

Отчет по РК № 1 по курсу
"Разработка Интернет-Приложений"

Выполнила:

Студентка группы

ИУ5-55Б

Богданова В.В.

Задание:

1. «Водитель» и «Автопарк» связаны соотношением один-ко-многим. Выведите список всех Водителей, у которых фамилия заканчивается на «ов», и названия их Автопарков.
2. «Водитель» и «Автопарк» связаны соотношением один-ко-многим. Выведите список Автопарков со средней зарплатой Водителей в каждом Автопарке, отсортированный по средней зарплате.
3. «Водитель» и «Автопарк» связаны соотношением многие-ко-многим. Выведите список всех Автопарков, у которых название начинается с буквы «А», и список работающих в них Водителей.

№ варианта	Класс 1	Класс 2
3	Водитель	Автопарк

Текст программы:

```
# используется для сортировки
from operator import itemgetter

class Driver:
    """Водитель"""
    def __init__(self, id, fio, sal, park_id):
        self.id = id
        self.fio = fio
        self.sal = sal
        self.park_id = park_id

class CarPark:
    """Автопарк"""
    def __init__(self, id, name):
        self.id = id
        self.name = name

class DrivCarPark:
    """
    'Водители автопарка' для реализации
    связи многие-ко-многим
    """
    def __init__(self, park_id, driv_id):
        self.park_id = park_id
        self.driv_id = driv_id

# Автопарки
parks = [
    CarPark(1, 'Яндекс такси'),
    CarPark(2, 'Ситимобил'),
    CarPark(3, 'Автогортранс'),
```

```

    CarPark(11, 'Делимобиль'),
    CarPark(22, 'Максим'),
    CarPark(33, 'Абер'),
]

# Водители
drivs = [
    Driver(1, 'Артамонов', 25000, 1),
    Driver(2, 'Петров', 35000, 2),
    Driver(3, 'Иваненко', 45000, 3),
    Driver(4, 'Иванов', 35000, 3),
    Driver(5, 'Иванин', 25000, 3),
]

parks_drivs = [
    DrivCarPark(1,1),
    DrivCarPark(2,2),
    DrivCarPark(3,3),
    DrivCarPark(3,4),
    DrivCarPark(3,5),

    DrivCarPark(11,1),
    DrivCarPark(22,2),
    DrivCarPark(33,1),
    DrivCarPark(33,2),
    DrivCarPark(33,5),
]

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

    # Соединение данных один-ко-многим
    one_to_many = [(d.fio, d.sal, p.name)
                   for p in parks
                   for d in drivs
                   if d.park_id==p.id]

    # Соединение данных многие-ко-многим
    many_to_many_temp = [(p.name, pd.park_id, pd.driv_id)
                          for p in parks
                          for pd in parks_drivs
                          if p.id==pd.park_id]

    many_to_many = [(d.fio, d.sal, park_name)
                    for park_name, park_id, driv_id in many_to_many_temp
                    for d in drivs if d.id==driv_id]

    print('Задание Д1')
    res_11 = [(d.fio, d.sal, p.name)
              for p in parks

```

```

        for d in drivs
            if d.park_id==p.id and d.fio.endswith('ов')]
print(res_11)

print('\nЗадание Д2')
res_12_unsorted = []
# Перебираем все автопарки
for p in parks:
    # Список водителей автопарка
    p_drivs = list(filter(lambda i: i[2]==p.name, one_to_many))
    # Если автопарк не пустой
    if len(p_drivs) > 0:
        # Зарплаты водителей автопарка
        d_sals = [sal for _,sal,_ in p_drivs]
        # Средняя зарплата водителей автопарка
        d_sals_mean = sum(d_sals) / len(d_sals)
        res_12_unsorted.append((p.name, d_sals_mean))

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

print('\nЗадание Д3')
res_13 = {}
# Перебираем все автопарки
for p in parks:
    if p.name.lower().startswith('а'):
        # Список водителей автопарка
        drivs_of_p = list(filter(lambda i: i[2]==p.name, many_to_many))
        # Только ФИО сотрудников
        drivs_names = [x for x,_,_ in drivs_of_p]
        # Добавляем результат в словарь
        # ключ - автопарк, значение - список фамилий
        res_13[p.name] = drivs_names

print(res_13)

if __name__ == '__main__':
    main()

```

Результат работы:

```

Задание Д1
[('Артамонов', 25000, 'Яндекс такси'), ('Петров', 35000, 'Ситимобил'), ('Иванов', 35000, 'Автогортранс')]

Задание Д2
[('Ситимобил', 35000.0), ('Автогортранс', 35000.0), ('Яндекс такси', 25000.0)]

Задание Д3
{'Автогортранс': ['Иваненко', 'Иванов', 'Иванин'], 'Абер': ['Артамонов', 'Петров', 'Иванин']}

```