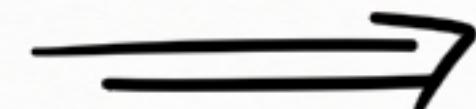
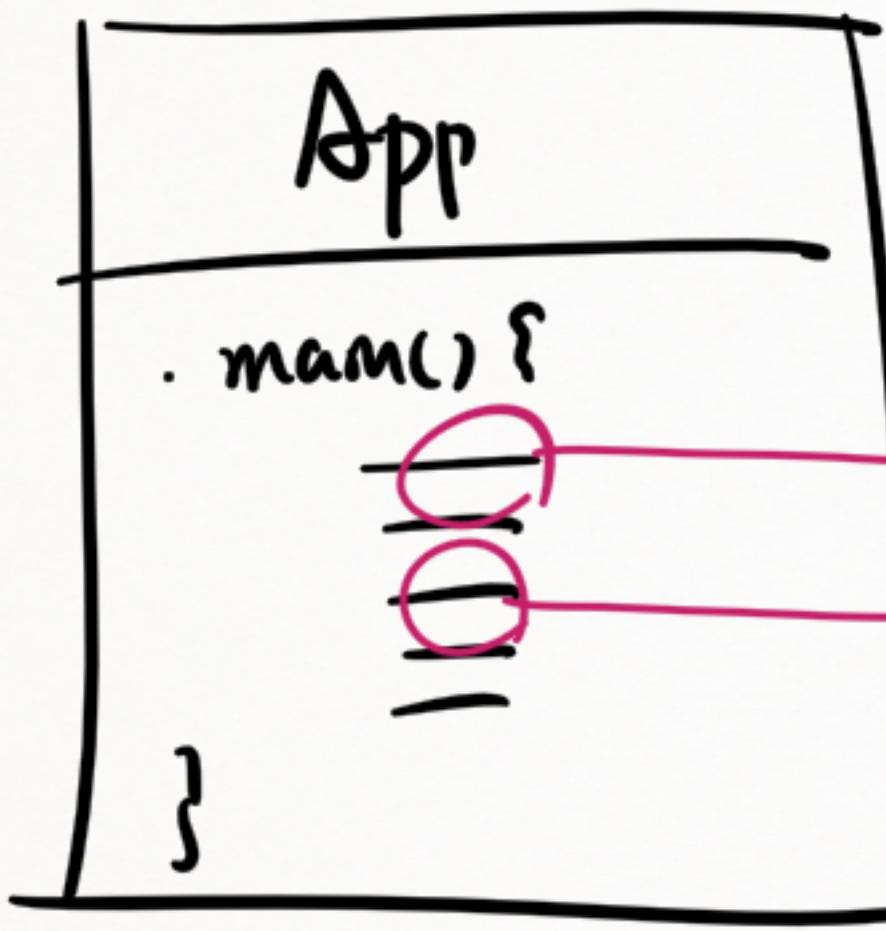
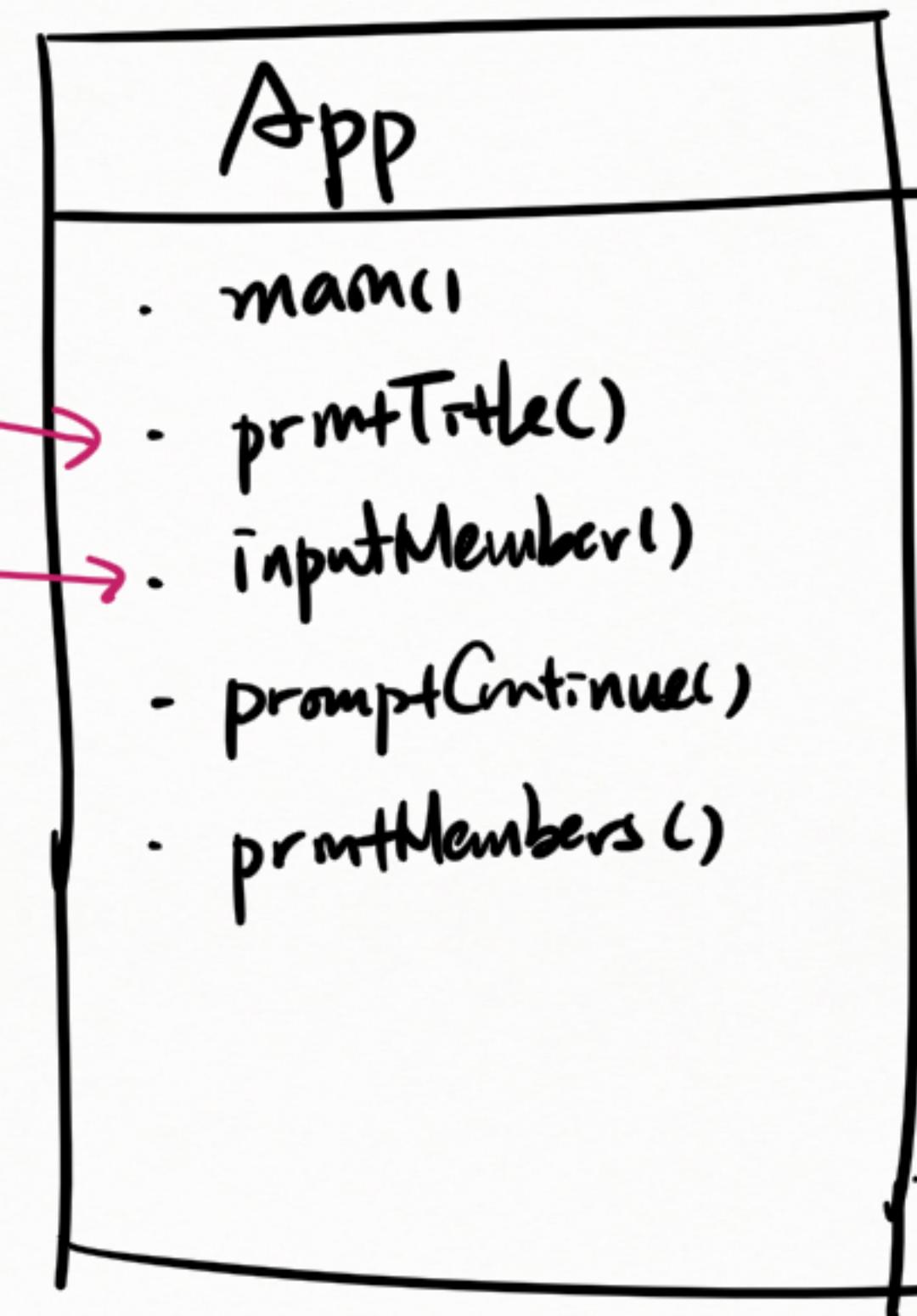


* 1. 디렉트 사용법

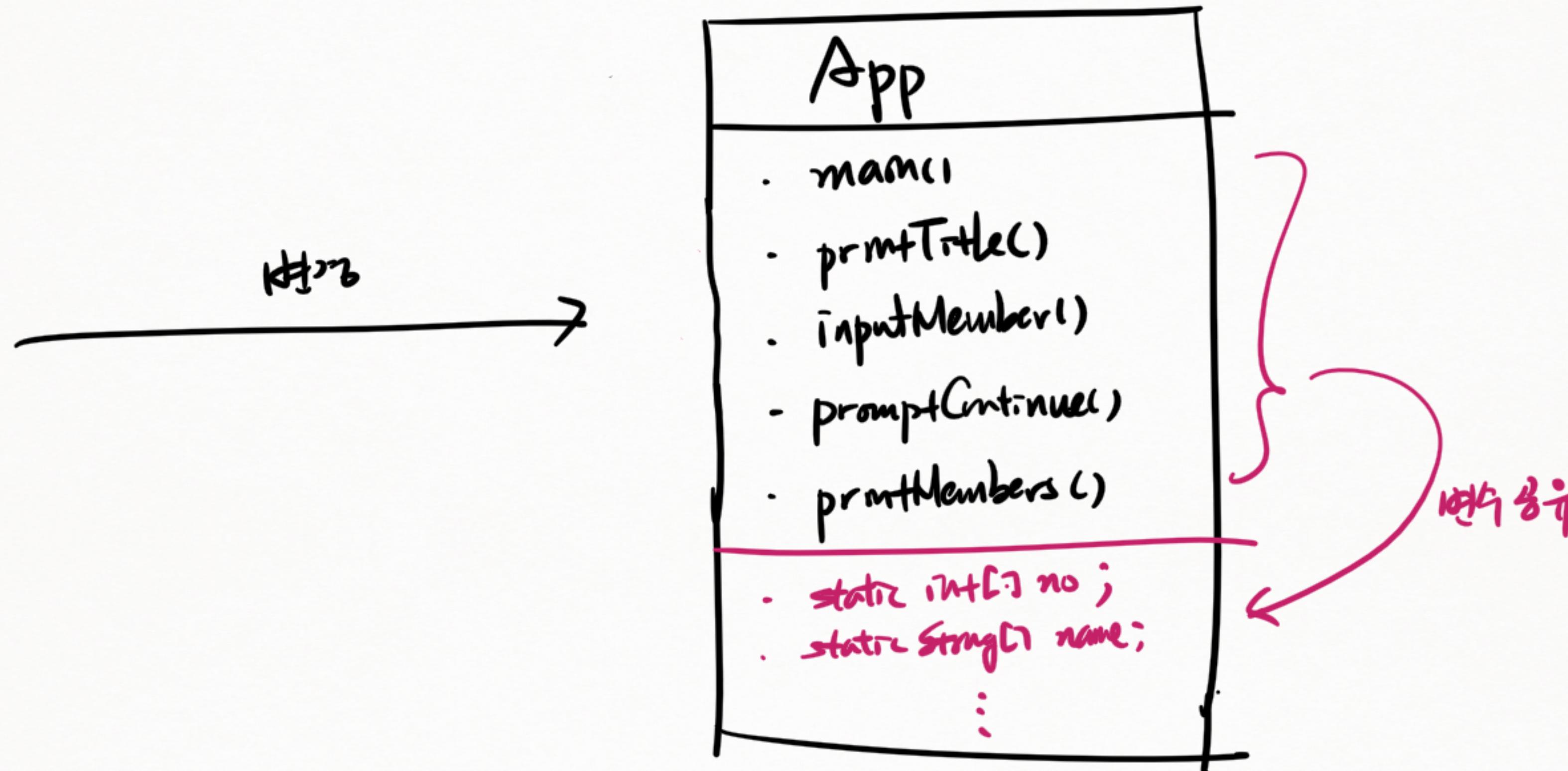
이전



변경

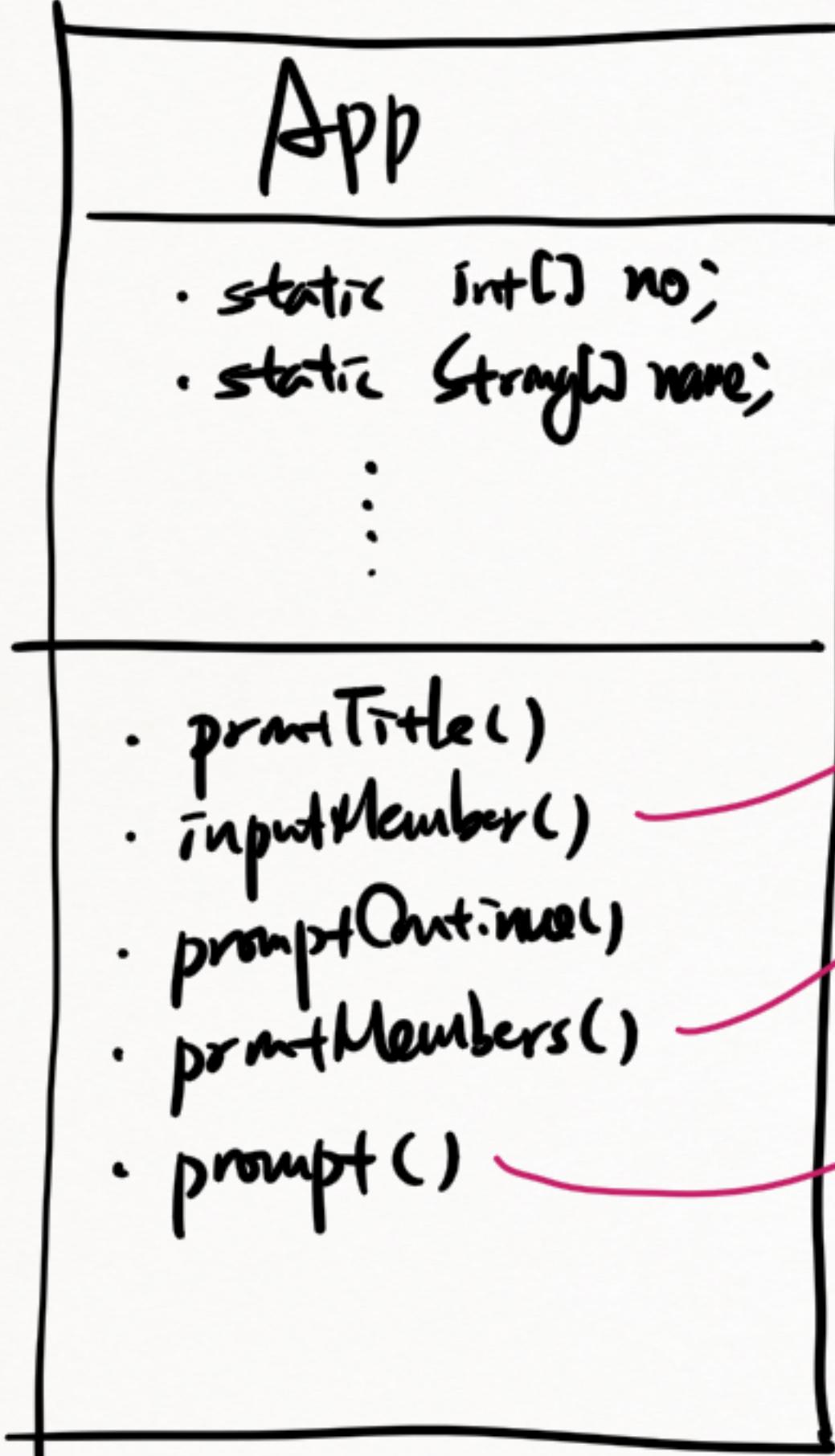


* Q. Lession with Array

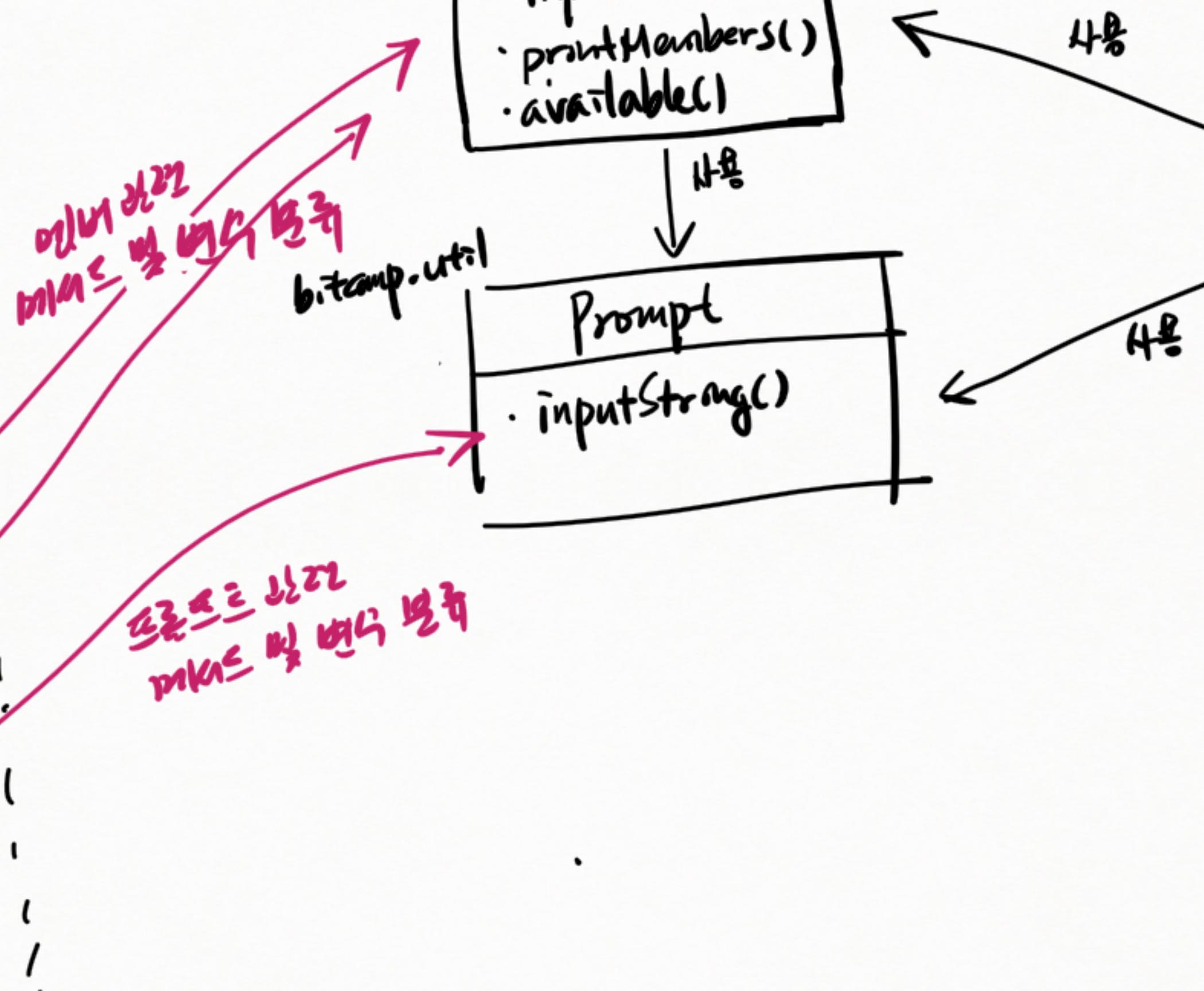
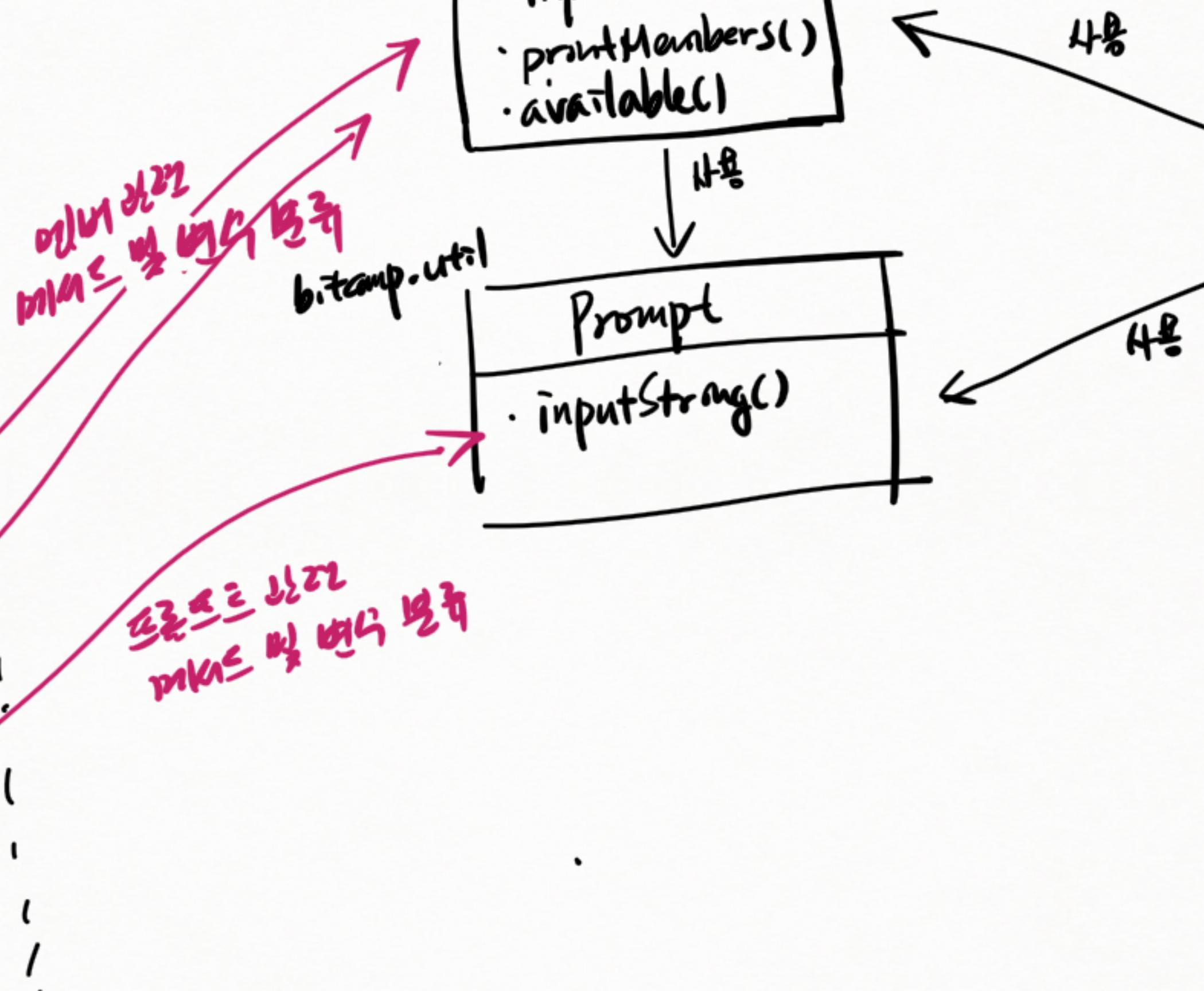
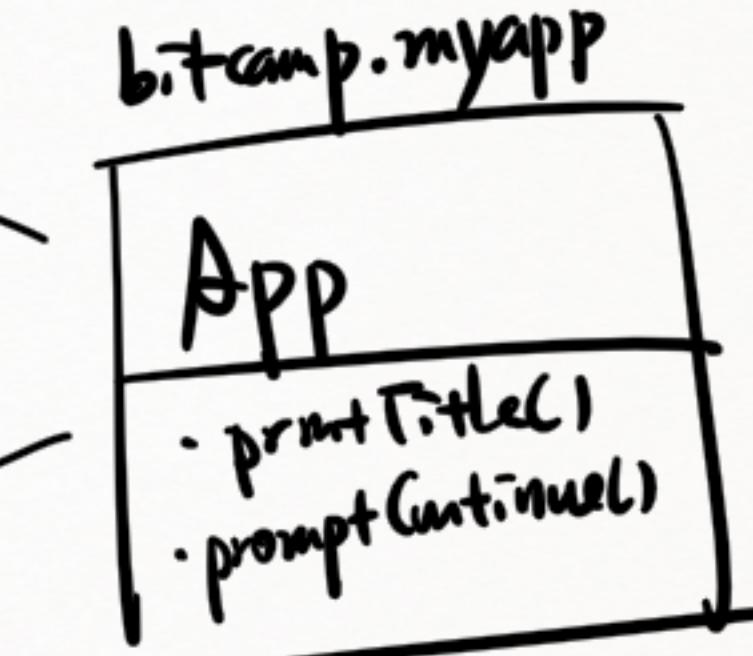
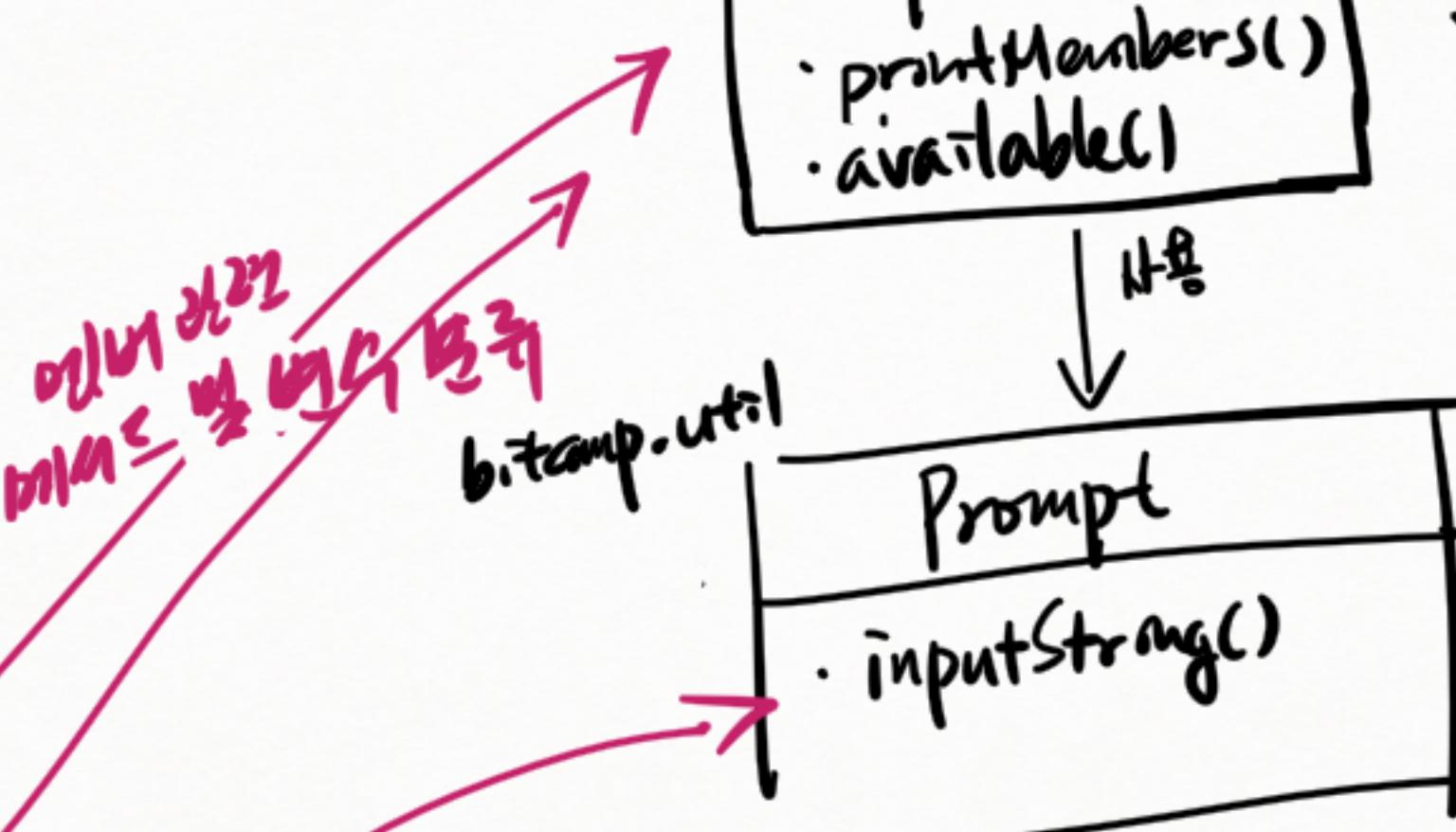
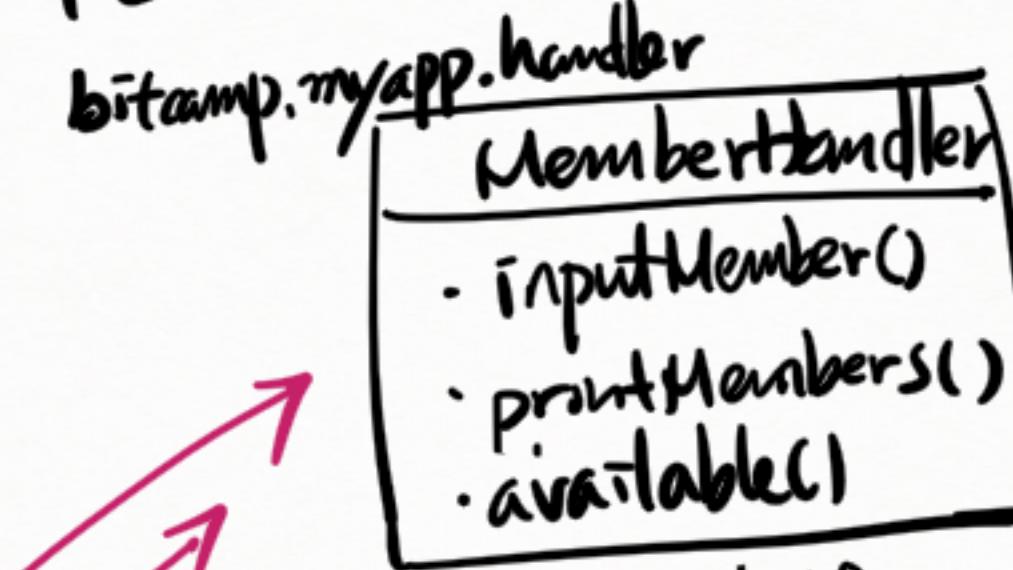


* 9. 클래스 및 패턴 학습

이전 구조
~
Architecture



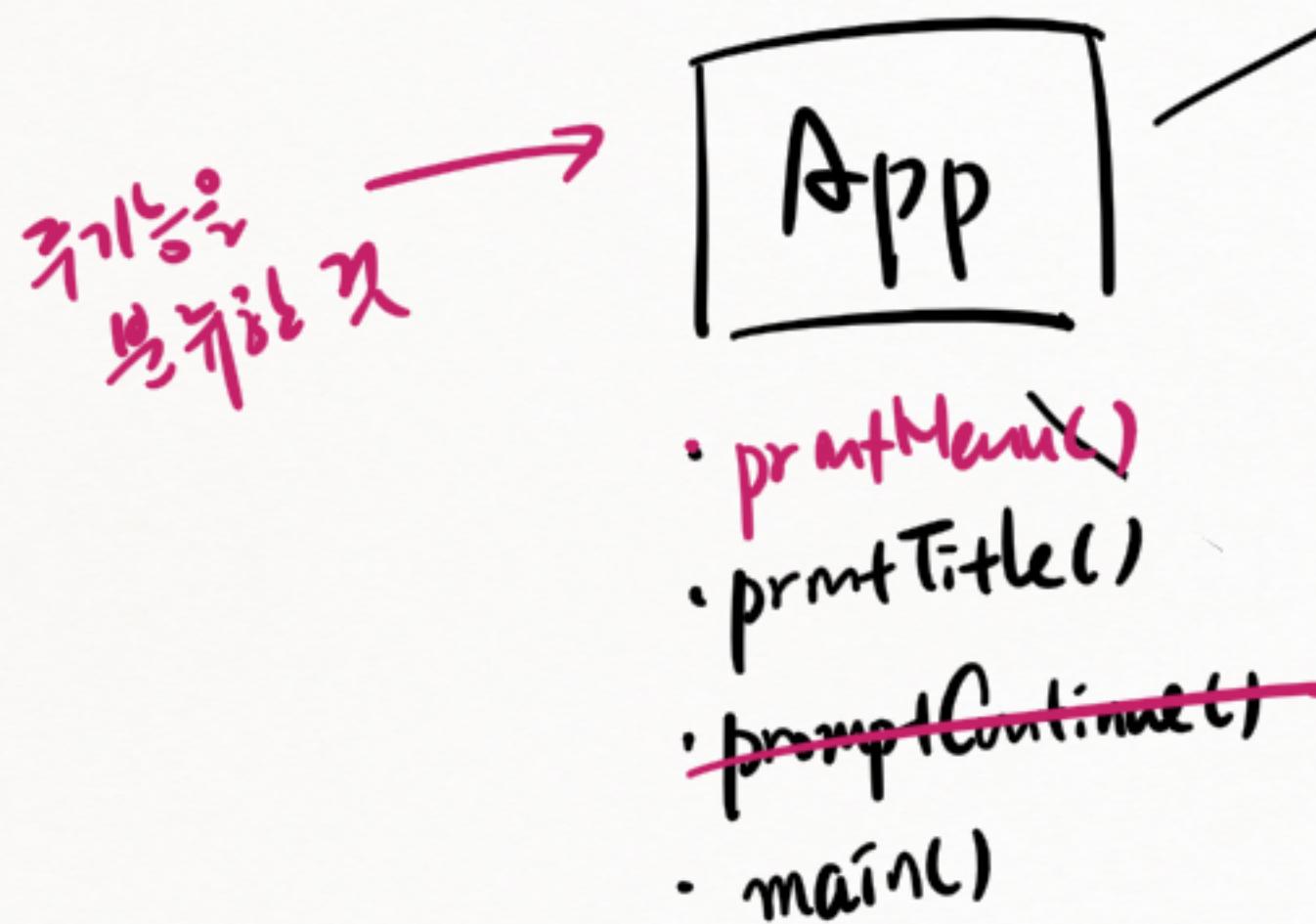
내 구조



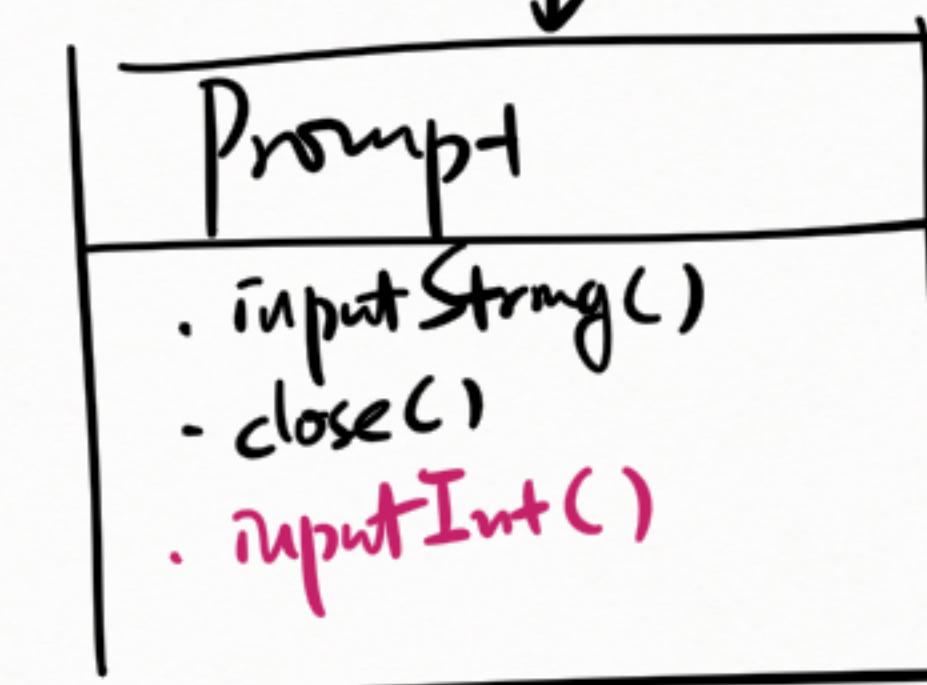
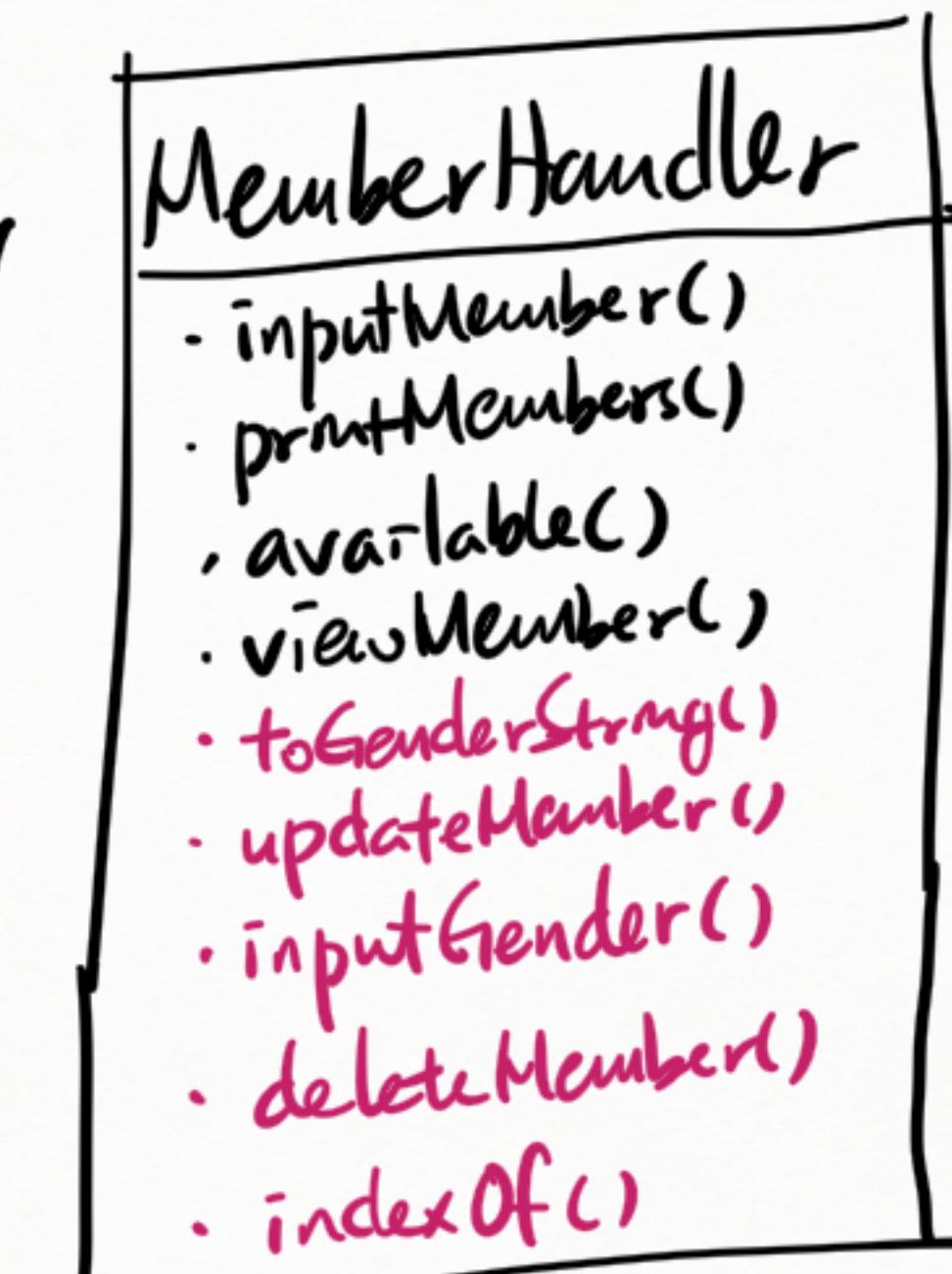
* 10. 멤버 및 CRUD 구현

* 클래스
↳ 애플리케이션 메서드를 분류할 것

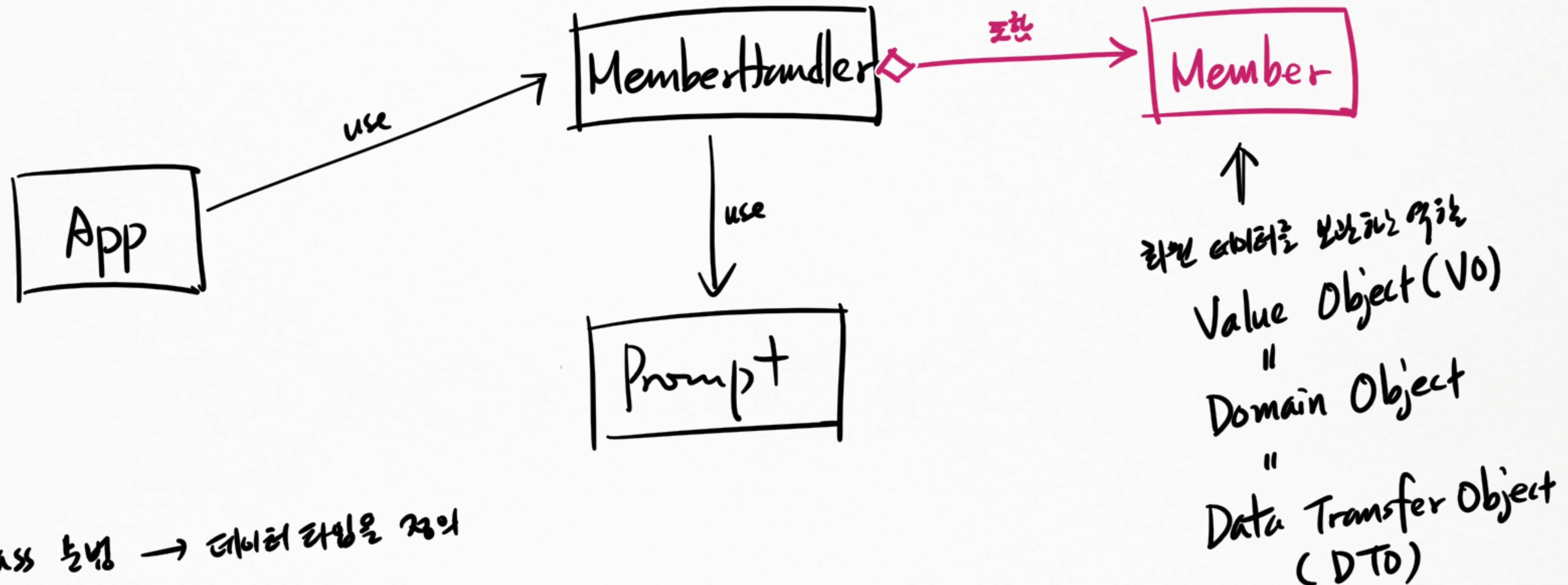
* 패키지
↳ 클래스를 분류할 것.



class 분류 → 메서드를 찾는 용도

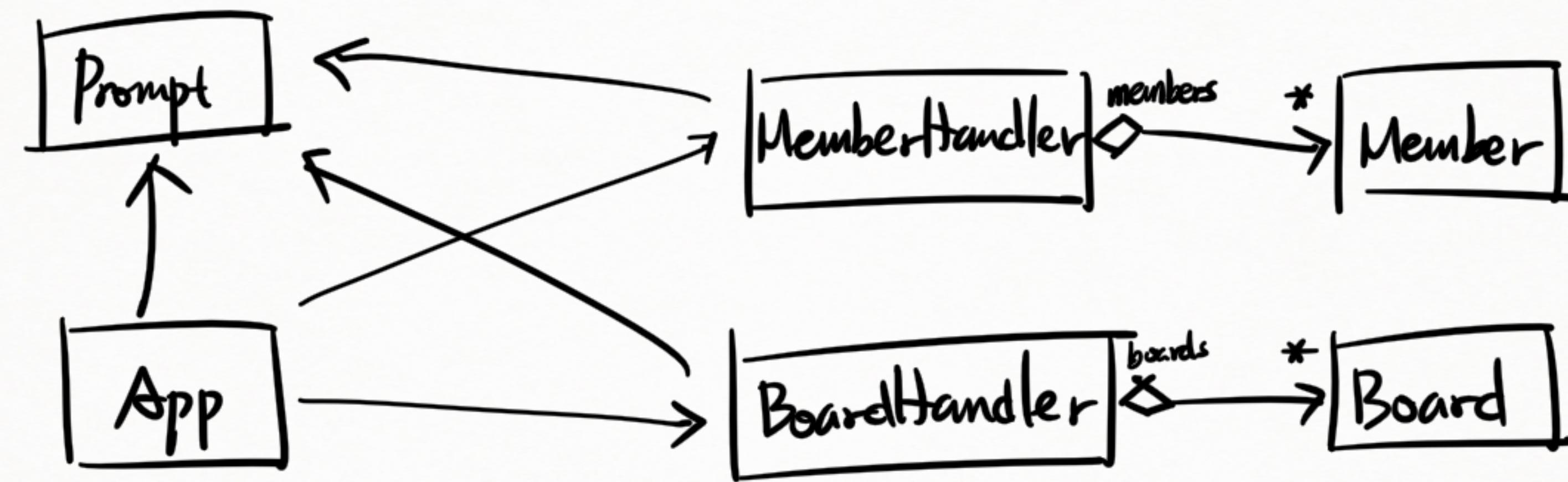


11. 사용자 정의 데이터 타입 만들기

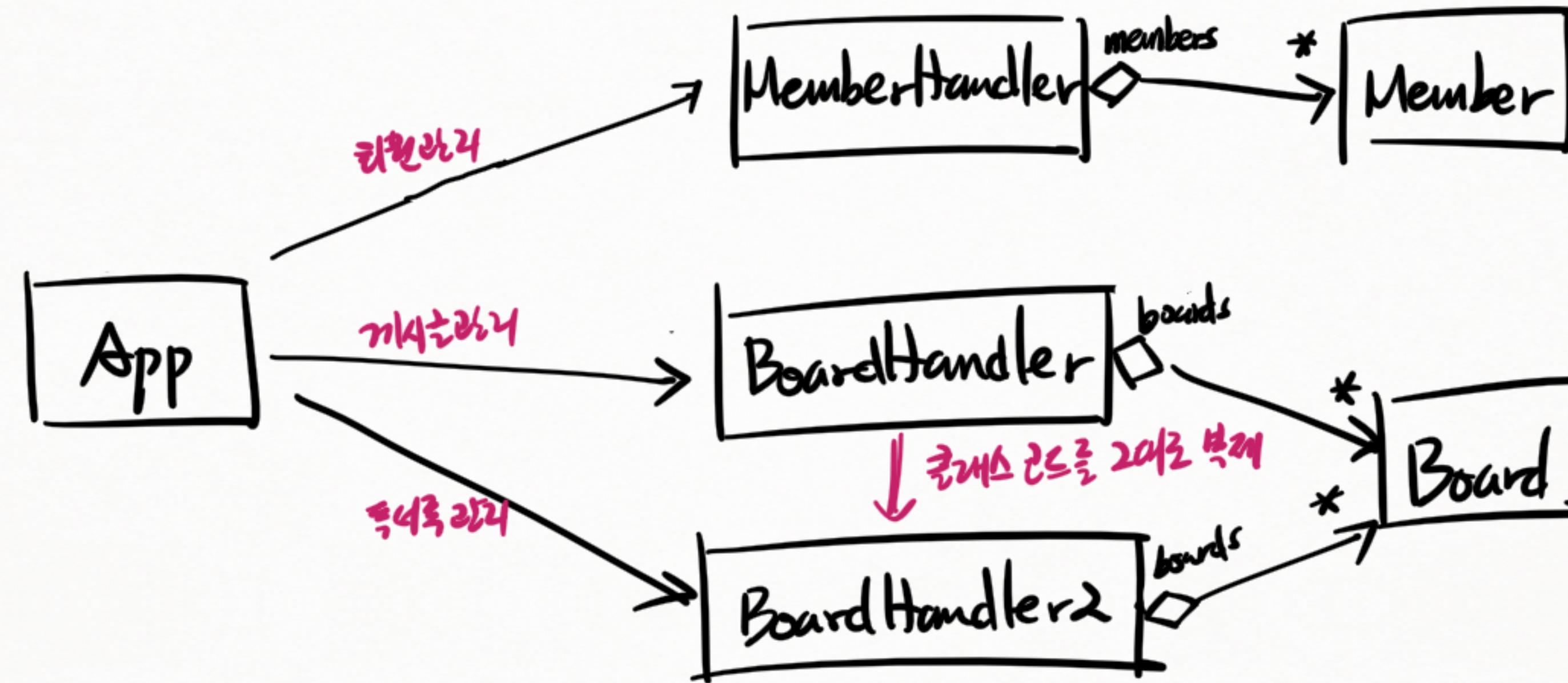


* class 늘기 → 데이터 타입을 확장

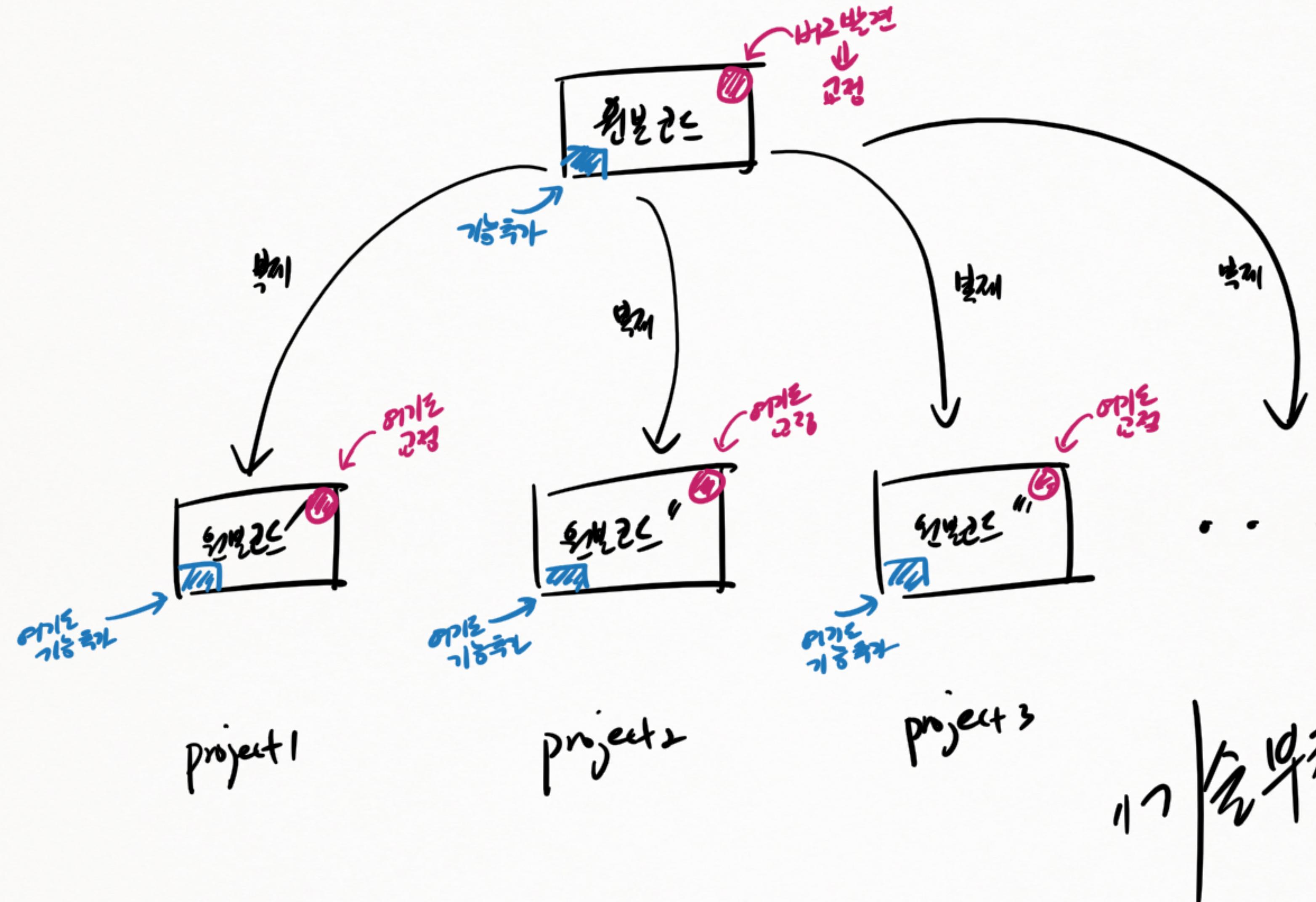
13. 깃허브 CRUD 추가



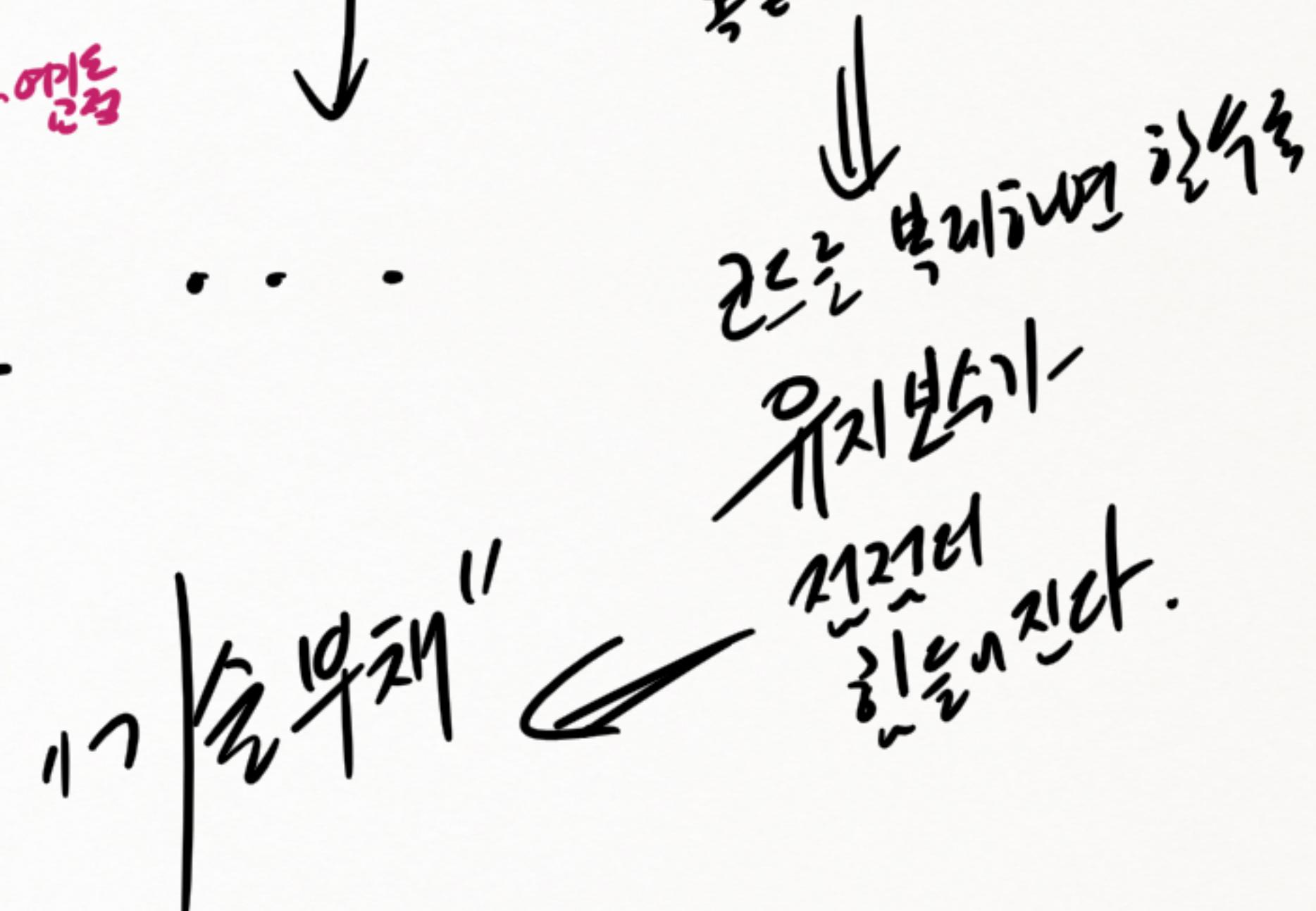
14. 토커를 CRUD 추가



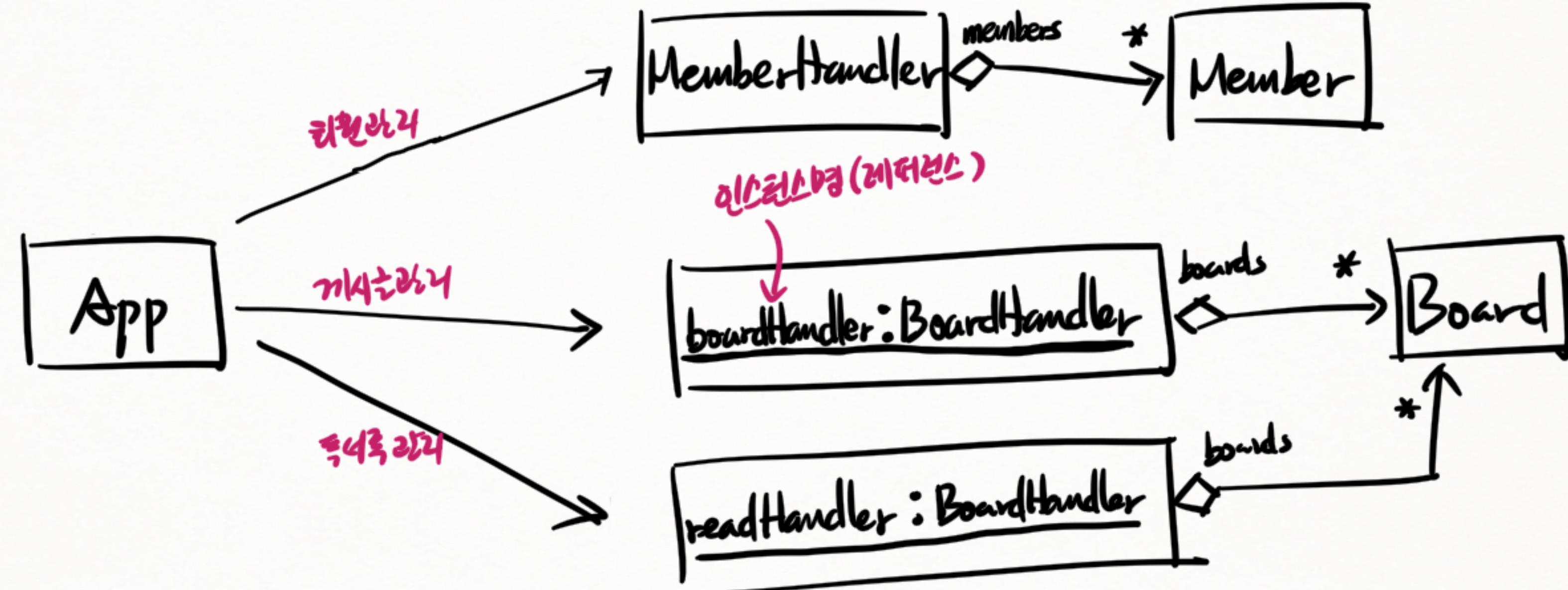
* 가로관 복제로써 새기능을 주입하는 예



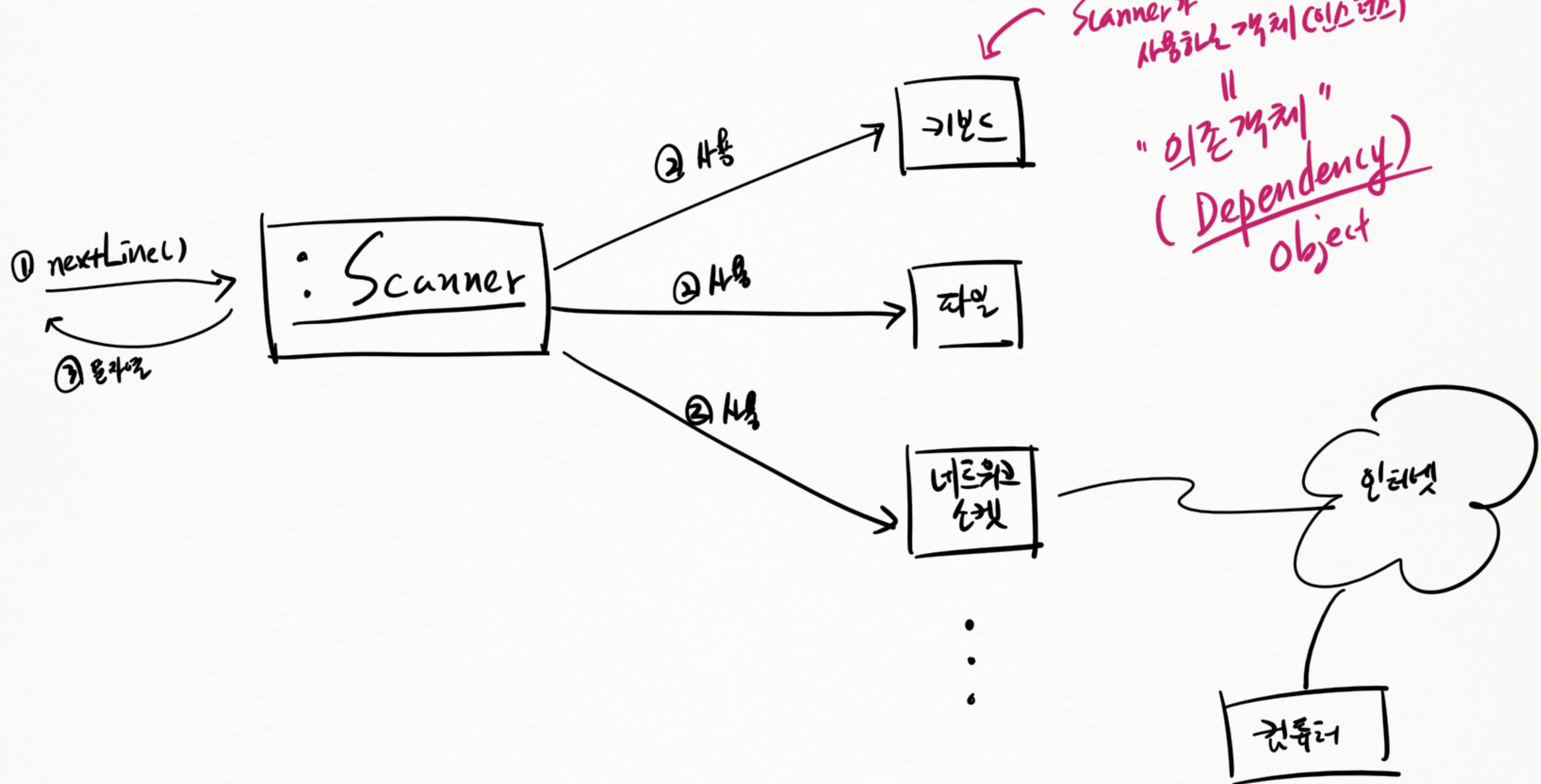
* API를 고정하여 기능을 추가하여 API를 복제한 코드의 디자인은 동일한 일을 수행하는 코드다



15. 인스턴스 있는 B2M에서 학습



Scanner 와 의존 객체



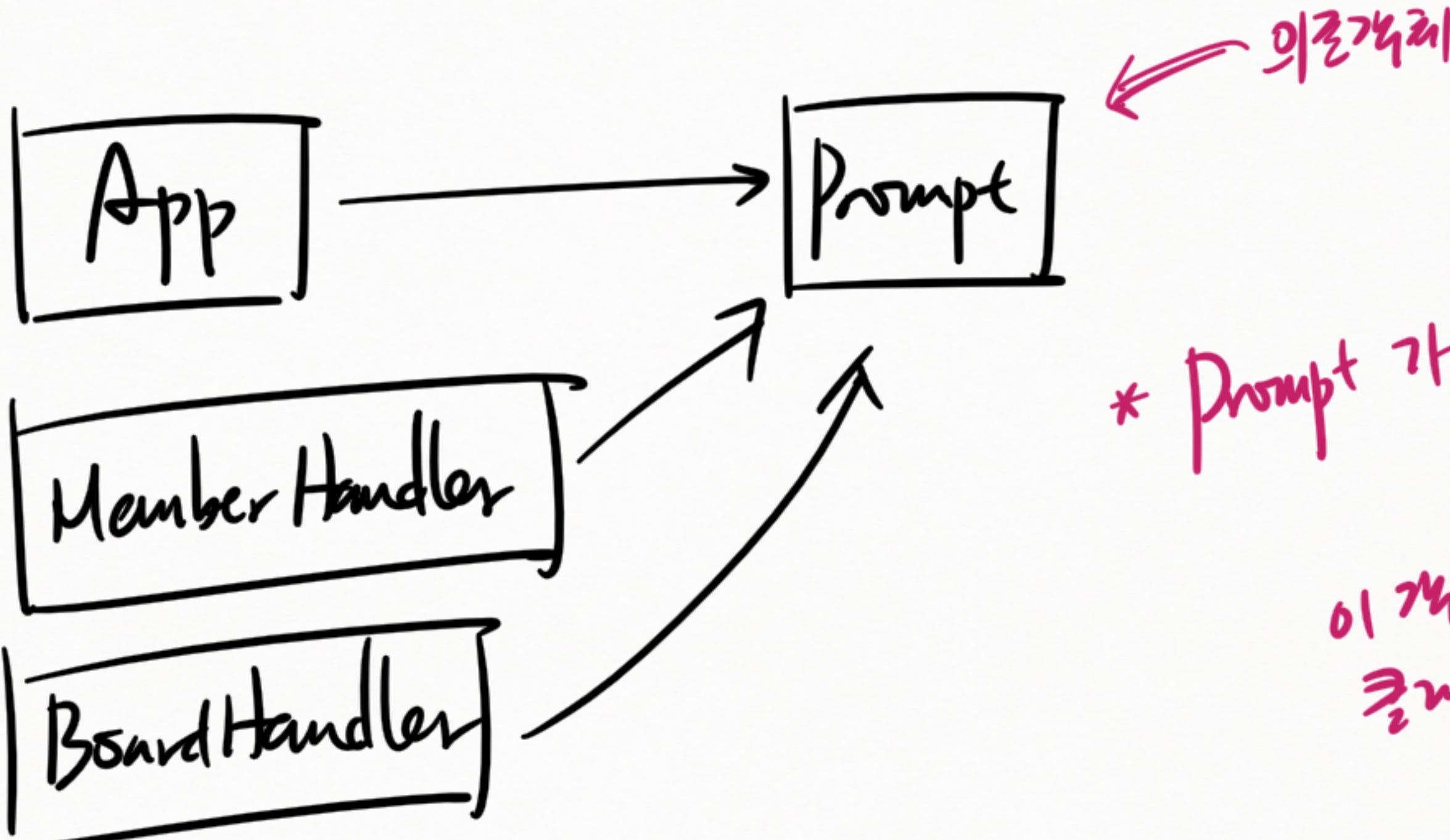
App, MemberHandler, BoardHandler et prompt

생성자 주입!



부모에
속해
하는가?

"생성자 주입"
Dependency Injection
(DI)

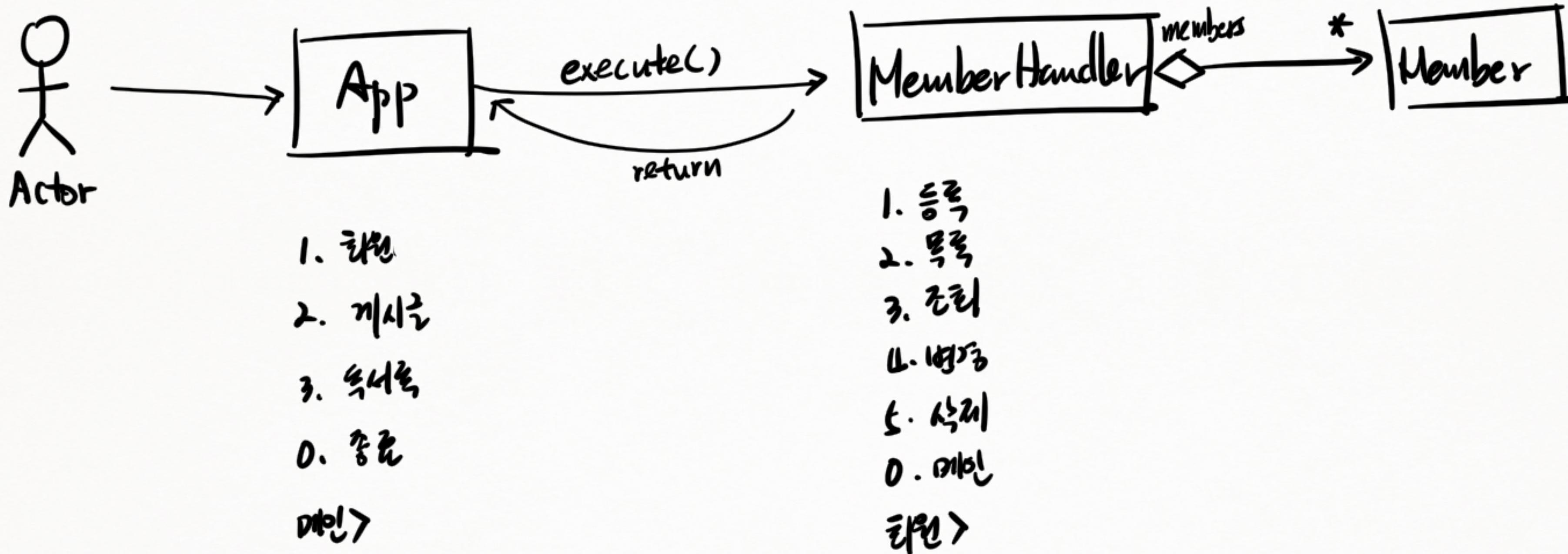


의존주입
* Prompt 가 인스턴스 복수로 만들기

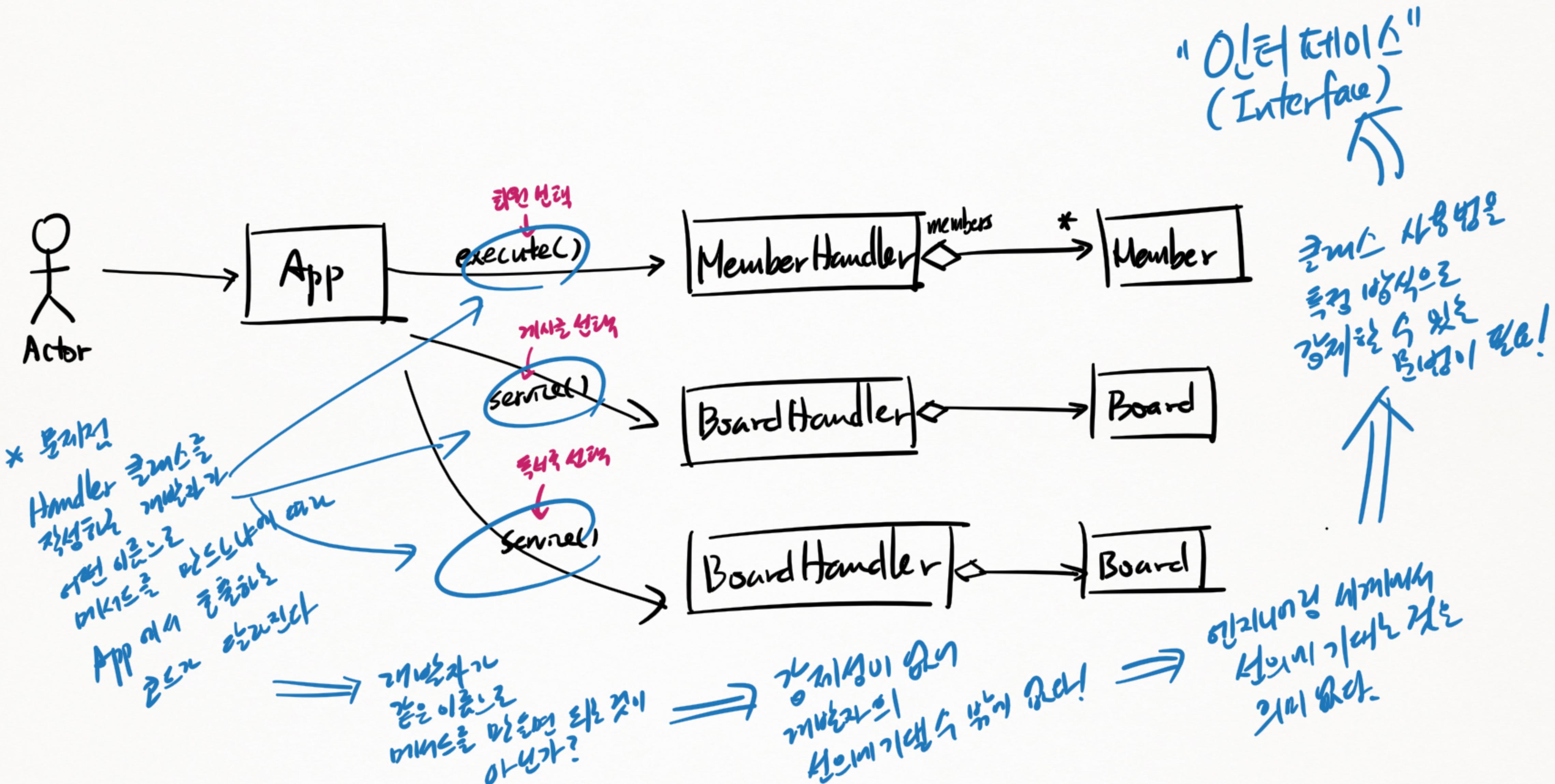
이 개체를 Algorithm
으로 변경되었는데
그럼으로

생성자를 주입
의존주입(Prompt)
주입!
Injection

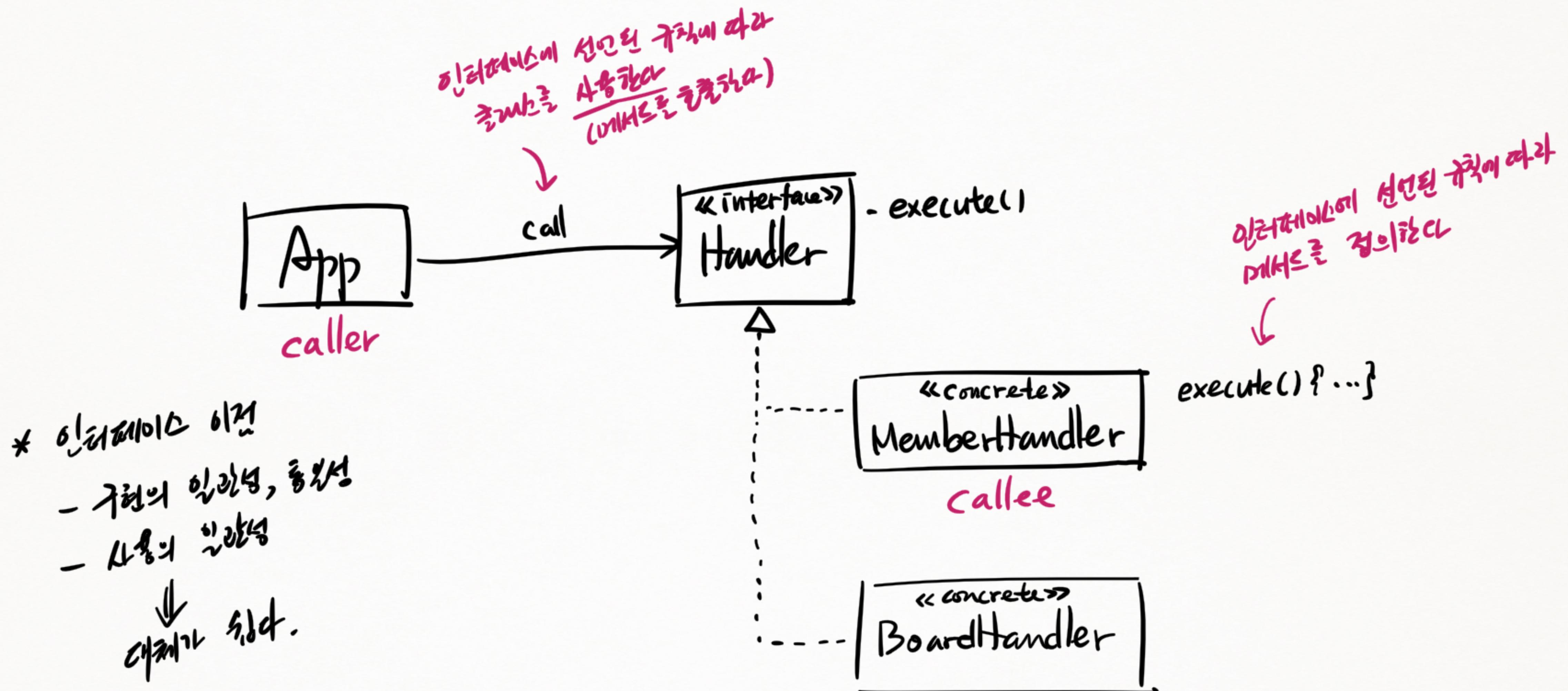
16. Handler에게 메뉴 기능을 위임



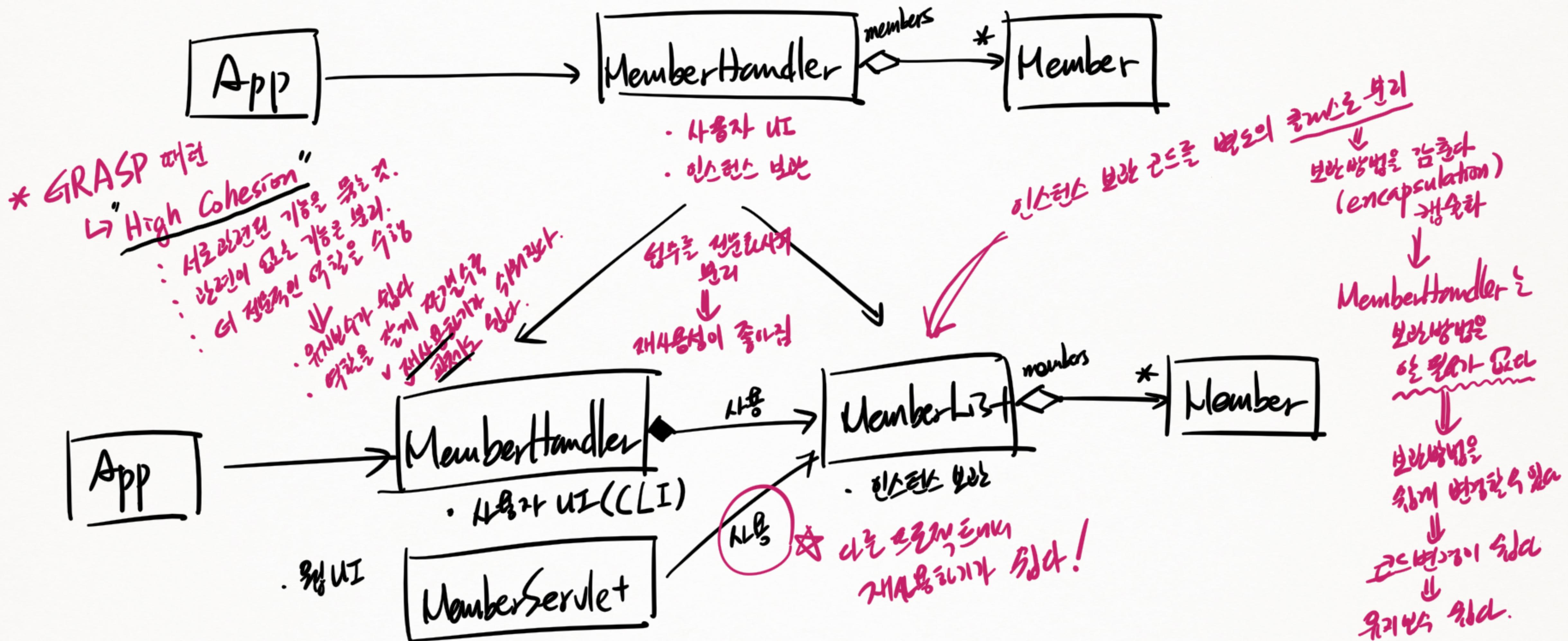
16. Handler에게 메뉴 기능을 위임



III. Handler의 사용 구조를 인터페이스로 정의하기

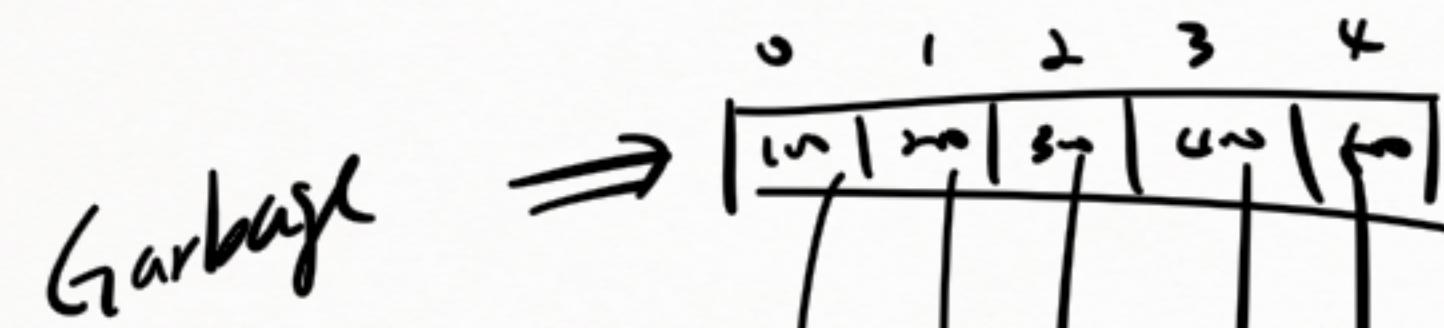


18. 인스턴스 목록을 다수로 코드를 빼는 축제로 보기

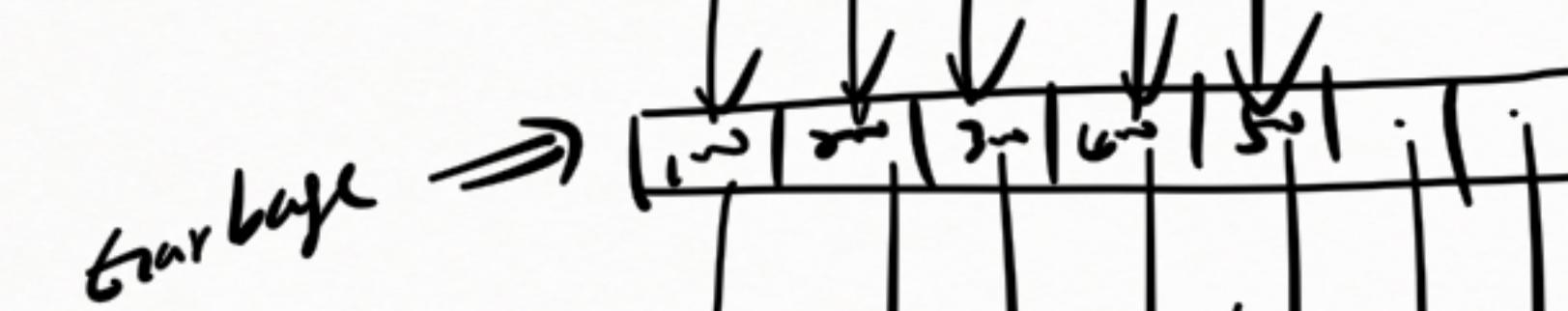


* 배포 흐름

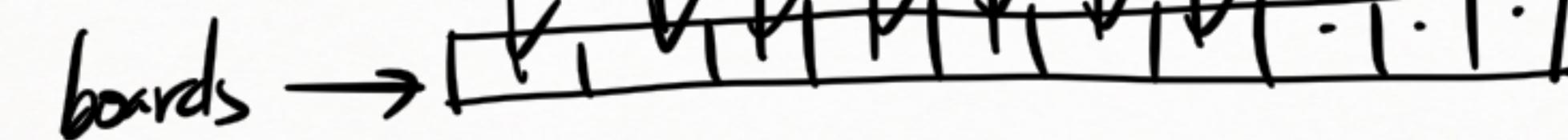
$$\underline{5/2} = \textcircled{2} \times$$



$$\underline{t+2} = 5$$

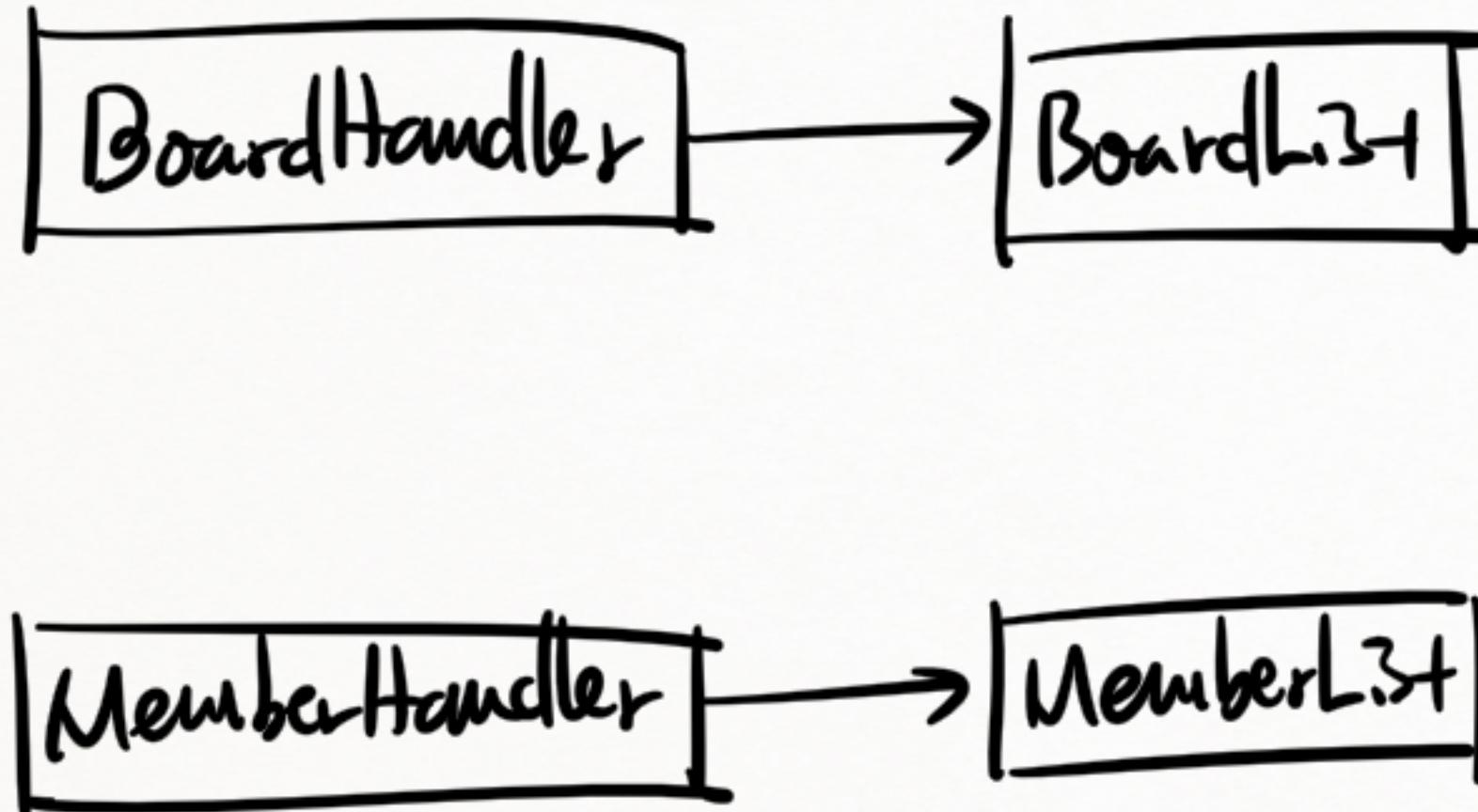


$$7 + \frac{(7/2)}{3} = 10$$

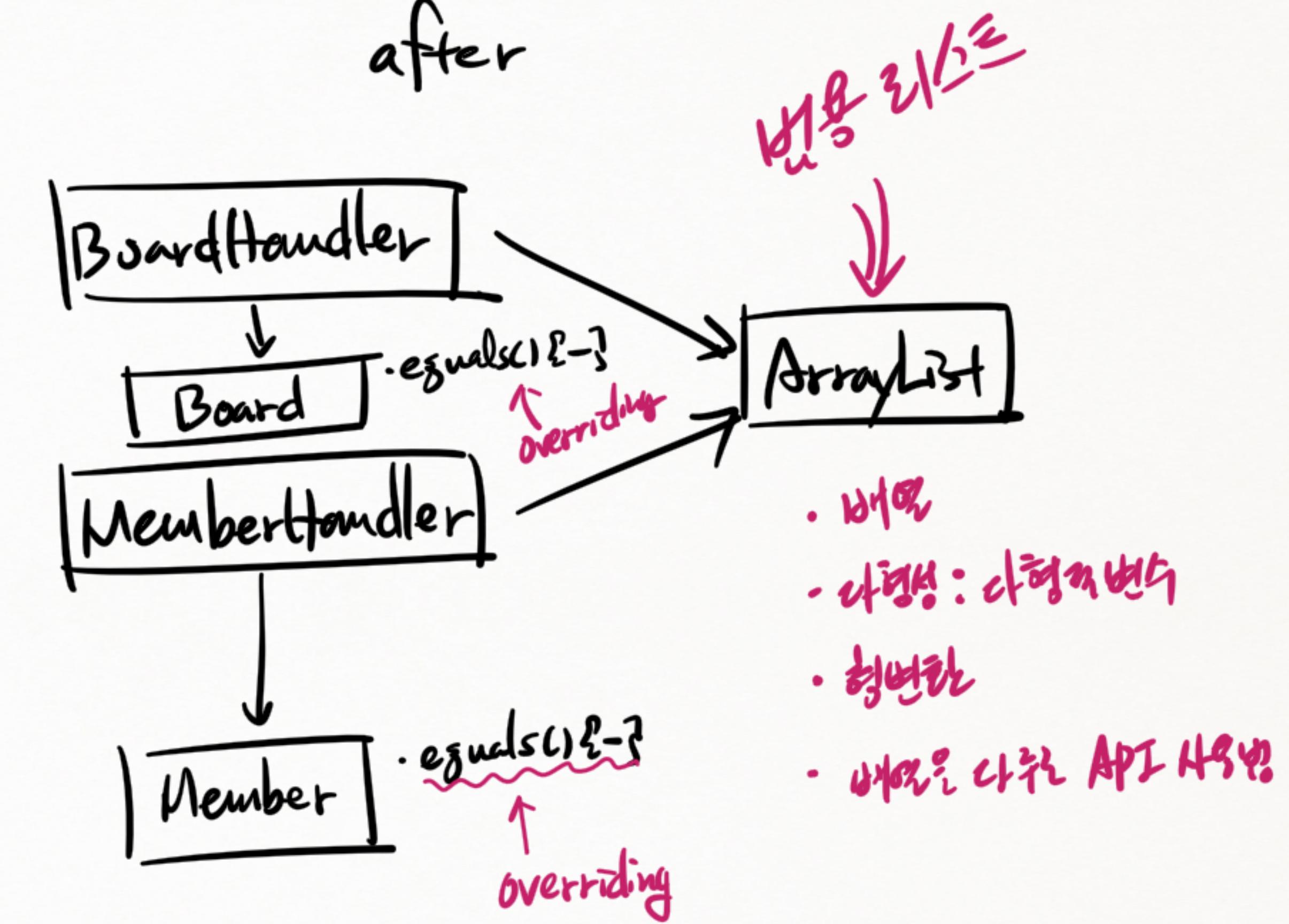


19. 범용리스트 만들기

before



after

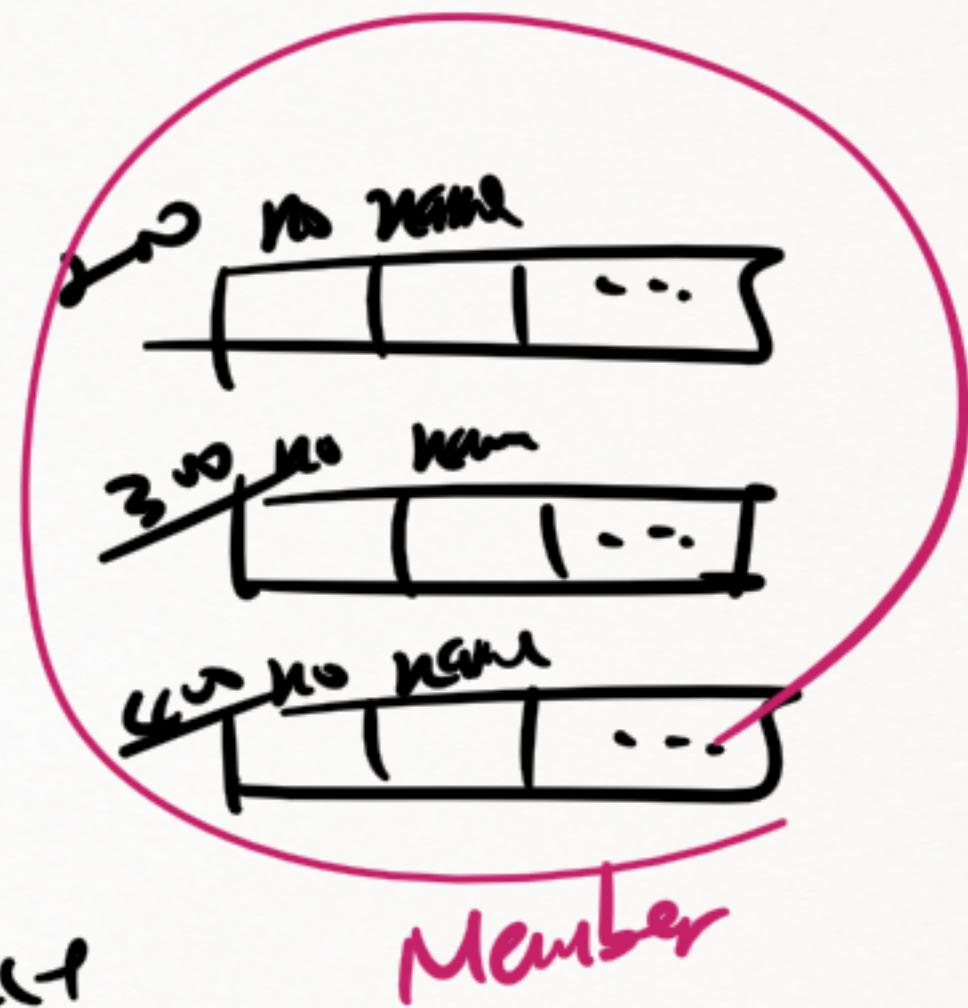
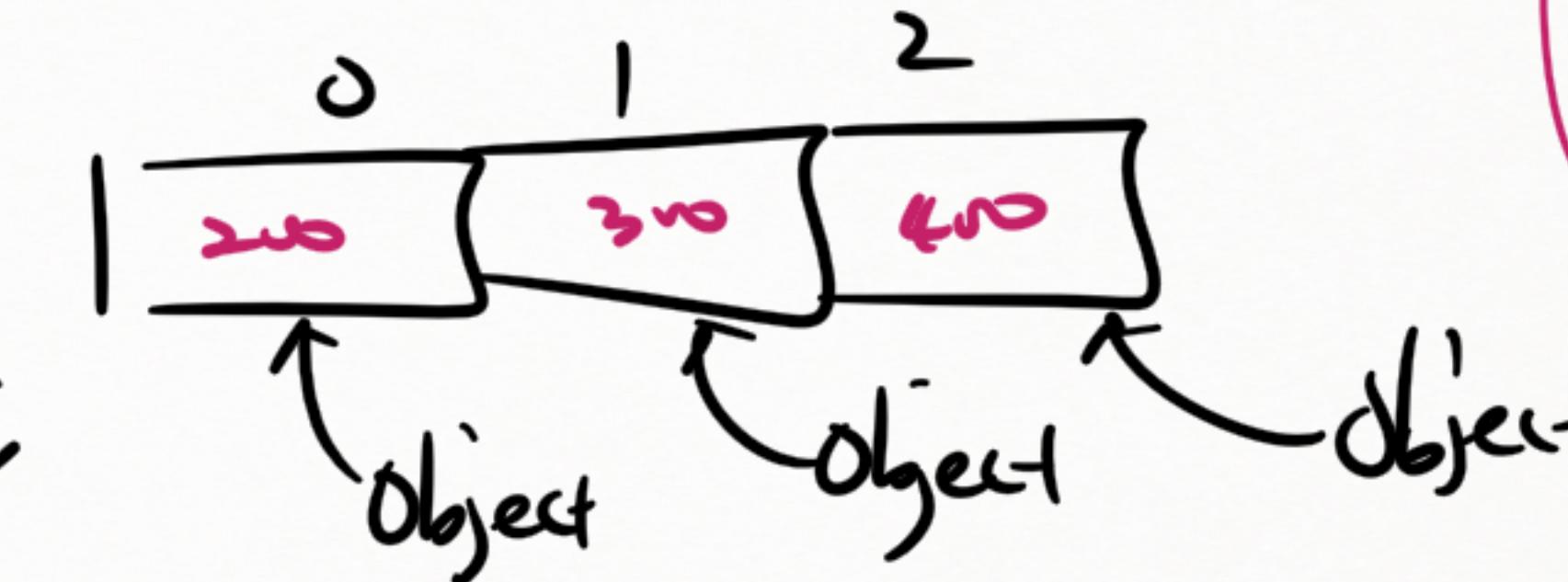


`Object[] arr1 = new Object[3];`

`arr1[0] = new Member();`

`obj1[1] = new Member();`

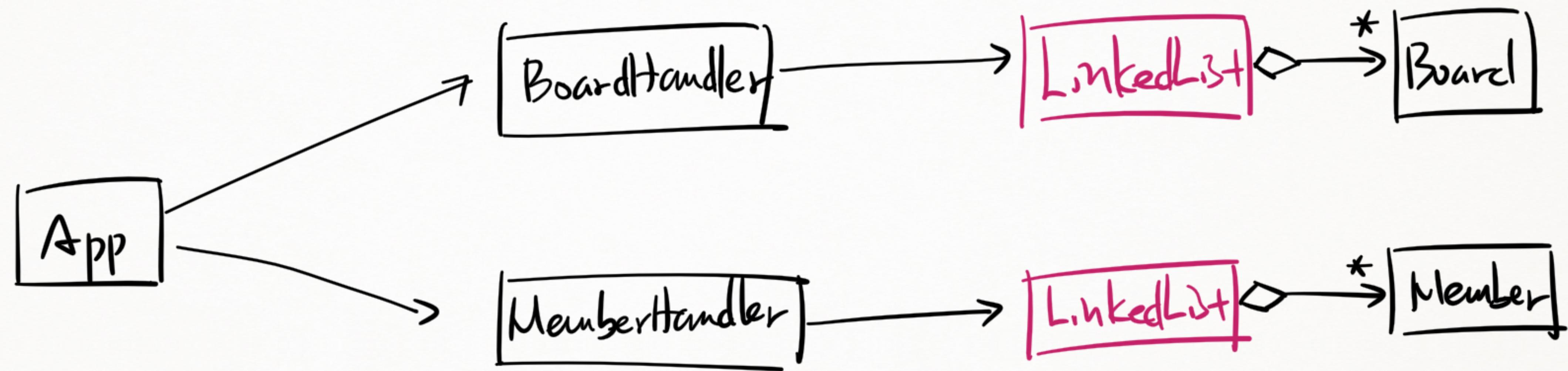
`obj1[2] = new Member();`



~~`Member[] arr2 = (Member[]) arr1;`~~

↑
arr1의 멤버 배열로 강제 타입 변환
Member 타입으로만 접근 가능.
Object 타입으로 접근하는 것은 가능.
Object 타입은 Member 타입을 확장한 타입이다.

* 20. LinkedList 자료구조 구현하기



ArrayList vs LinkedList

	ArrayList	LinkedList
추가/삭제	<u>일정 범위내</u> 랙보 (IMPO)	O 부터 시간
크기 증가	Yes → 가로세로로 가로나가면서	Yes → 가로이자 줄다.
검색	인덱스로 조회 (LinkedList는 불가능)	링크를 따라가야 한다 (ArrayList는 가능)
교환, 삽입, 삭제	삽입 → 배열값 옮기기 삭제 → 배열값 옮기기 <i>(LinkedList는 가능) (불가)</i>	해당 값을 찾고자 노드 추가/삭제 <i>(List는 가능) (불가)</i>

LinkedList - add()

tail 3100

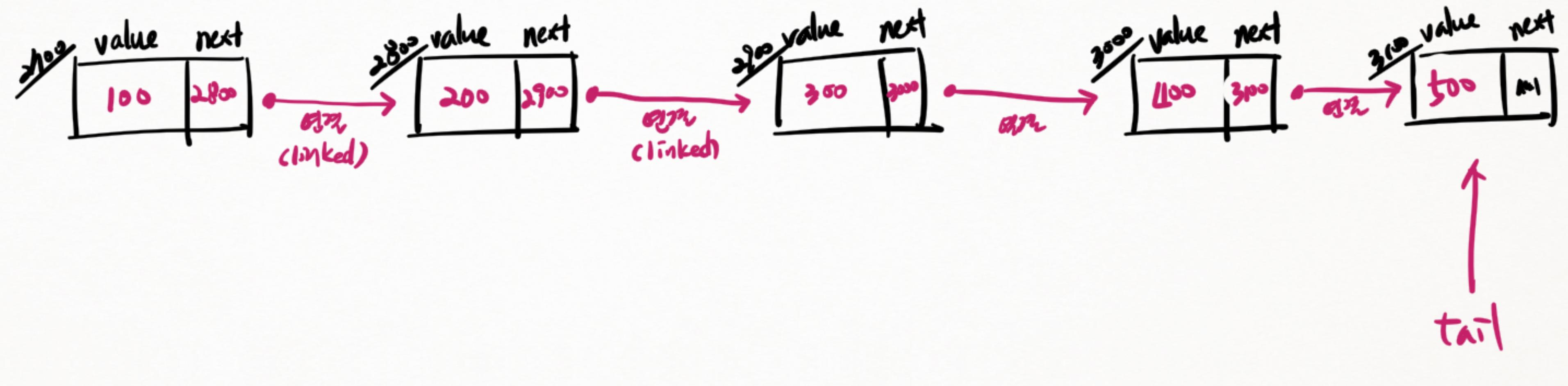
add(100);

add(200);

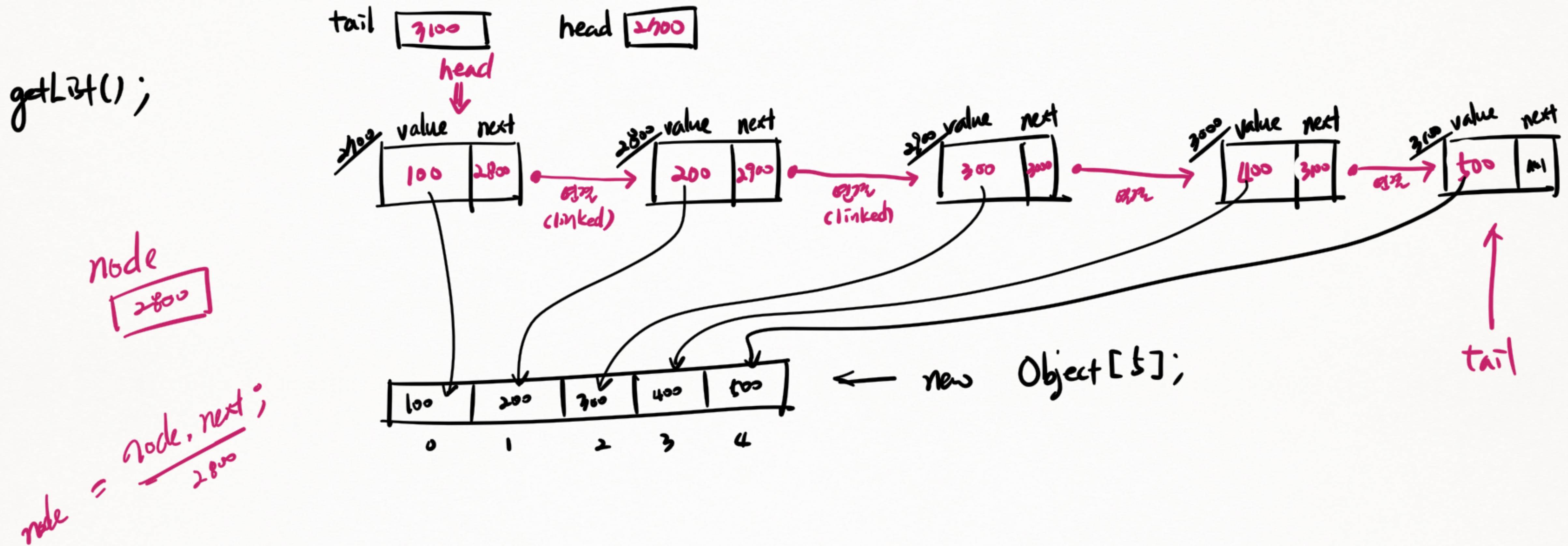
add(300);

add(400);

add(500);



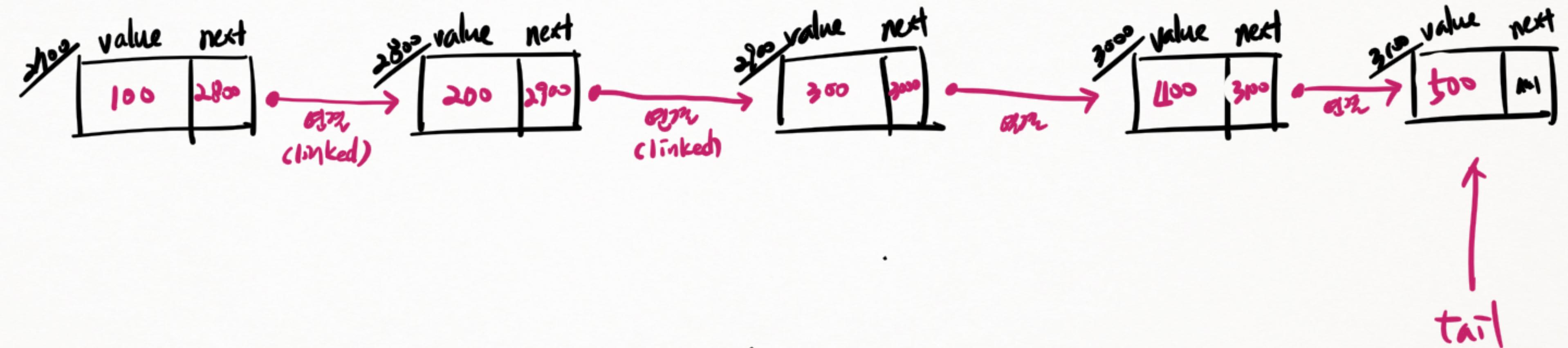
LinkedList - getList()



LinkedList - retrieve()

tail 3100

retrieve(100);



cursor

null

~~500.equals(100)~~

↑
tail

↑

cursor

LinkedList - remove(): 중간 항목 삭제

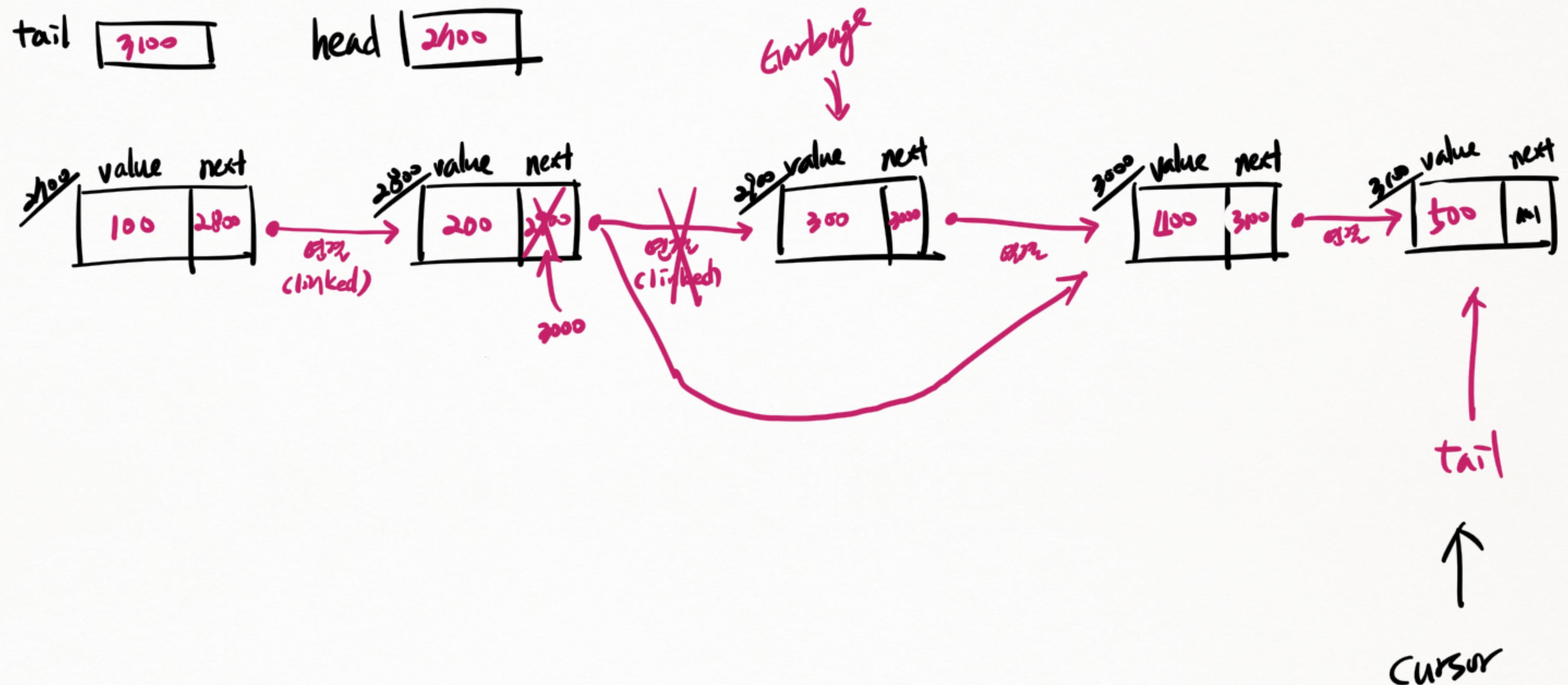
remove(300)

tail [3100]

head [2100]

prev [2800]

cursor [2900]



LinkedList - remove(): 정간, 흉목, 뒤지 \Rightarrow 가비지가 인스턴스를 가리킬 때 문제점.

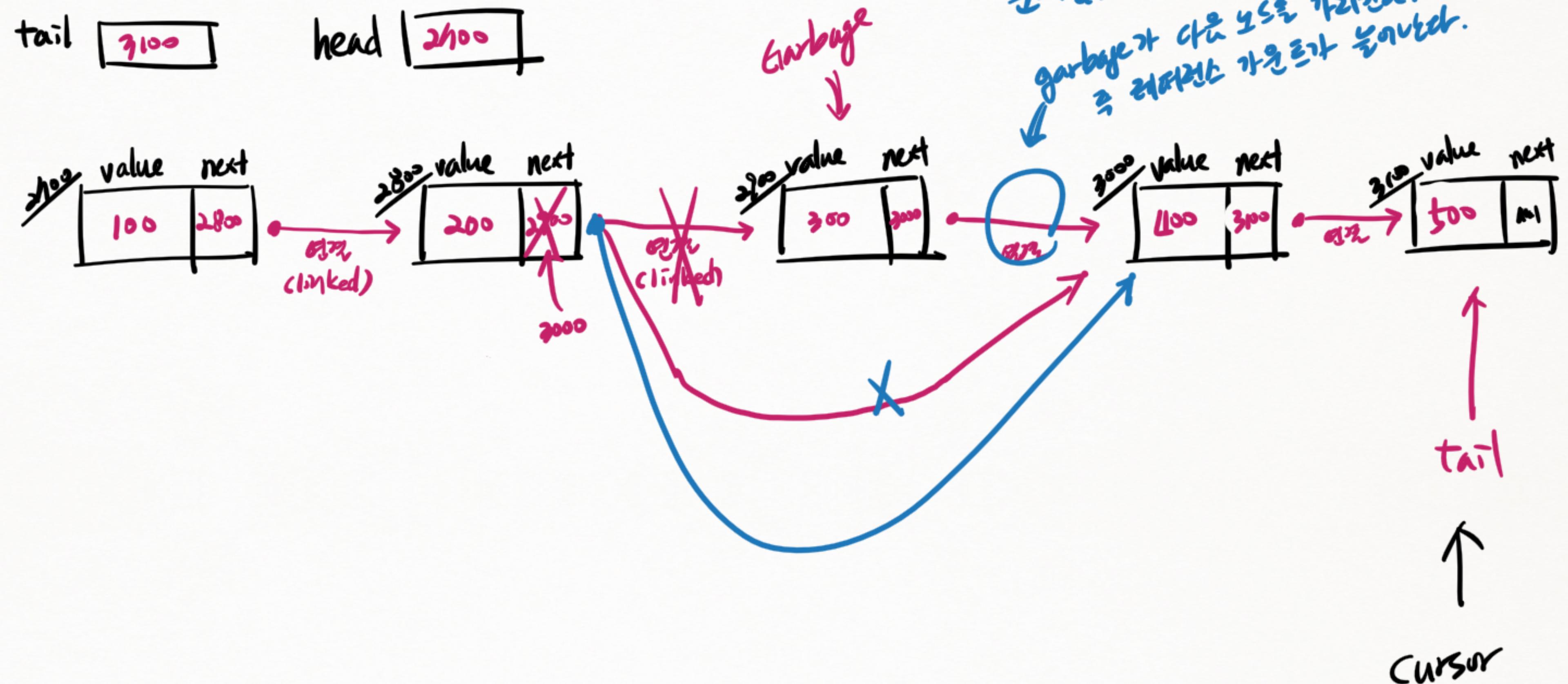
remove(300)

tail
3100

head
2100

prev
2800

cursor
2900



LinkedList - remove(): 정간, 흐름, 노드 ⇒ 가비지가 인스턴스를 가리킬 때 문제점.

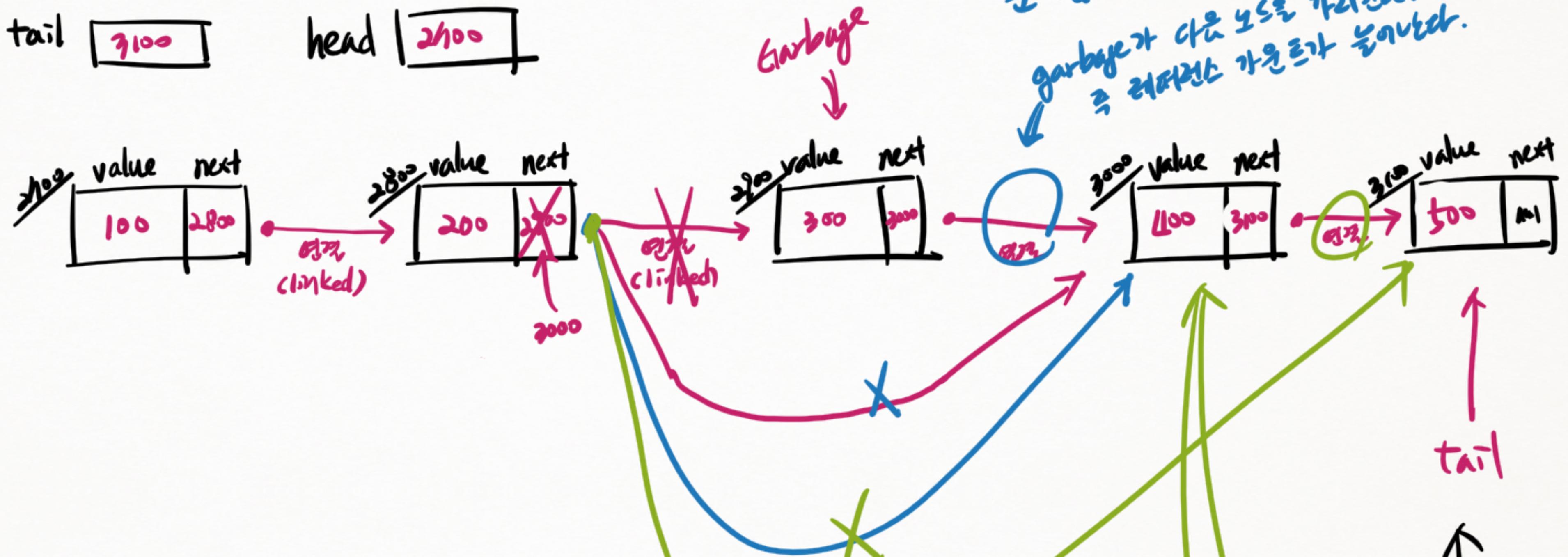
remove(300)

tail
3100

head
2100

prev
2800

cursor
2900

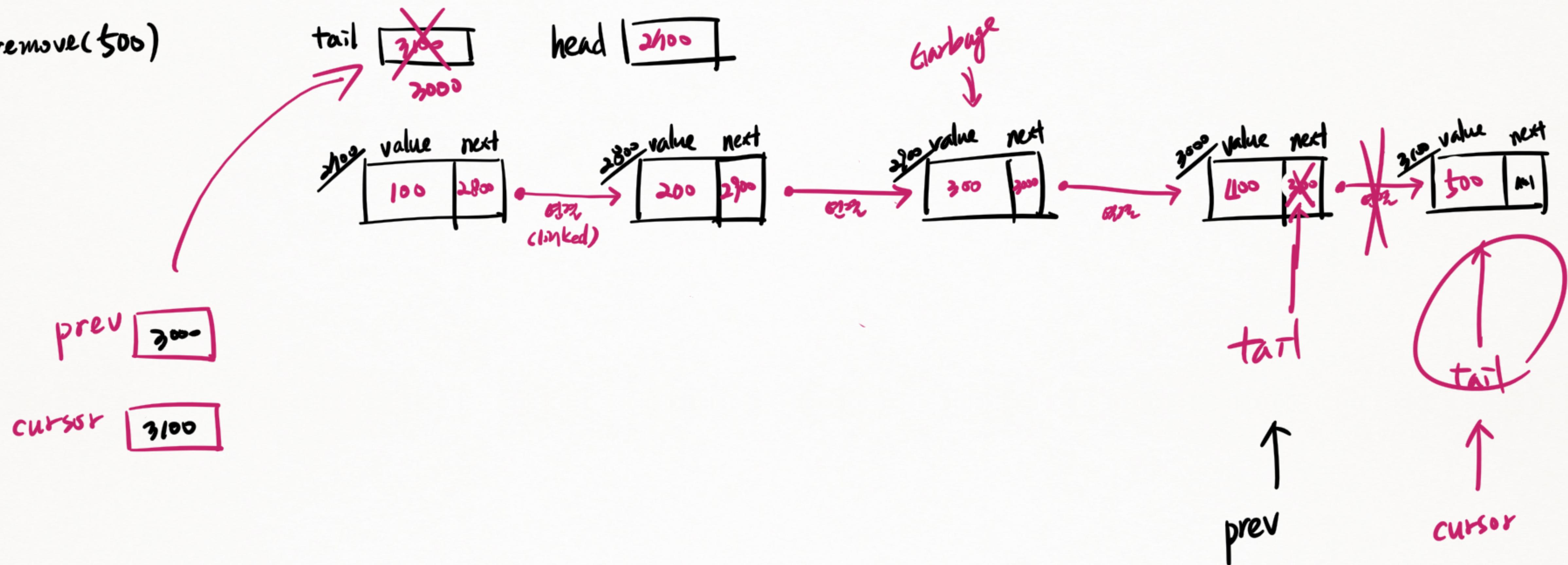


이 노드는
가비지가 되었나?
아? 이전에 속해있던 노드에서
이 노드를 가리키고 있다.

CURSOR

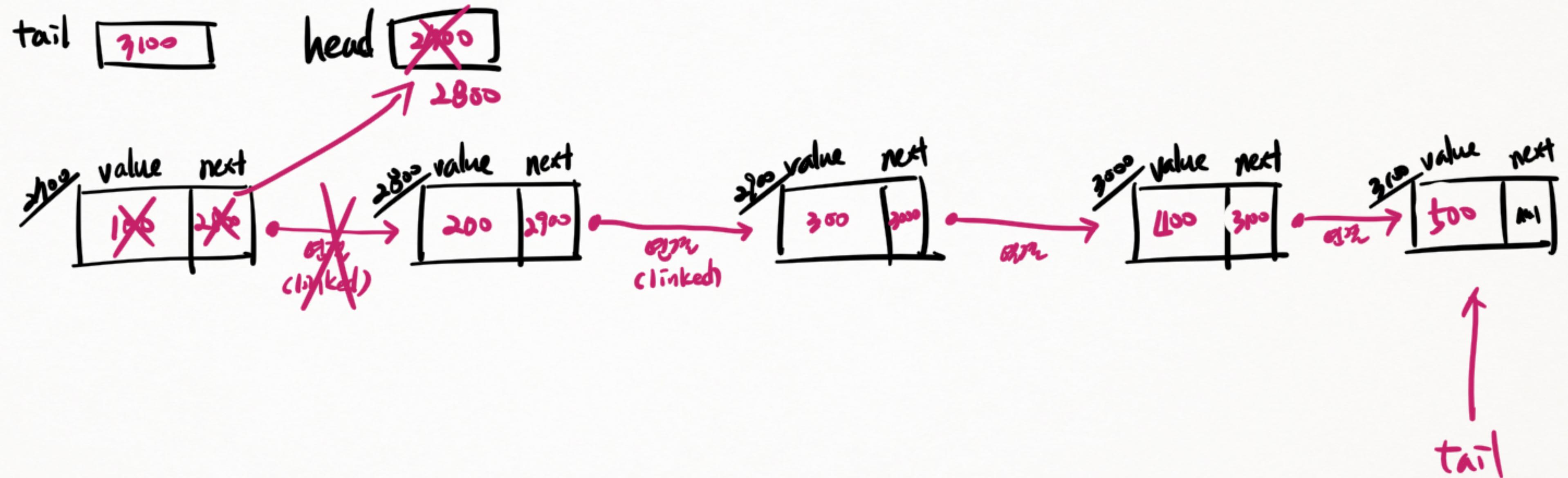
LinkedList - remove() : 끝 항목 삭제

remove(500)



LinkedList - remove() : 시작 노드

remove(100)

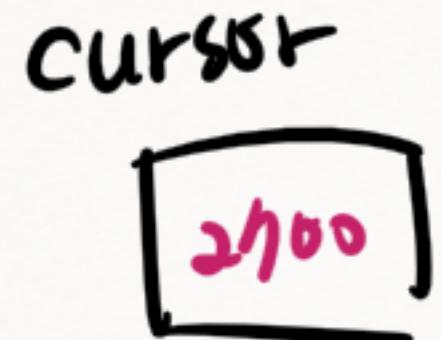
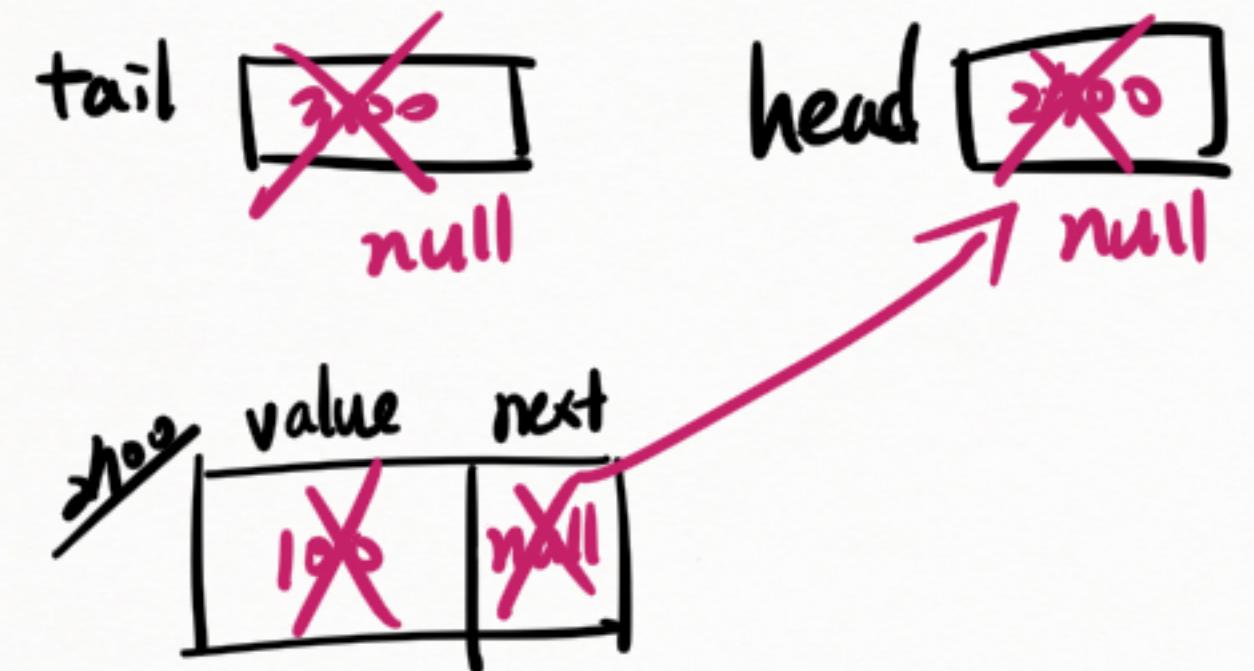


prev
[null]

CURSOR
[2700]

LinkedList - remove() : 시작노드 + 끝노드

remove(100)



21. 인터페이스로 목적관리 가능한 사용법을 기반으로 정의.

