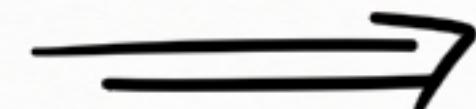
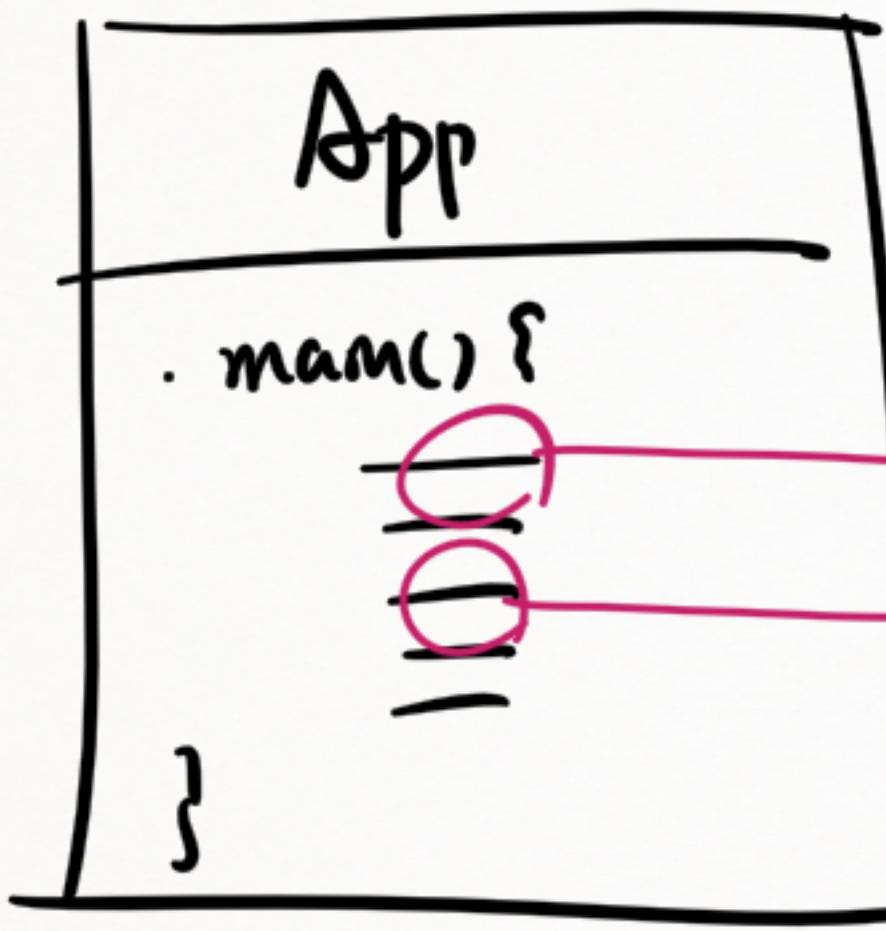
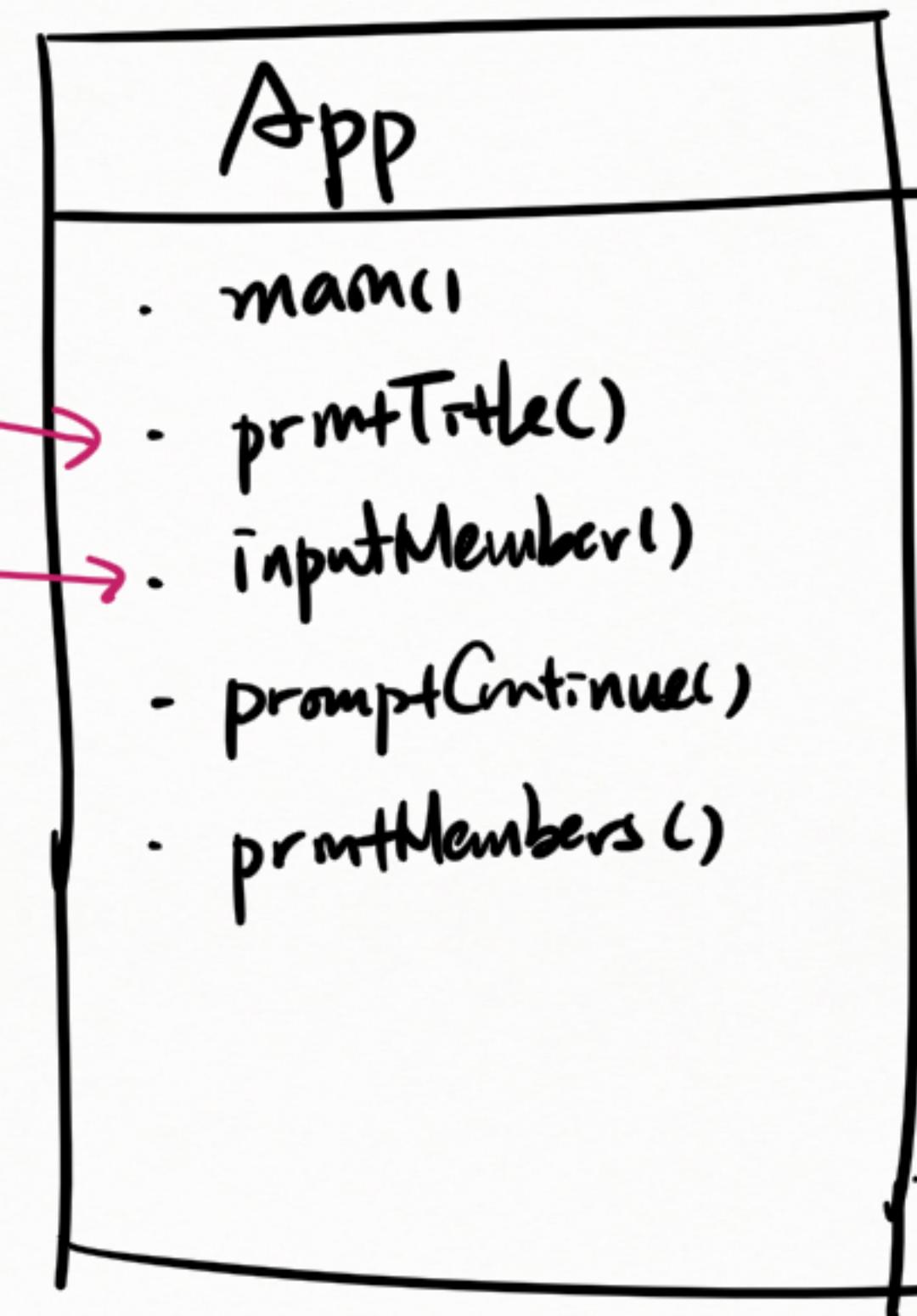


* 1. 디렉트 사용법

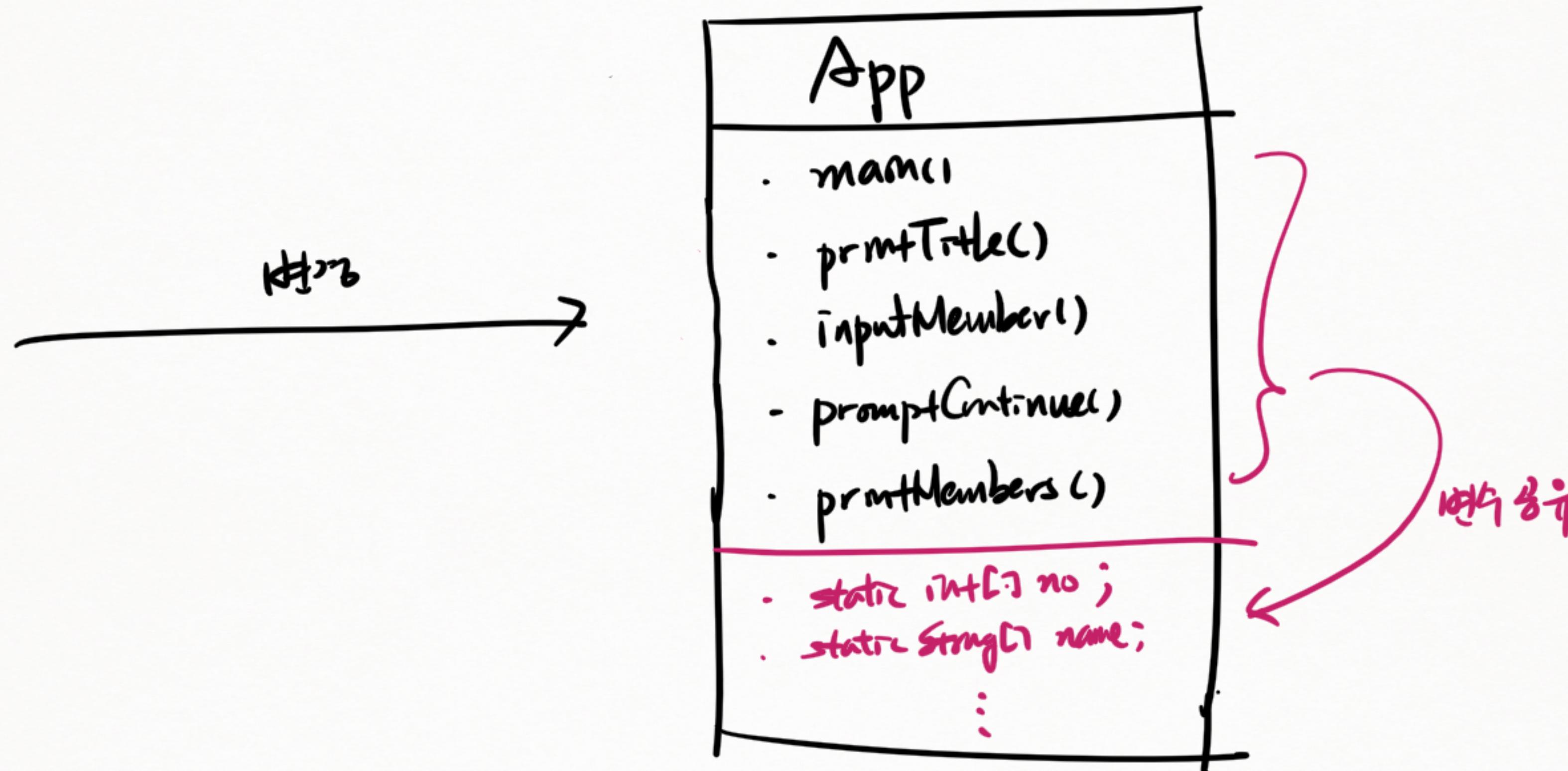
이전



변경

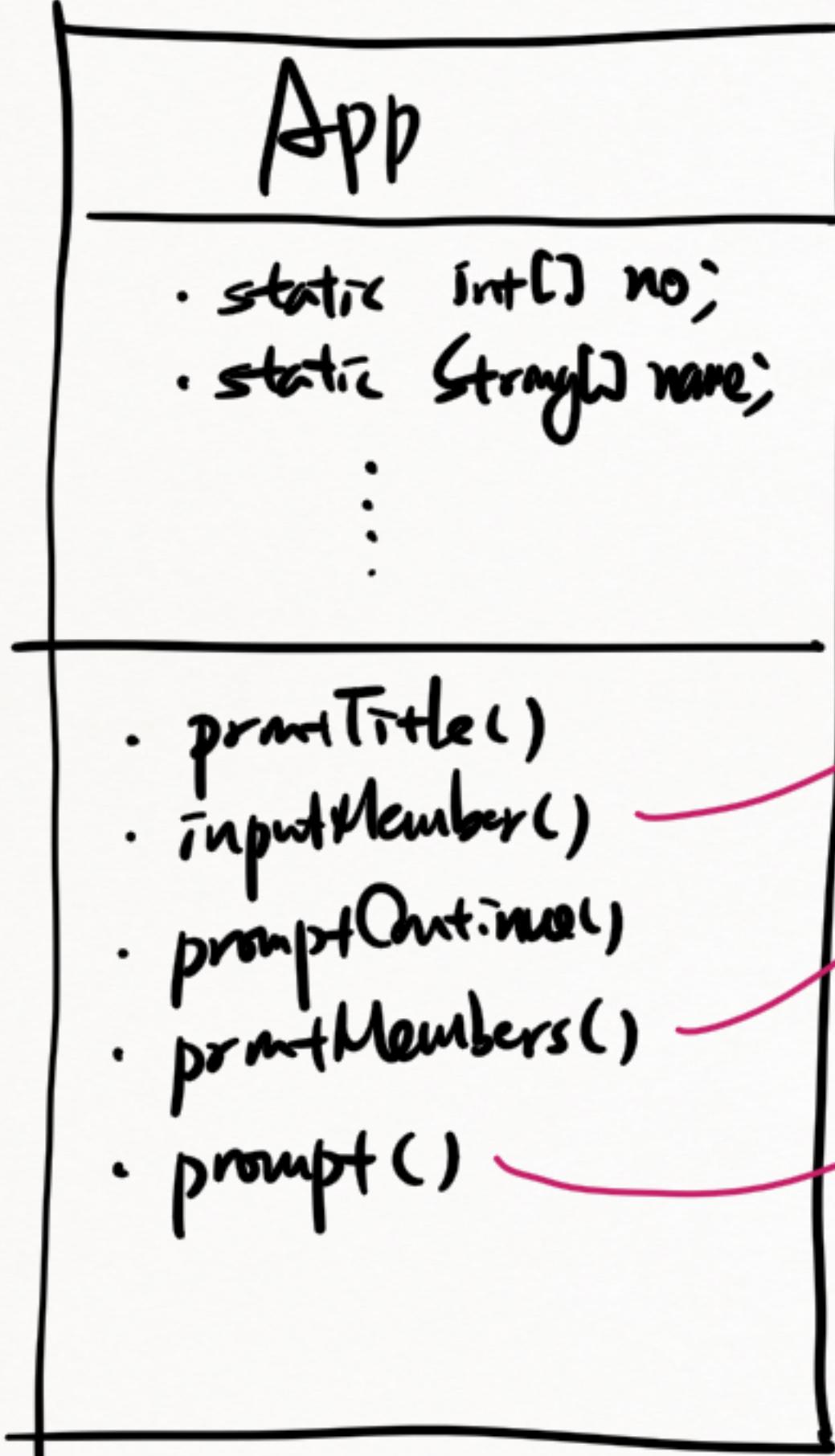


* Q. Lession with Array

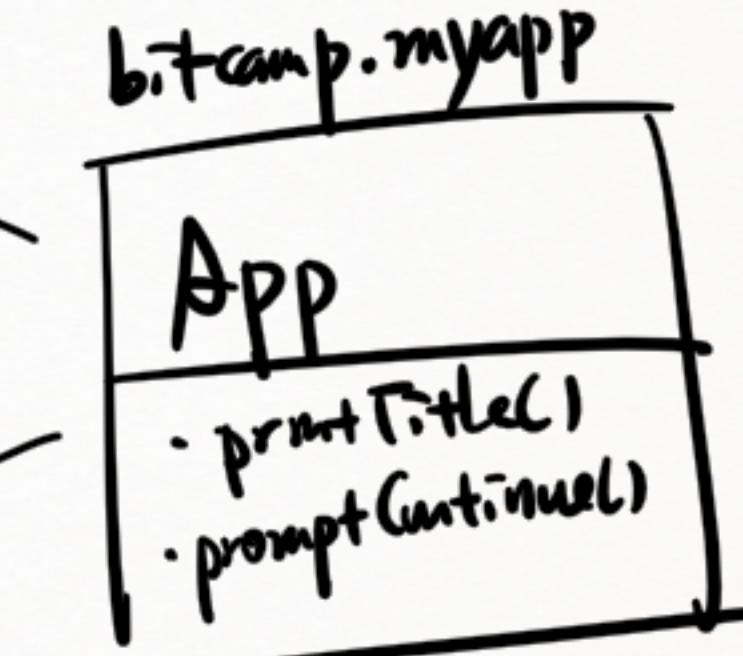
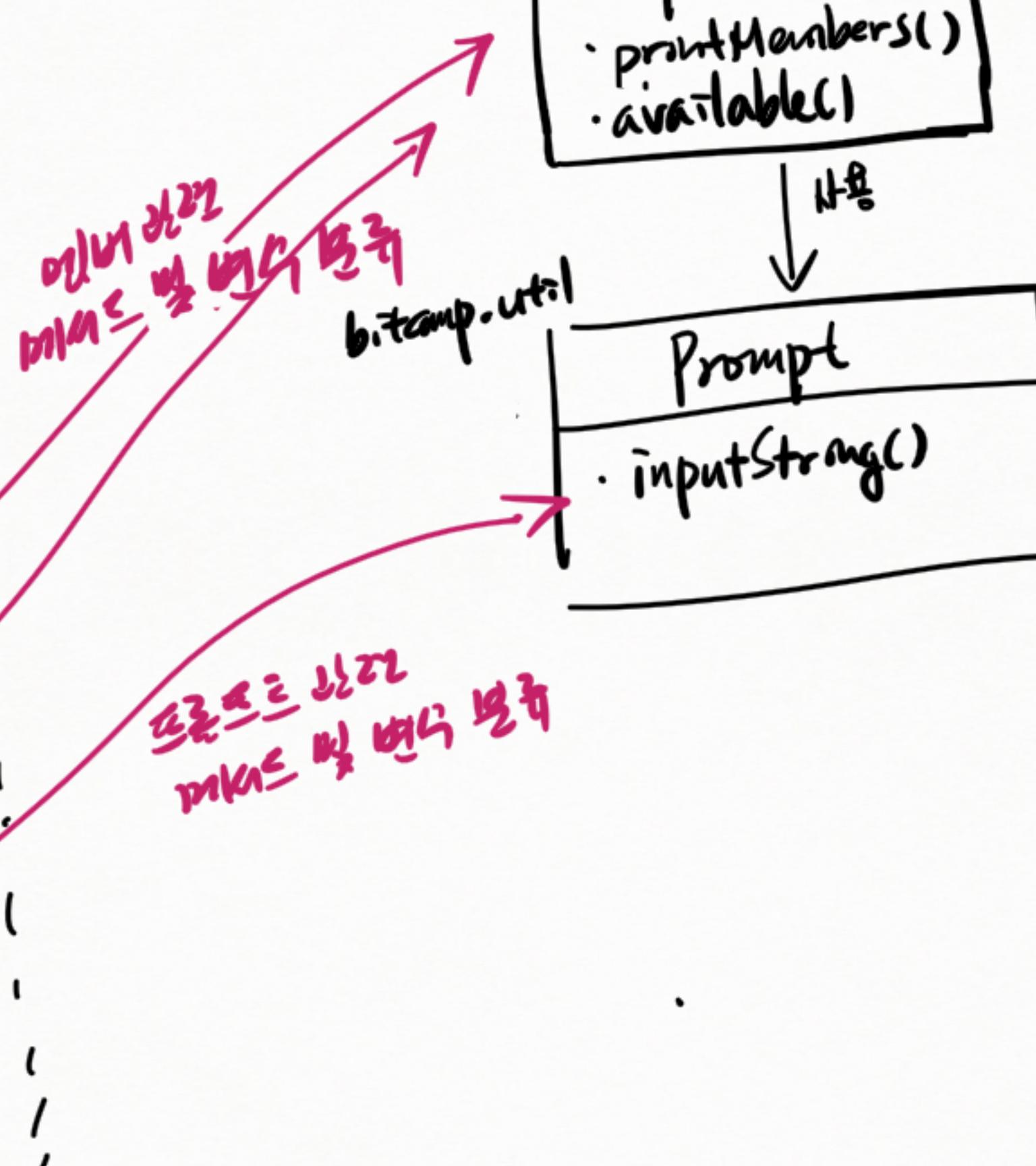
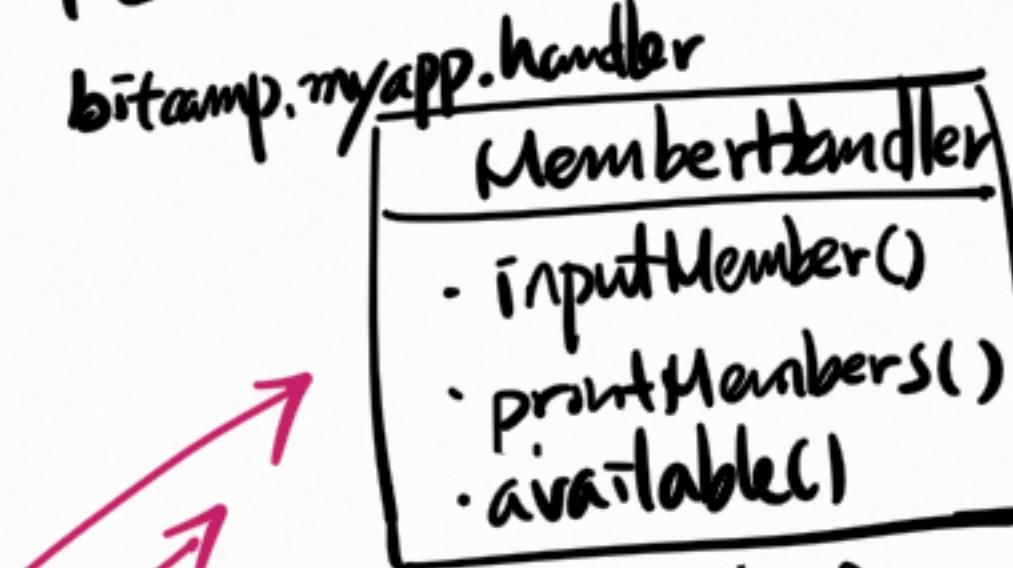


* 9. 클래스 및 패턴 학습

이전 구조
~
Architecture



내 구조



내 구조 : 멤버를 앱을 따라 복제 → 이동? 유지보수를 쉽게.
→ 개별비용 절감

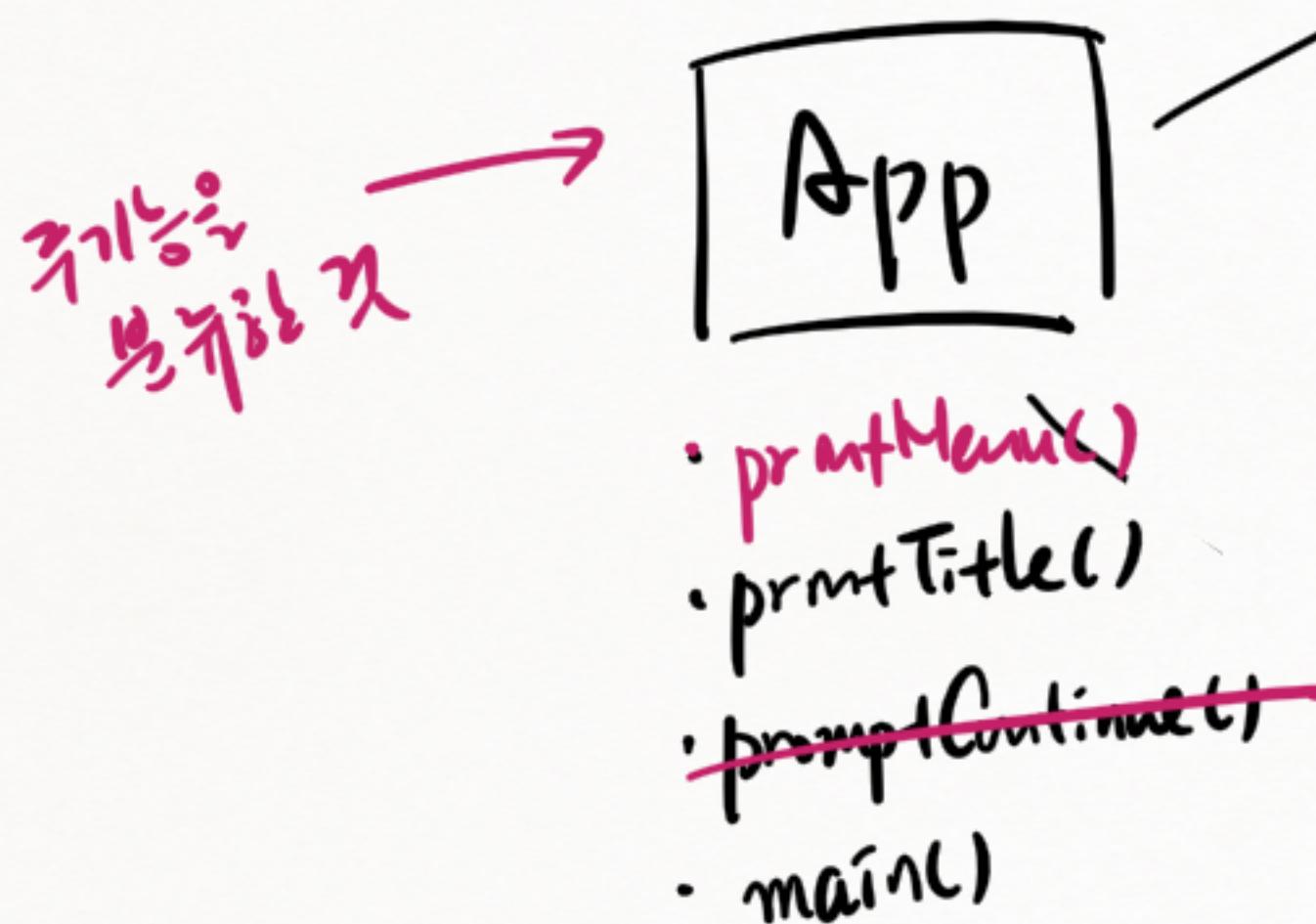
(대신 속도를 높운다.
→ 메모리 더 아낄.)

→ H/W 허용으로 처리

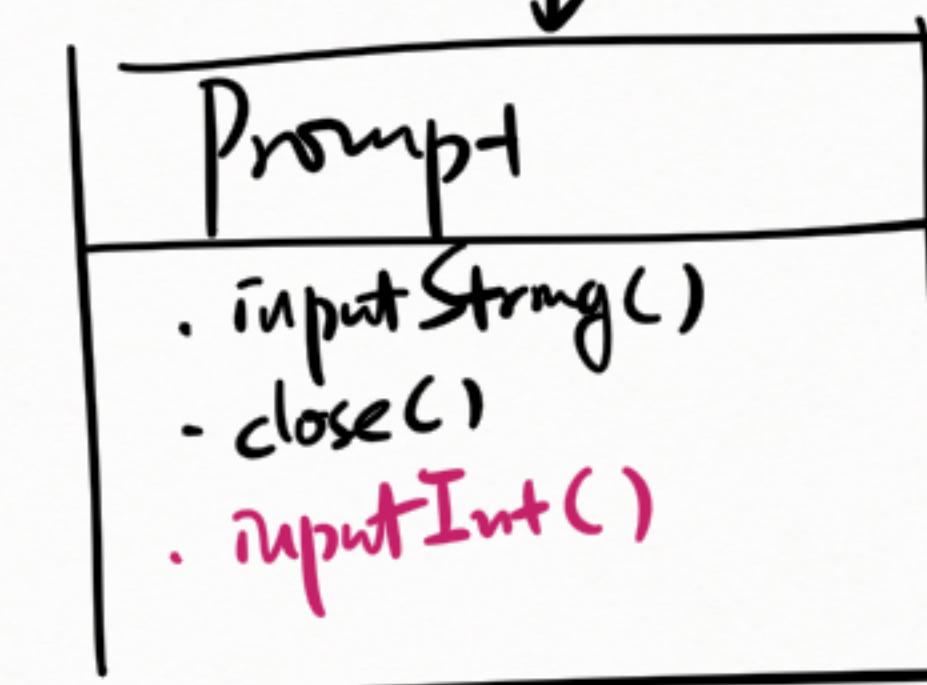
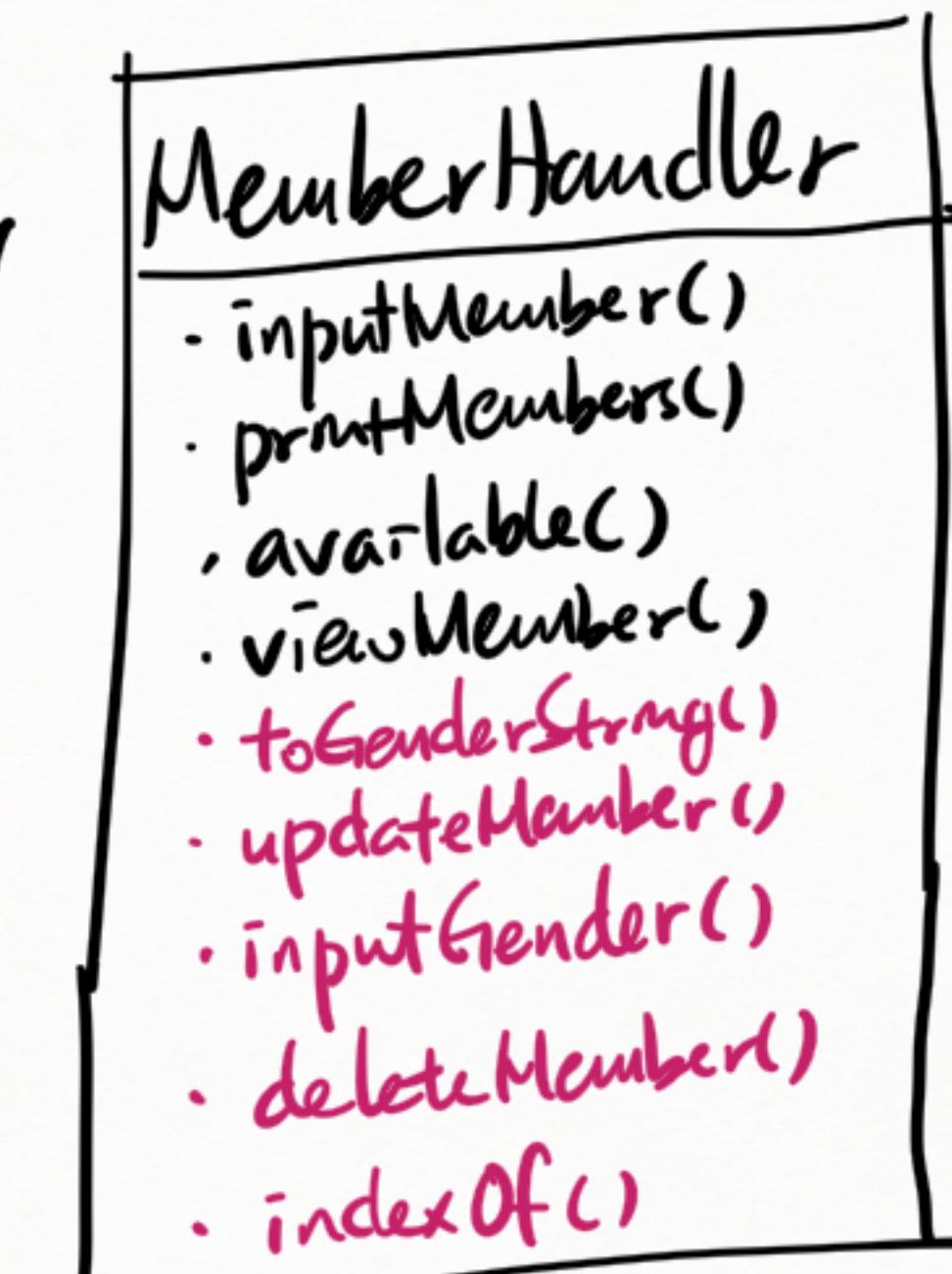
* 10. 멤버 및 CRUD 구현

* 클래스
↳ 애플리케이션 메서드를 분류할 것

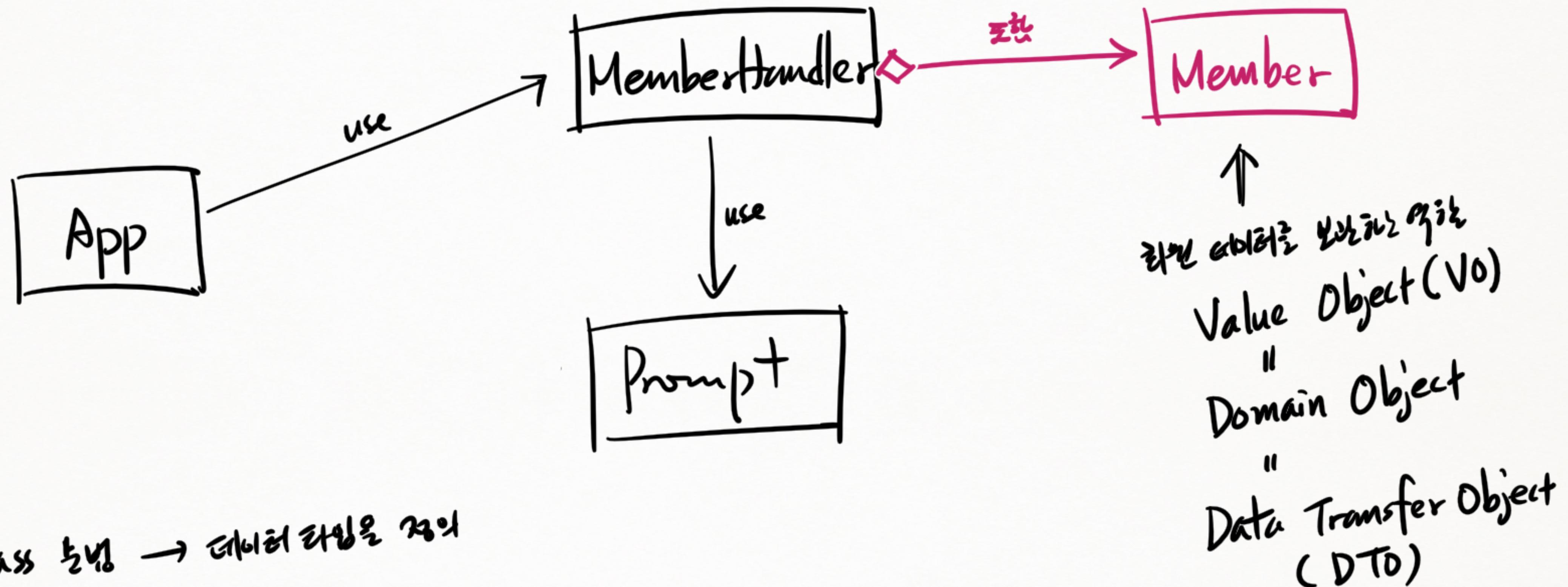
* 패키지
↳ 클래스를 분류할 것.



class 분류 → 메서드를 찾는 용도

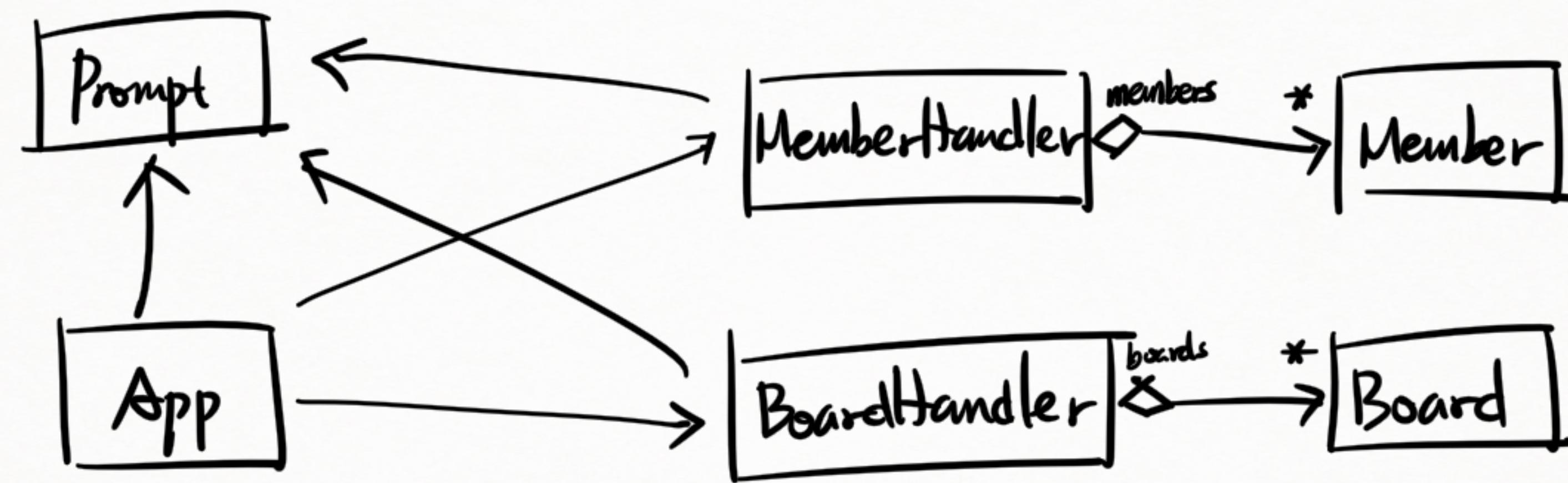


II. 사용자 정의 데이터 타입 만들기

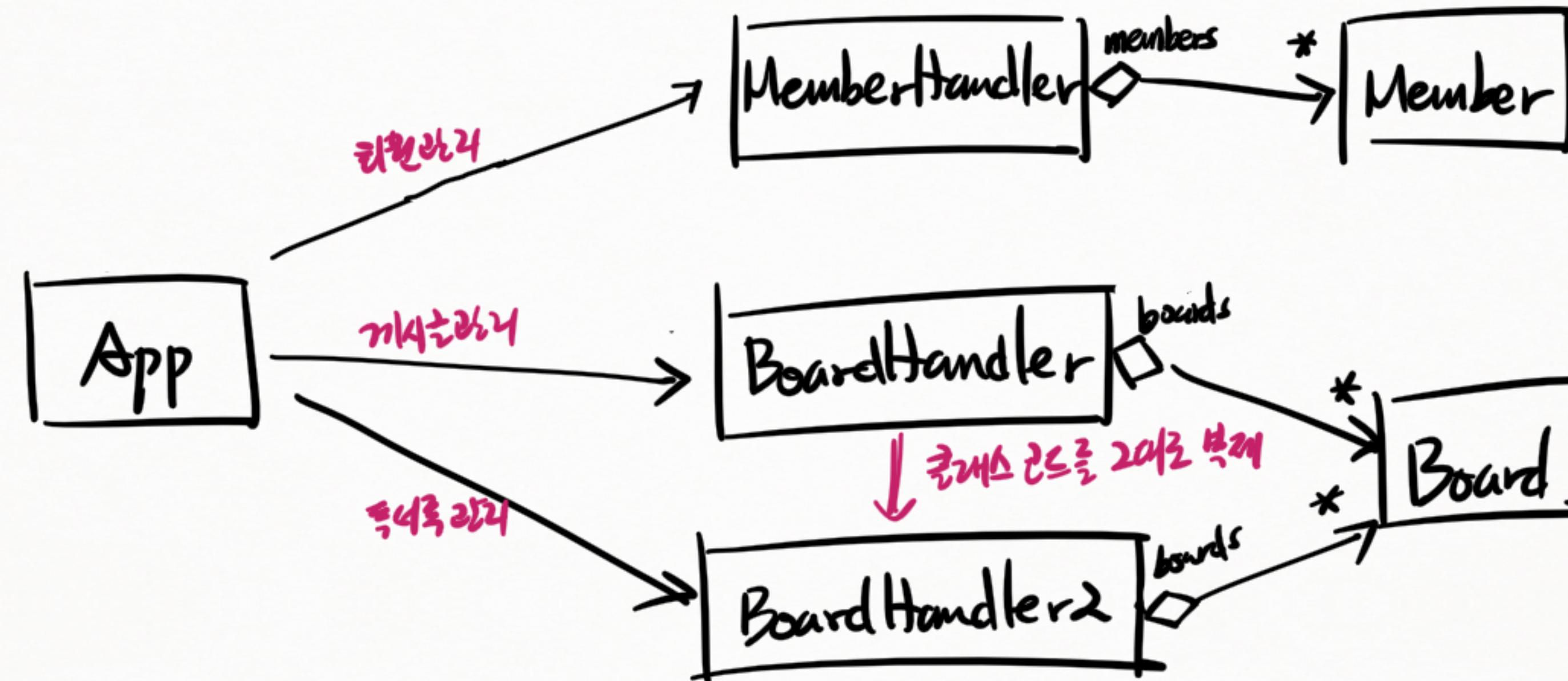


* class 늘임 → 데이터 타입을 확장

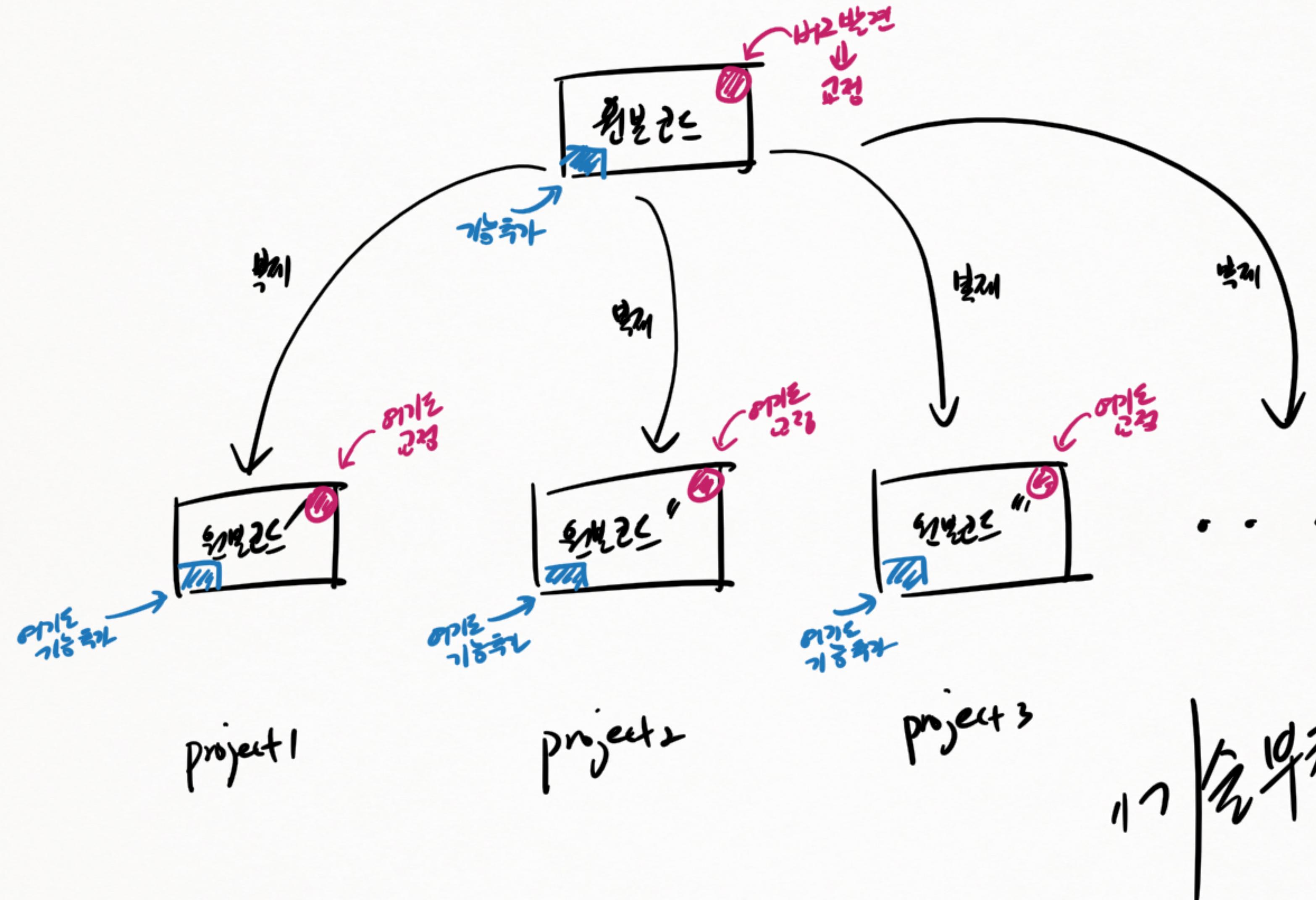
13. 깃허브 CRUD 추가



14. 토커를 CRUD 추가



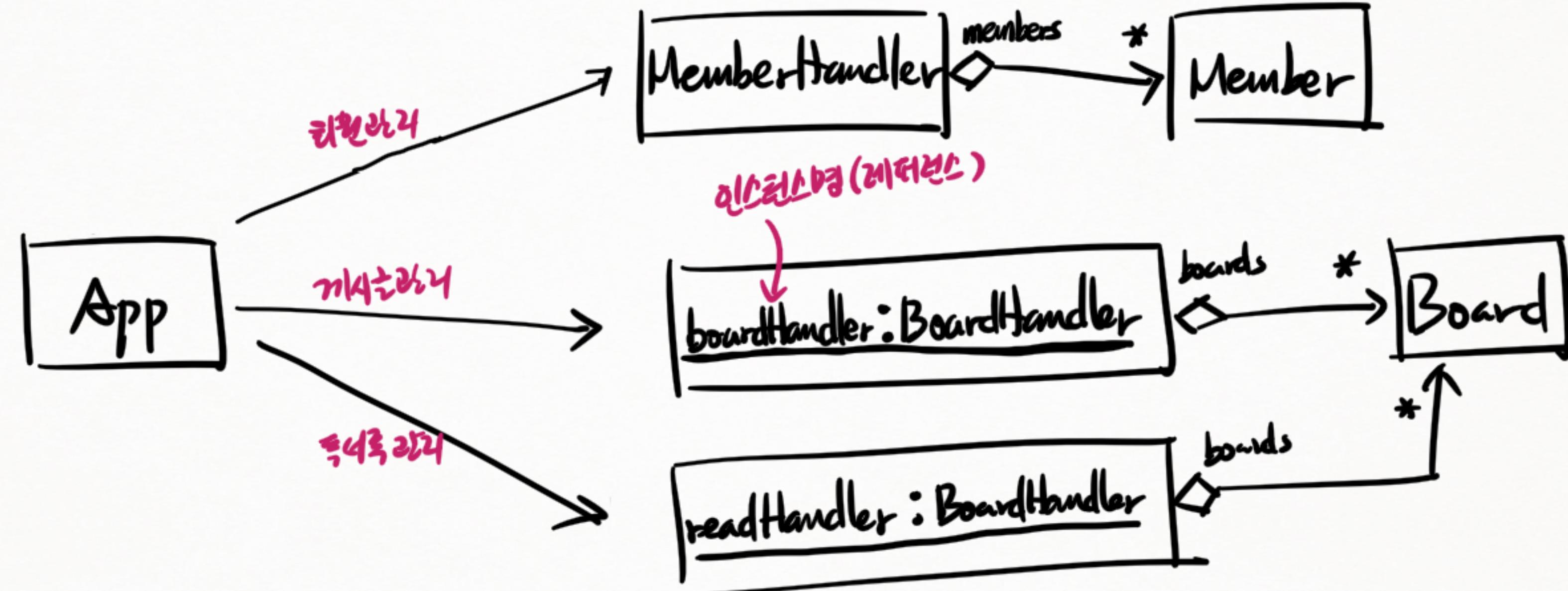
* 가로관 복제로써 새기능을 주입하는 예



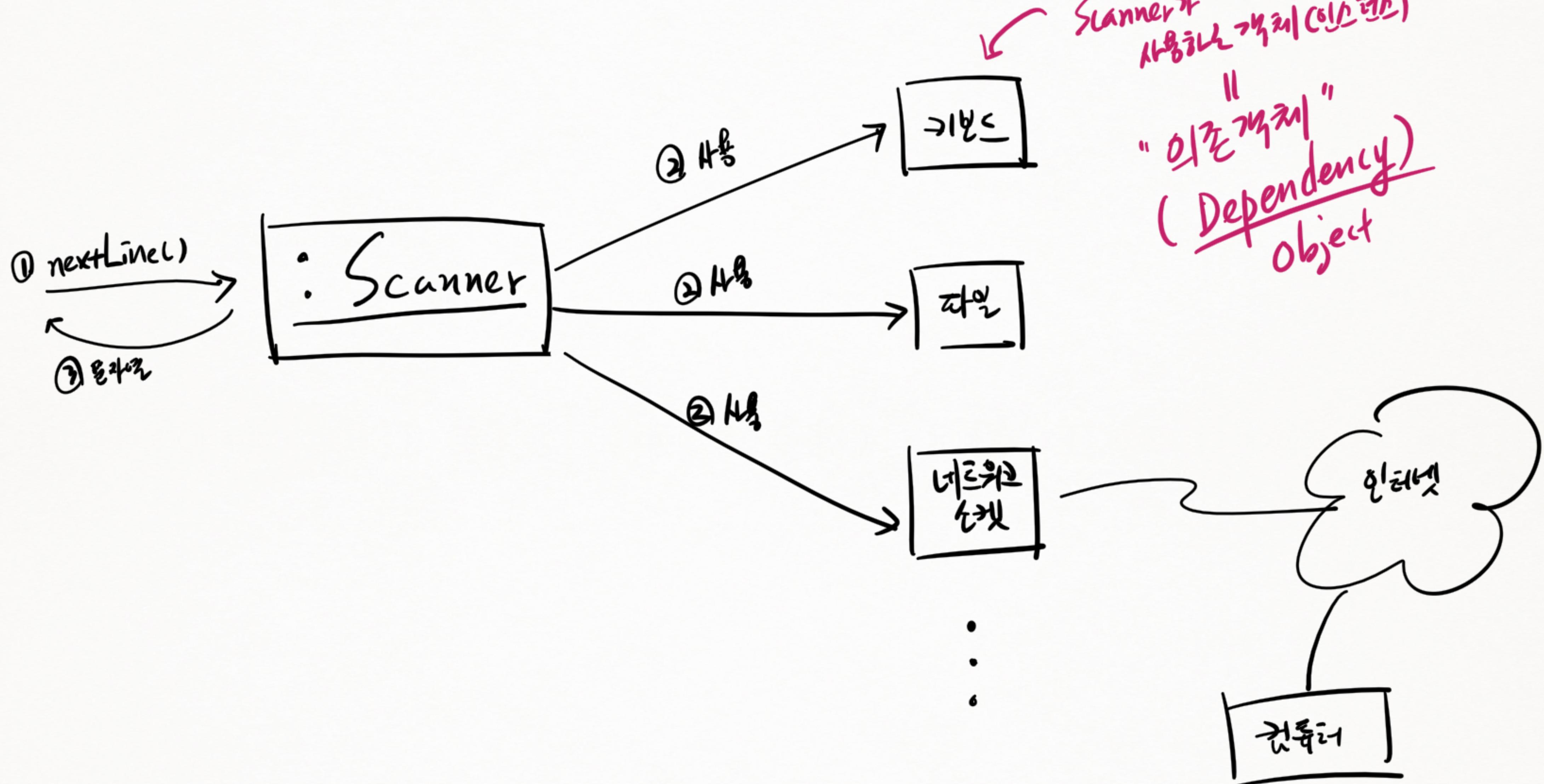
* API를 고정하여 기능을 추가하여 API를 복제한 코드의 디자인은 동일한 일을 수행합니다.

↓
코드는 복제이며 유지보수가 되어야 합니다.

15. 인스턴스 있는 디자인 패턴



Scanner 와 의존 객체



App, MemberHandler, BoardHandler et prompt

