

```

from operator import itemgetter

class driver:
    """Водитель"""

    def __init__(self, id, fio, sal, autopark_id):
        self.id = id
        self.fio = fio
        self.sal = sal
        self.autopark_id = autopark_id

class autopark:
    """Автопарк"""

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

# Автопарки
autoparks = [
    autopark(1, 'Yandex'),
    autopark(2, 'Uber'),
    autopark(3, 'Samokat'),
]

# Водители
drivers = [
    driver(1, 'Петров', 20000, 3),
    driver(2, 'Иванов', 35000, 3),
    driver(3, 'Сидоров', 45000, 2),
    driver(4, 'Голубчик', 35000, 1),
    driver(5, 'Смирнов', 25000, 1),
]

def main():
    """Основная функция"""

    # Соединение данных один-ко-многим
    one_to_many = [(d.fio, d.sal, a.name)
                    for a in autoparks
                    for d in drivers
                    if d.autopark_id == a.id]

    print('Задание 1')
    res_11 = sorted(one_to_many, key=itemgetter(2))
    print(res_11)

    print('\nЗадание 2')
    res_12_unsorted = []

```

```

# Перебираем все автопарки
for a in autoparks:
    # Список водителей автопарков
    d_drivers = list(filter(lambda i: i[2] == a.name,
one_to_many))
    # Если автопарк не пустой
    if len(d_drivers) > 0:
        # Зарплаты водителей автопарка
        a_sals = [sal for _, sal, _ in d_drivers]
        # Суммарная зарплата водителей автопарка
        a_sals_sum = sum(a_sals)
        res_12_unsorted.append((a.name, a_sals_sum))

# Сортировка по суммарной зарплате
res_12 = sorted(res_12_unsorted, key=itemgetter(1),
reverse=True)
print(res_12)

if __name__ == '__main__':
    main()

```

D:\WEB\venv\Scripts\python.exe D:\WEB\RK.py

Задание 1

[('Петров', 20000, 'Samokat'), ('Иванов', 35000, 'Samokat'), ('Сидоров', 45000, 'Uber'), ('Голубчик', 35000, 'Yandex'), ('Смирнов', 25000, 'Yandex

Задание 2

[('Yandex', 60000), ('Samokat', 55000), ('Uber', 45000)]

Process finished with exit code 0