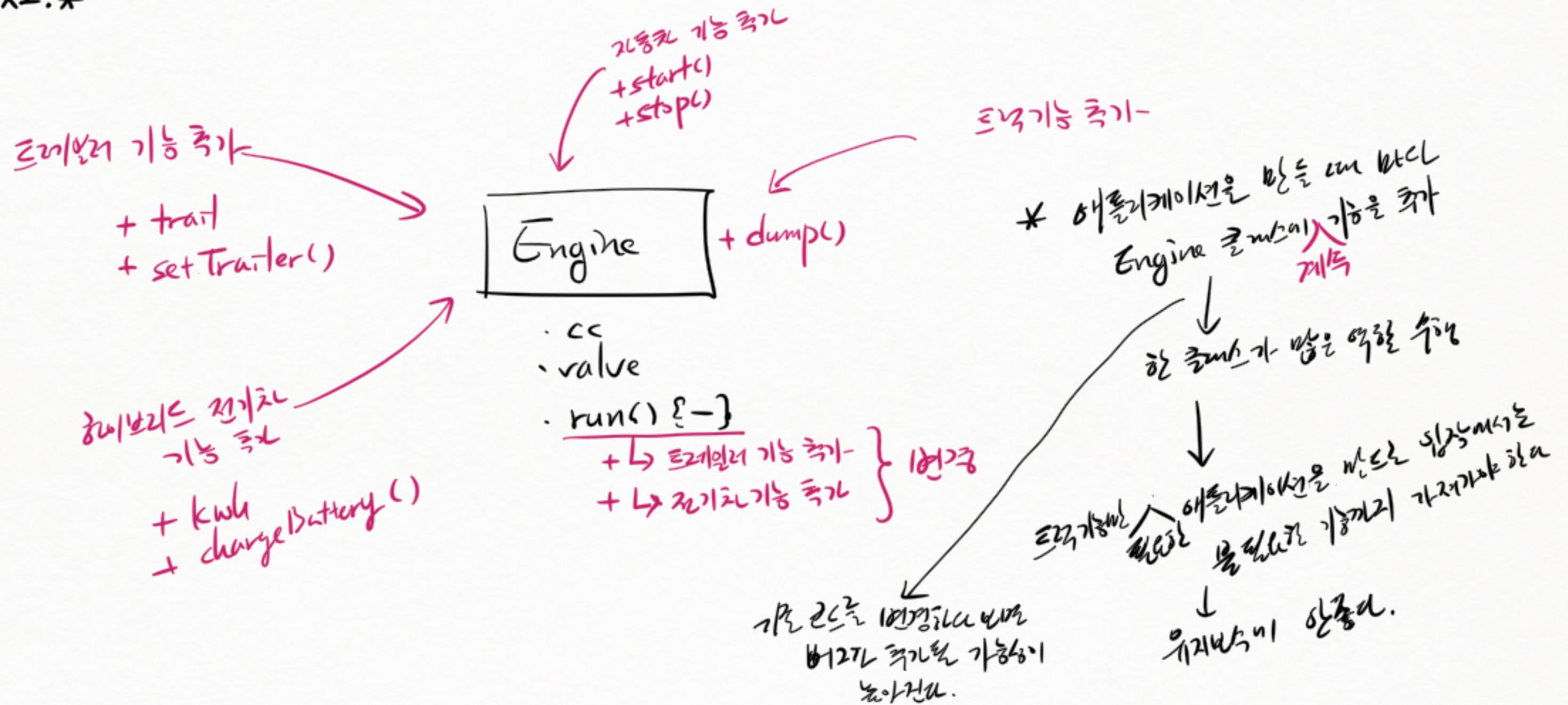


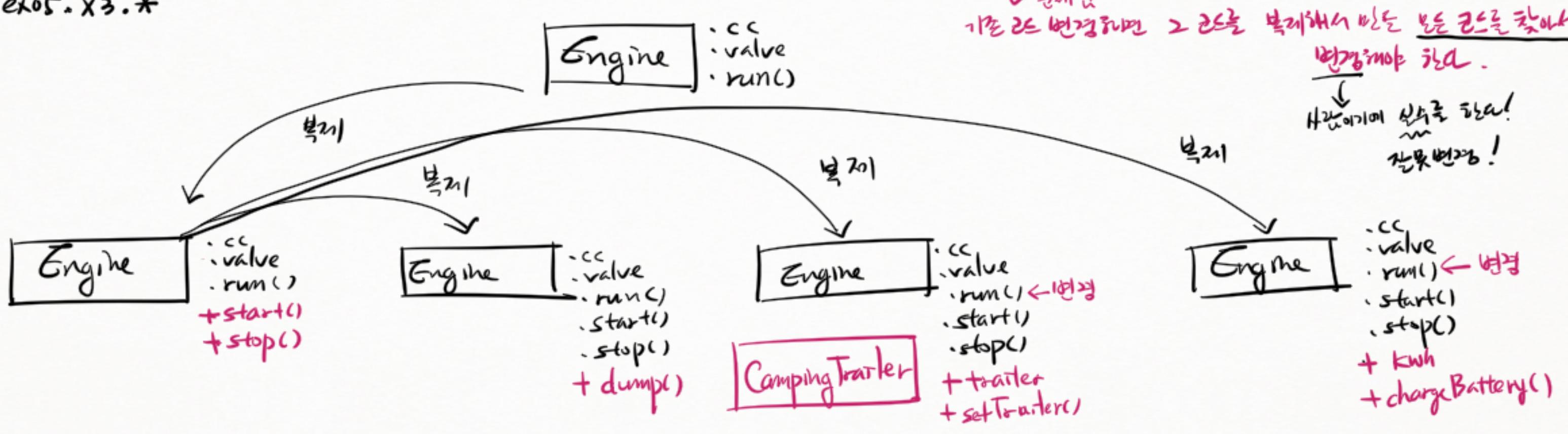
* 기능 확장 방법 1 - 기존 코드에 새 기능 추가

oop.ex5.x2.*



* 기능 확장 방법 2 - 복제한 코드에 새 기능 추가

oop.ex05.*



① 자동차 만들기
app1

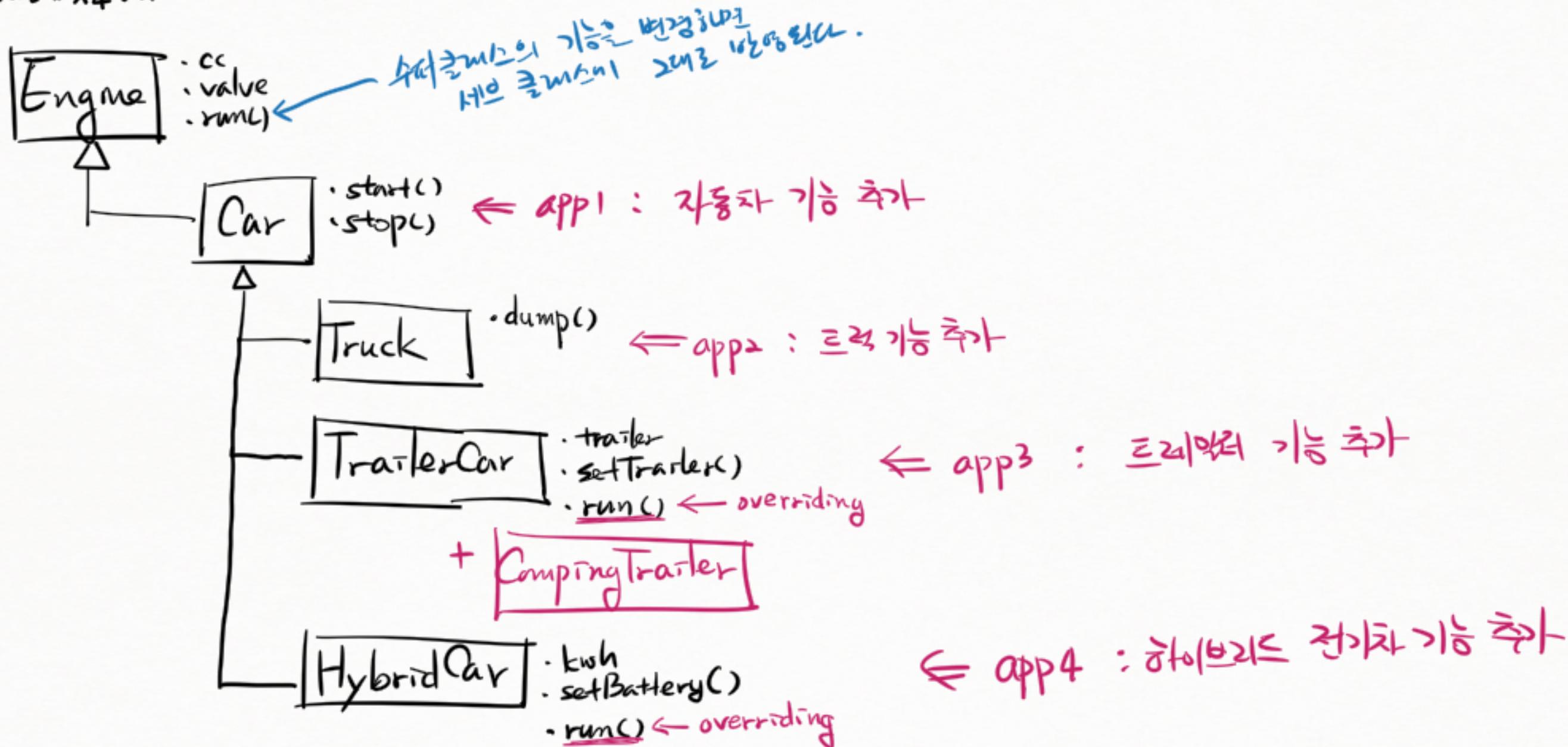
② 트레일러 만들기
app2

③ 캠핑카 만들기
app3

④ 헤드라이트 전자화!
app4

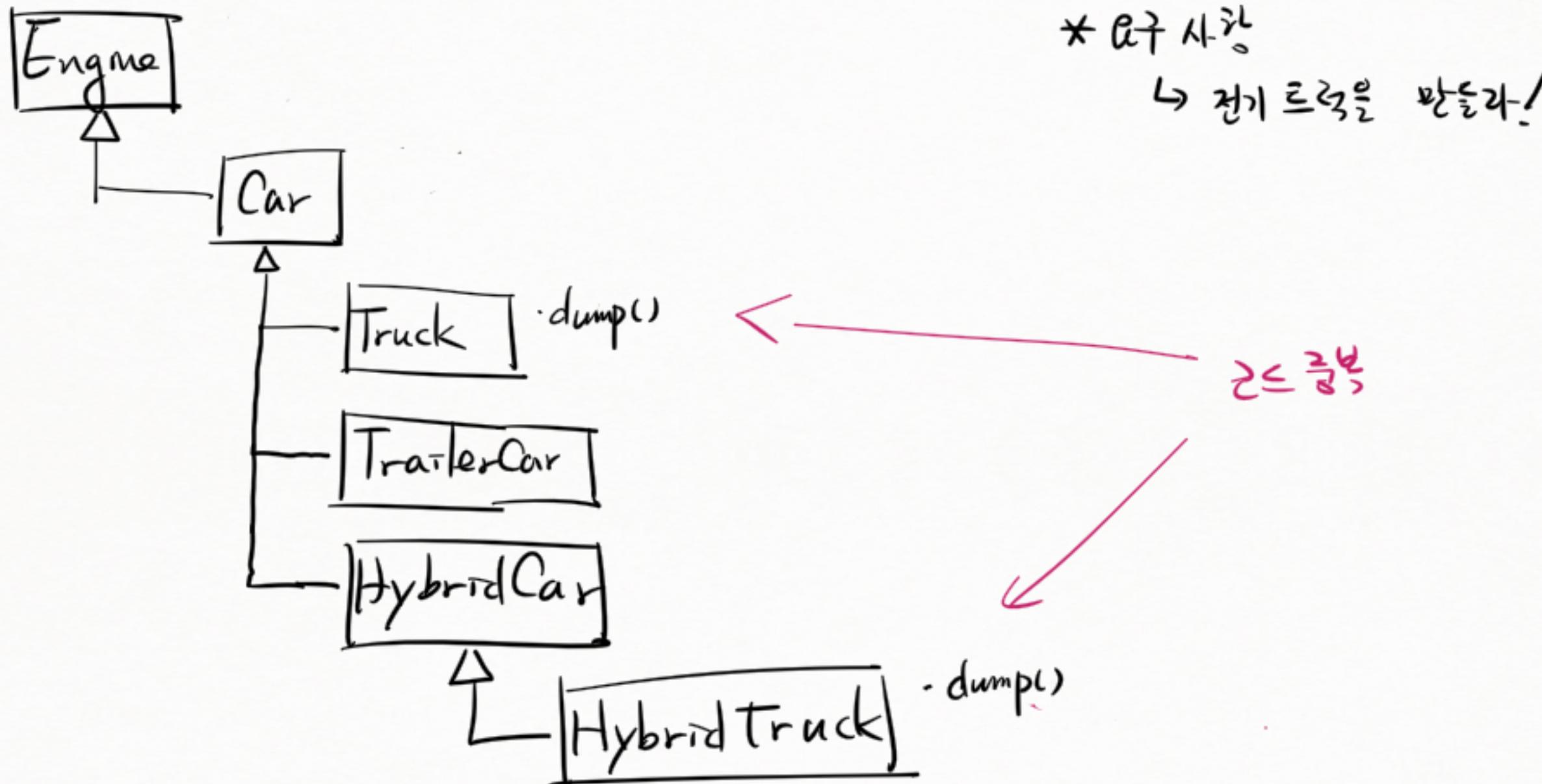
* 기능 확장 방법 3 - 상속을 이용한 확장.

oop.ex05.x4.*



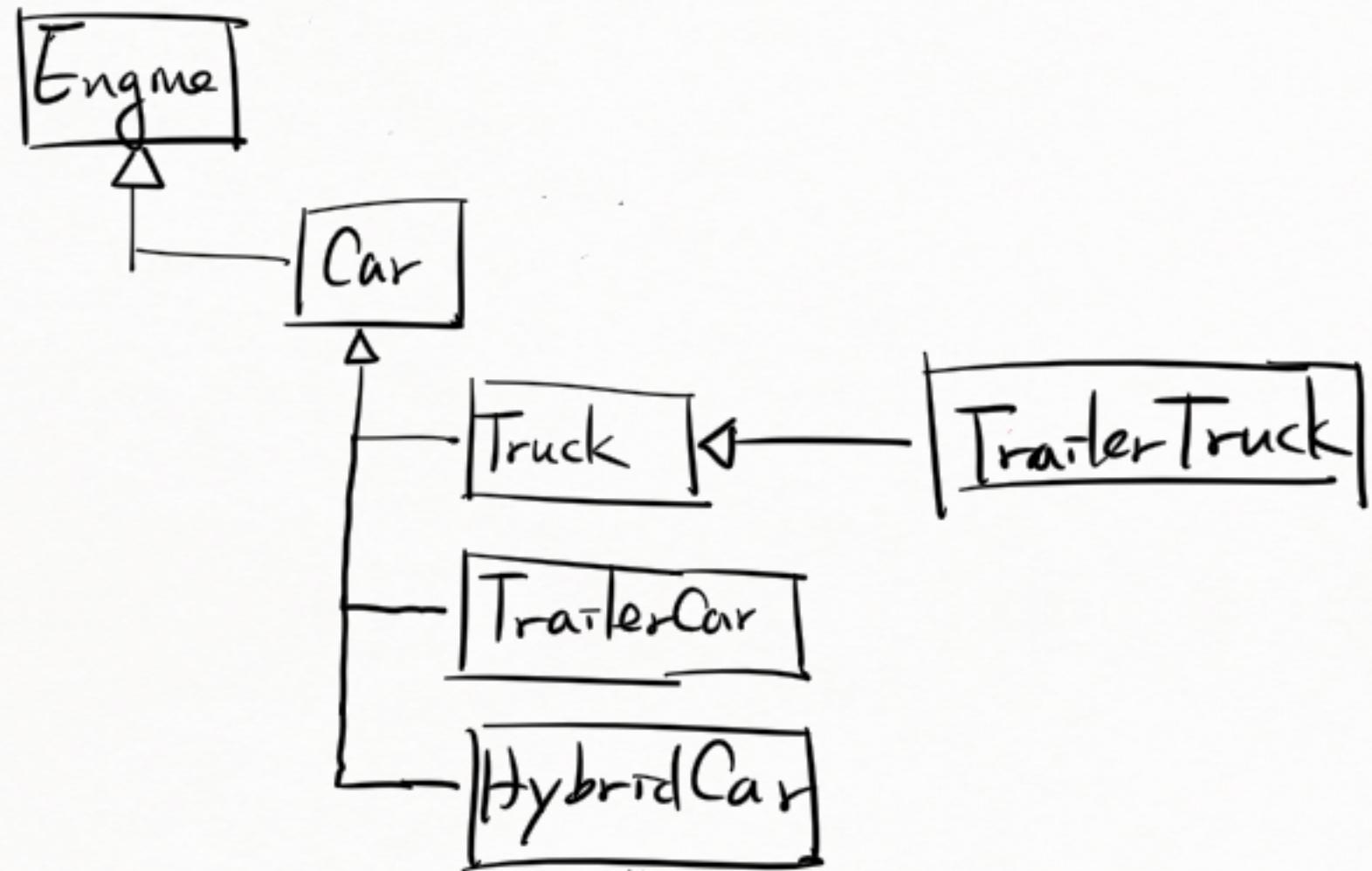
* 기능 확장 방법 3 - 상속을 통한 기능 확장의 한계

oop.ex05.x4.*



* 기능 확장 방법 3 - 상속을 통한 기능 확장의 한가지

oop. ex05. x4.*



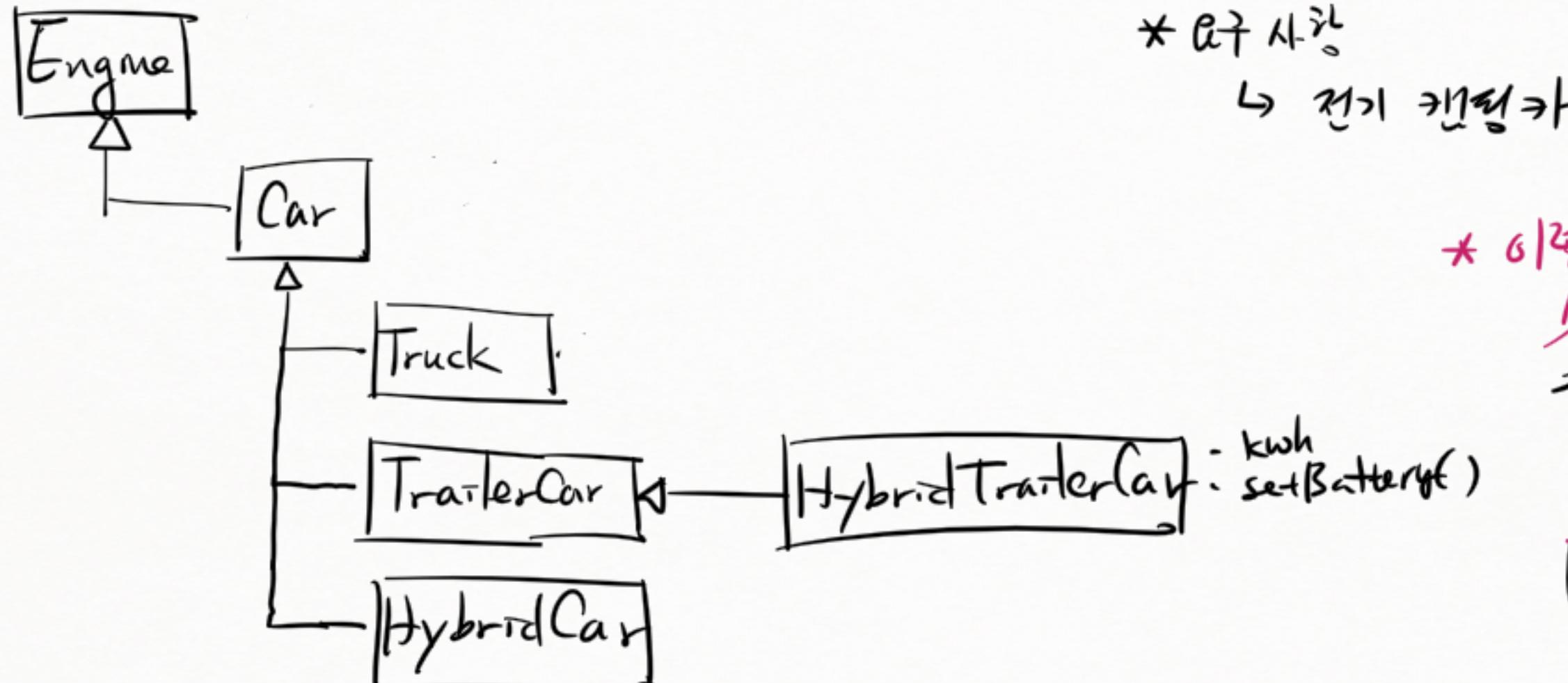
* @구사장

↳ 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 네트워크로 구성된다

(상속으로 대상간 기능이 조합된
개체를 만들기 어렵다.)

↳ 유지보수는 어렵지만...

* 기능학적 특성 4 : 디자인에서 기초 기능

기능적

Sedan

Truck

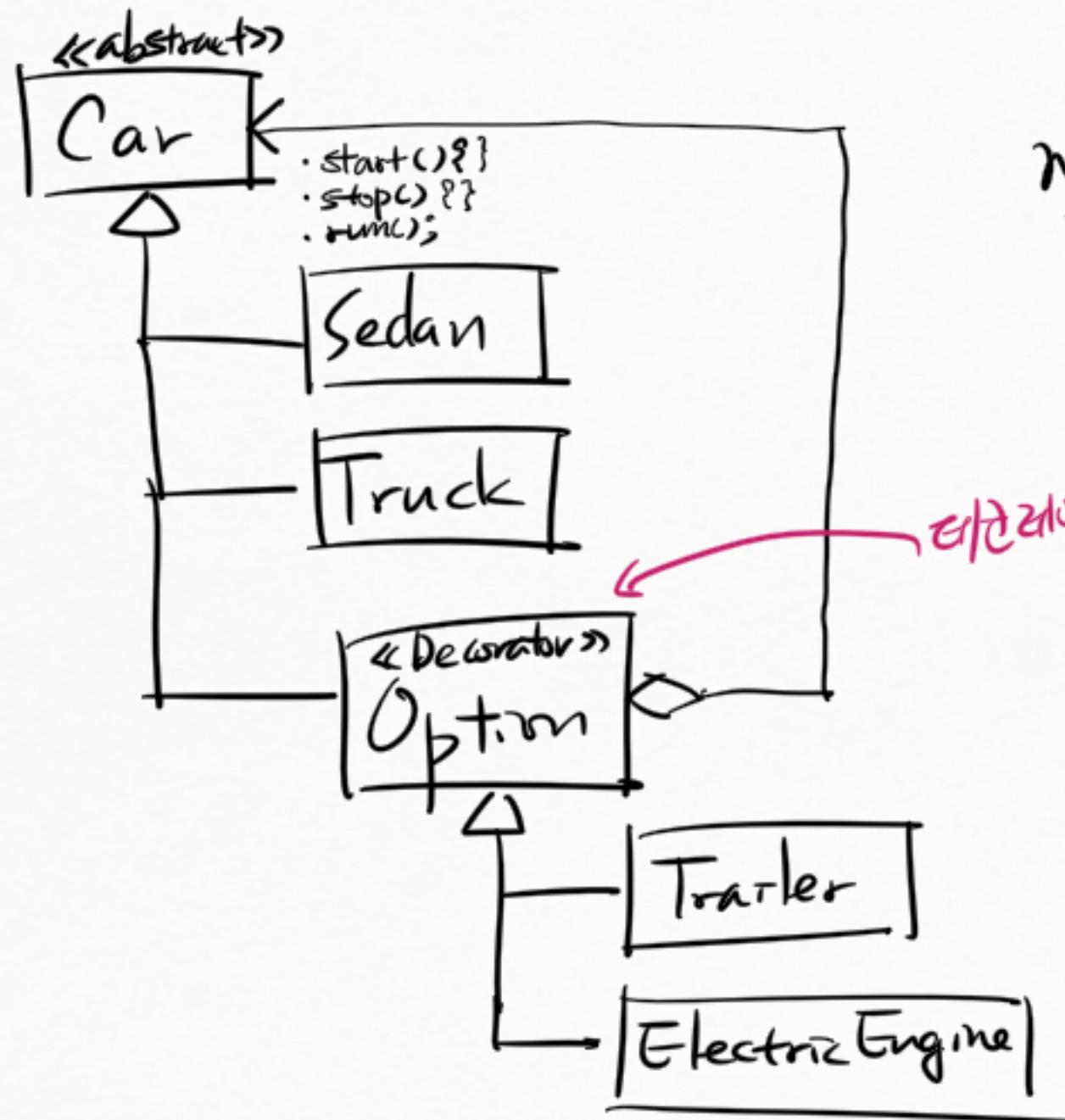
선택적

Gas Engine

Electric Engine

Trailer

* 테러리아 러닝 기법



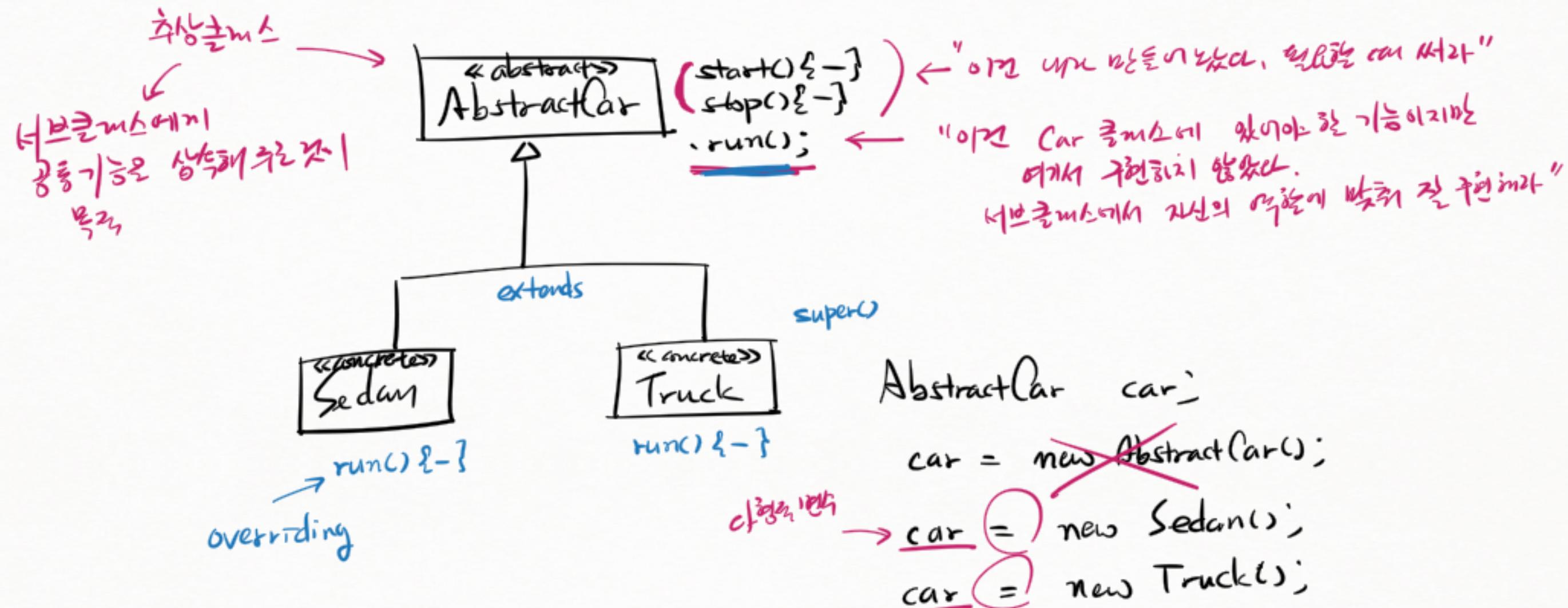
new Optim(new Sedan(new Optim(new ElectricEngine)))

new Optim(Truck)

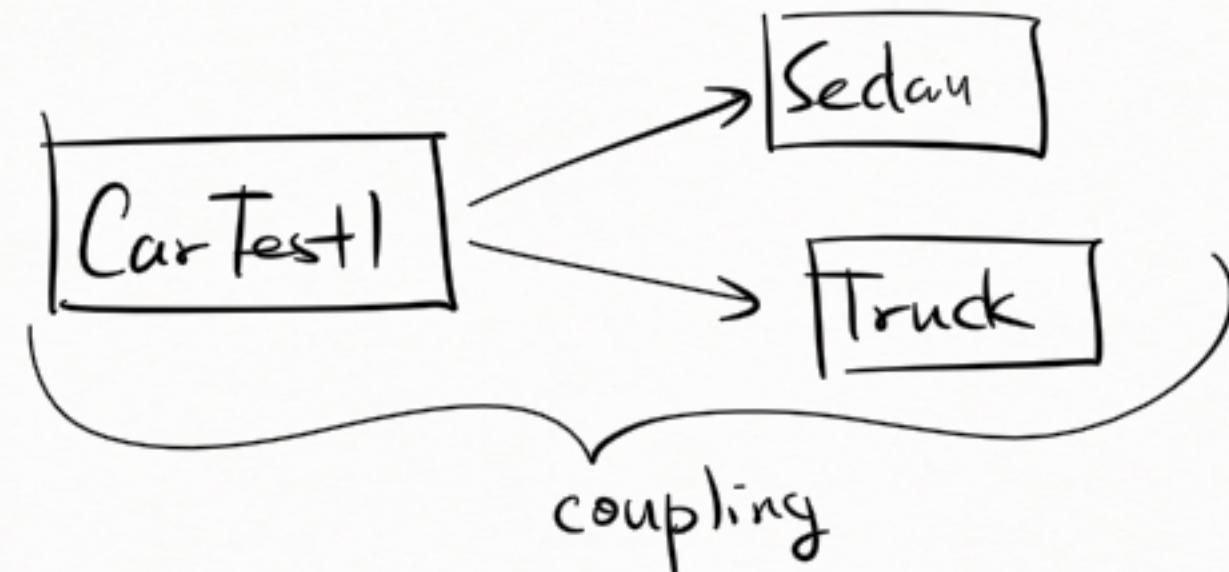
기존
클래스
추가
보다.

기존
클래스
선택 가능

* 추상클래스와 구현



* 실전 프로그래밍 1 단계



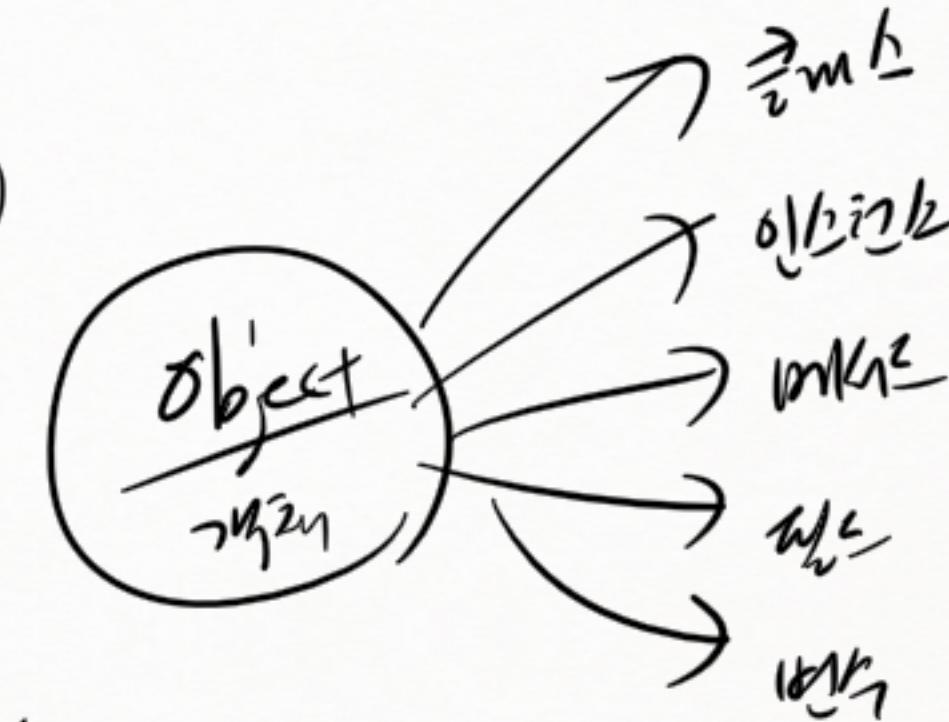
coupling

- openedSunroof
- auto
- **cc value**
- start() { - }
- stop() { - }
- ✓ · run() { - }
- openSunroof() { - }
- closeSunroof() { - }

Sedan

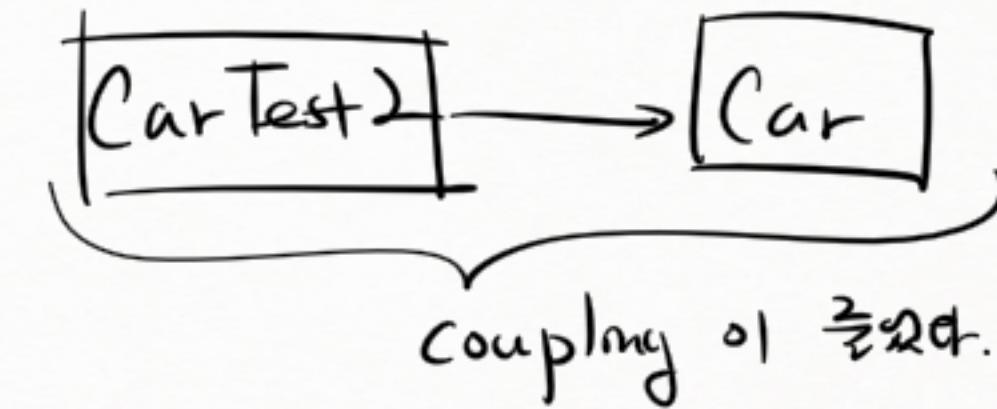
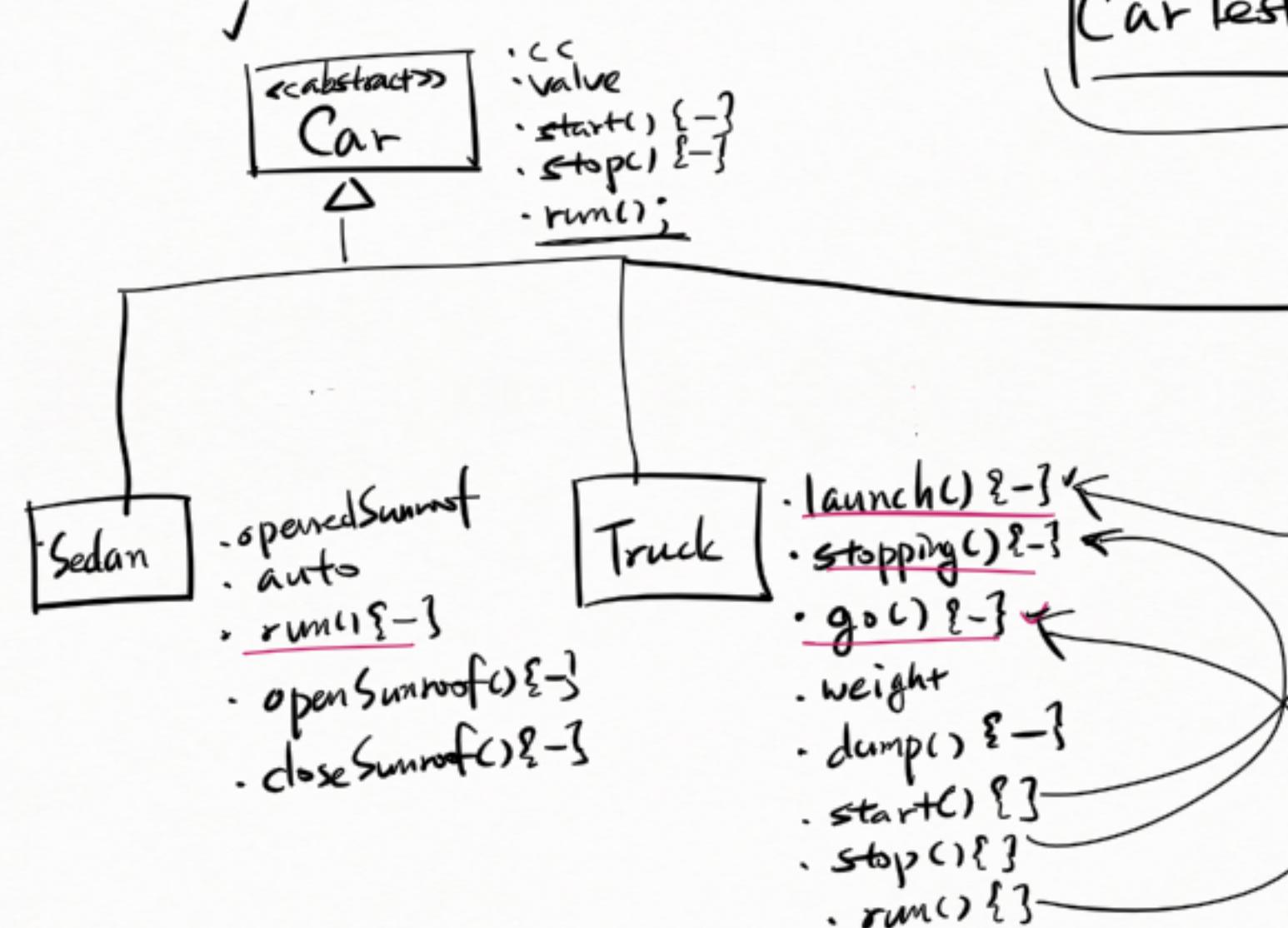
- **cc value**
- launch() { - } ✓
- stopping() { - } -
- go() { - } ✓
- weight
- dump() { - }

Truck



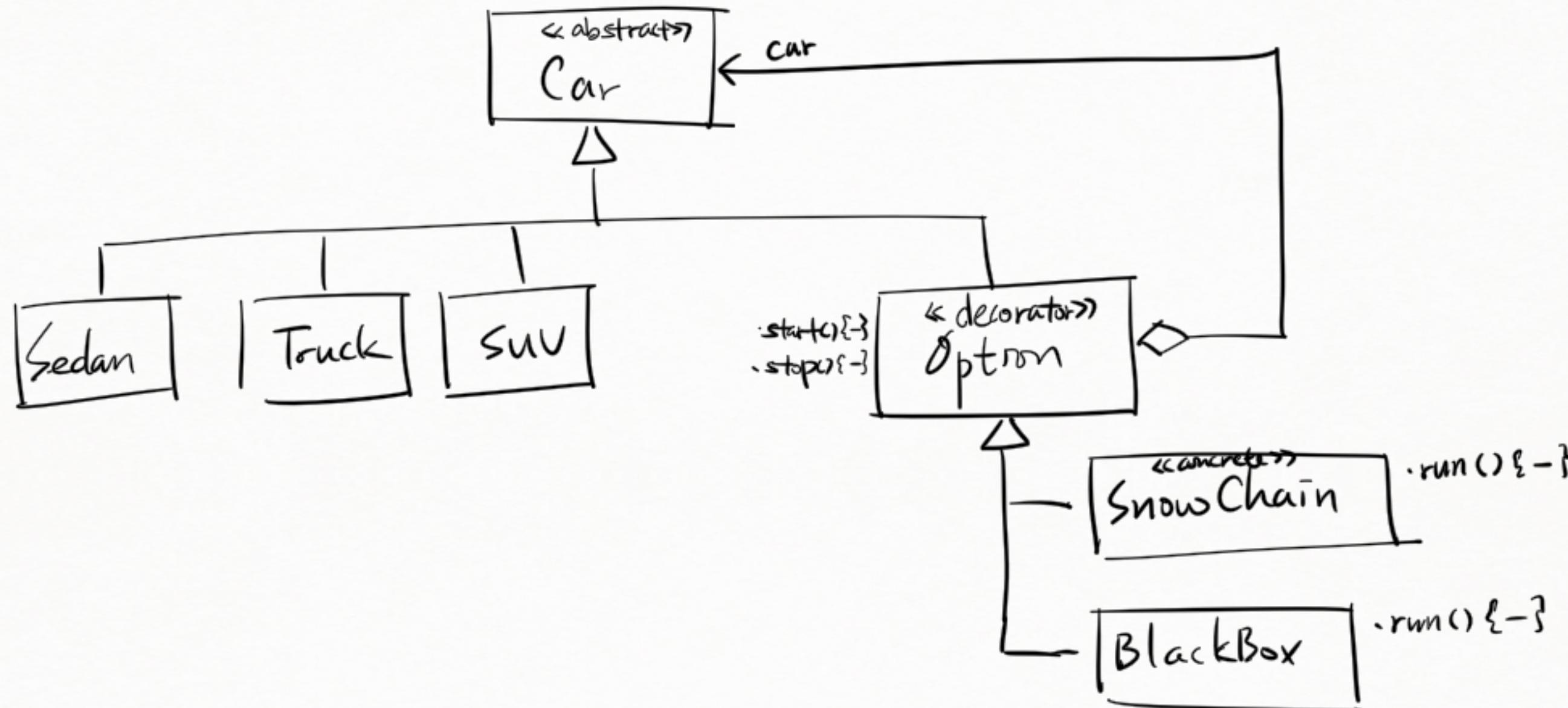
* 상 하 한 한 한 한 한 - 품종, 품질, 성능

\hookrightarrow : generalization

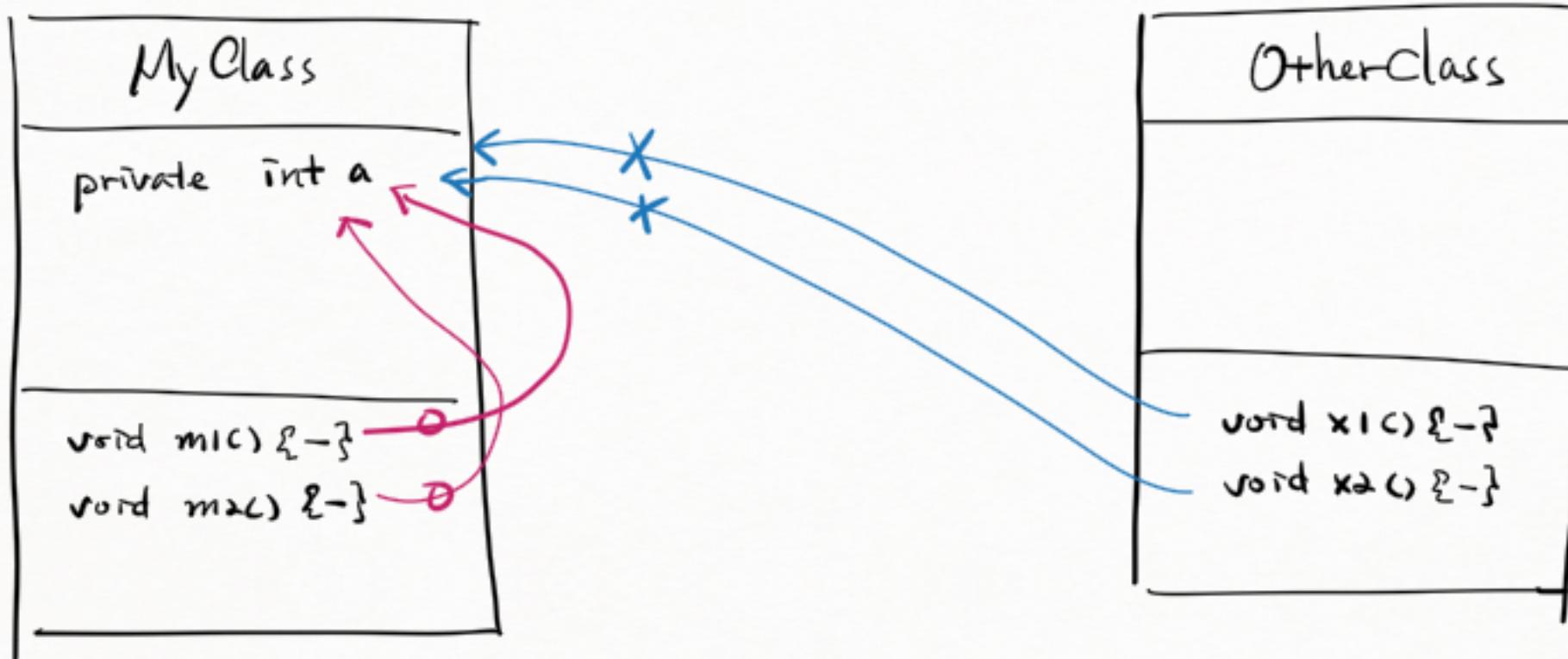


- enableTow: boolean
- activateTow(boolean)
- run() { }

* 차선을 3개로 나누면 차량이 1개로 통합

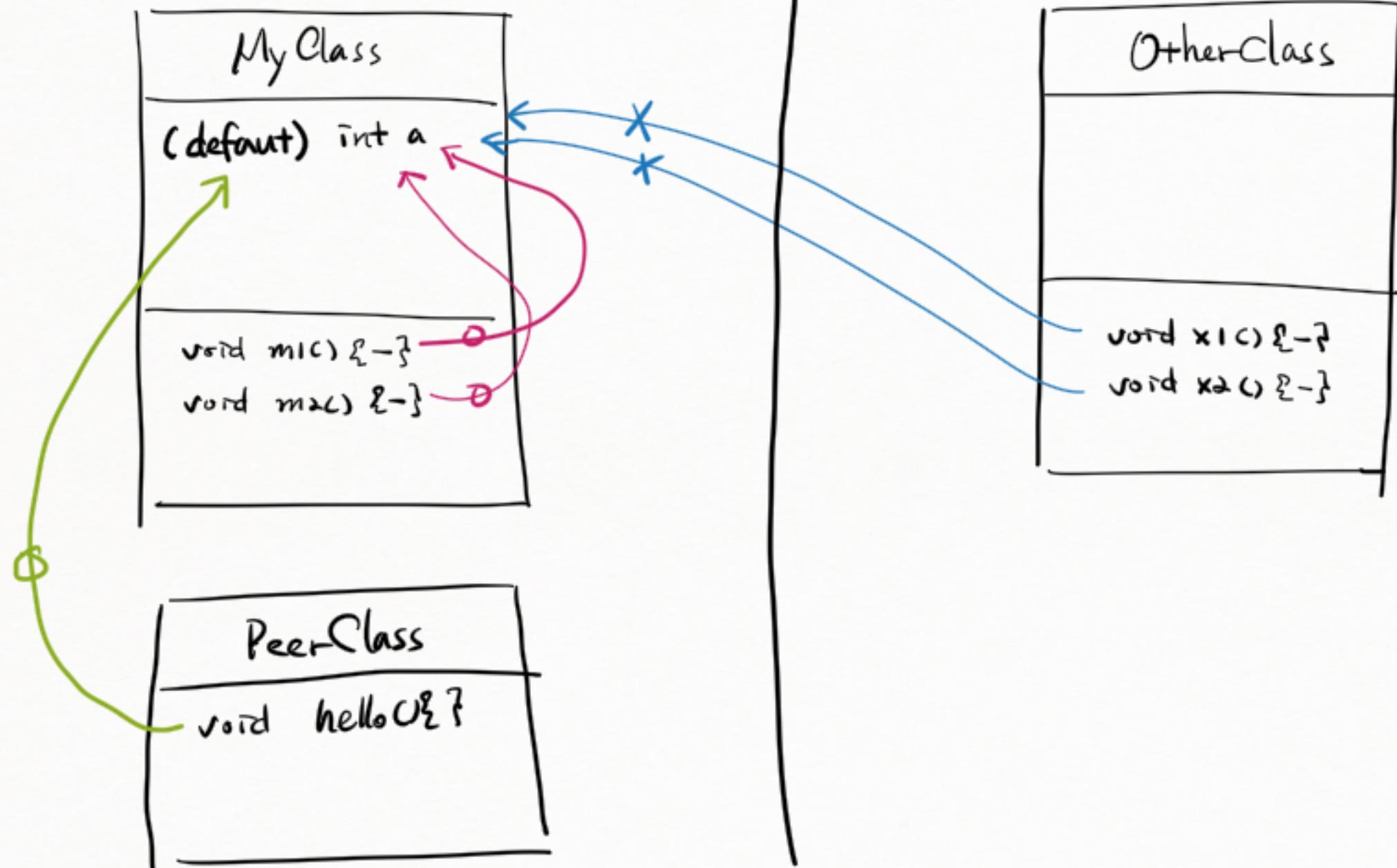


* private - 같은 클래스의 멤버만 접근 가능



* (default) - 같은 클래스의 멤버 접근 가능
+ 같은 패키지 속 클래스의 멤버 접근 가능

com.eomcs.test1



com.eomcs.test2

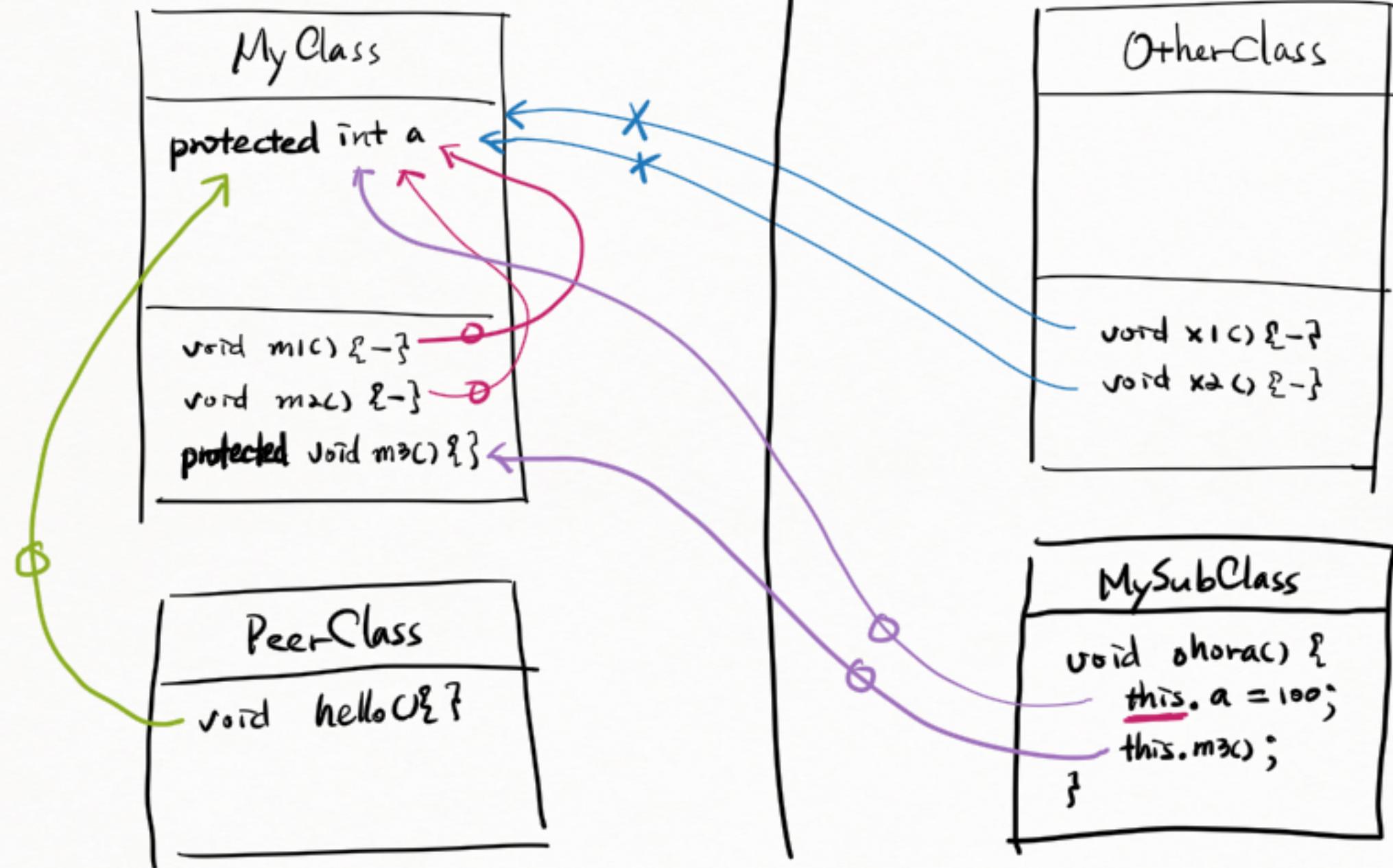
* protected - 같은 패키지 내의 멤버 접근 가능

+ 같은 패키지 속 다른 클래스의 멤버 접근 가능 + 다른 클래스 접근 가능

com.eomcs.test1

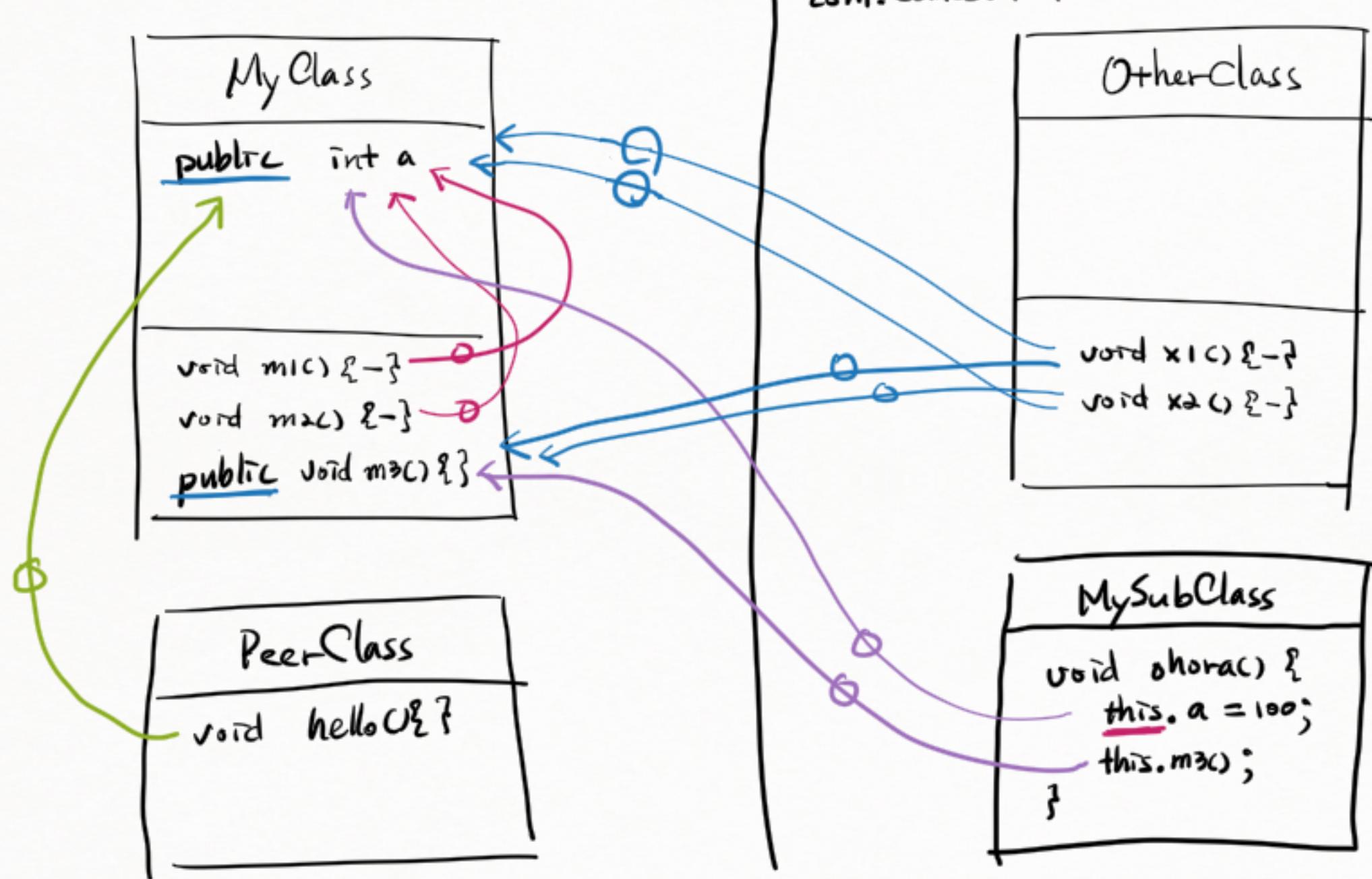
com.eomcs.test2

!!
자신이 상속 받은 멤버와 더불어
자신이 상속 받은 멤버와 더불어

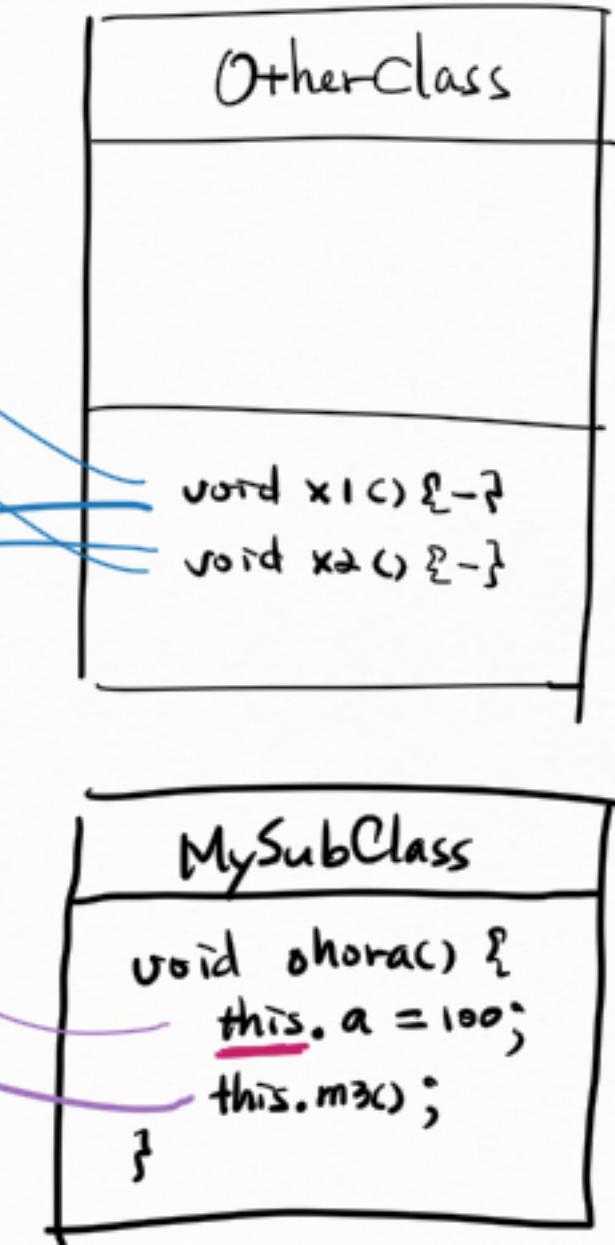


* public - 모든 멤버 접근 가능

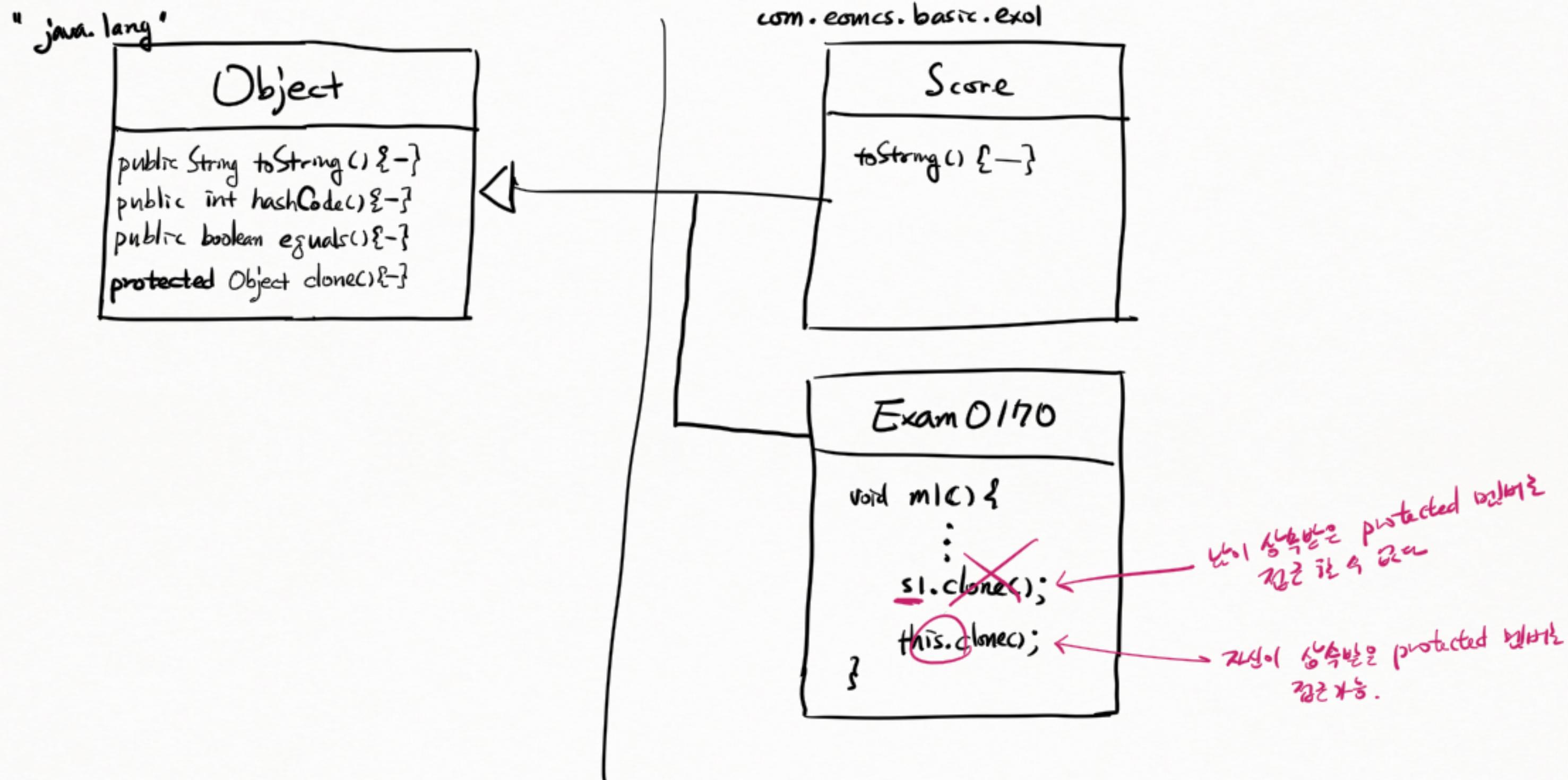
com.eomcs.test1



com.eomcs.test2



* clone 데일



* 추상클래스

the skeleton of an algorithm

