

week2

🚩Q1. 중간고사 기말고사 점수를 따로 받아 저장하는 클래스를 구현해보세요. 단, 생성자의 인스턴스는 private으로 선언되어야하며, 데코레이터를 이용해 데이터를 저장하고, 함수를 이용해 평균값을 출력해보세요.

- 자료형의 선언과 데코레이터를 이용해보세요.

```
# test score, mid : 50, final : 75

class Score():
    # TODO

#출력함수
score = Score(50, 75)
print((score.mid + score.final) / 2)
```

```
# 출력 예시
62.5
```

```
# test score, mid : 50, final :75

class Score():

def __init__ (self, mid , final ):

self.mid = mid

self.final = final

#출력함수

score = Score(50,75)

print((score.mid + score.final) / 2)
```

🚩Q2. 다양한 탈 것을 사용하는 게임을 만드는 중입니다. 빠른 구현을 위해서 이미 구현한 Car 클래스를 이용해서 Bike라는 클래스를 새로 제작하려고 합니다. Car 클래스를 상속받아서 새로운 Bike 클래스를 아래와 같이 출력되도록 구성해보세요.


- Bike class는 size 인자를 추가로 가집니다.

```
class Car():
    def __init__(self, fuel, wheels):
        self.fuel = fuel
        self.wheels = wheels
```

```
class Bike():
    # TODO
```

```
# 출력 예시
>>> bike = Bike("gas", 2, "small")
>>> print(bike.fuel, bike.wheels, bike.size)
gas 2 small
```

```
class Car():
    def __init__(self, fuel, wheels):
        self.fuel = fuel
        self.wheels = wheels
class Bike(Car):
    def __init__(self, fuel = str, wheels = int, size = str):
        super().__init__(fuel, wheels)
        self.size = size
bike = Bike("gas", 2, "small")
print(bike.fuel, bike.wheels, bike.size)
```

 **Q3. 이번 시험 결과에 대한 데이터를 학과 사무실에서 CSV파일로 전달해줬습니다. 우리는 이 파일을 이용해서 데이터 처리를 진행해야 합니다. 파일 입출력을 이용해 파일 데이터를 리스트로 만들어보세요.**

- 파일 입출력에 사용하는 open 함수를 이용해 CSV 파일 내부의 데이터를 읽어보세요
- *CSV파일은 아래 첨부되어있습니다.

```
# 파일의 경로를 file_path로 설정
# ex) file_path = "./data-01-test-score.csv"

def read_file(file_path):
    # TODO
```

```
# 출력 예시
>>> read_csv = ReadCSV(filepath)
>>> print(read_csv.read_file())
```

```
[['73', '80', '75', '152'],
 ['93', '88', '93', '185'],
 ['89', '91', '90', '180'],
 ['96', '98', '100', '196'],
 ['73', '66', '70', '142'],
 ['53', '46', '55', '101'],
 ['69', '74', '77', '149'],
 ['47', '56', '60', '115'],
 ['87', '79', '90', '175'],
 ['79', '70', '88', '164'],
 ['69', '70', '73', '141'],
 ['70', '65', '74', '141'],
 ['93', '95', '91', '184'],
 ['79', '80', '73', '152'],
 ['70', '73', '78', '148'],
 ['93', '89', '96', '192'],
 ['78', '75', '68', '147'],
 ['81', '90', '93', '183'],
 ['88', '92', '86', '177'],
 ['78', '83', '77', '159'],
 ['82', '86', '90', '177'],
 ['86', '82', '89', '175'],
 ['78', '83', '85', '175'],
 ['76', '83', '71', '149'],
 ['96', '93', '95', '192']]
```

```
f = open("data-01-test-score.csv", "r")
contents = f.readlines() #리스트로 만듦
print(contents)
```

data-01-test-score CSVQ 3,4,5에 사용되는 CSV파일

🔴Q4. 우리는 방금 CSV 파일을 읽는 함수를 구현해보았습니다. 하지만 이를 조금 더 효율적으로 사용하기 위해서 클래스로 구성을 진행하려고 합니다. 방금 구현한 함수를 포함한 클래스를 구현해보세요.

- merge list를 이용해 리스트 내 데이터의 합을 출력해보세요.

- 데이터를 합치기 전 데이터의 자료형을 변경해보세요.

```

# 파일의 경로를 file_path로 설정
# ex) file_path = "./data-01-test-score.csv"

class ReadCSV():
    def __init__(self, file_path):
        # TODO

    def read_file(self, #):
        # TODO

    def merge_list(self, #):
        # TODO

```

```

# 출력 예시
>>> read_csv = ReadCSV(filepath)
>>> print(read_csv.read_file())
>>> print(read_csv.merge_list())

[[73, 80, 75, 152],
 [93, 88, 93, 185],
 [89, 91, 90, 180],
 [96, 98, 100, 196],
 [73, 66, 70, 142],
 [53, 46, 55, 101],
 [69, 74, 77, 149],
 [47, 56, 60, 115],
 [87, 79, 90, 175],
 [79, 70, 88, 164],
 [69, 70, 73, 141],
 [70, 65, 74, 141],
 [93, 95, 91, 184],
 [79, 80, 73, 152],
 [70, 73, 78, 148],
 [93, 89, 96, 192],
 [78, 75, 68, 147],
 [81, 90, 93, 183],
 [88, 92, 86, 177],
 [78, 83, 77, 159],
 [82, 86, 90, 177],
 [86, 82, 89, 175],
 [78, 83, 85, 175],
 [76, 83, 71, 149],
 [96, 93, 95, 192]]

[380, 459, 450, 490, 351, 255, 369, 278,
 431, 401, 353, 350, 463, 384, 369, 470,
 368, 447, 443, 397, 435, 432, 421, 379, 476]

```

📌 Q5. 이전에 구현한 클래스에서 `merge_list`의 함수 동작을 변경해야 합니다. 단순합계가 아닌 평균을 구하는 함수로 변경해보세요.

- 리스트의 데이터를 다루는 함수를 이용해서 구현해보세요.

- 최종 평균을 구한 후 오름차순으로 정렬해주세요.

```
# 파일의 경로를 file_path로 설정
# ex) file_path = "./data-01-test-score.csv"

class ReadCSV():
    def __init__(self, file_path):
        # TODO

    def read_file(self, #):
        # TODO

    def merge_list(self, #):
        # TODO
```

```
# 출력 예시
>>> read_csv = ReadCSV(filepath)
>>> print(read_csv.merge_list())

[63.75, 69.5, 87.5, 87.75, 88.25, 92.0, 92.25,
92.25, 94.75, 95.0, 96.0, 99.25, 100.25, 105.25,
107.75, 108.0, 108.75, 110.75, 111.75, 112.5,
114.75, 115.75, 117.5, 119.0, 122.5]
```