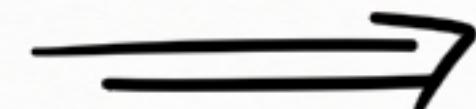
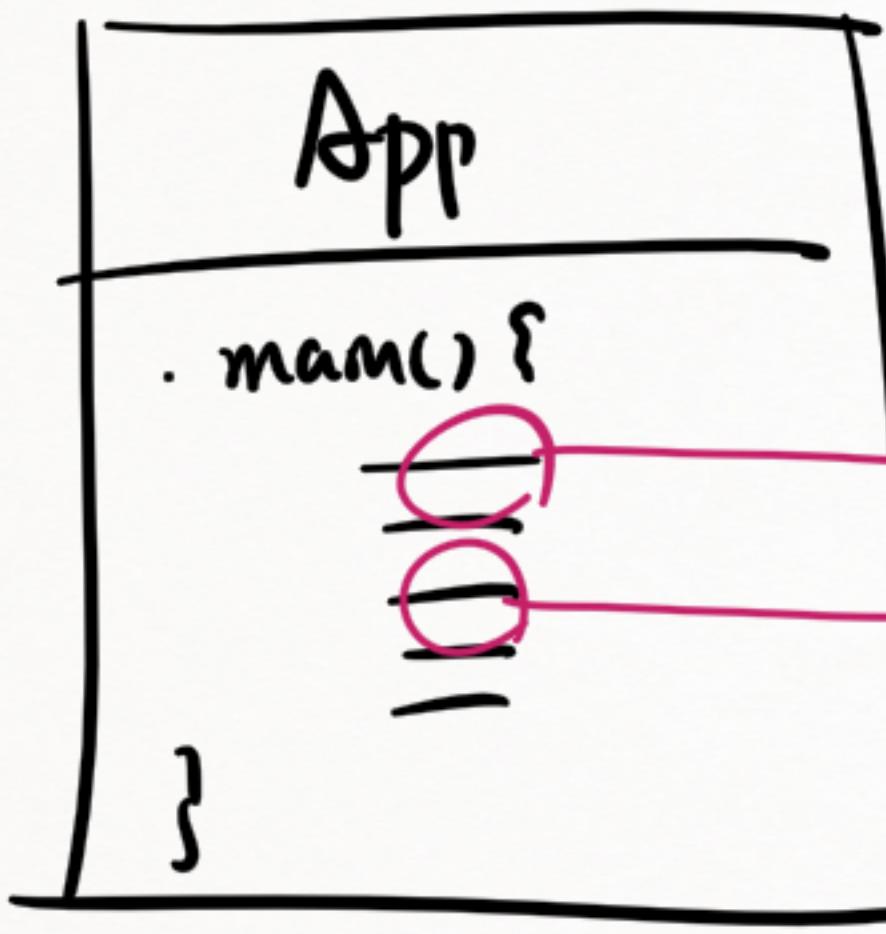
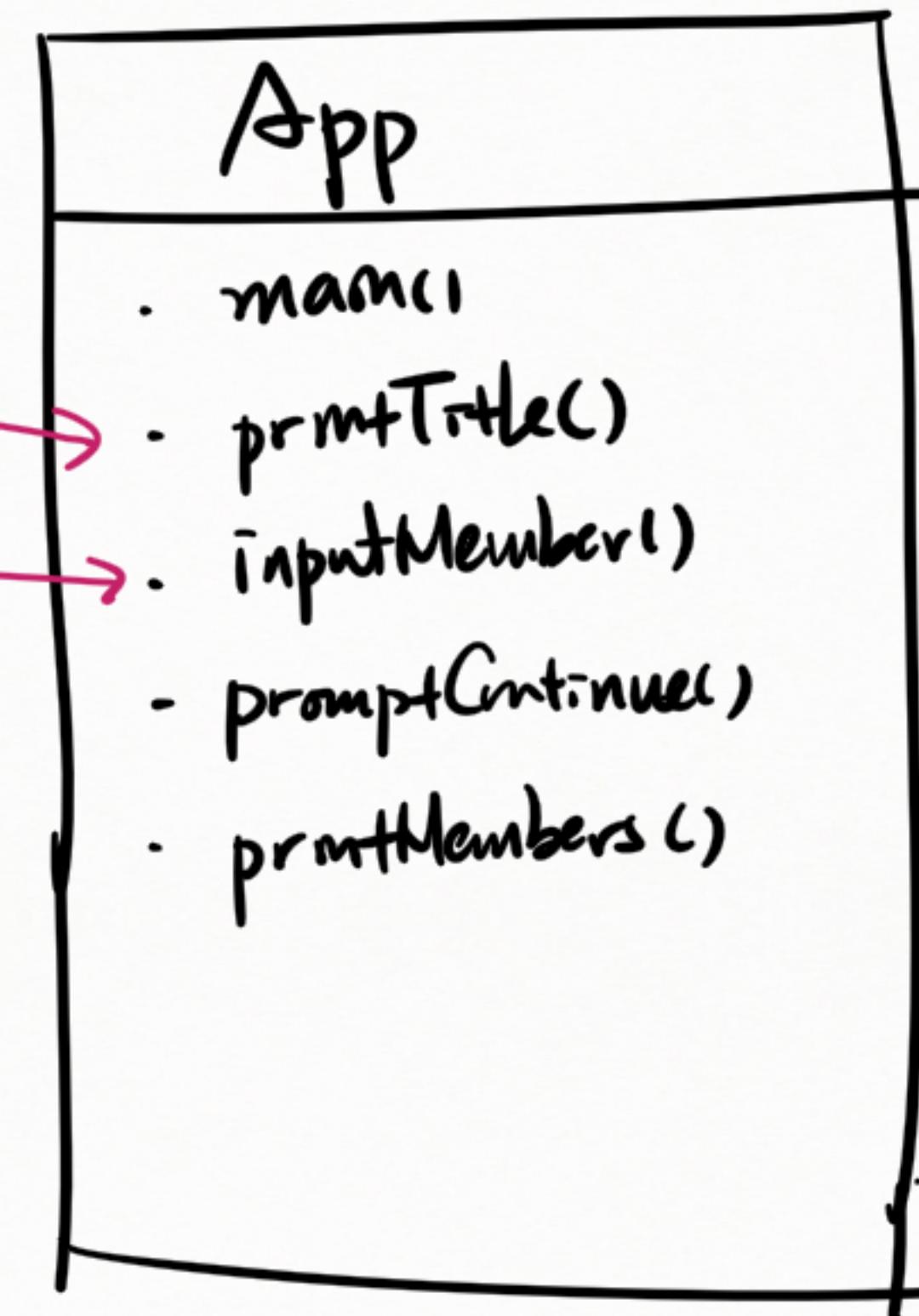


\* 1. 디렉트 사용법

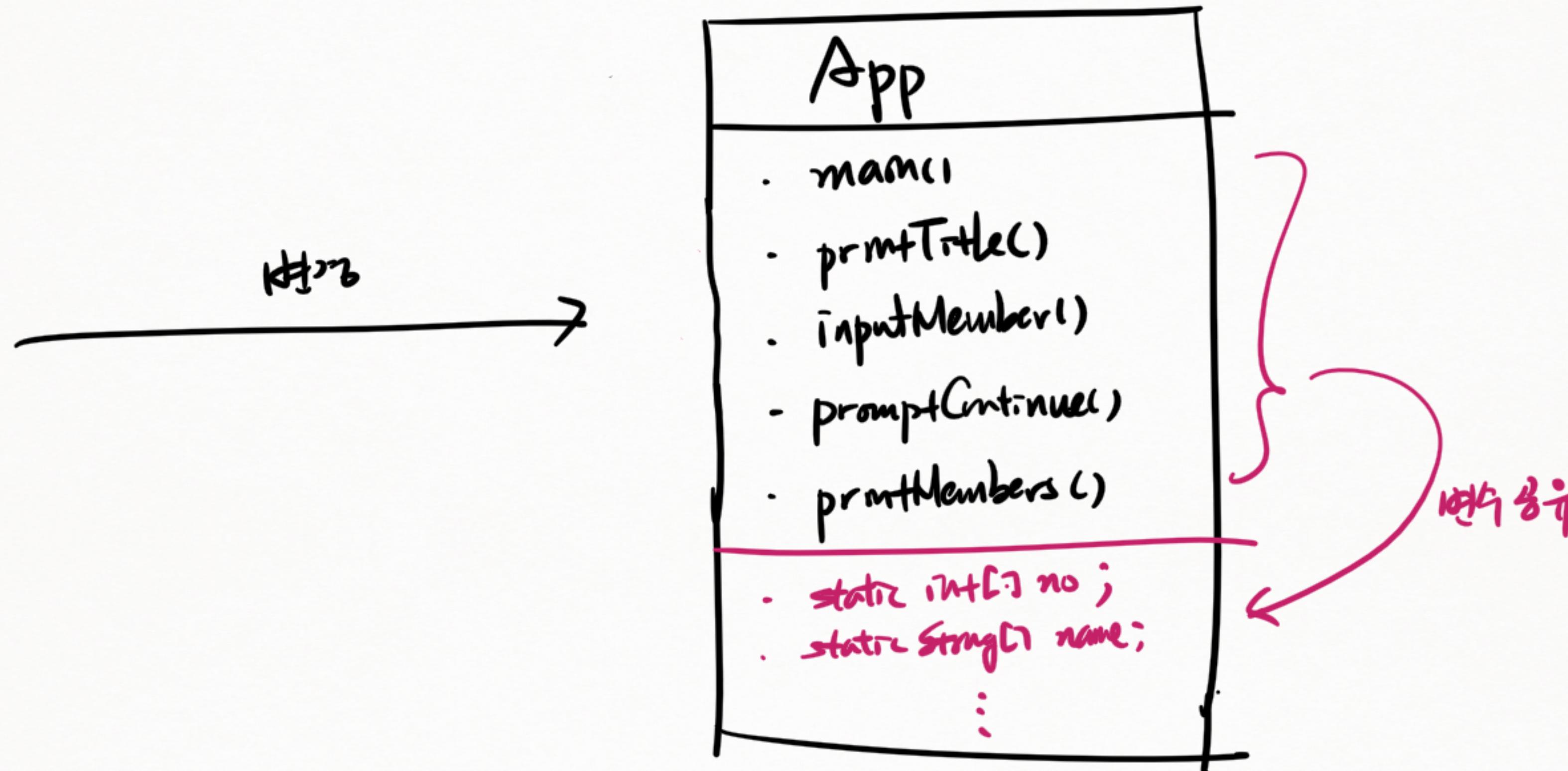
이전



변경

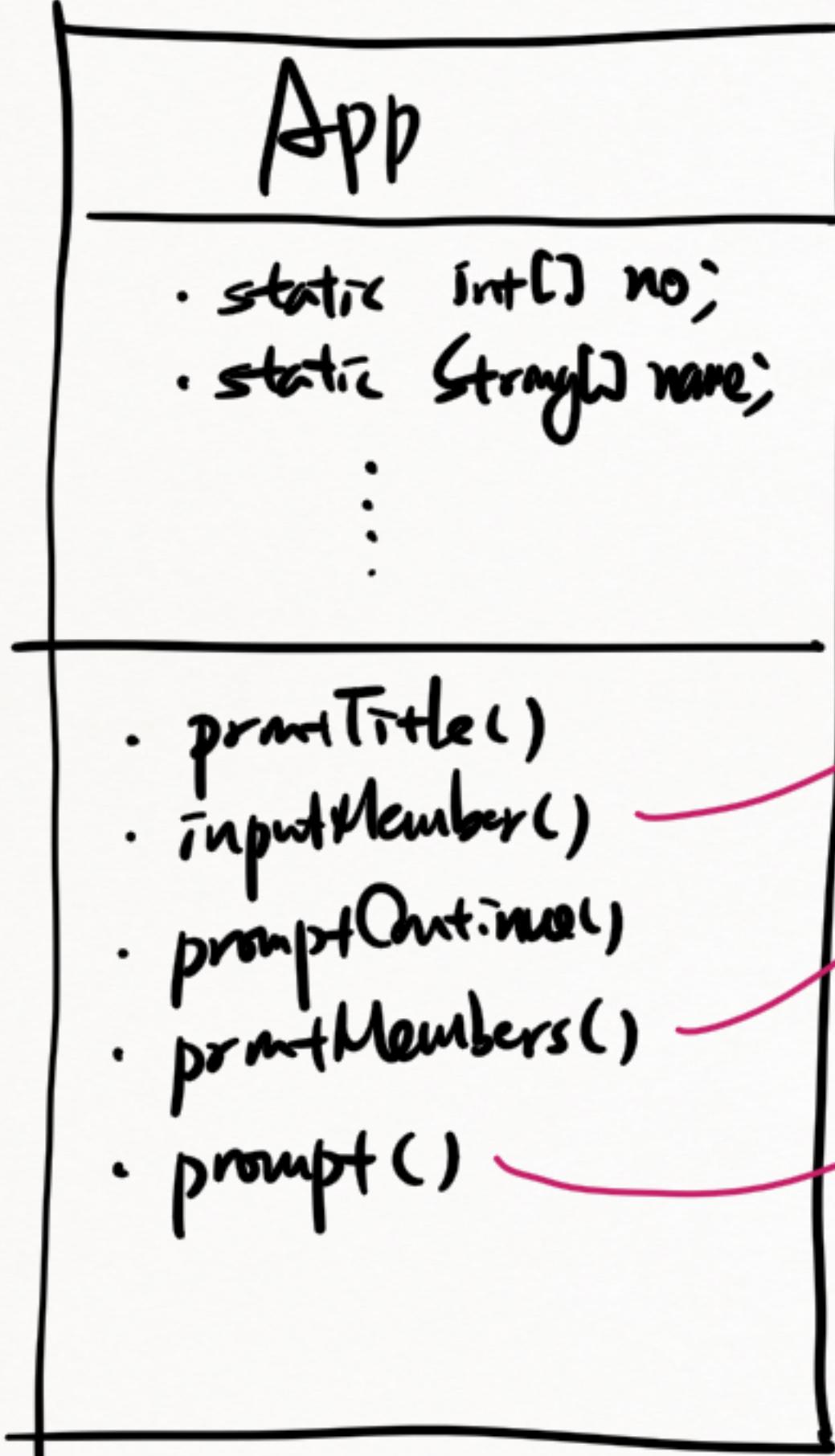


## \* Q. Lession with Array

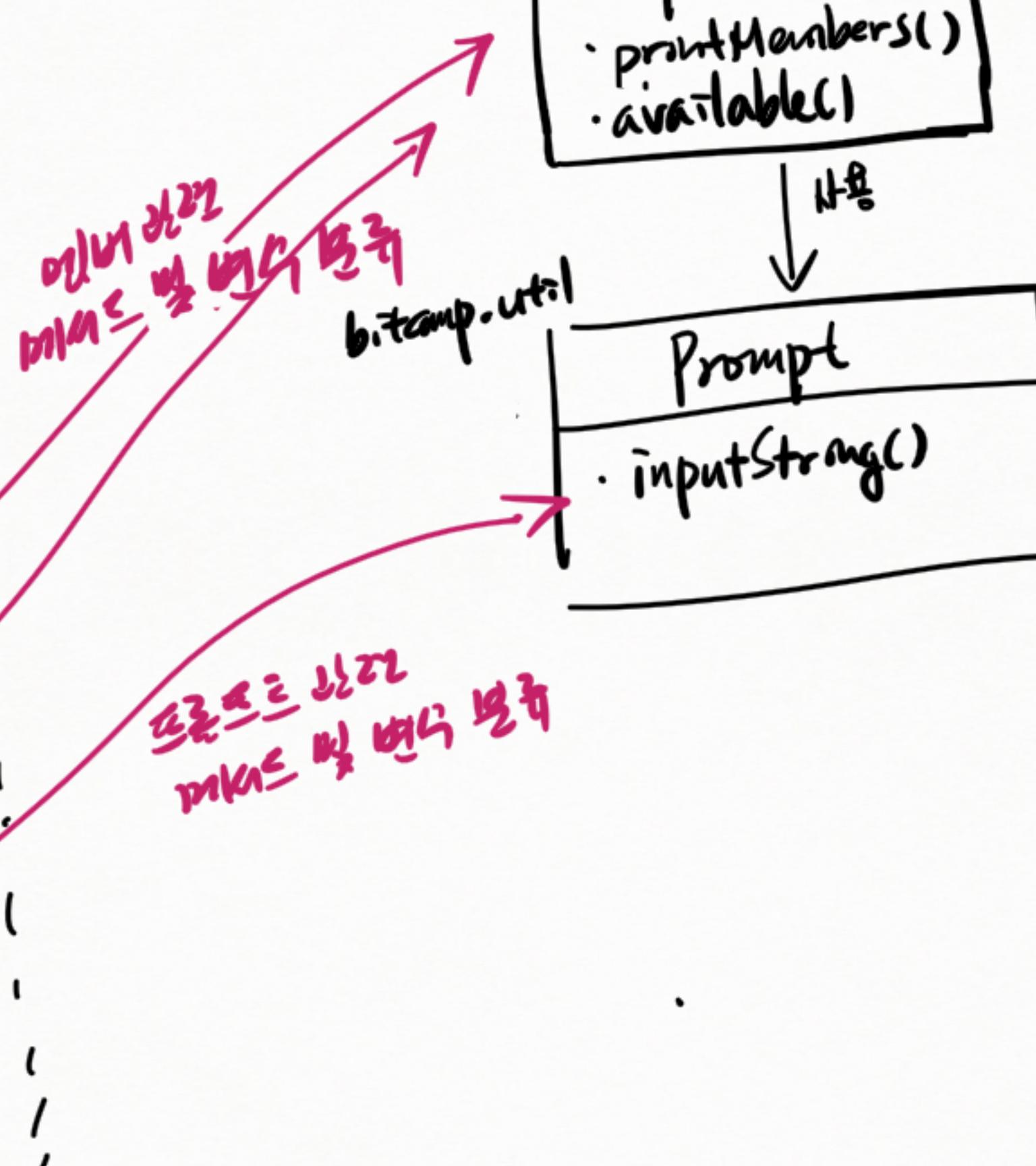
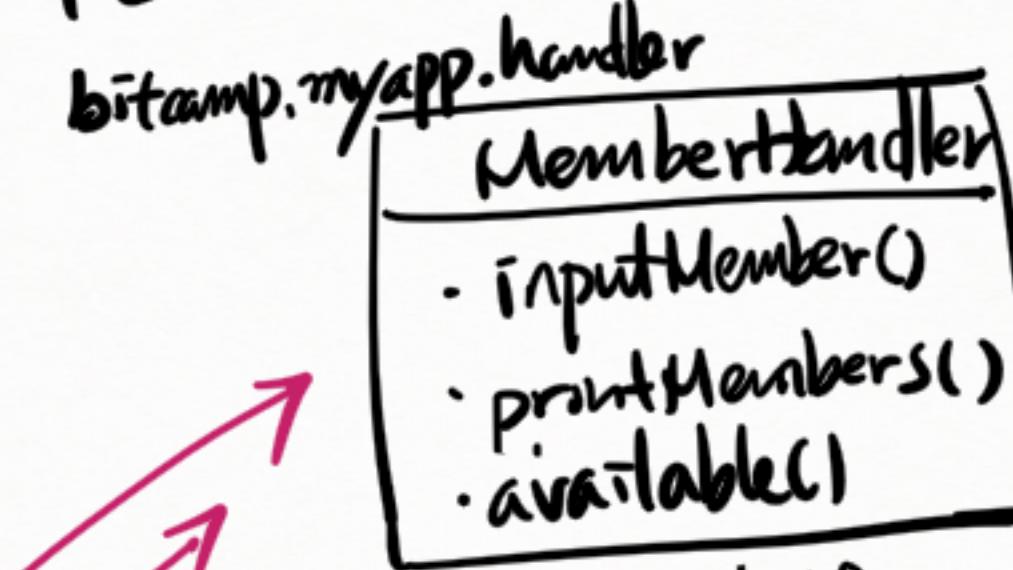


## \* 9. 클래스 및 패턴 학습

이전 구조  
~  
Architecture

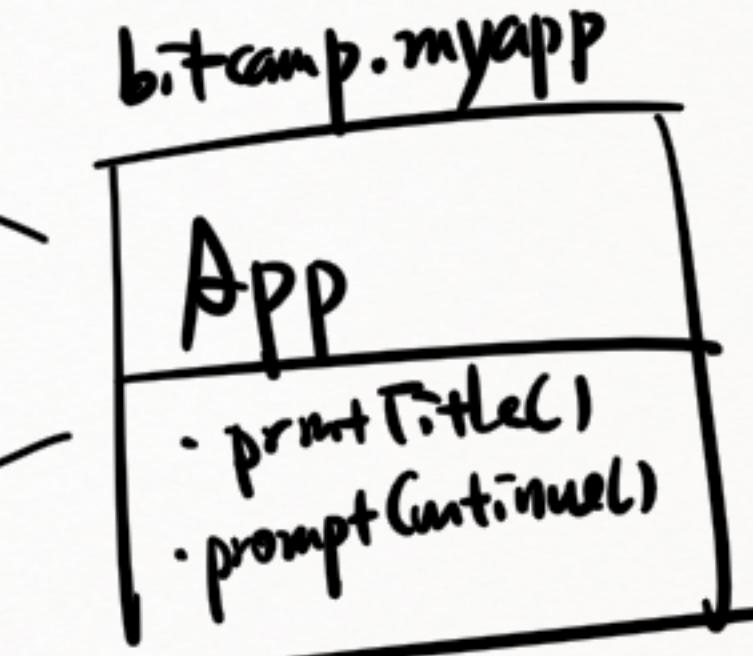


내 구조



MemberHandler를  
인스턴스에  
따라  
분류 →  
이유?  
유지보수를  
쉽게.

(  
다른  
속도를  
얻을  
수 있다.  
메모리  
节约.  
})

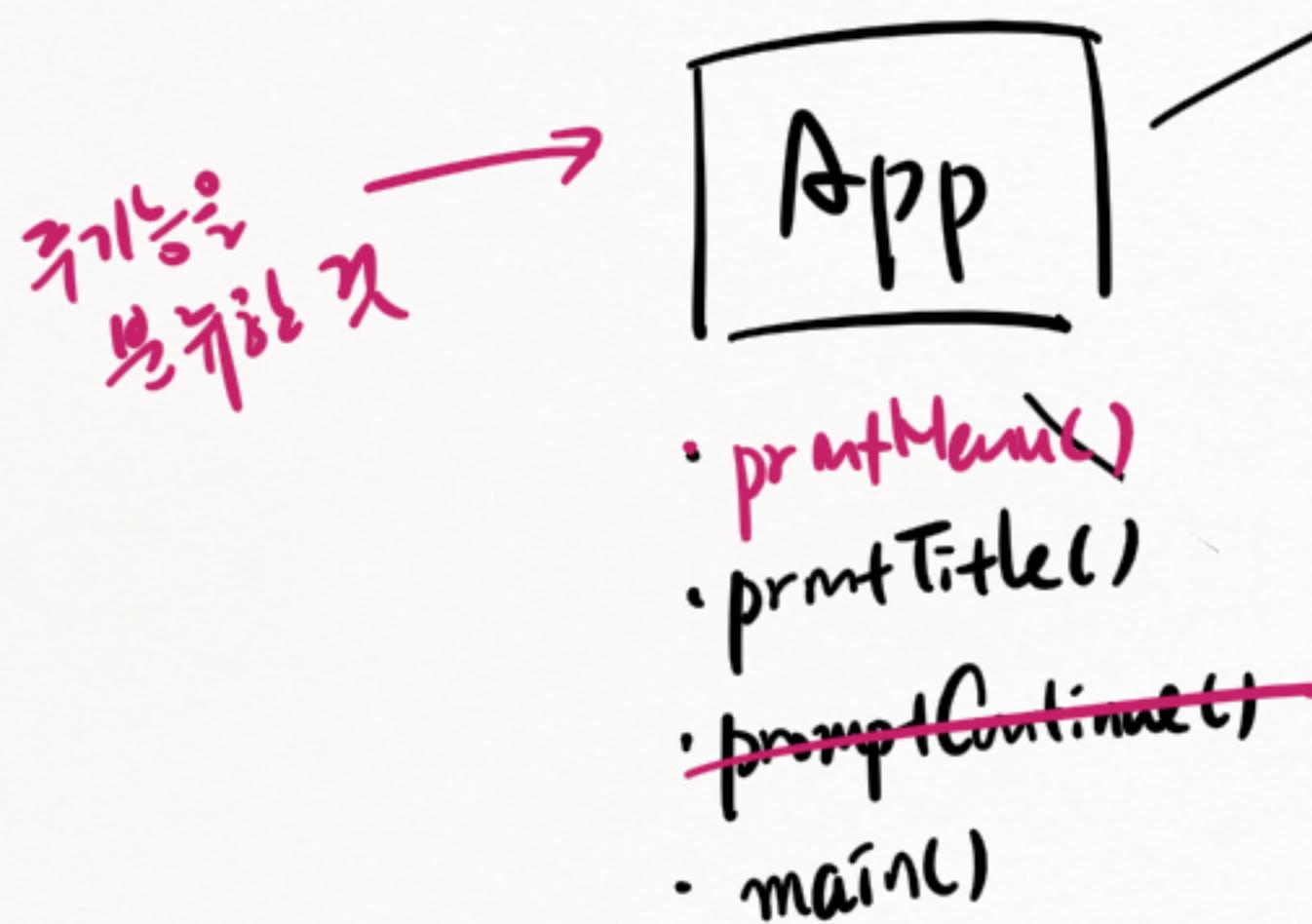


내부  
비용  
절감  
→  
H/W  
환경으로  
제작

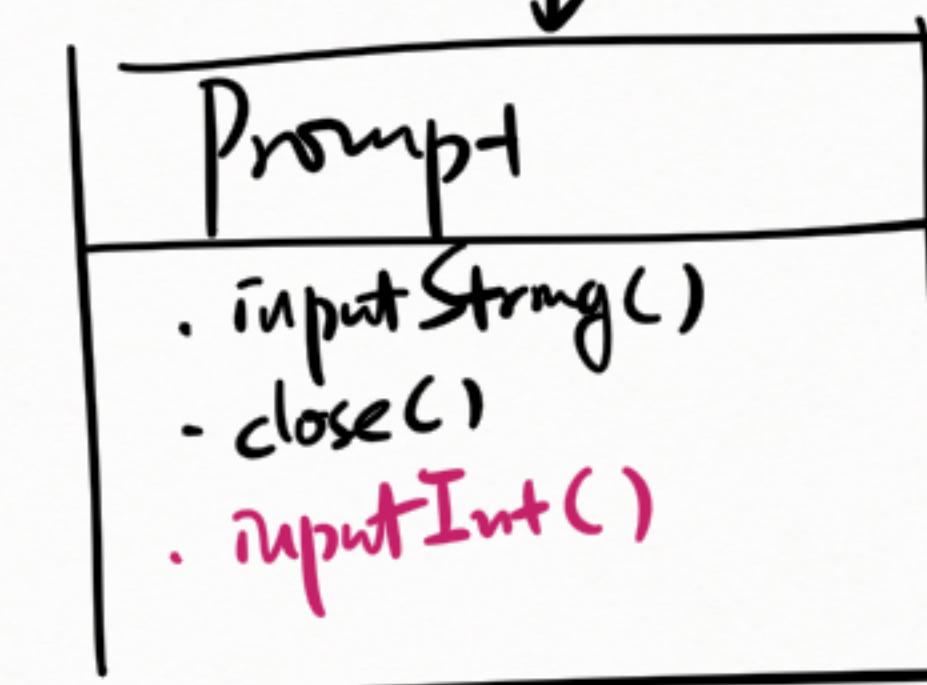
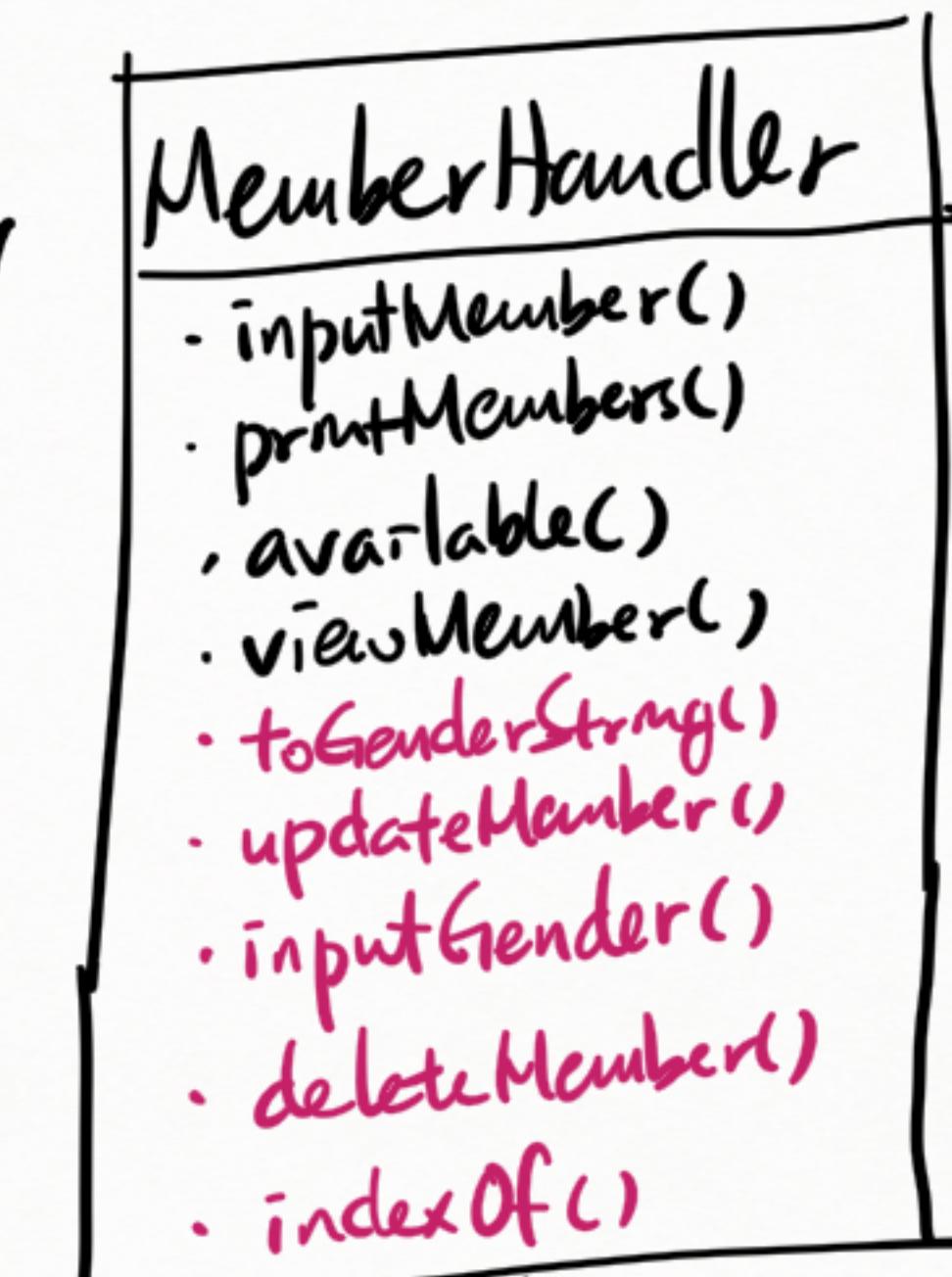
## \* 10. 멤버 및 CRUD 구현

\* 클래스  
↳ 애플리케이션 메서드를 분류할 것

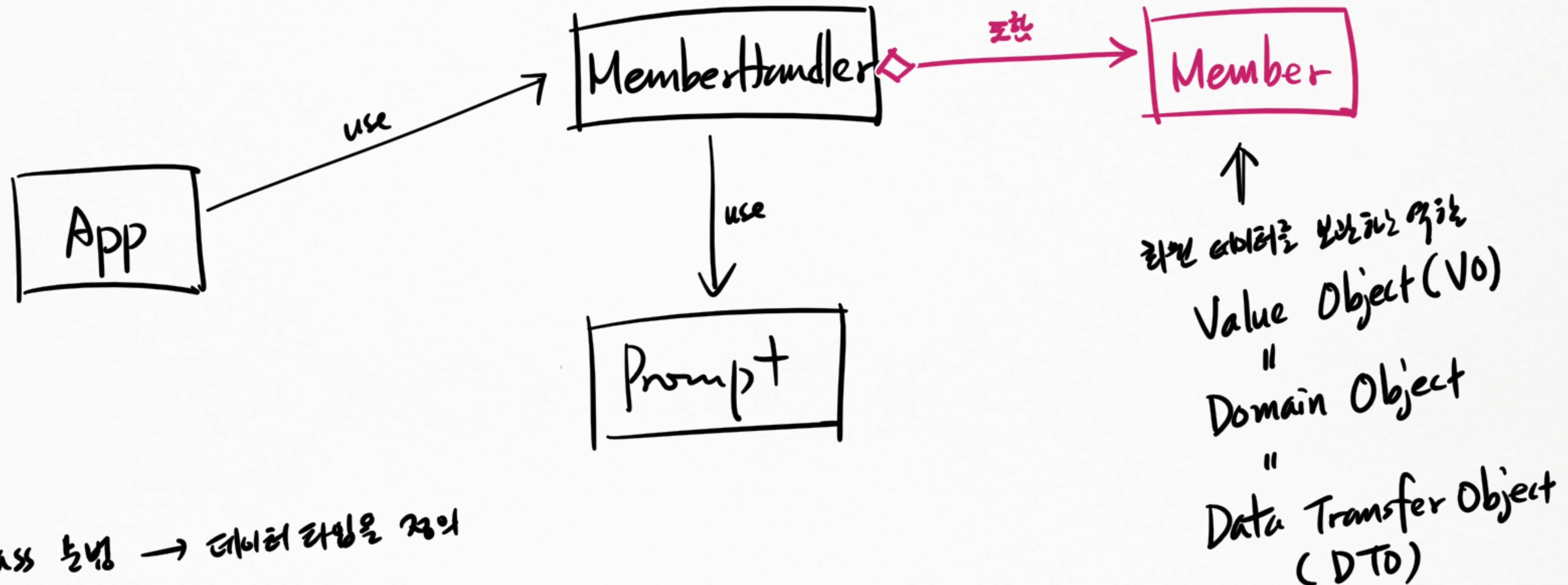
\* 패키지  
↳ 클래스를 분류할 것.



class 분류 → 메서드를 찾는 용도

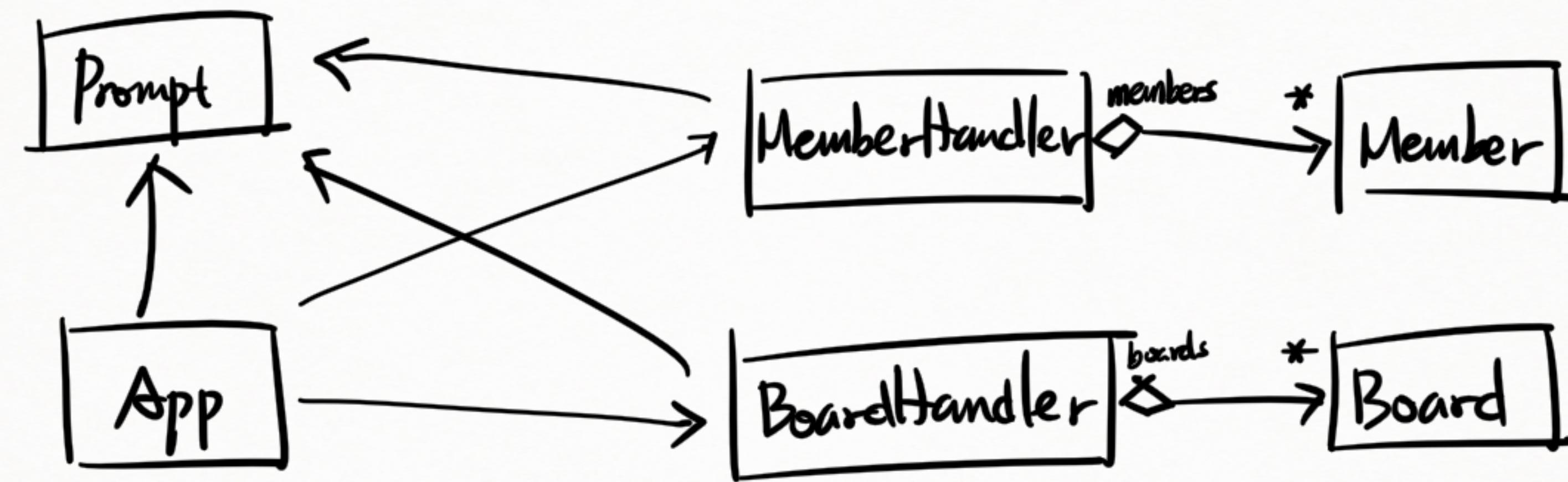


## 11. 사용자 정의 데이터 타입 만들기

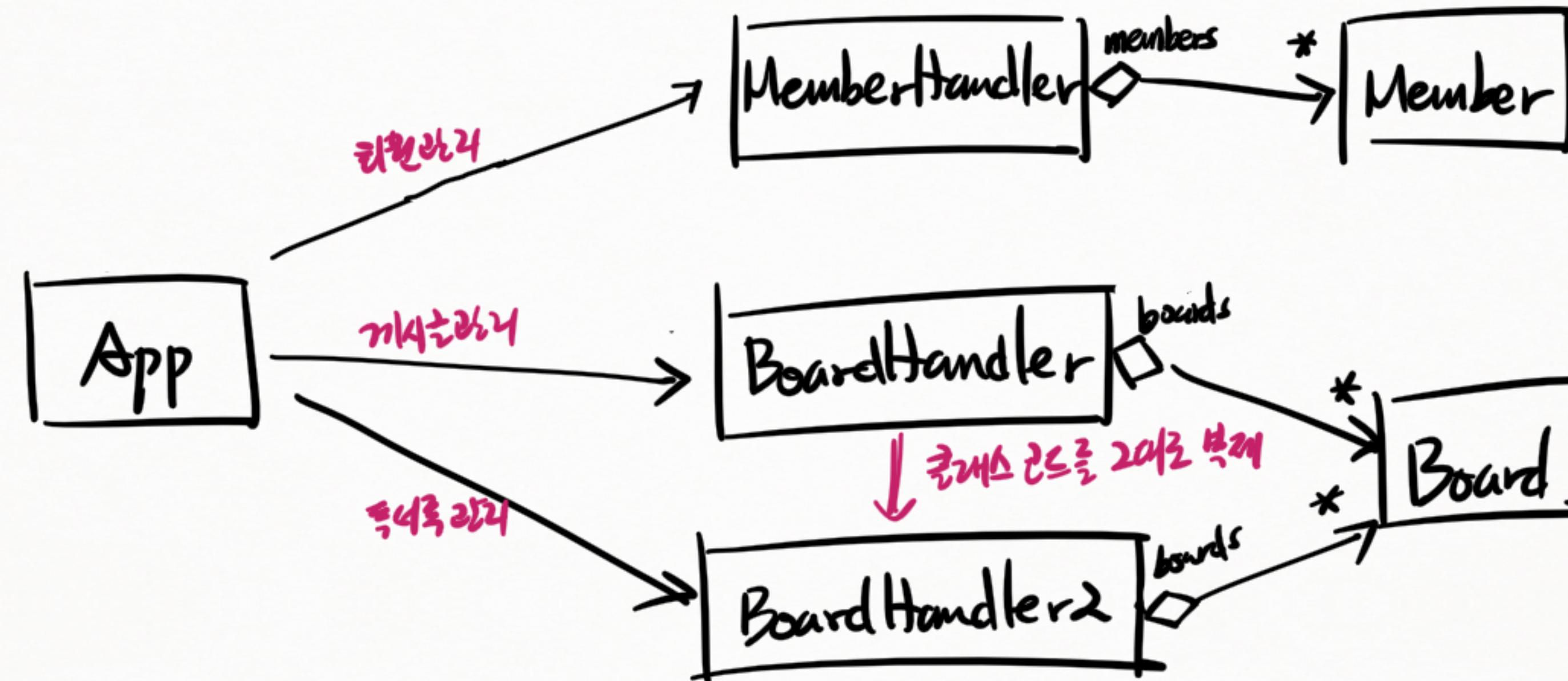


\* class 늘기 → 데이터 타입을 확장

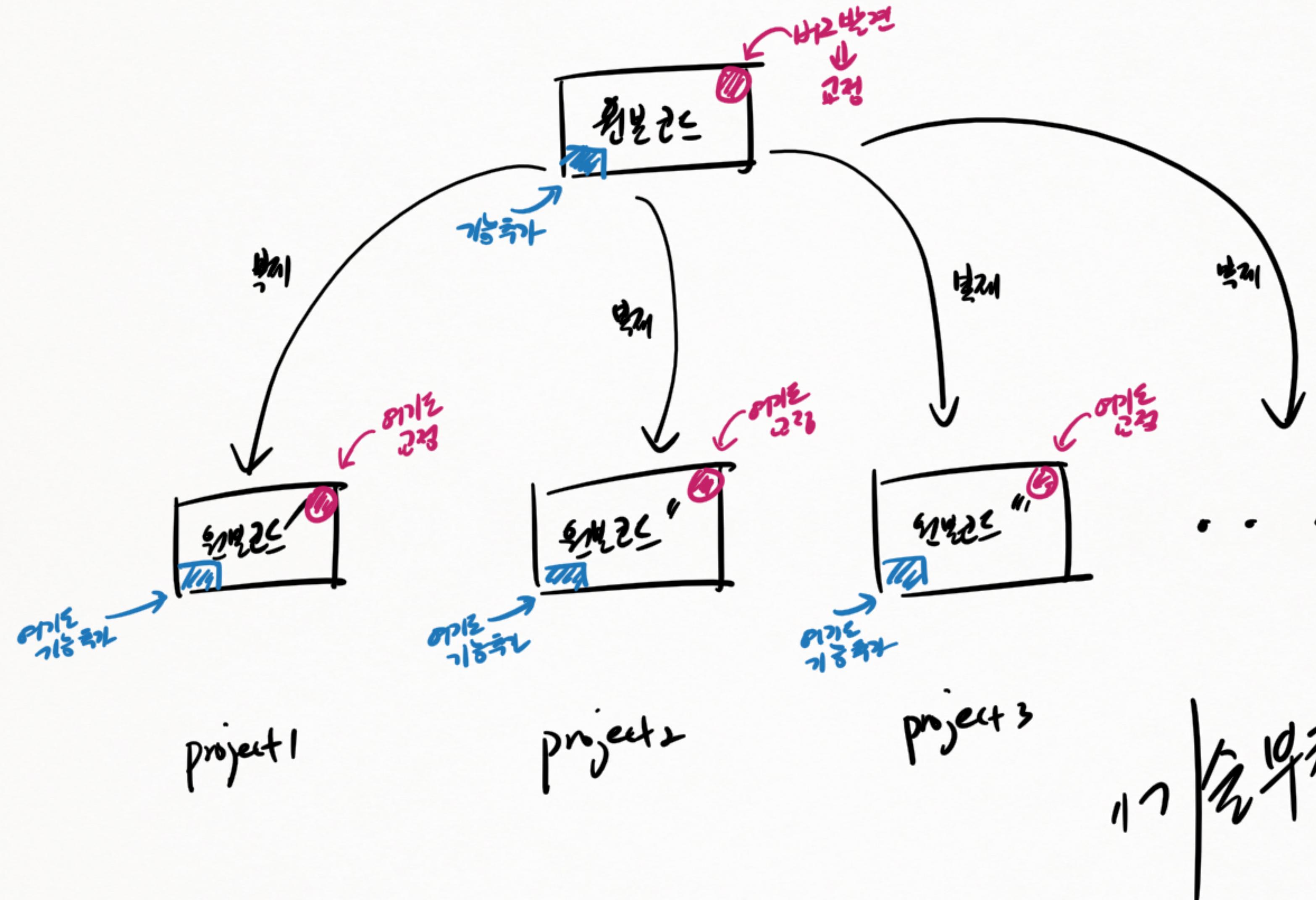
### 13. 깃허브 CRUD 추가



## 14. 토커를 CRUD 추가



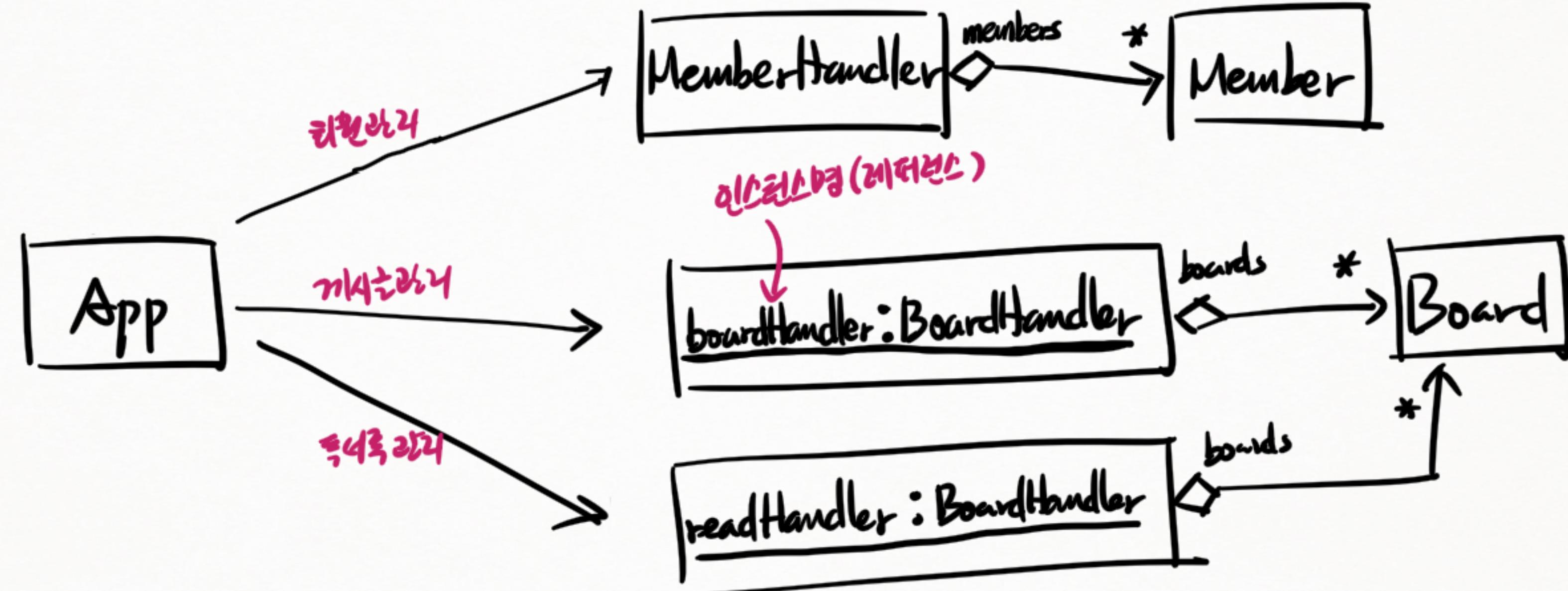
\* 가로관 복제로써 새기능을 주입하는 예



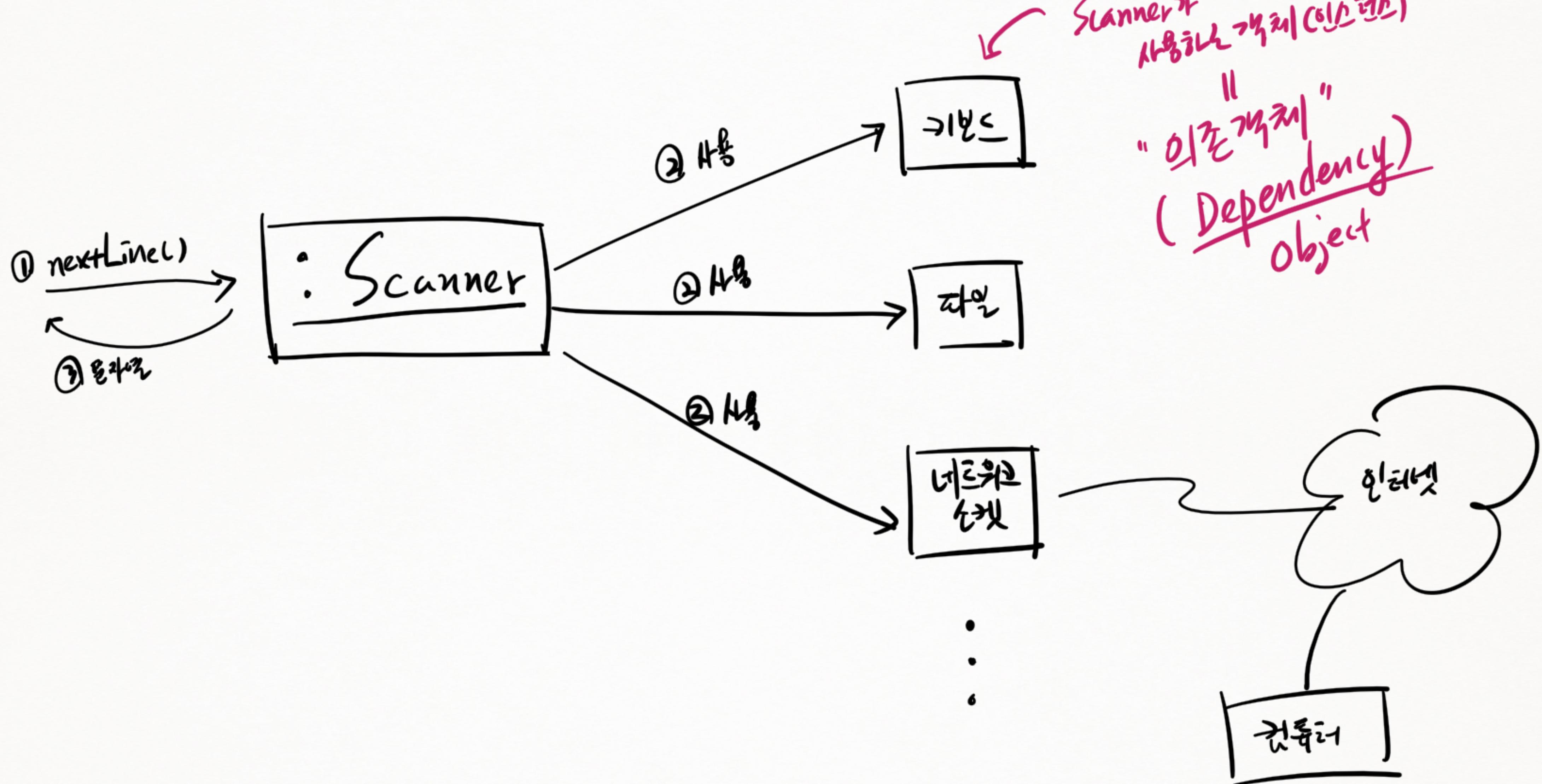
\* API를 고정하여 기능을 추가하여 API를 복제한 코드의 디자인은 동일한 일을 수행하는 코드다

코드는 복제이며 유지보수가 어렵습니다.

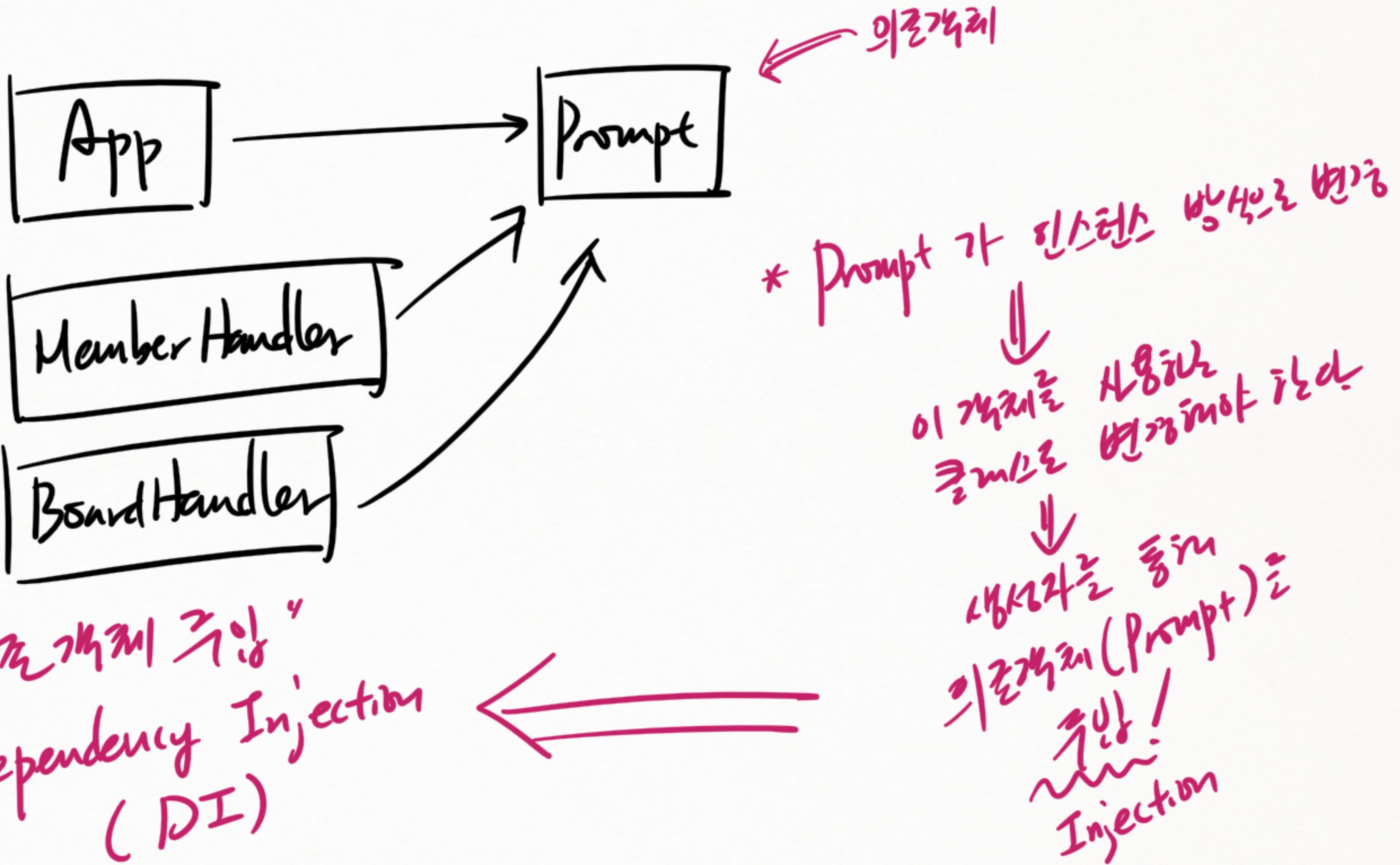
## 15. 인스턴스 있는 디자인 패턴



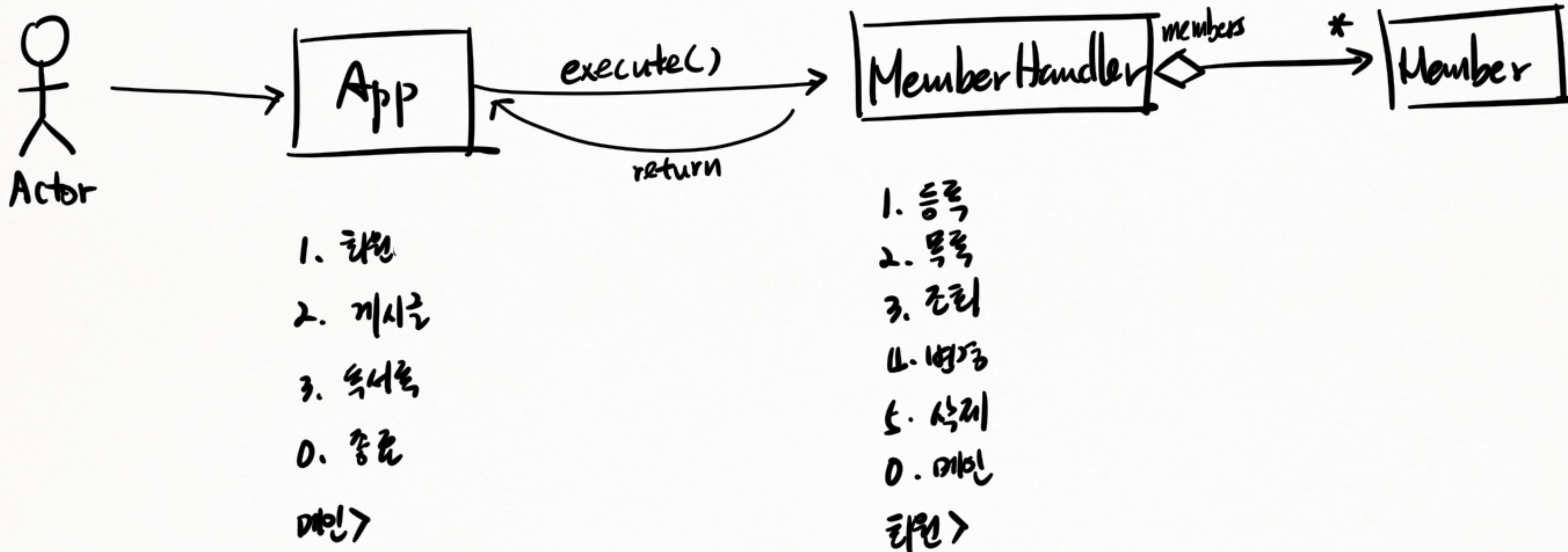
## Scanner 와 의존 객체



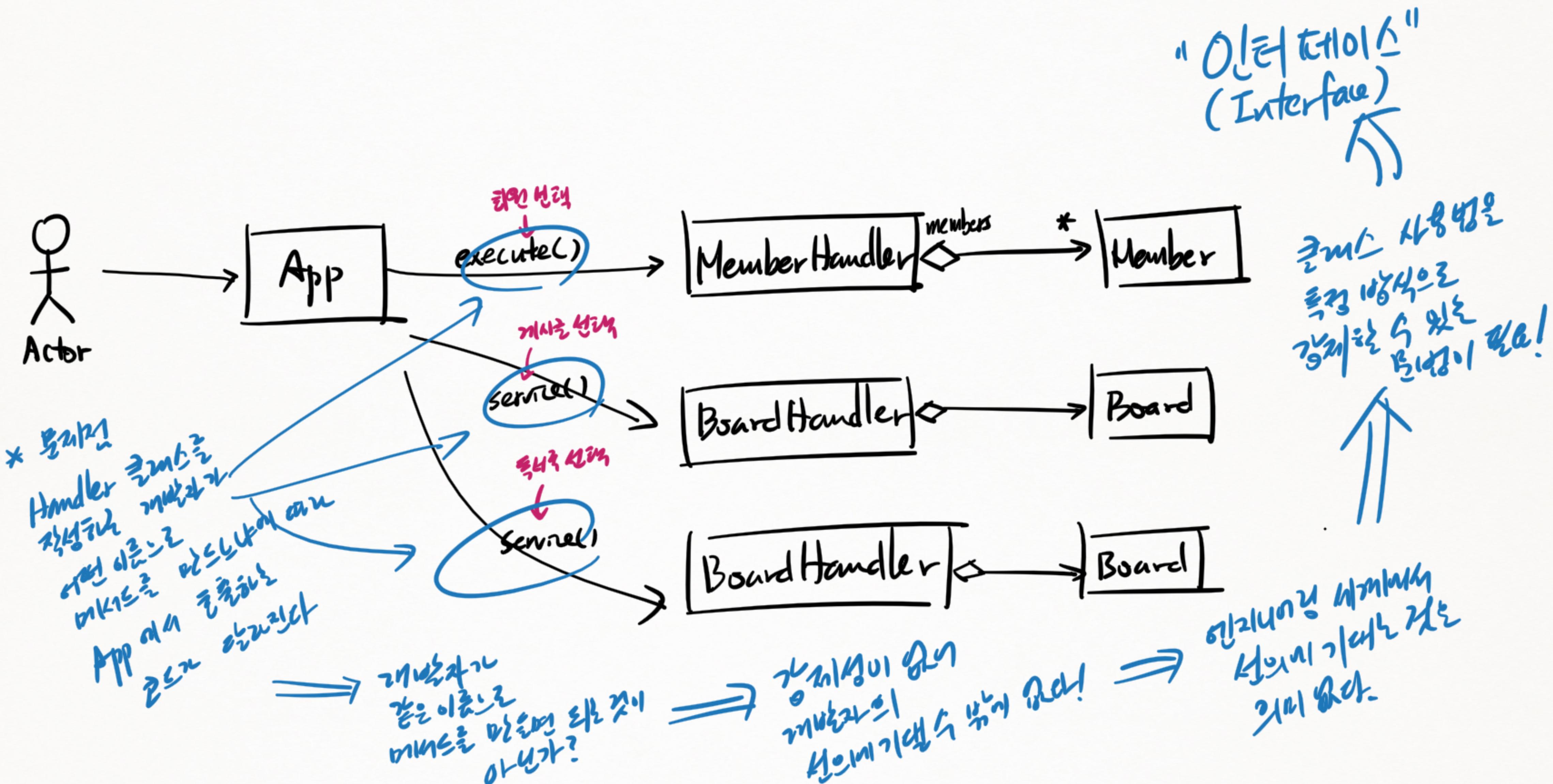
# App, MemberHandler, BoardHandler et prompt



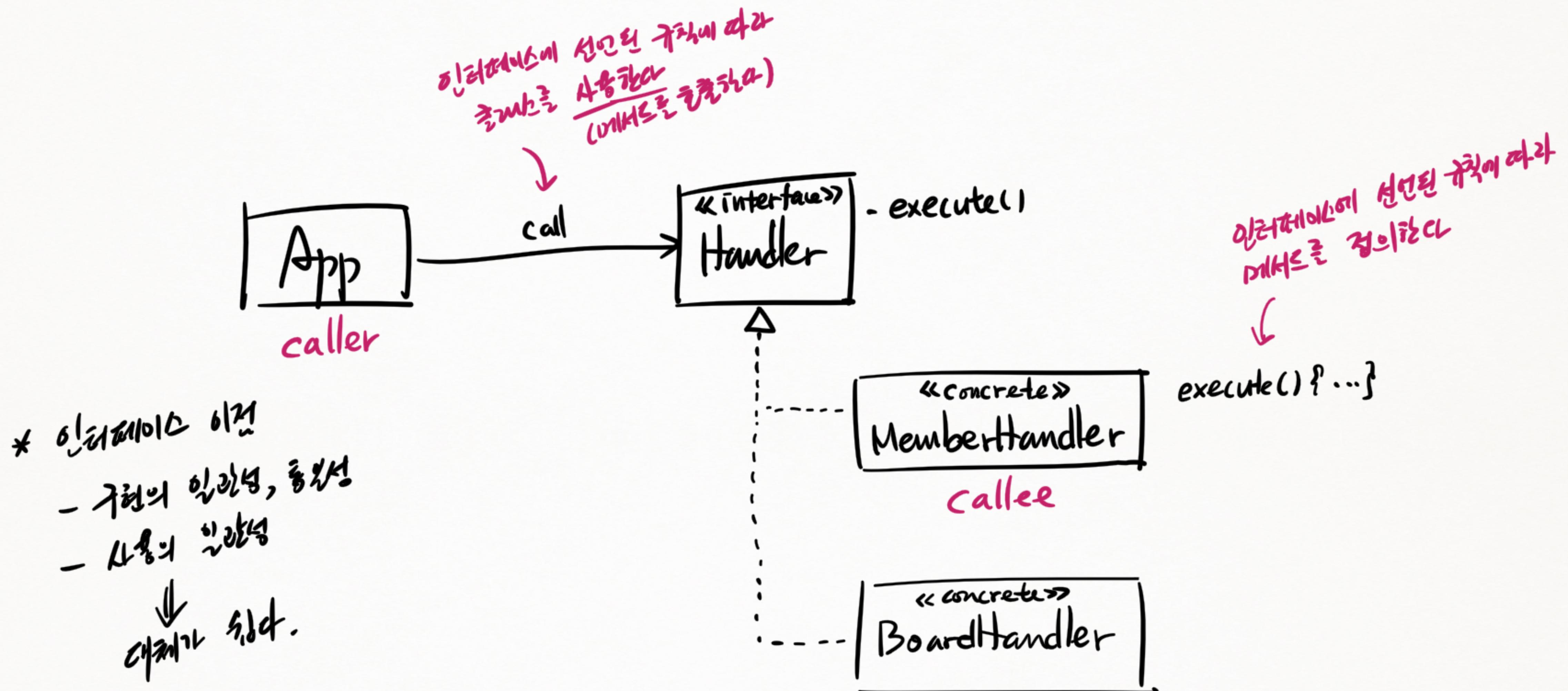
## 16. Handler에게 메뉴 기능을 위임



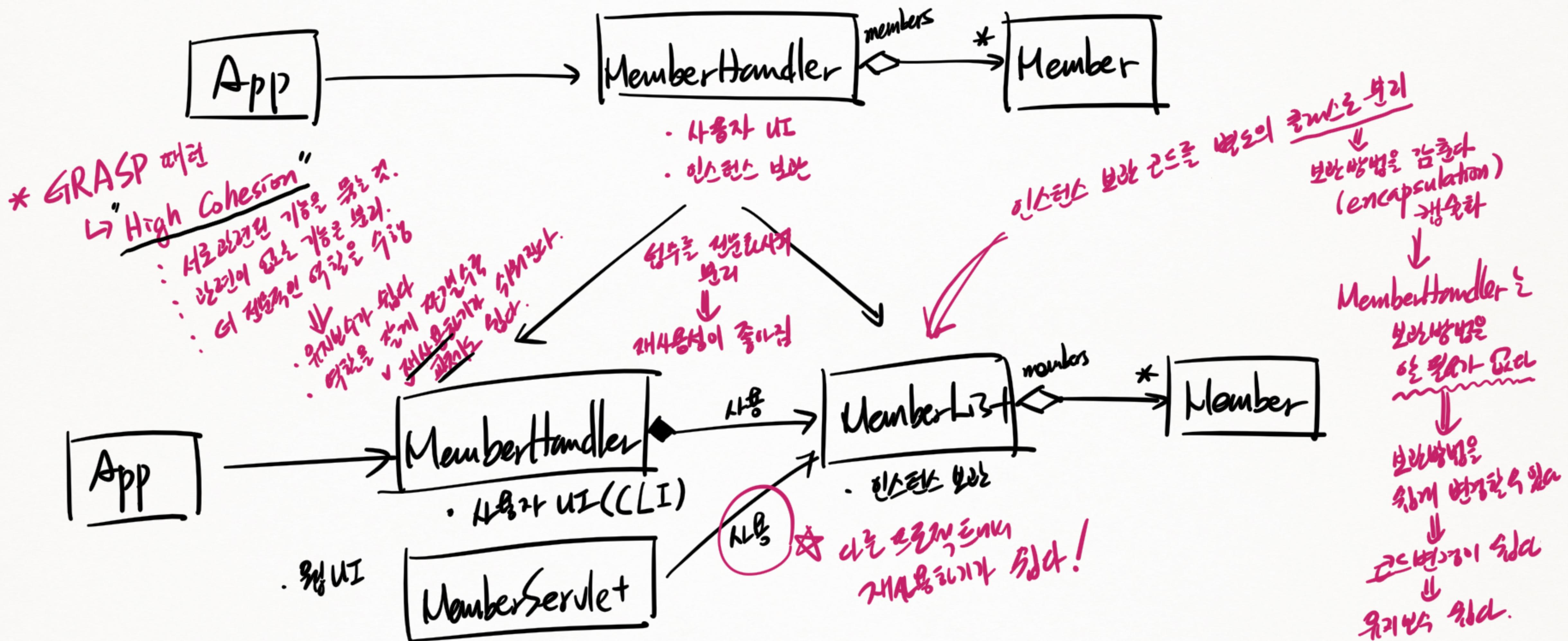
16. Handler에게 메뉴 기능을 위임



### III. Handler의 사용 구조를 인터페이스로 정의하기



## 18. 인스턴스 목록을 다수로 코드를 빼는 축제로 보기



## \* 배운 늘하기

$$\frac{5}{2} = \cancel{2} \cancel{\times}$$

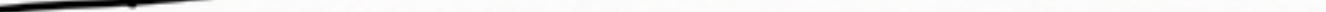
Garbage  $\Rightarrow$ 

0	1	2	3	4
100	200	300	400	500

$$\underline{t+2} = 4$$

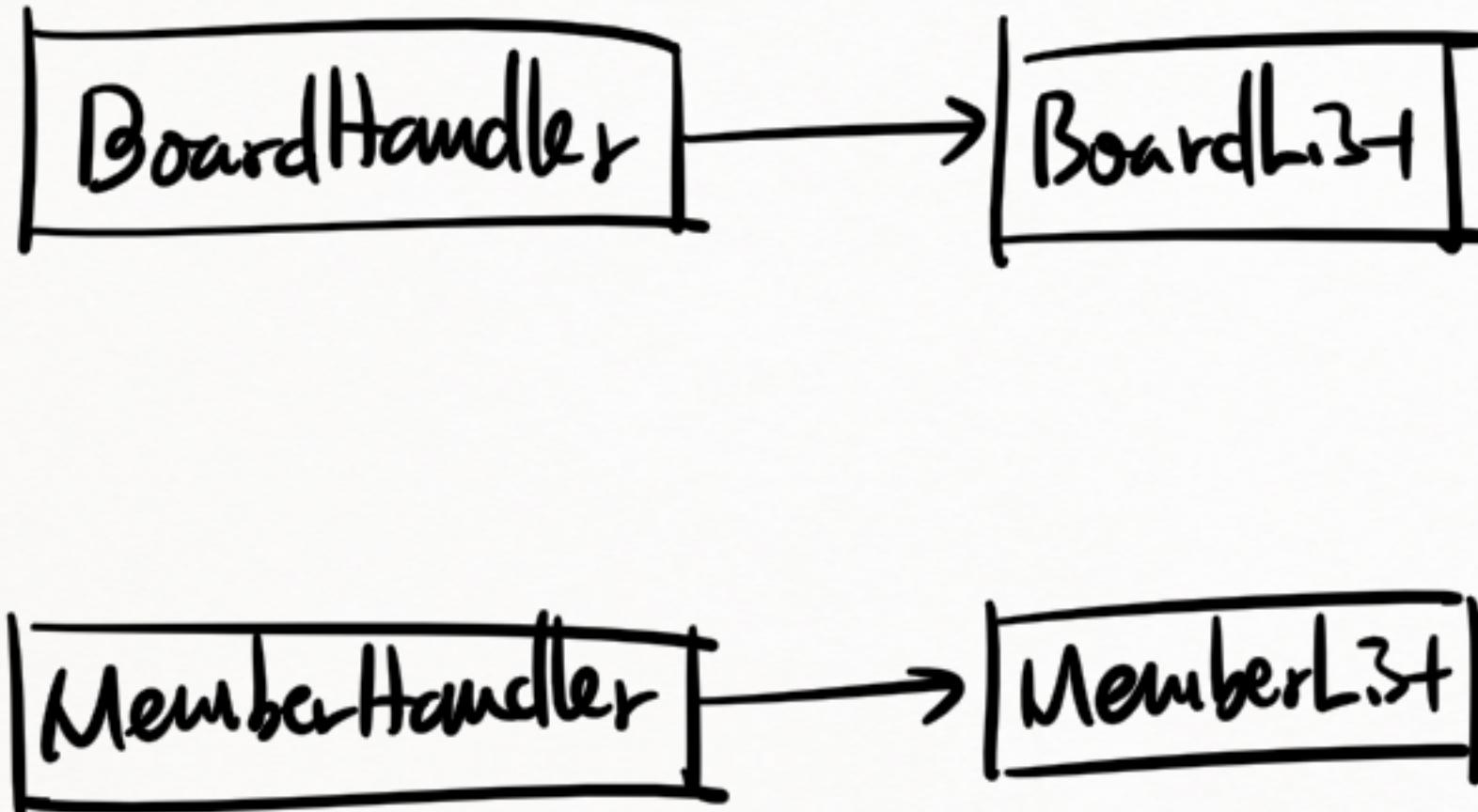
garbage

$$7 + \frac{1}{3} = 10$$

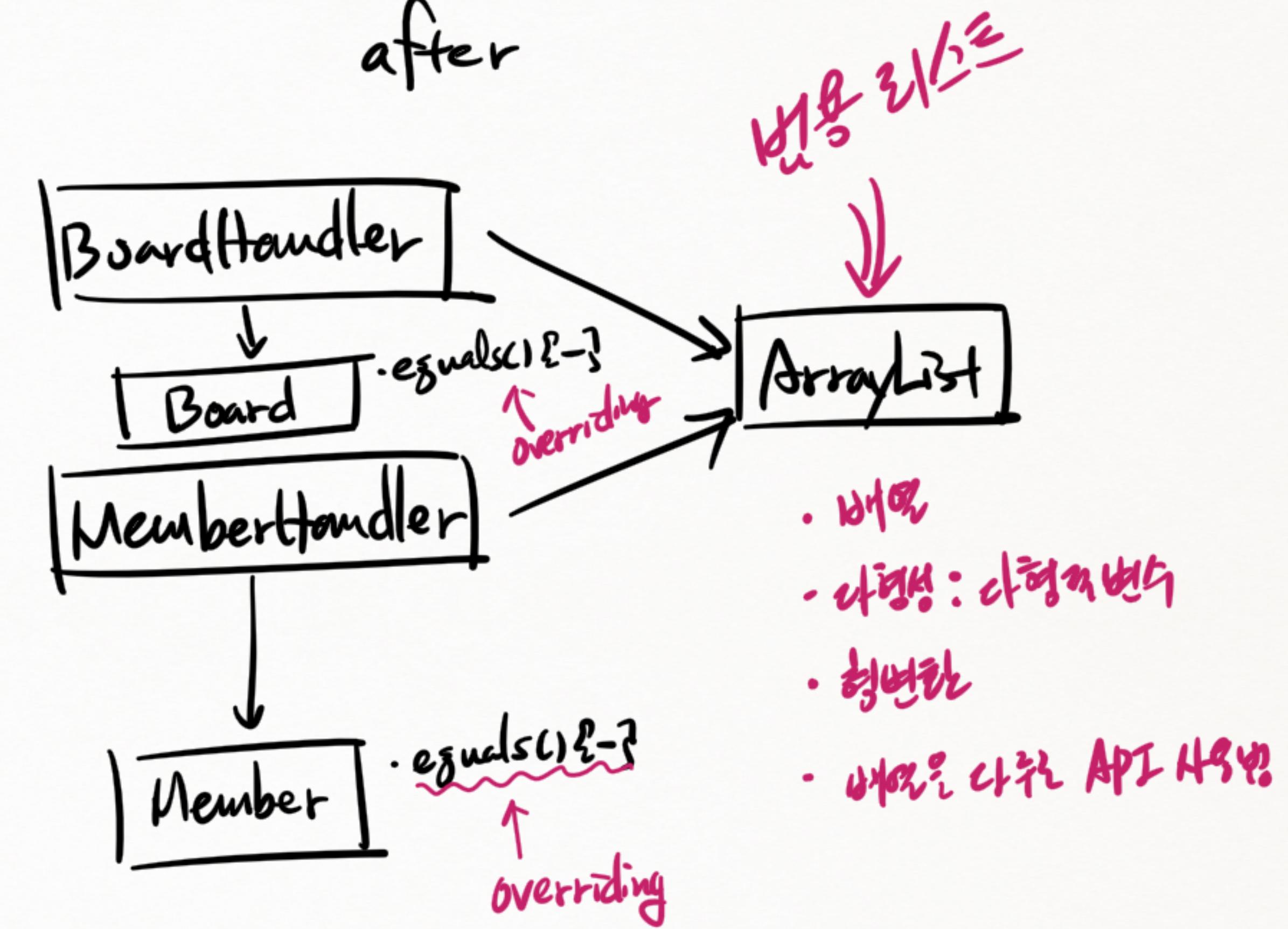
boards  $\rightarrow$  

## 19. 범용리스트 만들기

before



after

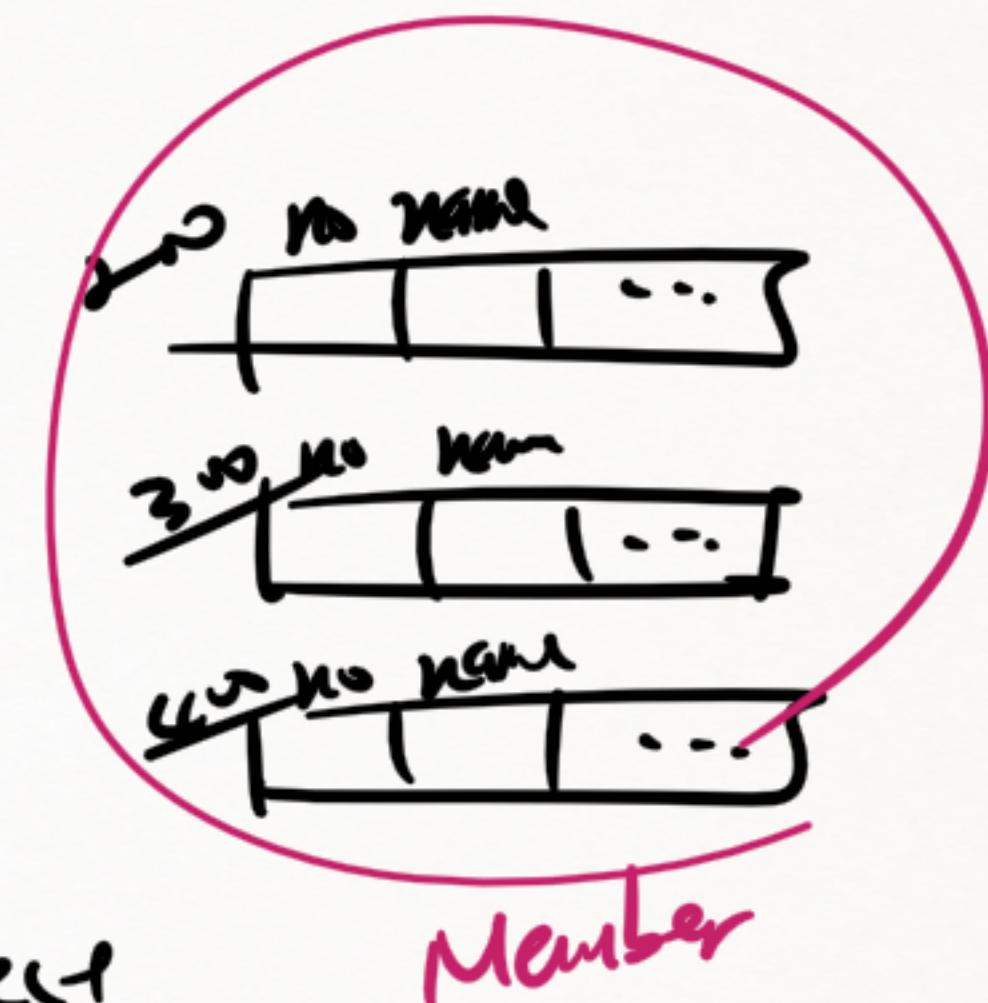
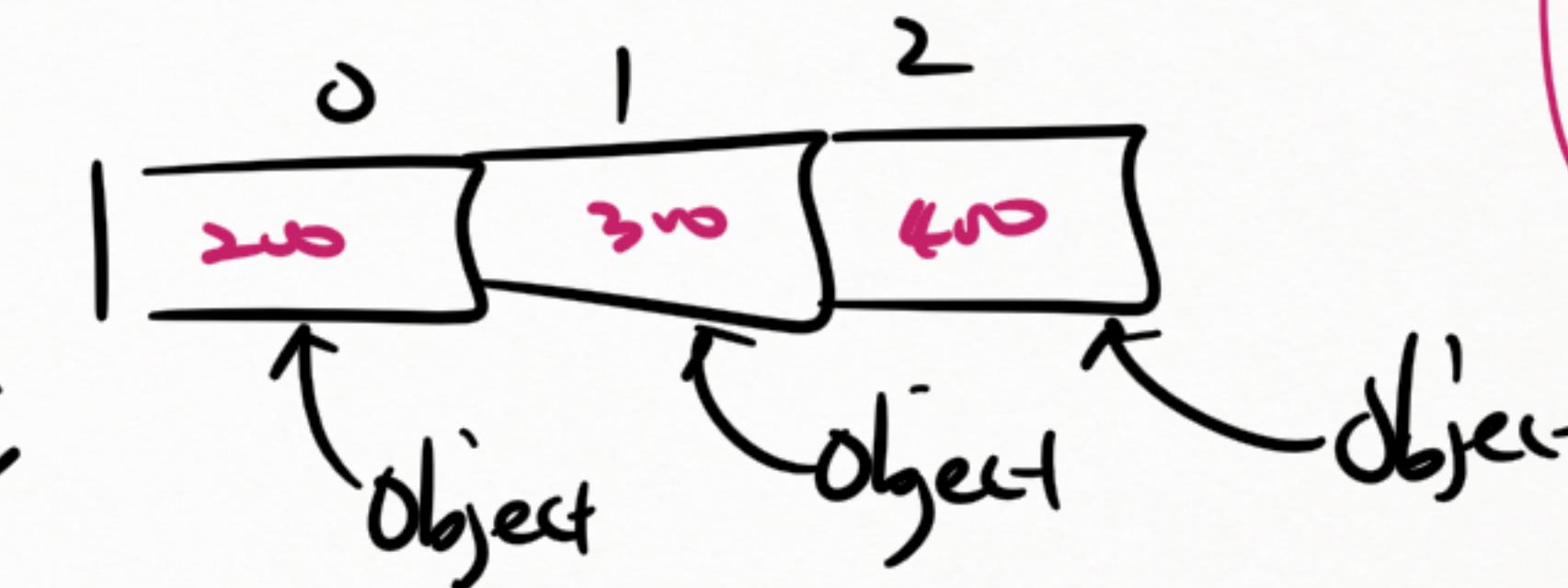


`Object[] arr = new Object[3];`

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

`obj[1] = new Member()`

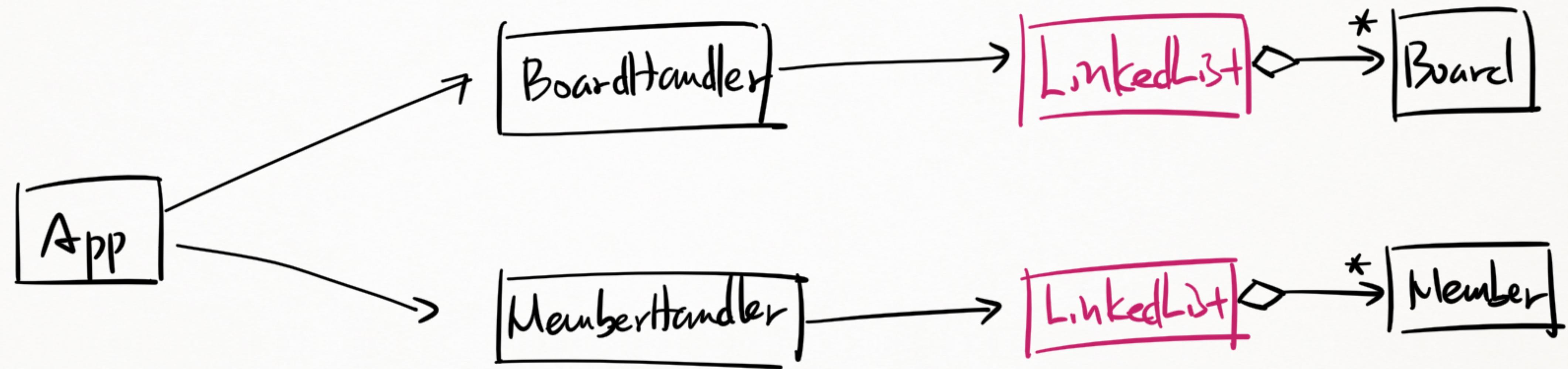
obj[2] = new Member()



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

[1) arrl ;  
↑  
arrl 이 가리키는 것 ~  
Member 라이브러리  
Object 라이브러리  
자주 사용하는  
변수

\* 20. LinkedList 자료구조 구현하기



## ArrayList vs LinkedList

	ArrayList	LinkedList
추가/삭제	<u>일정 범위내</u> 빠르 (IMPL)	O 부터 시작
크기 증가	Yes → 가로세로로 가로나가면서	Yes → 가로이거나 세로
검색	인덱스로 조회 (LinkedList는 불가능)	링크를 따라가야 한다 (ArrayList는 가능)
교환, 삽입, 삭제	삽입 → 배열길이 늘기 삭제 → 배열길이 줄기기  <i>(LinkedList는 가능) (불가능)</i>	해당 값을 찾고자 노드 추가/삭제  <i>(List는 가능) (불가능)</i>

## LinkedList - add()

tail 3100

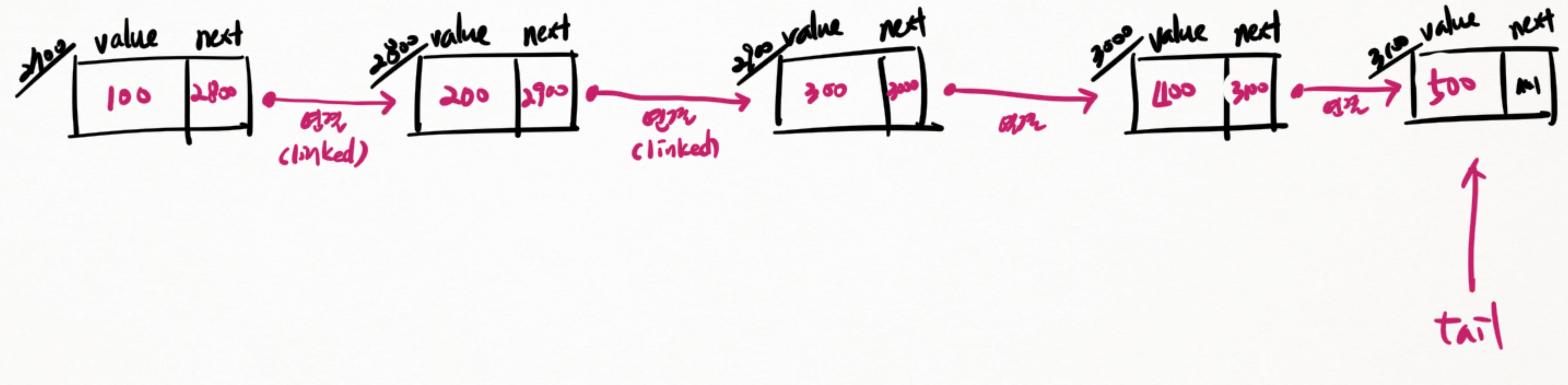
add(100);

add(200);

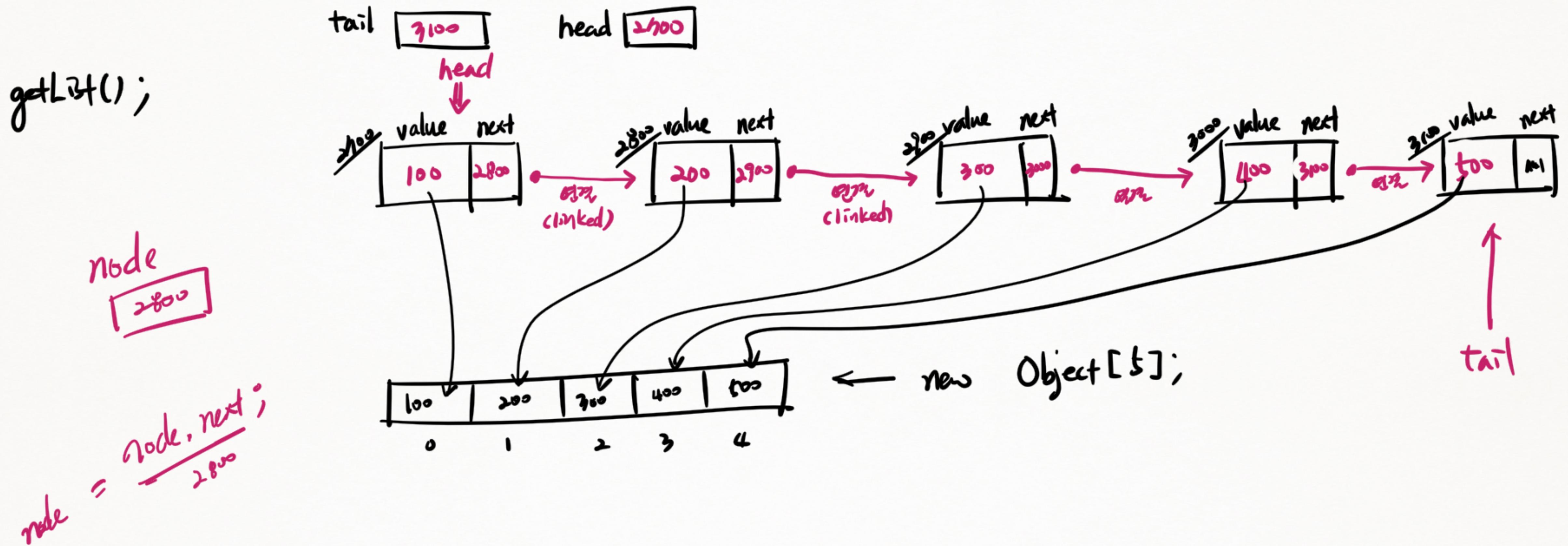
add(300);

add(400);

add(500);



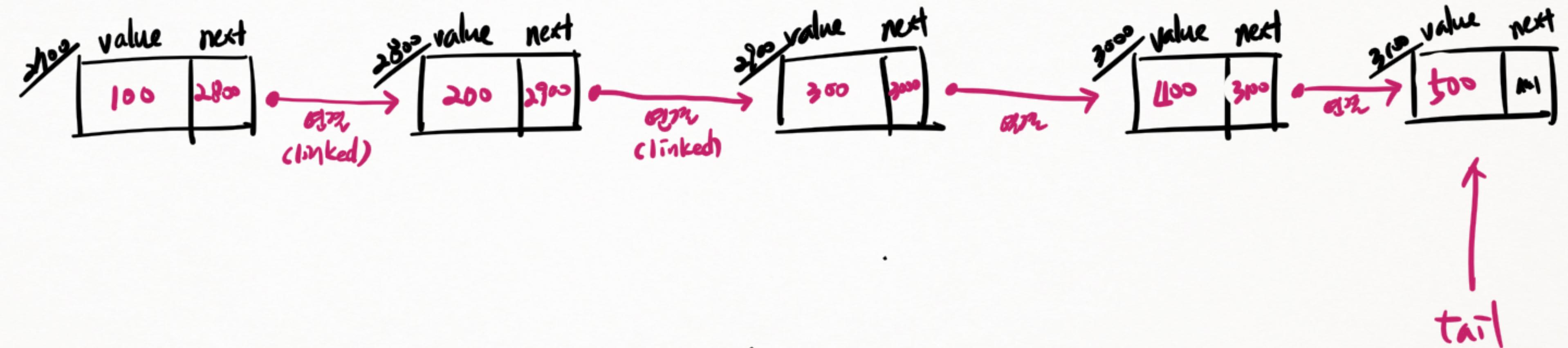
## LinkedList - getList()



## LinkedList - retrieve()

tail 3100

retrieve(100);



~~500.equals(100)~~

↑  
tail

↑

cursor

## LinkedList - remove(): 중간 항목 삭제

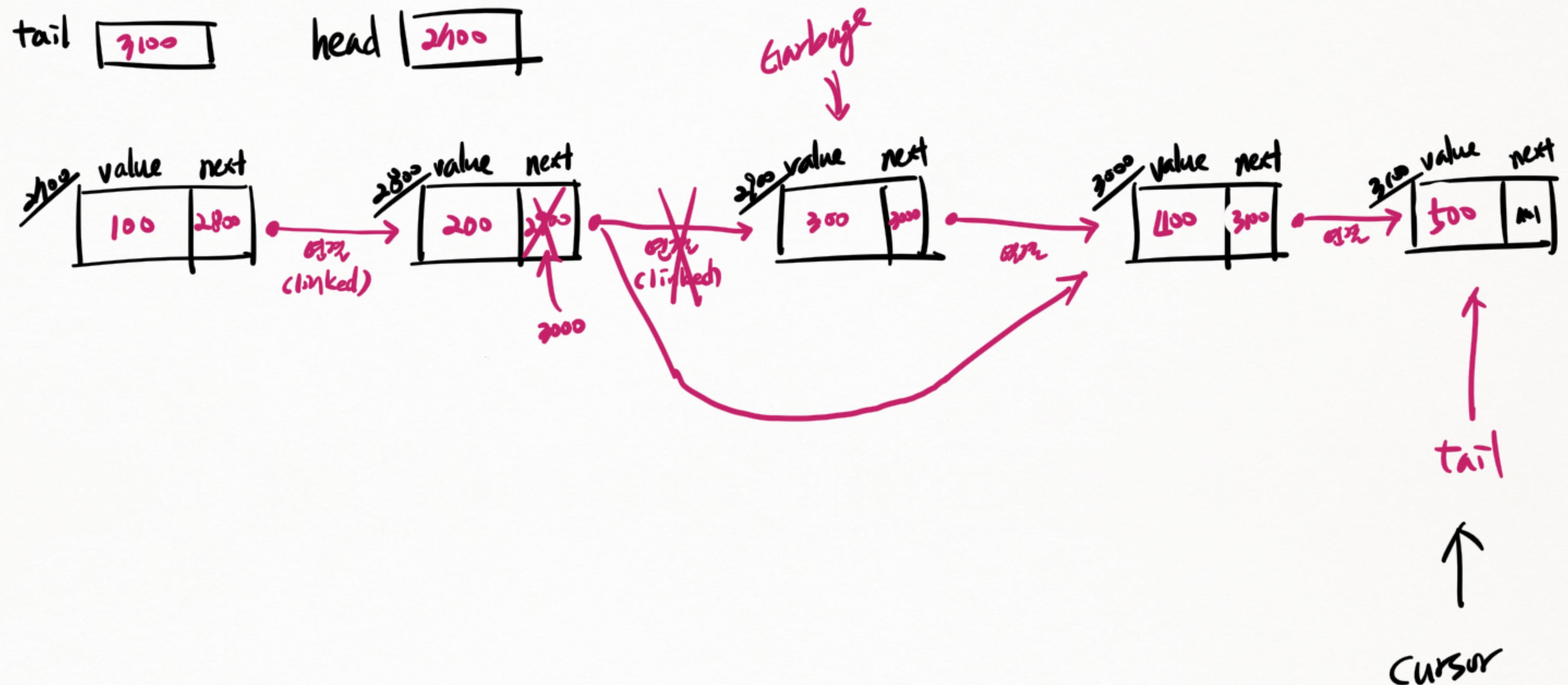
remove(300)

tail [3100]

head [2100]

prev [2800]

cursor [2900]



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

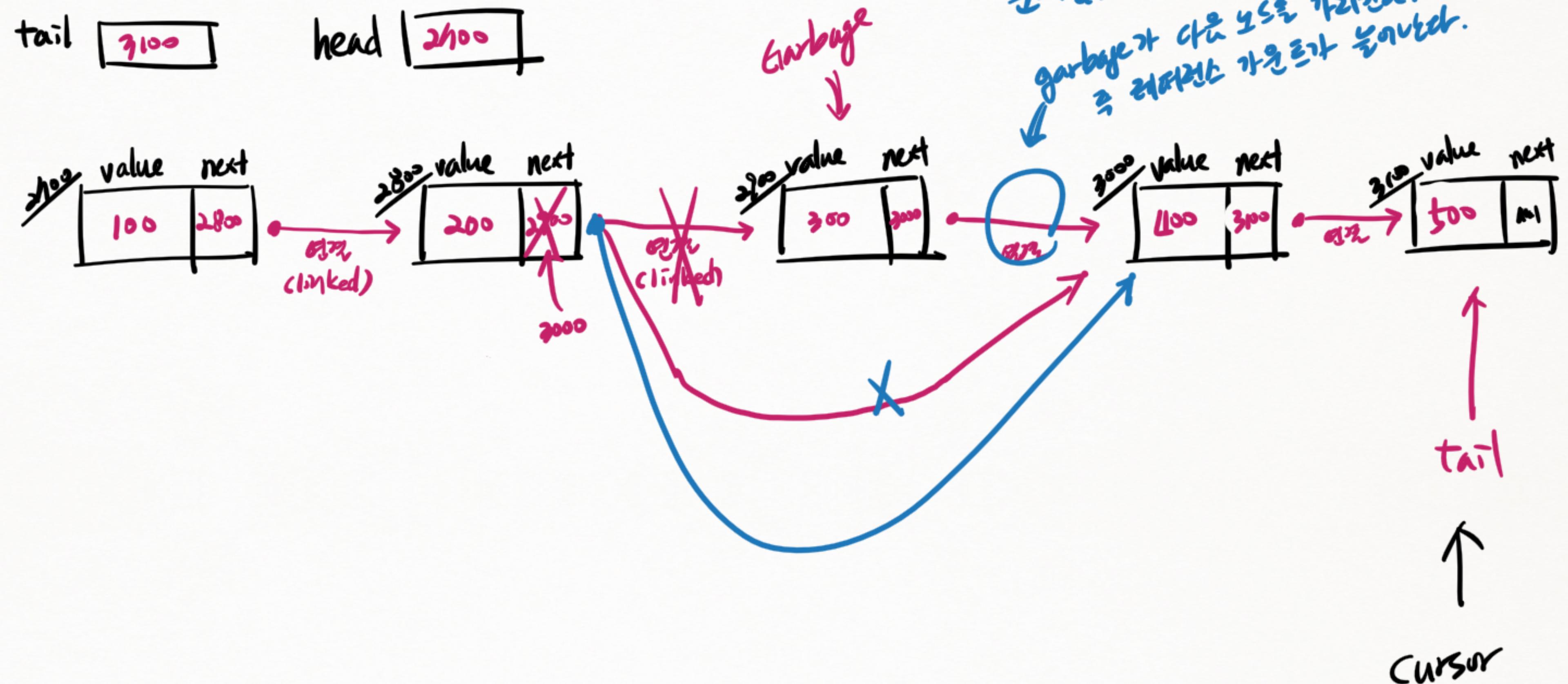
remove(300)

tail  
3100

head  
2100

prev  
2800

cursor  
2900



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

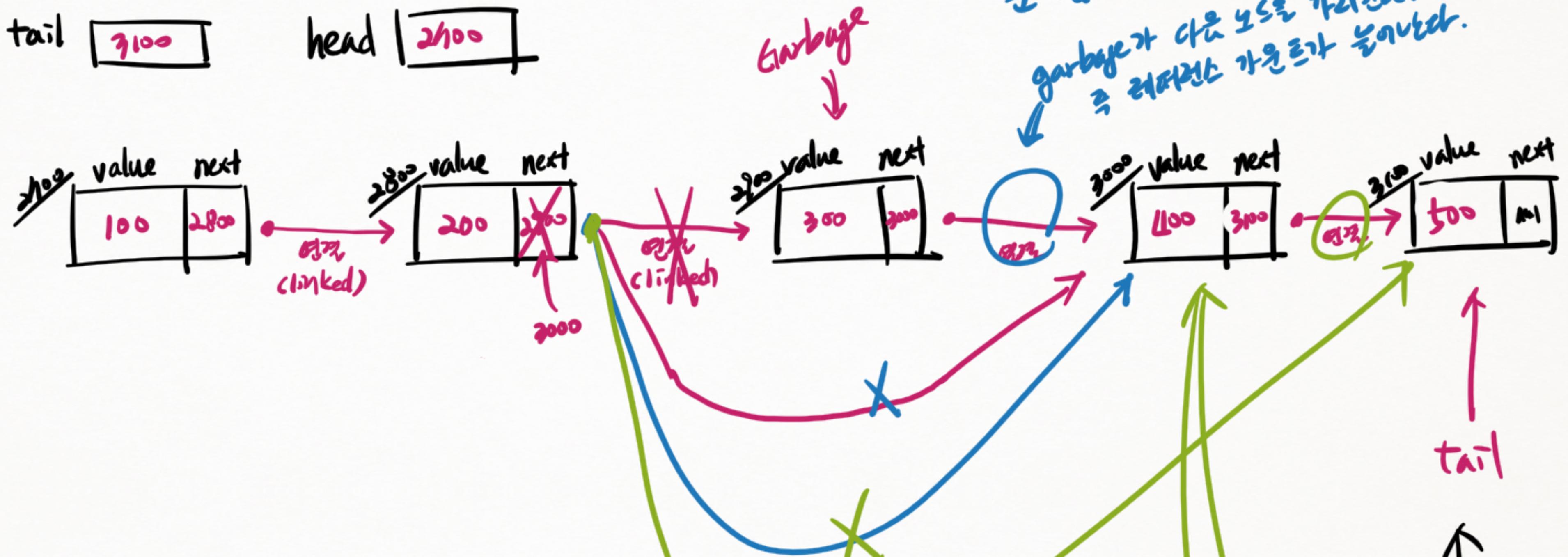
remove(300)

tail  
3100

head  
2100

prev  
2800

cursor  
2900

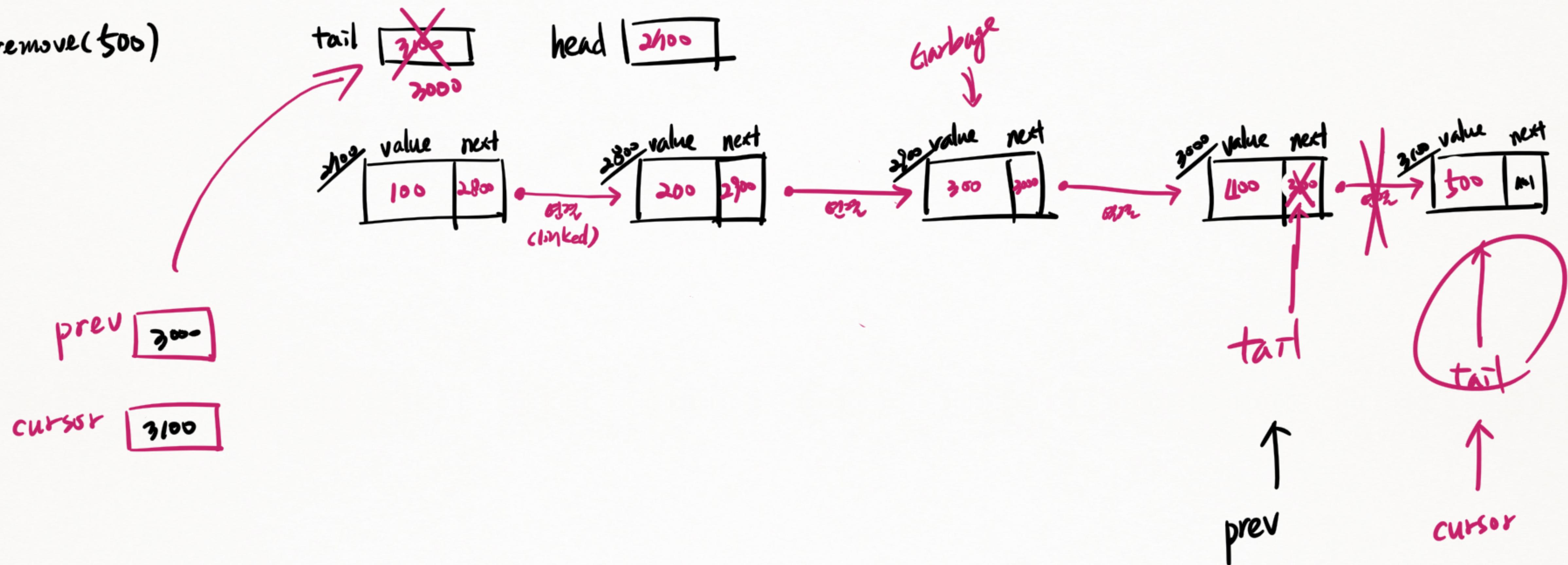


이 노드는  
가비지가 되었나?  
아니? 이전에 속해있던 노드에서  
이 노드를 가리키고 있다.

CURSOR

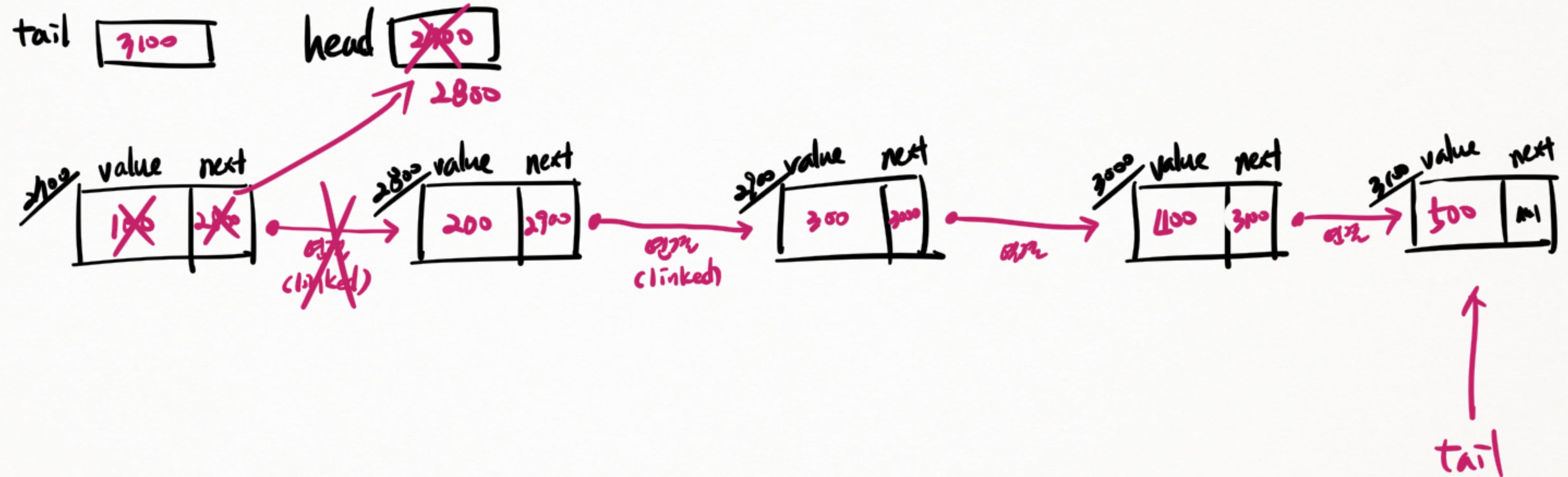
# LinkedList - remove() : 끝 항목 삭제

remove(500)



## LinkedList - remove() : 시작 노드

remove(100)



prev  
[null]

CURSOR  
[200]

LinkedList - remove() : 시작노드 + 끝노드

remove(100)

