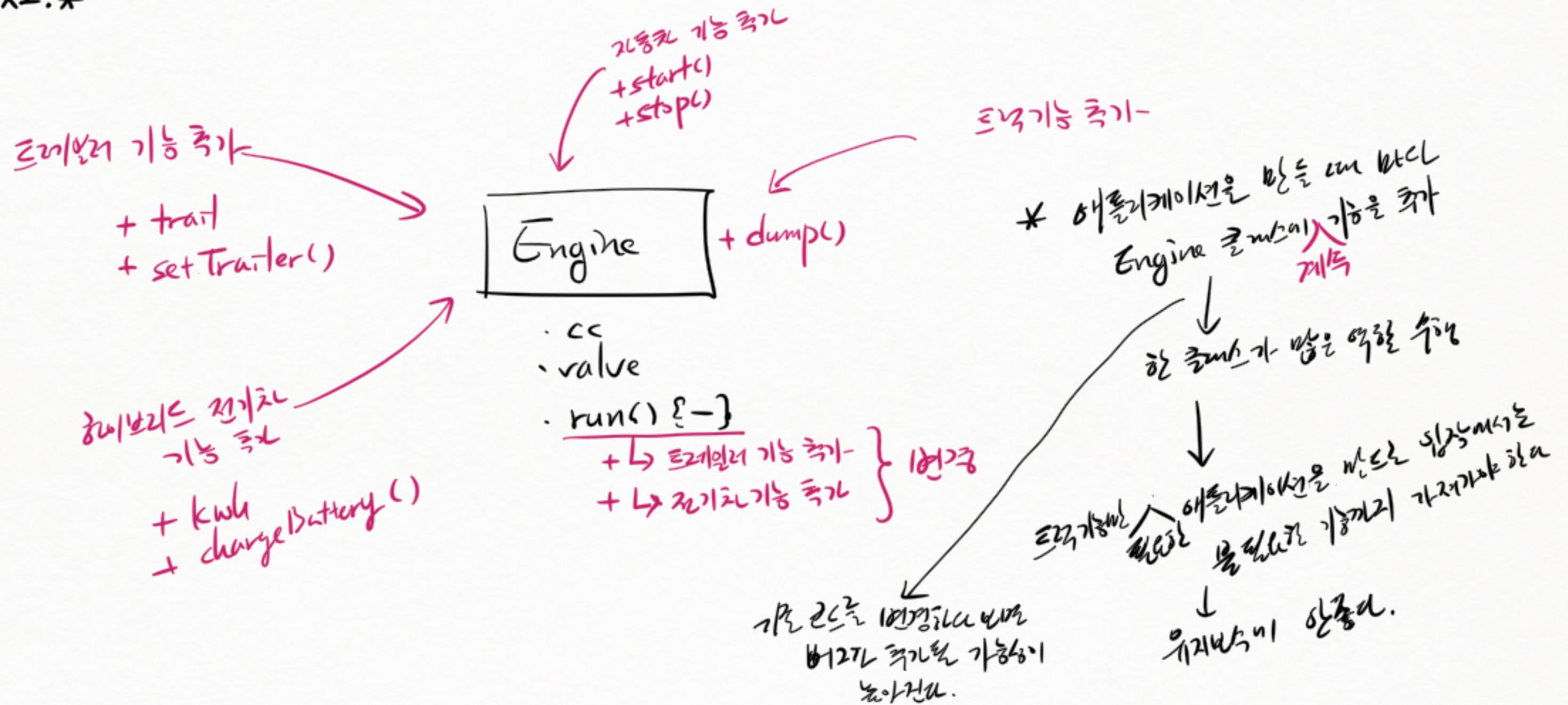


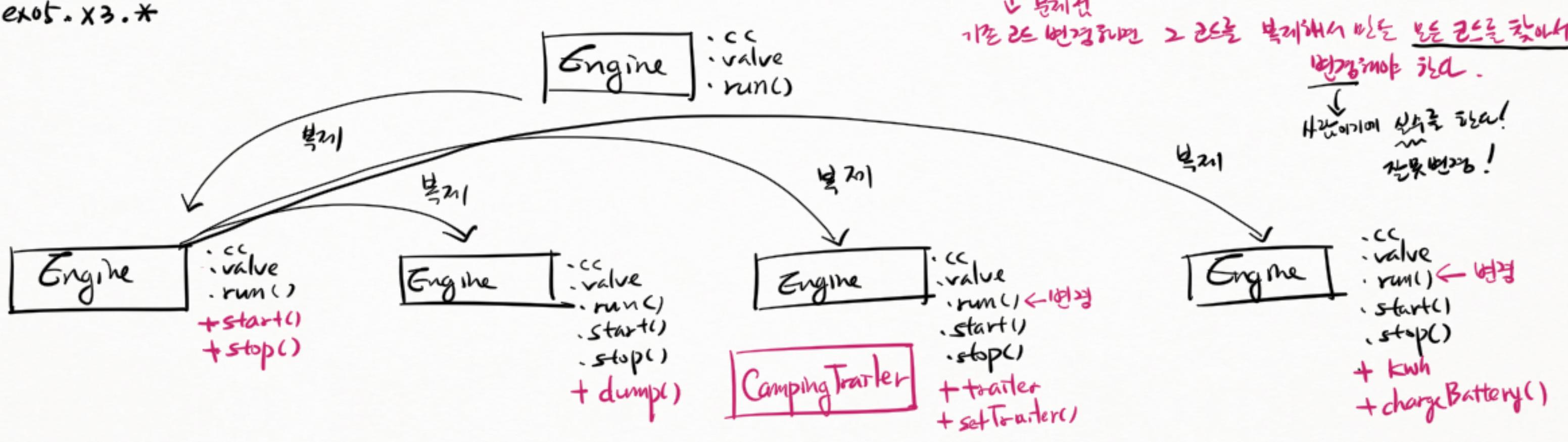
* 기능 확장 방법 1 - 기존 코드에 새 기능 추가

oop.ex5.x2.*



* 기능 확장 방법 2 - 복제한 코드에 새 기능 추가

oop.ex05.*



① 자동차 만들기
app1

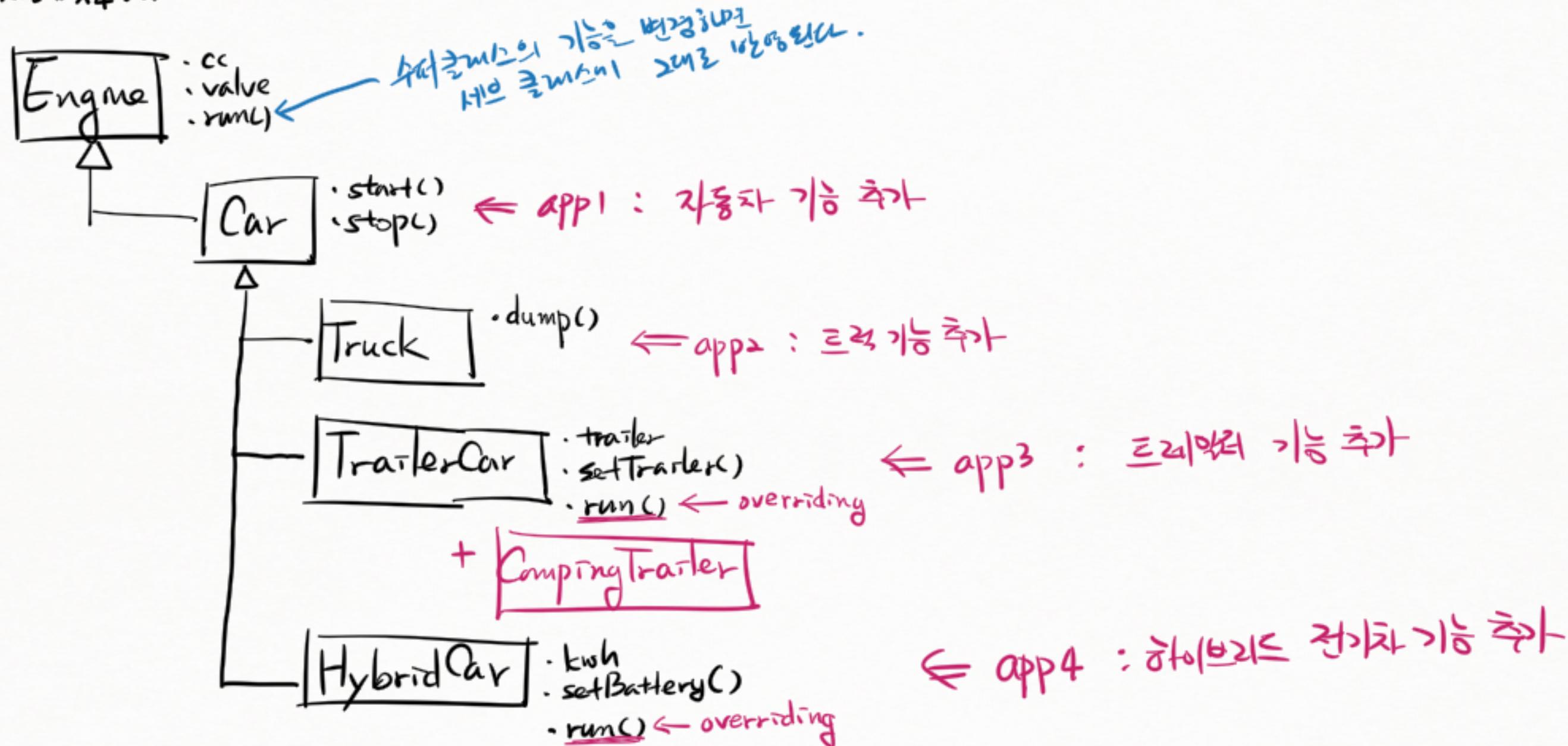
② 트레일러 만들기
app2

③ 캠핑카 만들기
app3

④ 하이브리드 전기차 만들기
app4

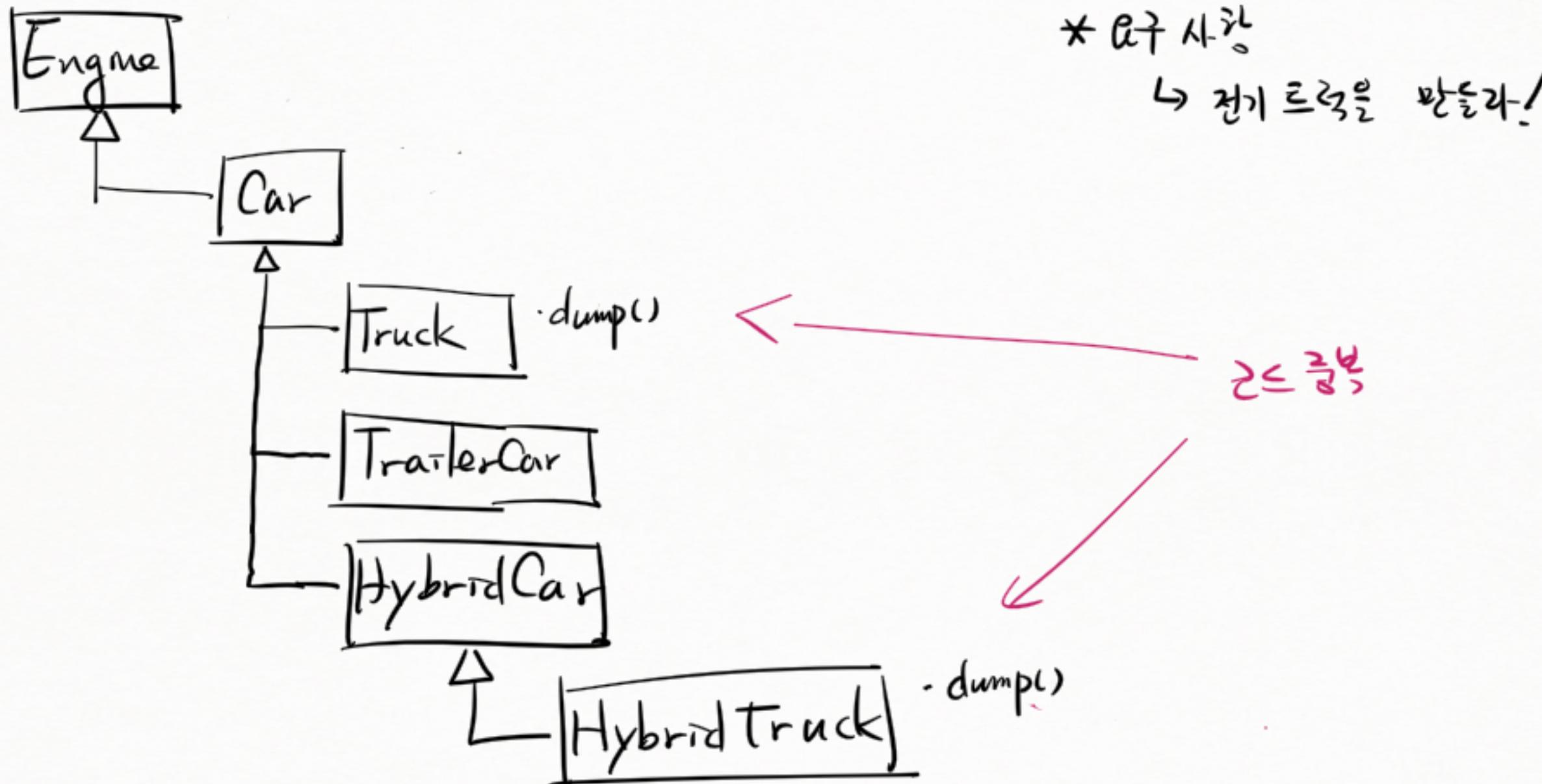
* 기능 확장 방법 3 - 상속을 이용한 확장.

oop.ex05.x4.*



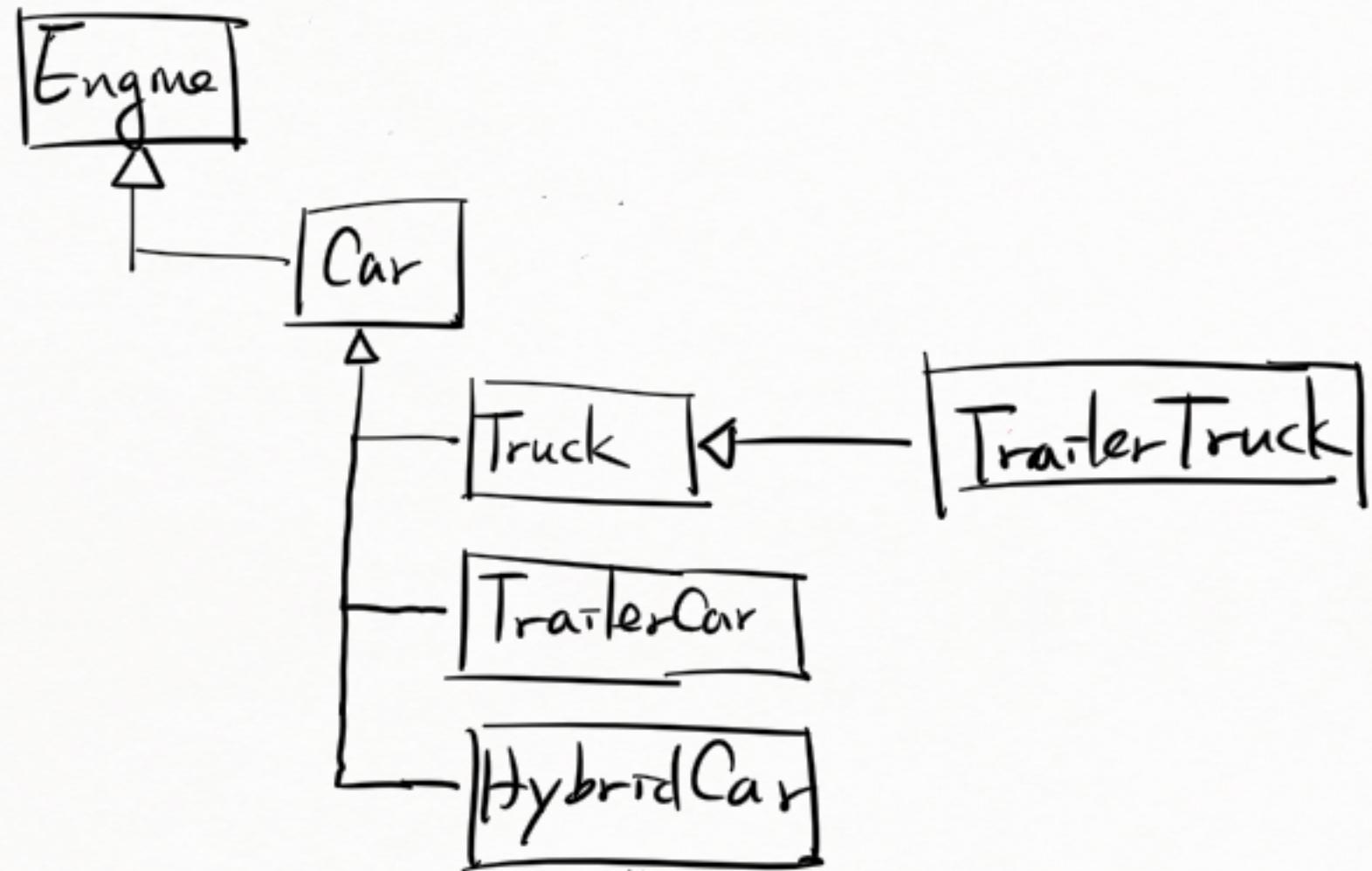
* 기능 확장 방법 3 - 상속을 통한 기능 확장의 한계

oop.ex05.x4.*



* 기능 확장 방법 3 - 상속을 통한 기능 확장의 한가지

oop. ex05. x4.*



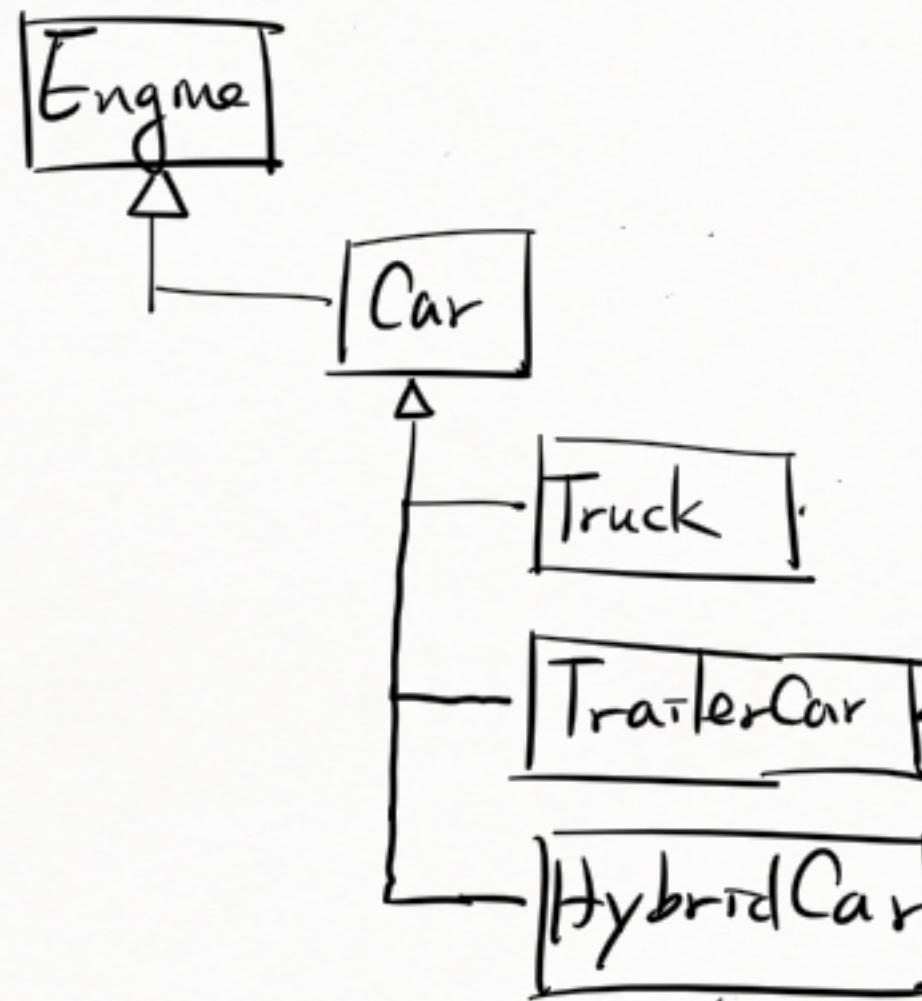
* 구조 사용

↳ 트레일러를 뒤에 두 수 있는 트럭

· trailer
· setTrailer()

* 기능 학습 18주 3 - 상속을 통한 기능 학습의 한계

oop. ex05. x4.*



* 예시 사용

↳ 전기 캐리어카

* 이렇게 기능 조합을 하려면
수행은 퍼블릭 메소드가 대체된다

- kwh
setBattery()

(상속으로 대상은 기능이 조합된
개체로 만든다면 가능)

↳ 유지보수는 어렵지만

* 기능학적 특성 4 : 디자인에서 기초 기능

기능적

Sedan

Truck

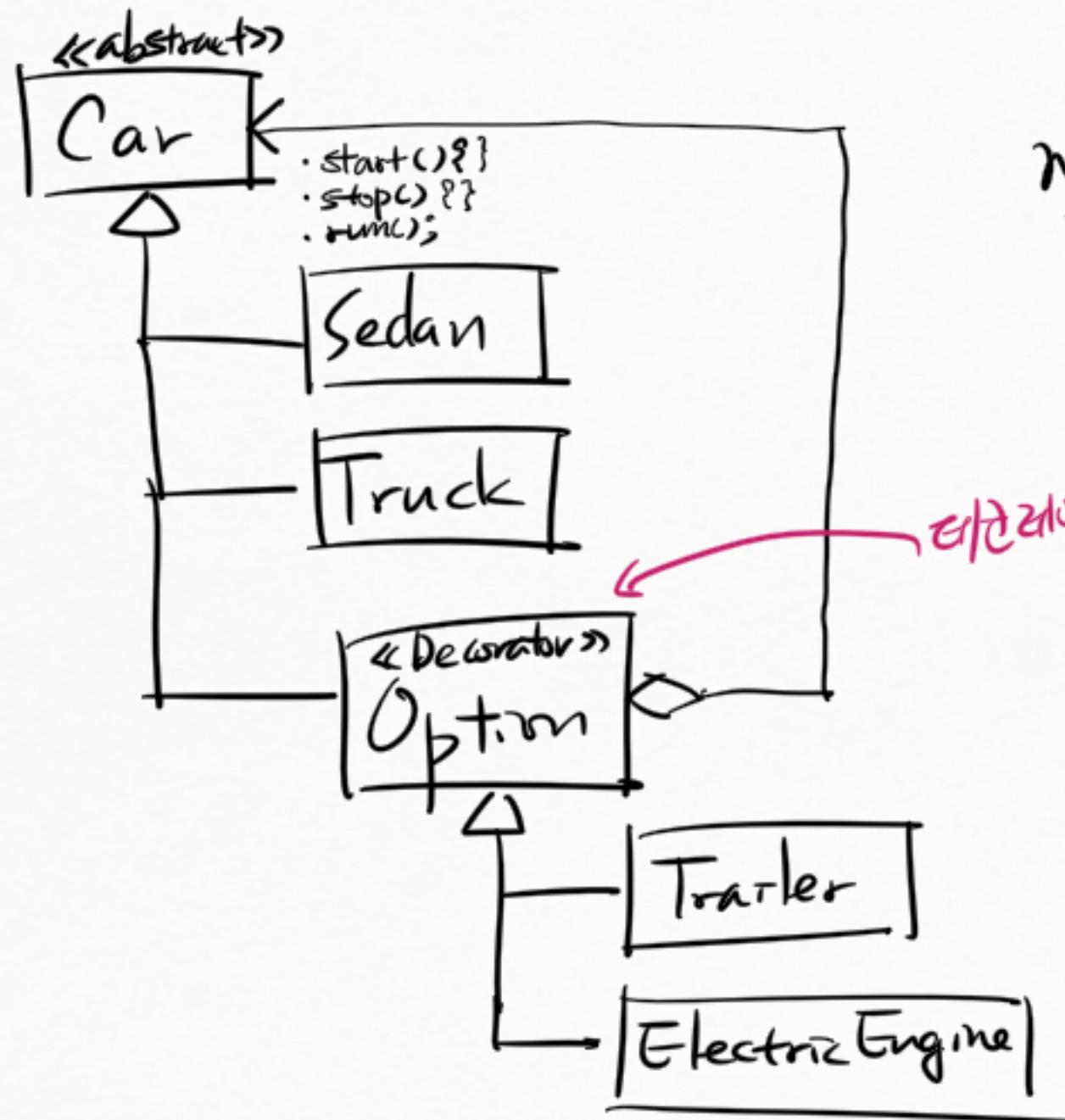
선택적

Gas Engine

Electric Engine

Trailer

* 테러리터 km 기능



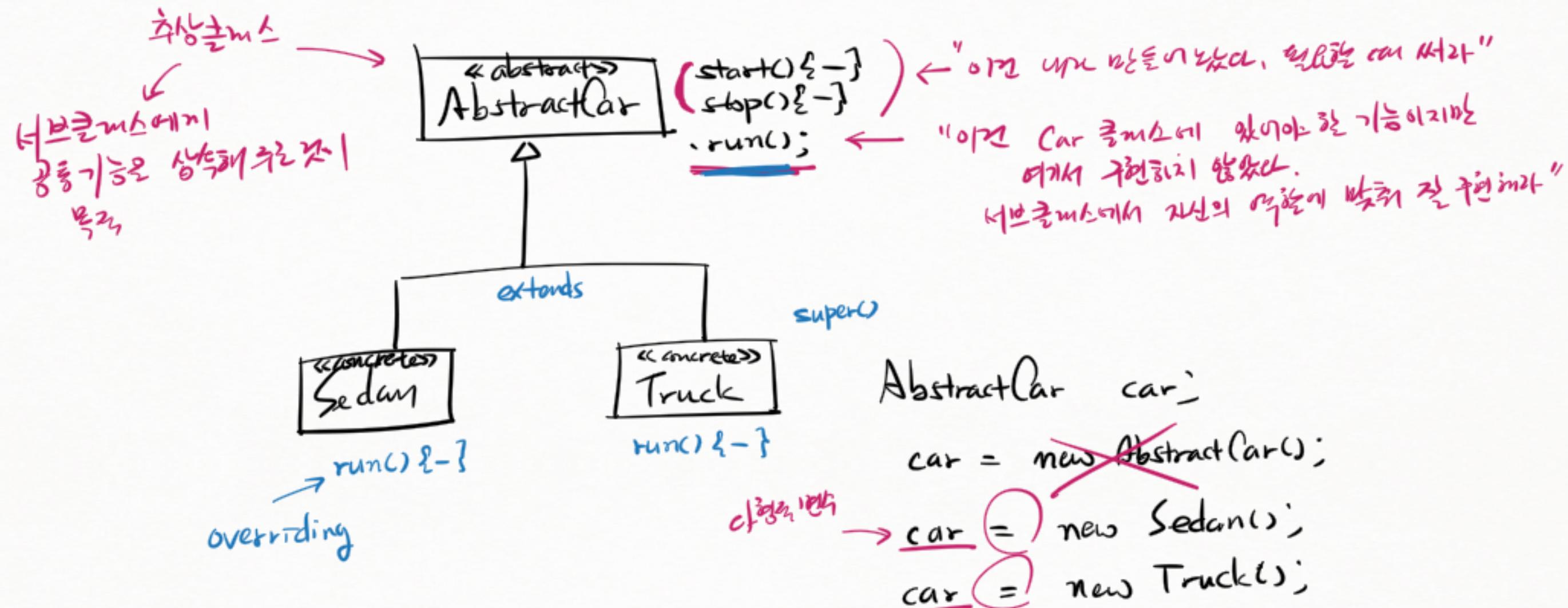
new Optim(new Optim(new Optim(주행기)))

new Optim(주행기)

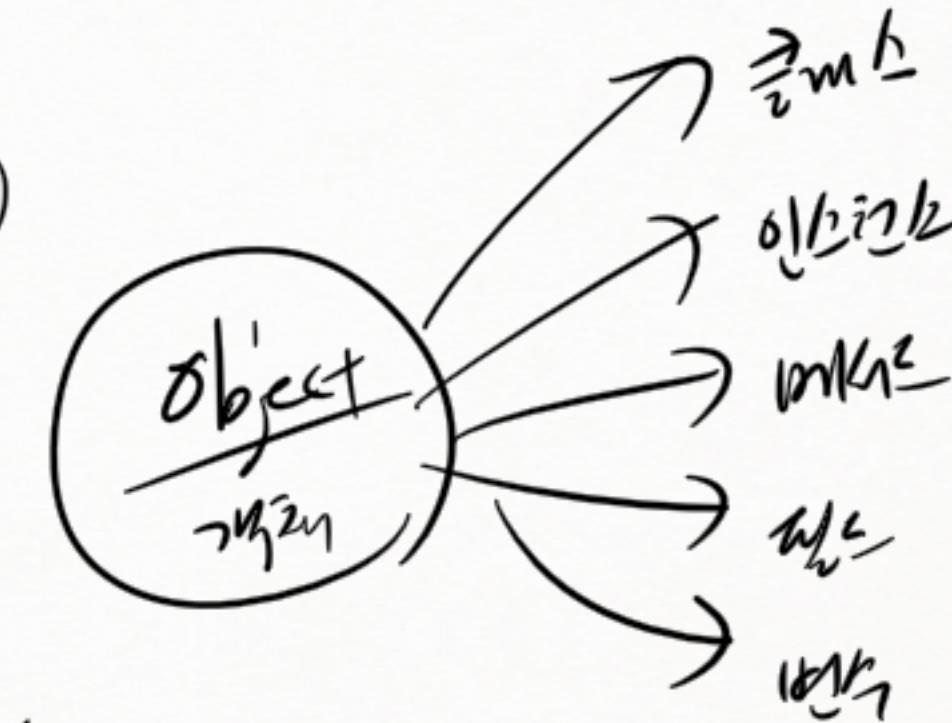
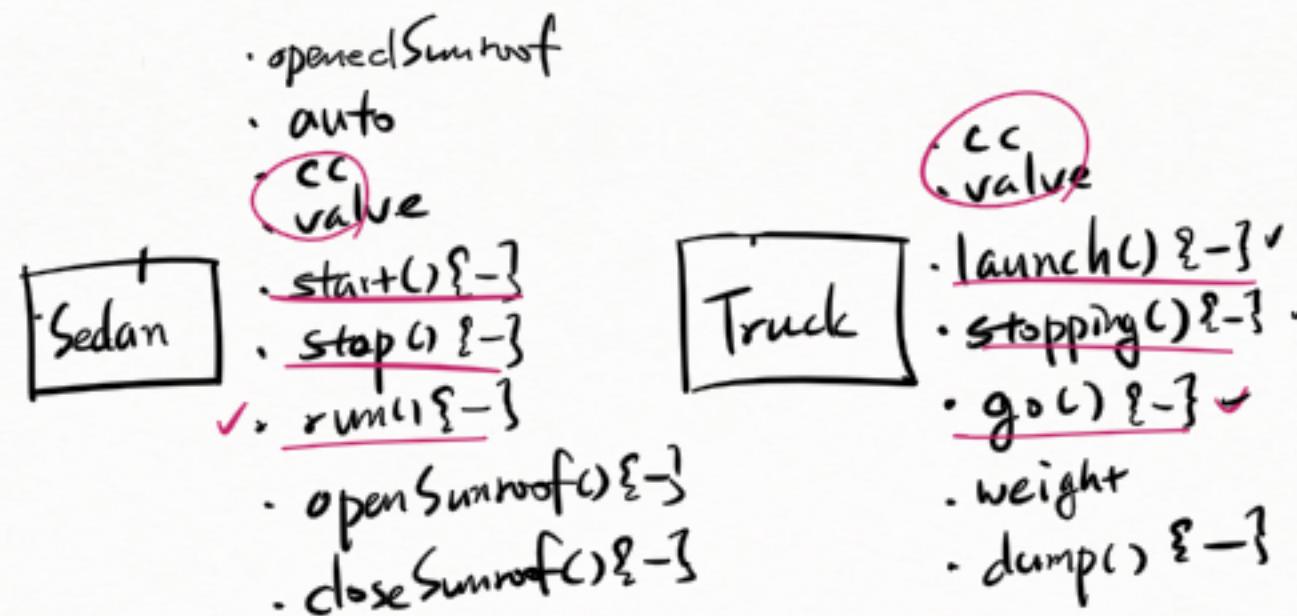
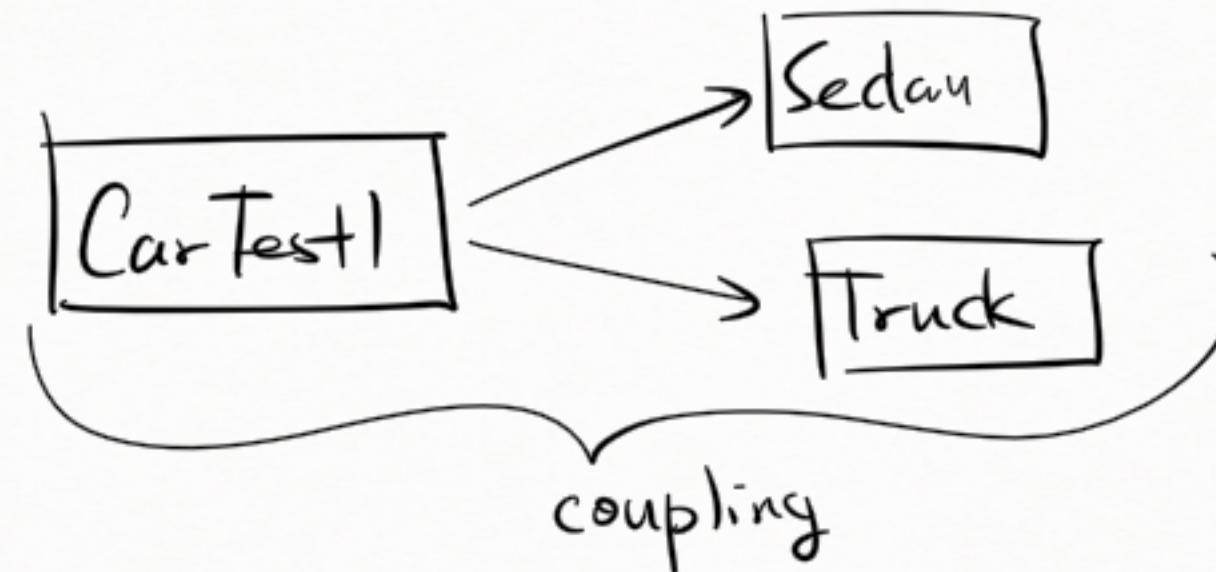
기본
클래스
주행기
Optim

디코레이터 : 주행기에게 물이로 선택 가능

* 추상클래스와 구현

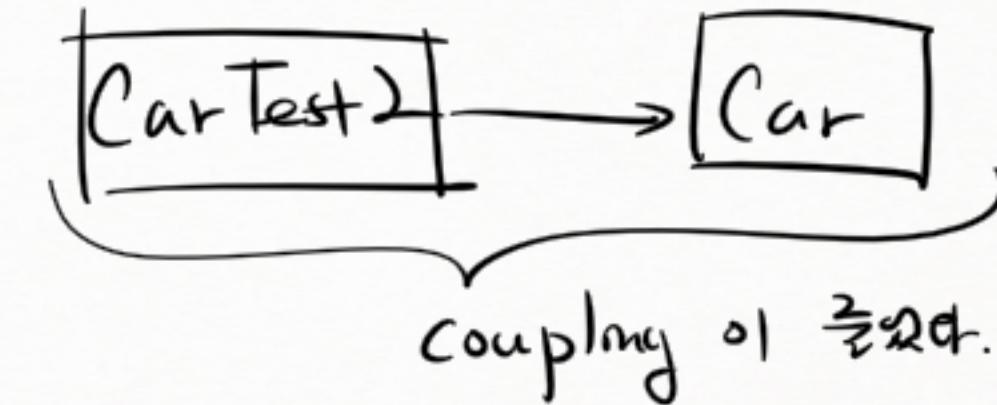
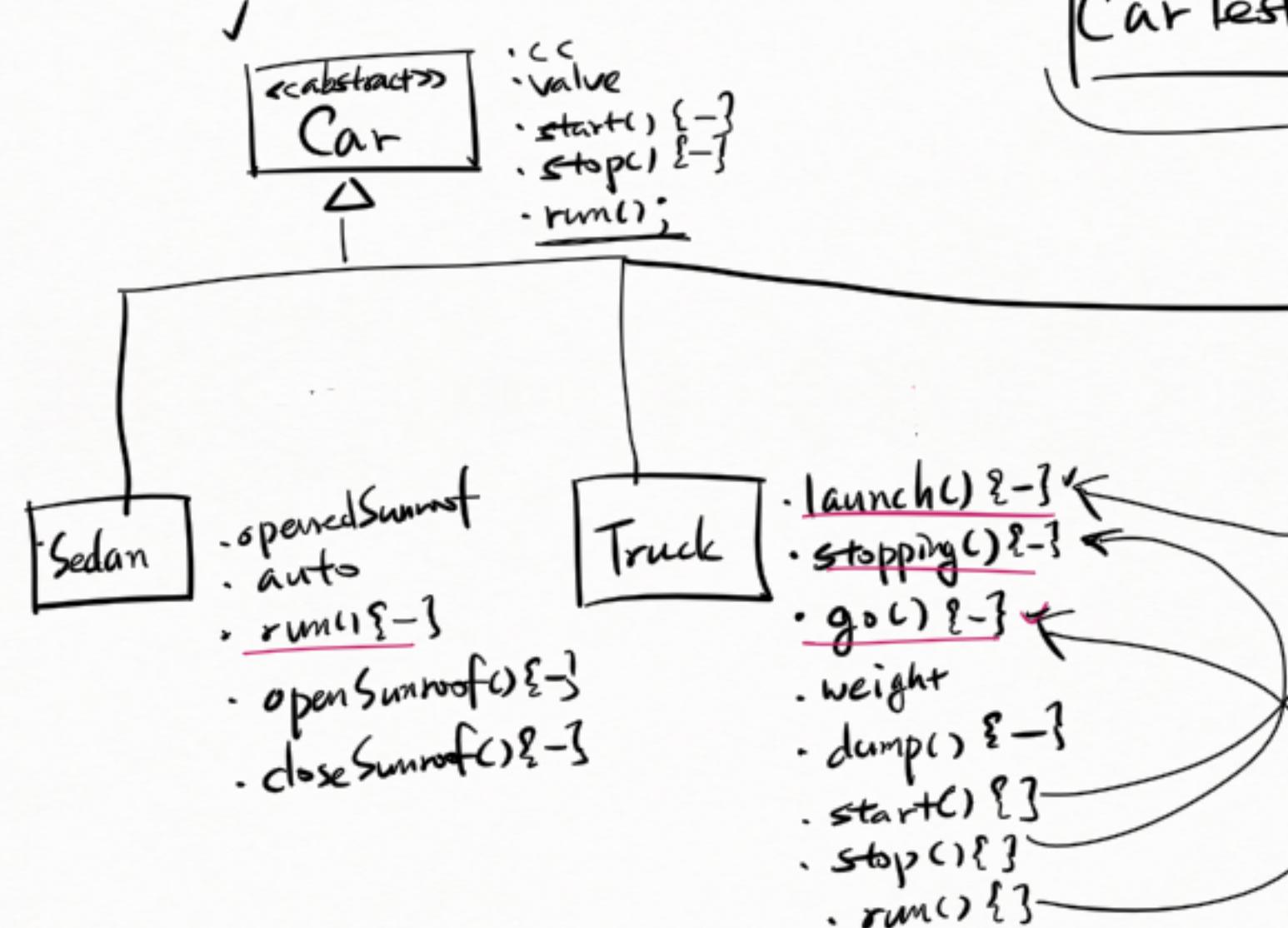


* 실전 프로그래밍 1 단계



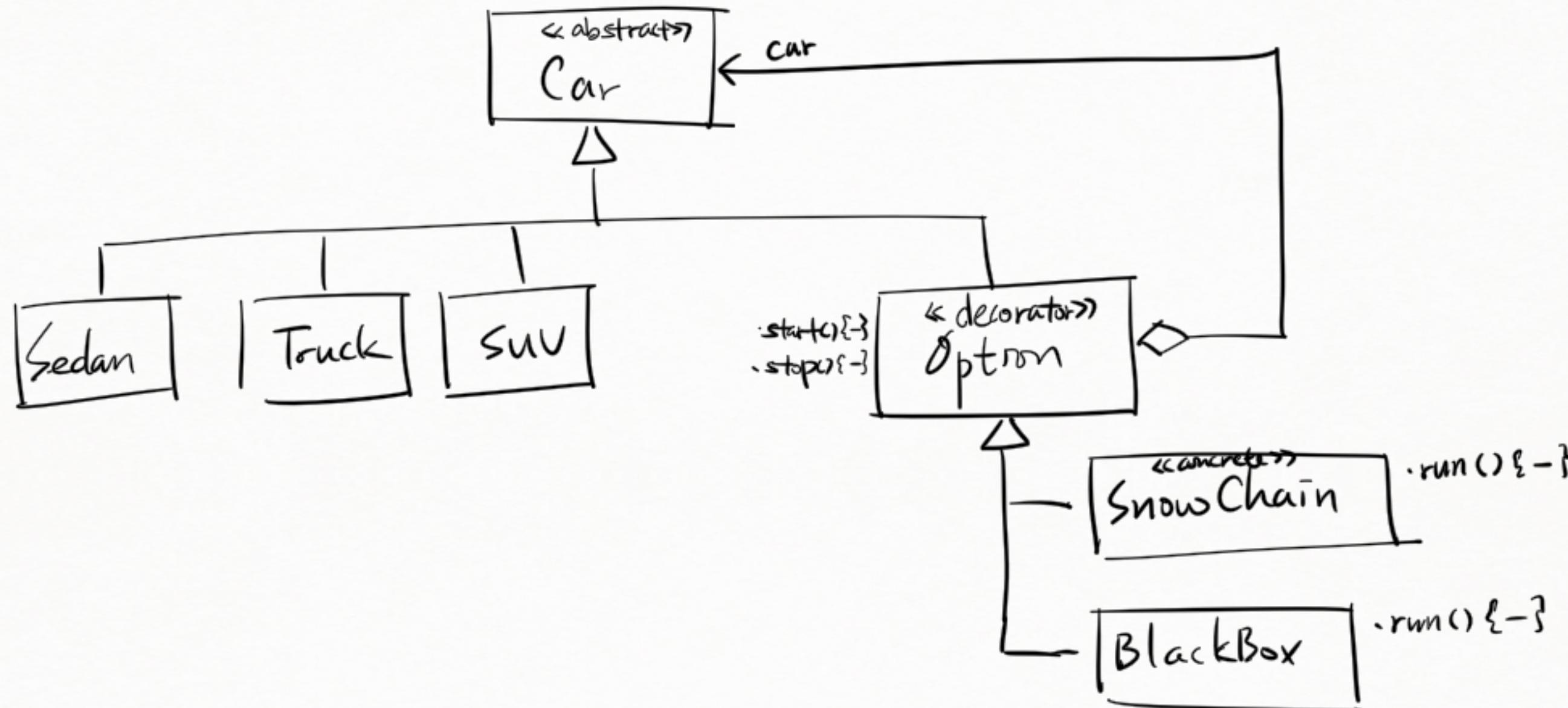
* 상 하 한 한 한 한 한 - 품종, 품질, 성능

\hookrightarrow : generalization

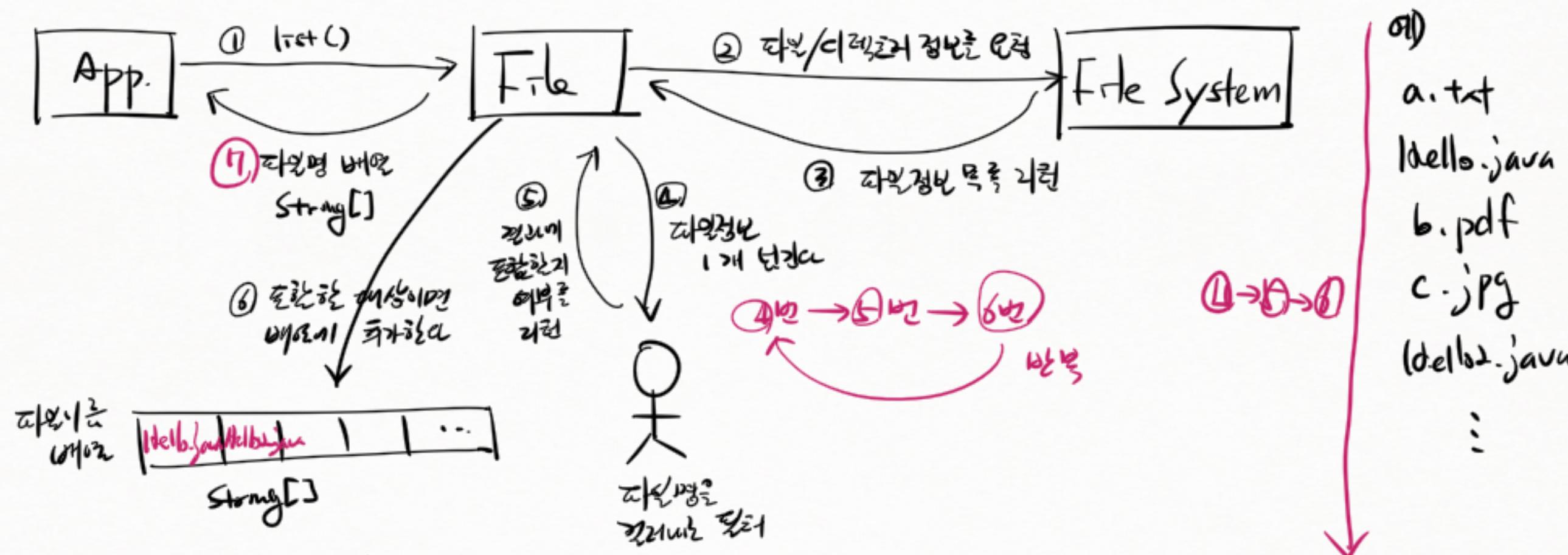


- enableLwd: boolean
- activateLwd(boolean)
- run() {-}

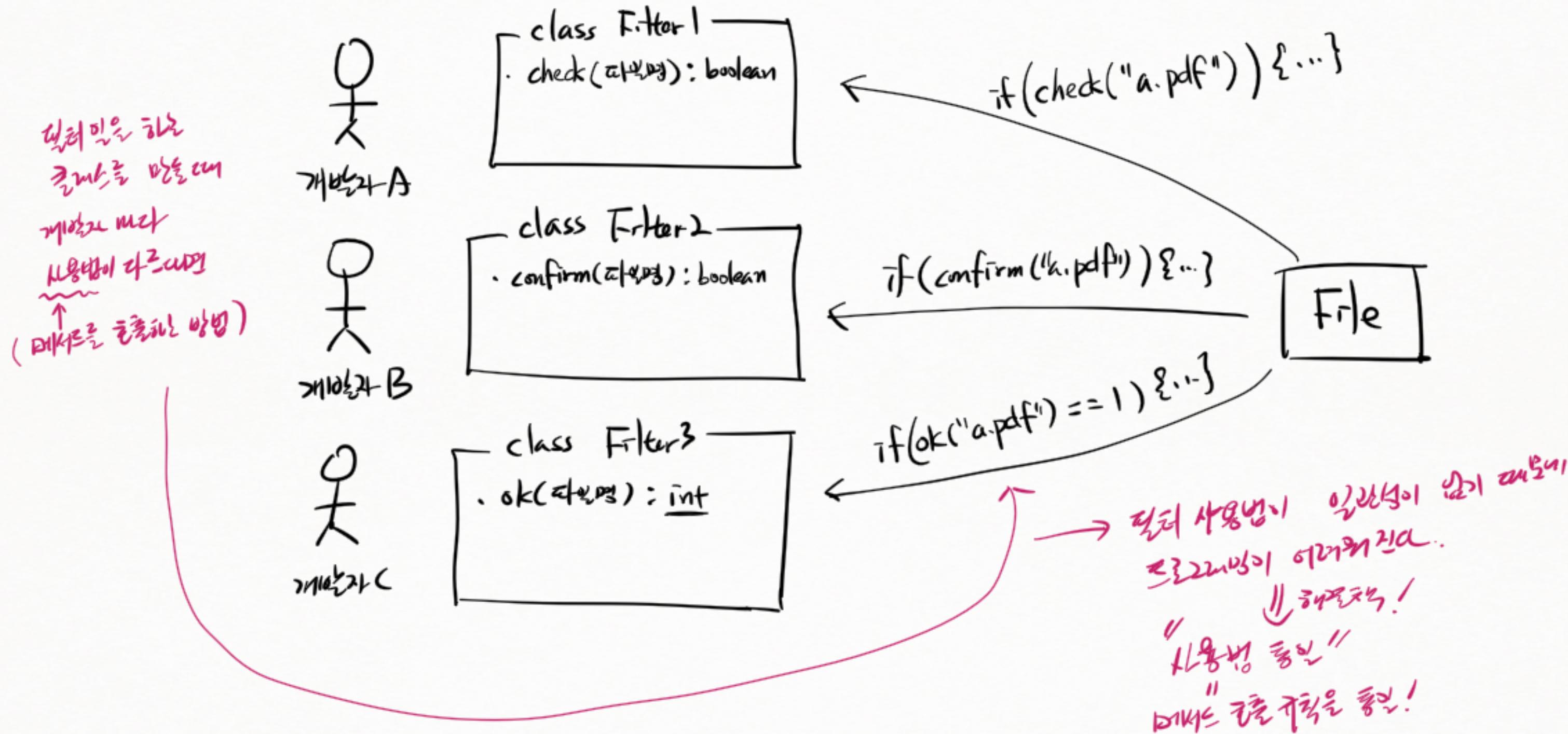
* 차선을 3개로 나누면 차량이 1개로 통합



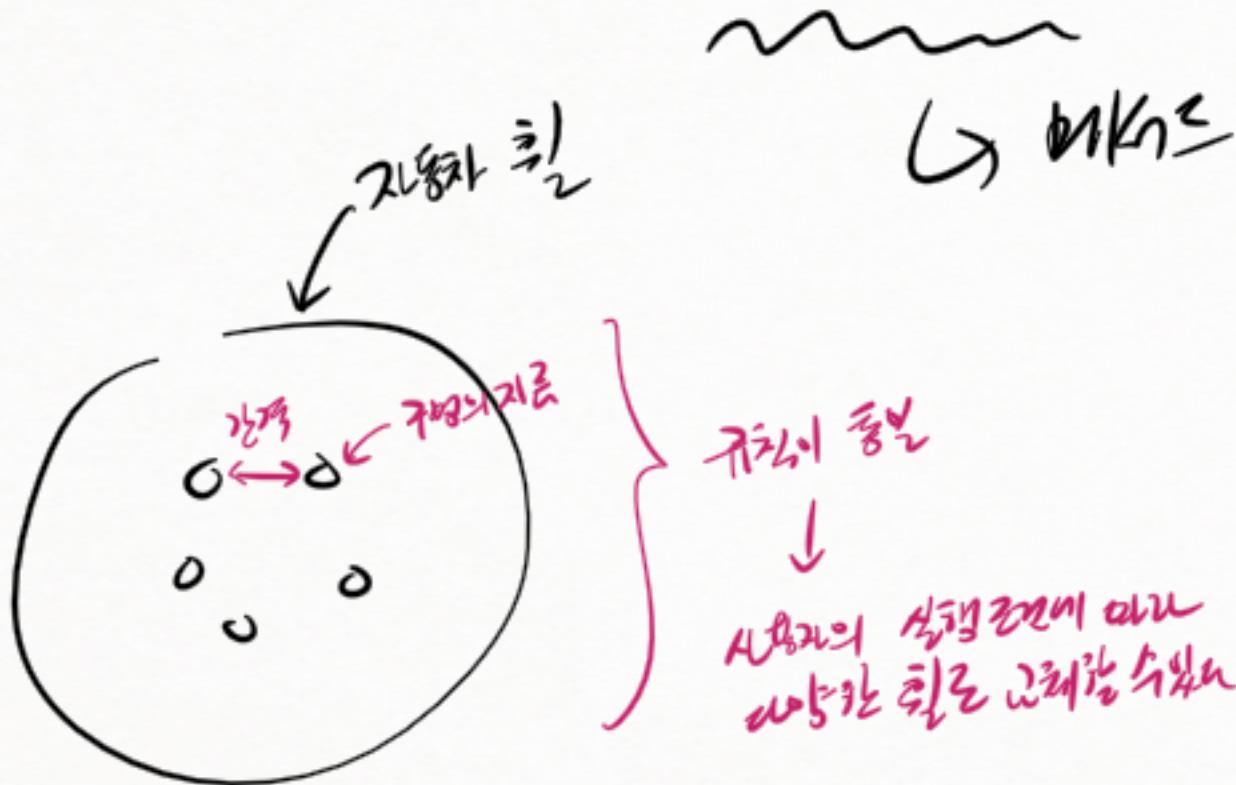
* 파일 개체와 파일명 풀터의 작업 과정.



* 파일 검색과 필터의 관계



* 인터페이스



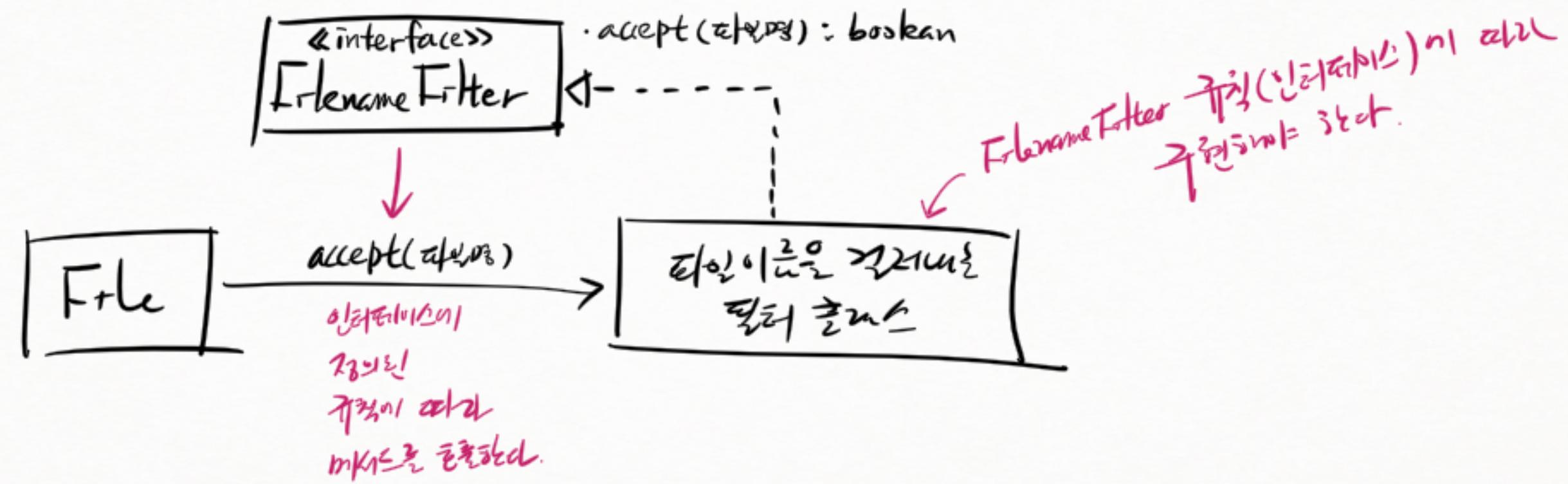
↳ 비클래스 허용 규칙을 정의하는 방법

- 비클래스를 클래스로 사용하기 상관없이
동일한 시그니처(비클래스, 라이브러리, 라이브린)를 갖는 비클래스를 정의한다.

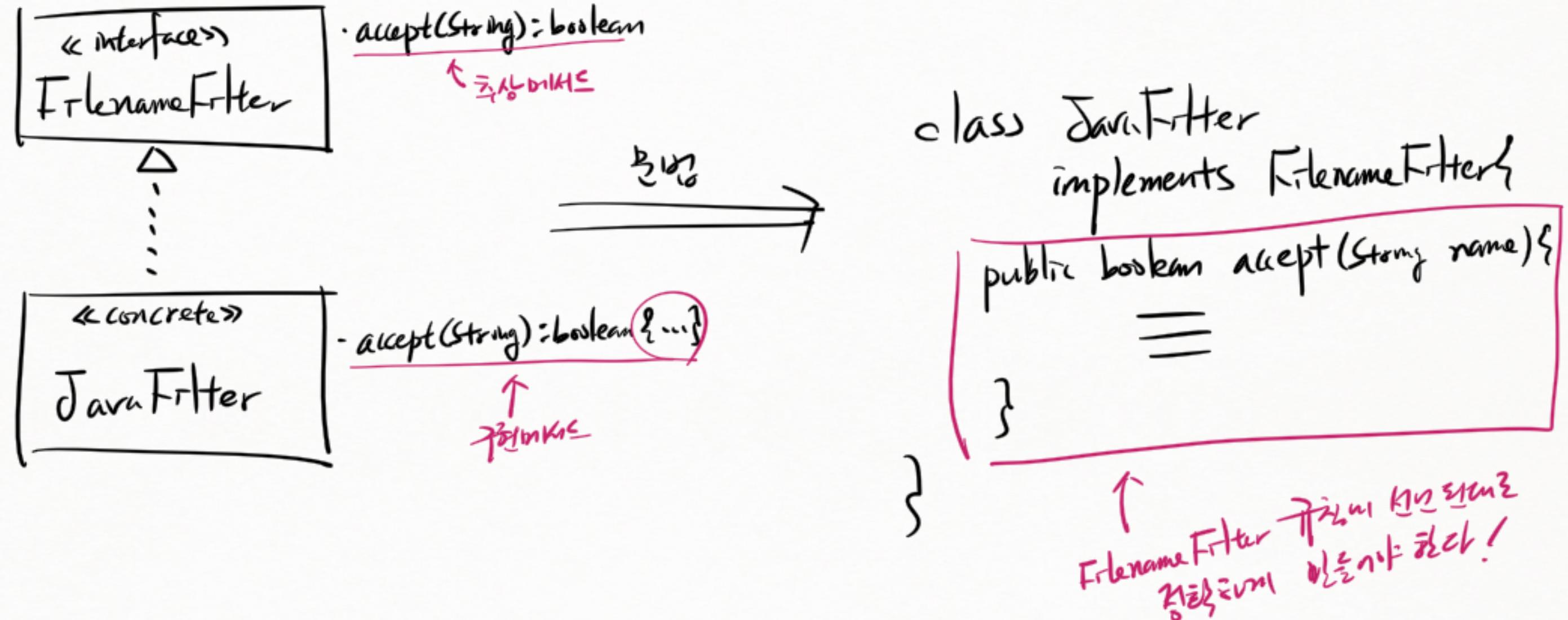
↳ 비클래스를 사용하는 클래스가 상관없이
일반적 형식으로 사용할 수 있도록
존재하는가 수월진다.
기체의 고체가 존재.

비클래스를 사용하는 클래스가 상관없이
일반적 형식으로
사용할 수 있다!
↓
비클래스 고체는 가능!

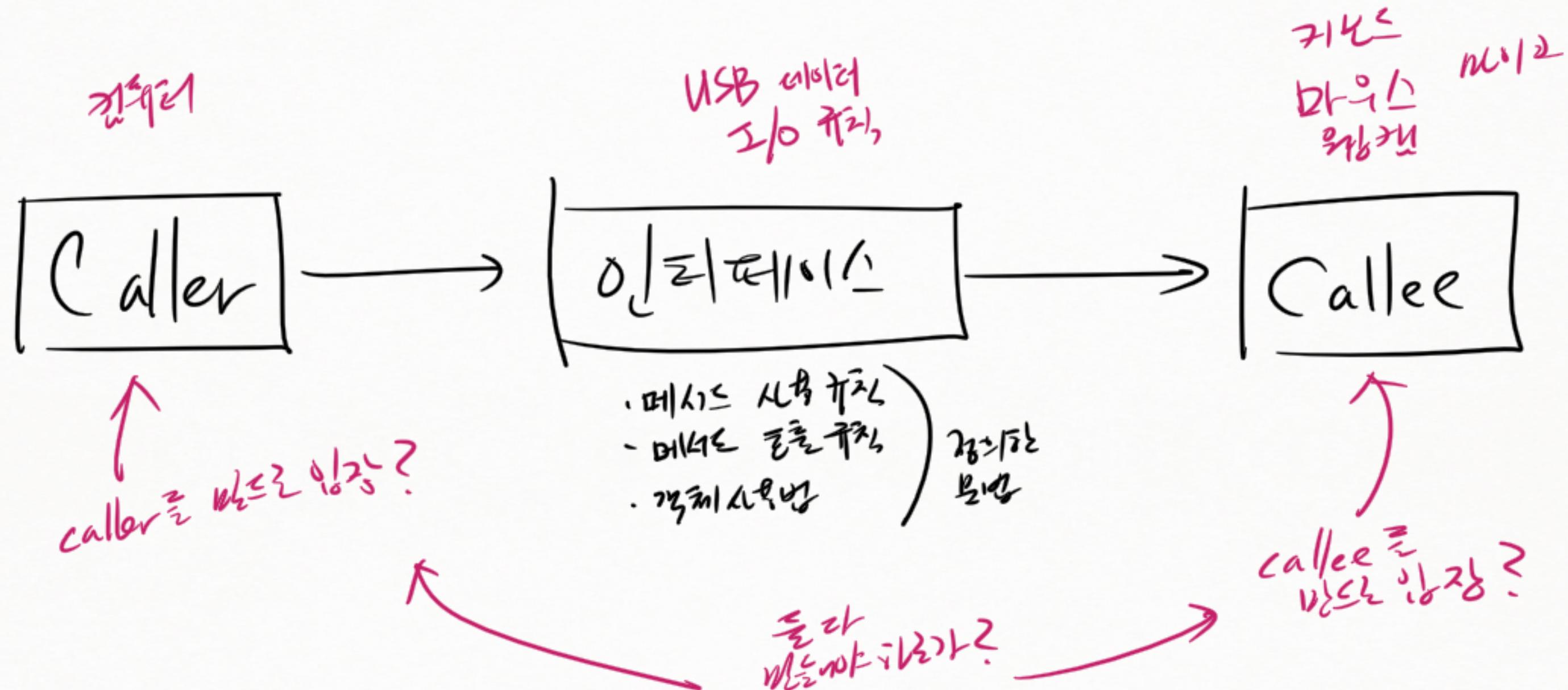
* 자료 필터와 인터페이스



* FilenameFilter 인터페이스 구현



* OOP 인터페이스



* FilenameFilter 인터페이스

