

## Код

```
from operator import itemgetter

class Driver:
    """Водитель"""

    def __init__(self, id, name, sal, CarPark_id):
        self.id = id
        self.name = name
        self.sal = sal
        self.CarPark_id = CarPark_id

class CarPark:
    """Автопарк"""

    def __init__(self, id, name):
        self.id = id
        self.name = name

class DriverCarPark:

    def __init__(self, park_id, driver_id):
        self.park_id = park_id
        self.driver_id = driver_id

car_parks = [
    CarPark(1, 'Mercedes'),
    CarPark(2, 'BMW'),
    CarPark(3, 'Audi'),
    CarPark(4, 'Porsche'),
    CarPark(5, 'Hyundai'),
    CarPark(6, 'KIA'),
]

drivers = [
    Driver(1, 'Саргсян', 100000, 1),
    Driver(2, 'Иванов', 130000, 6),
    Driver(3, 'Сомов', 100000, 5),
    Driver(4, 'Петренко', 75000, 4),
    Driver(5, 'Зеленский', 60000, 3),
    Driver(6, 'Пивоваров', 150000, 2),
    Driver(7, 'Леонов', 95000, 5),
    Driver(8, 'Матвеев', 102000, 1),
    Driver(9, 'Гутова', 120000, 2),
]

drivers_cars = [
    DriverCarPark(1, 1),
    DriverCarPark(6, 2),
```

```

        DriverCarPark(3, 3),
        DriverCarPark(5, 4),
        DriverCarPark(2, 5),
        DriverCarPark(6, 6),
        DriverCarPark(7, 9),
        DriverCarPark(8, 7),
        DriverCarPark(9, 9)
    ]

def main():
    one_to_many = [
        (c.name, c.sal, d.name)
        for d in car_parks
        for c in drivers
        if c.CarPark_id == d.id
    ]

    many_to_many_temp = [
        (d.name, cl.CarPark_id, cl.id)
        for d in car_parks
        for cl in drivers
        if d.id == cl.CarPark_id
    ]

    many_to_many = [
        (c.name, car_parks_name)
        for car_parks_name, car_parks_id, drivers_id in many_to_many_temp
        for c in drivers
        if c.id == drivers_id
    ]

    print('Задание Б1')
    res_1 = sorted(one_to_many, key=itemgetter(0))
    for i in res_1:
        print(i, end="\n")

    print('\nЗадание Б2')
    res_2_unsorted = []
    for b in car_parks:
        d_driver = list(filter(lambda i: i[2] == b.name, one_to_many))
        res_2_unsorted.append((b.name, len(d_driver)))

    res_2 = sorted(res_2_unsorted, key=itemgetter(1), reverse=True)
    print(res_2)

    print('\nЗадание Б3')
    res_3 = {}
    for d in drivers:
        if str(d.name).endswith('ов'):
            d_Drivers = list(filter(lambda i: i[0] == d.name, many_to_many))
            d_Drivers_names = [x for _, x in d_Drivers]
            res_3[d.name] = d_Drivers_names

```

```
print(res_3)

if __name__ == '__main__':
    main()
```

C:\Lab\_Python\Lab\_Python\RK1\venv\Scripts\python.exe

C:/Lab\_Python/Lab\_Python/RK1/RK1.py

Задание Д1

(‘Гутова’, 120000, ‘BMW’)

(‘Зеленский’, 60000, ‘Audi’)

(‘Иванов’, 130000, ‘Kia’)

(‘Леонов’, 95000, ‘Hyundai’)

(‘Матвеев’, 102000, ‘Mercedes’)

(‘Петренко’, 75000, ‘Porsche’)

(‘Пивоваров’, 150000, ‘BMW’)

(‘Саргсян’, 100000, ‘Mercedes’)

(‘Сомов’, 100000, ‘Hyundai’)

Задание Д2

[('Mercedes', 2), ('BMW', 2), ('Hyundai', 2), ('Audi', 1), ('Porsche', 1), ('Kia', 1)]

Задание Д3

{‘Иванов’ : [‘Kia’], ‘Сомов’ : [‘Hyundai’], ‘Пивоваров’ : [‘BMW’] }

Process finished with exit code 0

Задание Б1

```
('Гутова', 120000, 'BMW')  
('Зеленский', 60000, 'Audi')  
('Иванов', 130000, 'KIA')  
('Леонов', 95000, 'Hyundai')  
('Матвеев', 102000, 'Mercedes')  
('Петренко', 75000, 'Porsche')  
('Пивоваров', 150000, 'BMW')  
('Саргсян', 100000, 'Mercedes')  
('Сомов', 100000, 'Hyundai')
```

Задание Б2

```
[('Mercedes', 2), ('BMW', 2), ('Hyundai', 2), ('Audi', 1), ('Porsche', 1), ('KIA', 1)]
```

Задание Б3

```
{'Иванов': ['KIA'], 'Сомов': ['Hyundai'], 'Пивоваров': ['BMW'], 'Леонов': ['Hyundai'], 'Гутова': ['BMW']}
```