

## Бирюкова Екатерина 3А.

### Вариант 3.

Класс 1 – водитель, класс 2 – автопарк.

### Вариант запроса А.

1. «Автопарк» и «Водитель» связаны соотношением один-ко-многим. Выведите список всех связанных водителей и автопарков, отсортированный по автопаркам, сортировка по водителям произвольная.
2. «Автопарк» и «Водитель» связаны соотношением один-ко-многим. Выведите список автопарков с суммарной зарплатой водителей в каждом автопарке, отсортированный по суммарной зарплате.
3. «Автопарк» и «Водитель» связаны соотношением многие-ко-многим. Выведите список всех автопарков, у которых в названии присутствует слово «Московский», и список работающих в них водителей.

### Текст программы.

```
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 Park:
```

```
    """Автопарк"""
```

```
    def __init__(self, id, name):
```

```
        self.id = id
```

```
        self.name = name
```

```
class Driver_Park:
```

```
    """Водители автопарка"""
```

```
    def __init__(self, driver_id, park_id):
```

```
        self.driver_id = driver_id
```

```
        self.park_id = park_id
```

```
# Автопарки
```

```
parks = [
```

```
    Park(1, 'Нижегородский автопарк'),
```

```
    Park(2, 'Московский автопарк'),
```

```
    Park(3, 'Костромской автопарк'),
```

```
    Park(11, 'Нижегородский (другой) автопарк'),
```

```
    Park(22, 'Московский (другой) автопарк'),
```

```
    Park(33, 'Костромской (другой) автопарк'),
```

```
]
```

```

# Водители
drivers = [
    Driver(1, 'Петров', 10000, 1),
    Driver(2, 'Сидоров', 15000, 2),
    Driver(3, 'Иванов', 20000, 3),
    Driver(4, 'Абакумов', 25000, 1),
    Driver(5, 'Большов', 30000, 2),
]

drivers_parks = [
    Driver_Park(1,1),
    Driver_Park(2,2),
    Driver_Park(3,3),
    Driver_Park(4,1),
    Driver_Park(5,2),

    Driver_Park(1,22),
    Driver_Park(2,33),
    Driver_Park(3,11),
    Driver_Park(4,22),
    Driver_Park(5,33),
]

def main():
    one_to_many = [(d.fio, d.sal, p.name)
                   for p in parks
                   for d in drivers
                   if d.park_id==p.id]

    many_to_many_temp = [(p.name, dp.park_id, dp.driver_id)
                          for p in parks
                          for dp in drivers_parks
                          if p.id==dp.park_id]

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

    #Задание 1
    print("Задание A1")
    res_11 = sorted(one_to_many, key=itemgetter(2))
    print(res_11)

    #Задание 2
    print("\nЗадание A2")
    res_12_unsorted = []
    for p in parks:
        drivers_of_park = list(filter(lambda i: i[2]==p.name, one_to_many))
        if len(drivers_of_park) > 0:
            p_sals = [sal for _,sal,_ in drivers_of_park]

```

```

        p_sals_sum = sum(p_sals)
        res_12_unsorted.append((p.name, p_sals_sum))
    res_12 = sorted(res_12_unsorted, key=itemgetter(1), reverse=True)
    print(res_12)

#Задание 3
print("\nЗадание А3")
res_13 = { }
for p in parks:
    if 'Московский' in p.name:
        drivers_of_park = list(filter(lambda i: i[2]==p.name, many_to_many))
        drivers_of_park_names = [x for x,_,_ in drivers_of_park]
        res_13[p.name] = drivers_of_park_names
print(res_13)

if __name__ == '__main__':
    main()

```

## Результаты выполнения.

### Задание А1

```

[('Иванов', 20000, 'Костромской автопарк'), ('Сидоров', 15000, 'Московский автопарк'),
('Большов', 30000, 'Московский автопарк'), ('Петров', 10000, 'Нижегородский автопарк'),
('Абакумов', 25000, 'Нижегородский автопарк')]

```

### Задание А2

```

[('Московский автопарк', 45000), ('Нижегородский автопарк', 35000), ('Костромской автопарк', 20000)]

```

### Задание А3

```

{'Московский автопарк': ['Сидоров', 'Большов'], 'Московский (другой) автопарк': ['Петров', 'Абакумов']}

```