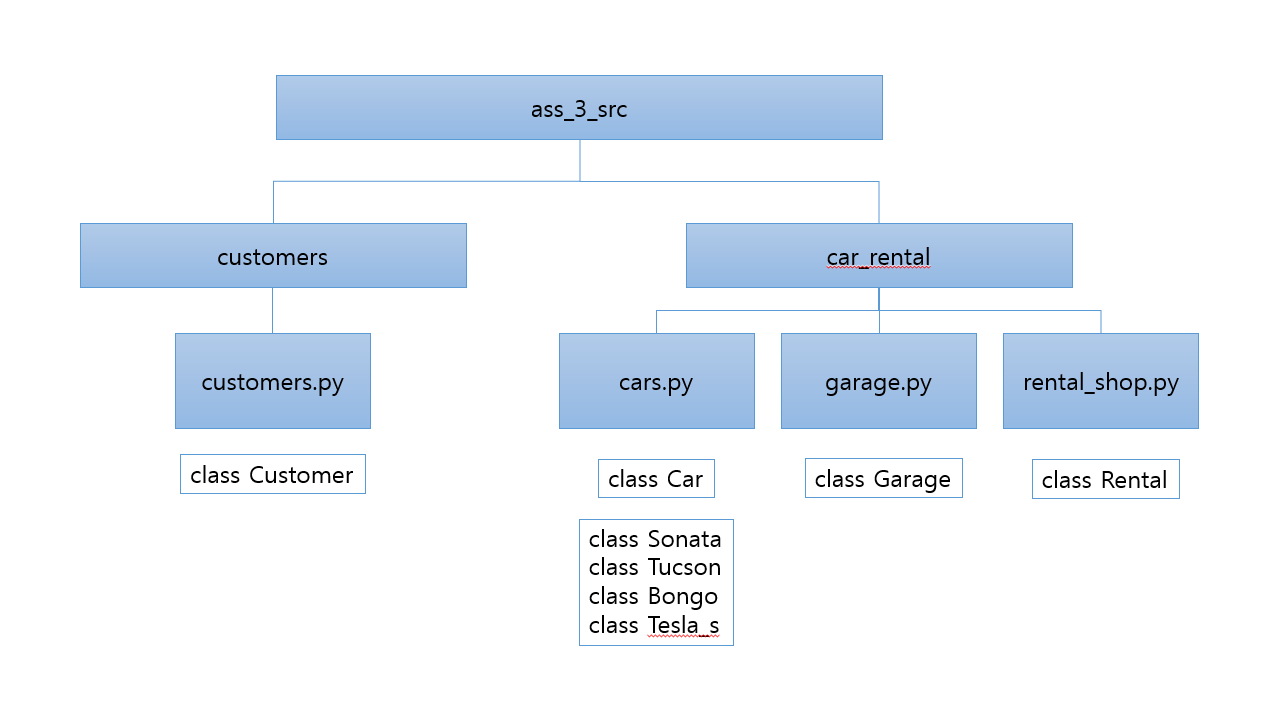
**<구성>**

2017111762 이희경



**<각 클래스별 메소드와 변수에 대한 간단한 설명>**

클래스 이름 옆에 세모를 누르면 목록이 펼쳐집니다!

**Car(name, True or False)**

**True : 신차, False : 중고차를 의미합니다.**

<변수>

max\_fuel : 최대 주유량

fuel\_economy : 최대 연비

max\_passenger : 최대 승객 수

max\_load : 최대 적재량

passenger : 현재 승객 수

load : 현재 적재량

economy: 현재 연비

m\_record : 주행량을 기록하기 위한 리스트

r\_index : 주행량을 기록하기 위한 리스트의 인덱스

name : 자동차의 이름

isnew : 자동차가 신차인지 아닌지를 결정하는 변수

recent\_quantity : 가장 최근의 주유량 또는 주행량을 임시 저장하는 변수

<메소드>

get\_name() : 자동차의 이름(별칭)을 리턴

get\_fuel() : 현재 자동차의 연료량을 리턴

get\_max\_fuel() : 이 자동차의 최대연료량을 리턴

fill\_up() : 자동차를 최대연료량까지 주유

fill(x) : x만큼 주유

set\_recent\_fill(quantity) : 주유 후 주유한량을 저장, 가장 최근 주유량만 저장

get\_recent\_fill() : 가장 최근 주유량을 리턴

ride(x) : 차에 승객 x명 승차

get\_off(x) : 승객 x명 하차

loading(x) : 짐 x kg 적재

unloading(x) : 짐 x kg 하차

get\_range(x) : 주행가능거리 리턴

update\_economy(passenger and/or load): 승객이나 짐이 변화할 때마다 현재 연비 업데이트

move(x) : x km 주행

set\_recent\_mileage(q) : 최근 주행량 저장

get\_recent\_mileage() : 최근 주행량 리턴

get\_mileage() : 현재까지 주행량 리턴

move\_record(x, economy) : 주행량, 연비 기록

fuel\_economy\_100() : 최근 100km 주행간 연비 리턴

show\_dash() : 대쉬보드 출력

get\_type() : 자동차 종류 리턴

**Garage()**

<변수>

garage\_list : 차고에 있는 차들의 목록

garage\_list={car.get\_name() :[car, 0 or 1]

0이면 현재 차고에 없고(대여 중이고), 1이면 현재 차고에 있다.

<메소드>

add\_car(car) : 차고에 자동차 추가

remove\_car(car) : 차고에 자동차 제거

renting(car) : 차 대여

get\_back(car) : 차 반납

available\_list() : 현재 대여가능한 차 목록 출력

retrieve() : 현재 자동차 현황 출력

search(ask) : 고객 요구사항(차 종류)에 맞는 대여가능한 차 목록 출력

**Rental(year, month, day)**

**오픈날짜 입력**

<변수>

opening: 렌탈샵을 오픈한 날짜

rent\_list: 현재 렌터카 렌트현황

garage : 렌탈샵의 차고

recent\_fill : 렌탈된 차들의 주유량을 저장

recent\_fill={날짜:주유량}

recent\_mileage : 렌탈된 차들의 주행량 저장

recent\_mileage={날짜:주행량}

<메소드>

buy\_car(car) : 렌탈샵이 차를 구매한 후 차고에 추가

remove\_car(car) : 폐차

renting(car, who, howlong) : 고객에게 차 대여, 대여,반납시 연료는 50%이상이어야함, 대여시 50%이하이면 연료를 충전하고 대여함.

get\_back(car) : 고객이 차 반납, 연료가 50%이하면 반납할 수 없음

monitor() : 현재 자동차 현황 출력

* 일주일(조회날로부터 1일전~7일전)

agg\_recent\_fill(x) : 일주일간 주유량 집계, 고객이 차를 주유할 때마다 업데이트됨

week\_fill() : 일주일간 주유량 조회

agg\_recent\_mileage(x) : 일주일간 운행량 집계, 고객이 운행할 때마다 업데이트 됨

week\_mileage(x) : 일주일간 운행량 조회

recommendation(ask) : 고객의 요구사항에 맞는 차 매칭

* 가능한 고객의 요구사항 종류 → 매칭되는 차

Passenger car → Sonata

SUV → Tucson

Truck → Bongo

Electric → Tesla\_s

**Customer(name)**

<변수>

name : 고객의 이름,

d : 빌리는 기간

(rent를 했을 때 생기는 변수)

car:빌린 차

where:차를 빌린 렌탈샵

<메소드>

get\_name() : 이름 리턴

rent(car, d, where) : 어떤 렌탈샵에서 어떤차를 몇일동안 빌림

give\_back() : 빌린차를 반납. 고객은 한번에 차를 한번만 빌릴 수 있음

fill\_up() : 자동차 가득 주유 후 렌탈샵에 정보 업데이트

fill(x) : x 만큼 주유 후 렌탈샵에 정보 업데이트

ride\_passenger(x) : 운전자 말고 다른 탑승객 x명 탑승

get\_off\_passenger(x) : 탑승객 x명 하차

loading(x) : 짐 x kg 적재

lunloading(x) : 짐 x kg 하차

driving(x) : x km 주행하고 주행 후 렌탈샵에 정보 업데이트

retrieve\_dash() : 자동차 대쉬보드 조회

ask(ask, where) : 어떤 렌탈샵에 요구사항 질문

* 요구사항 = ‘Passenger car’,’SUV’,’Truck’,’Electric’