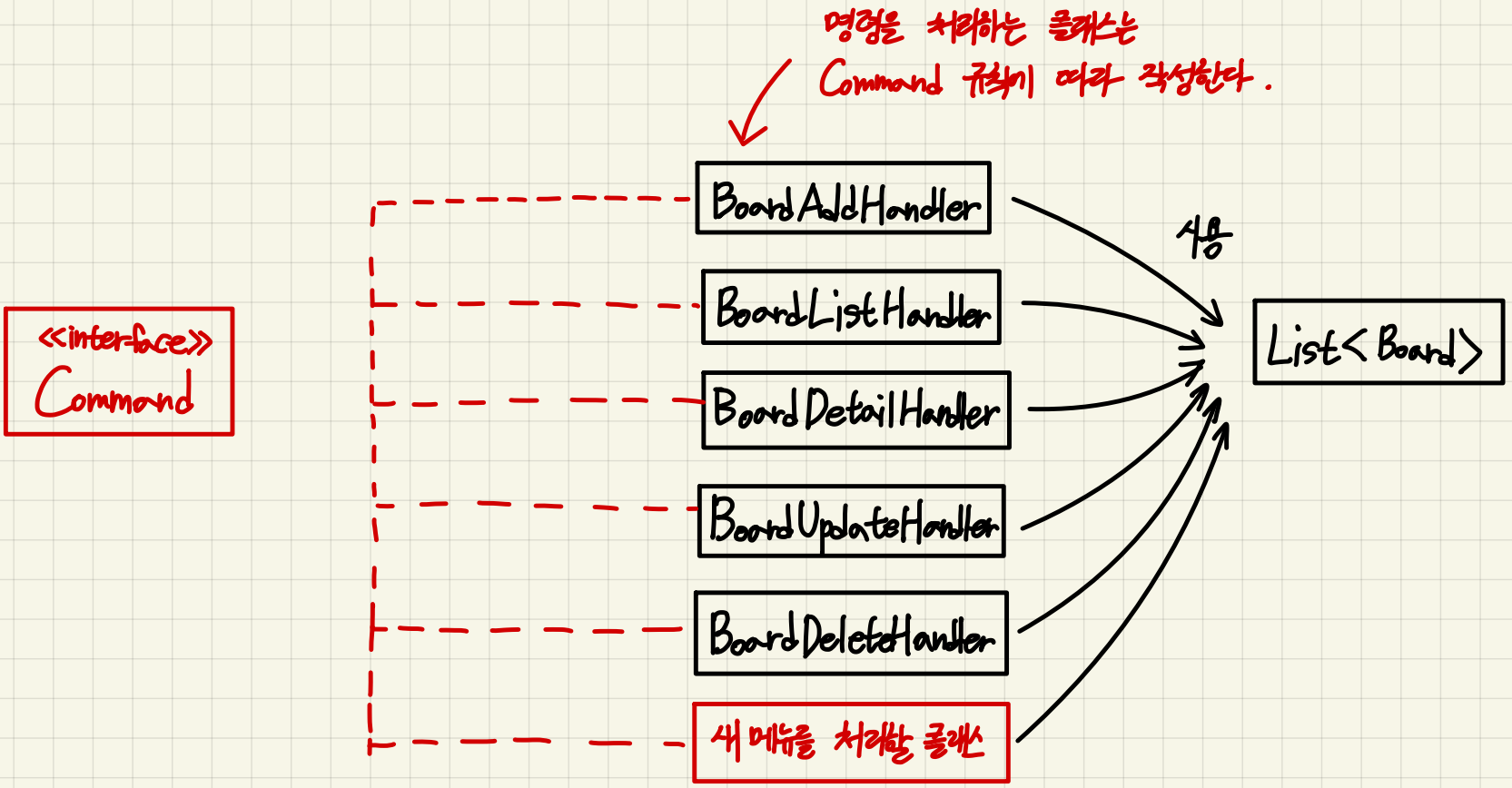
메서드를 별도의 클래스로 분리!

⇓ 설계기법

`Command' 디자인 패턴

2021.09.01 (47일차)

* 14 - x Command 디자인 패턴 적용



2021.09.01 (47일차) 11:30 * 14 - X Command 디자인 패턴 적용