



## РУБЕЖНЫЙ КОНТРОЛЬ №1

2020 г.

```
# используется для сортировки
```

```
from operator import itemgetter
```

```
class Musician:
```

```
    def __init__(self, id, fio, instrument, orch_id):
```

```
        self.id = id
```

```
        self.fio = fio
```

```
        self.instrument = instrument
```

```
        self.orch_id = orch_id
```

```
class Orchestra:
```

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

```
        self.id = id
```

```
        self.name = name
```

```
class MusOrch:
```

```
    def __init__(self, mus_id, orch_id):
```

```
        self.mus_id = mus_id
```

```
        self.orch_id = orch_id
```

```
orchs = [
```

```
    Orchestra(1, 'Royal'),
```

```
    Orchestra(2, 'Berlin'),
```

```
    Orchestra(3, 'Vienna'),
```

```
]
```

```
musicians = [
```

```
    Musician(1, 'fio1 Иванов', 'guitar', 1),
```

```
    Musician(2, 'fio2 Петров', 'violin', 1),
```

```
    Musician(3, 'fio3 Сидоров', 'piano', 2),
```

```
    Musician(4, 'fio4 Кравченко', 'vocal', 2),
```

```
Musician(5, 'fio5 Стацюк', 'viola', 2),  
Musician(6, 'fio6 Стриженко', 'violin', 3),  
Musician(7, 'fio7 Мельничевец', 'violin', 3),  
Musician(8, 'fio8 Варламов', 'piano', 3),  
]
```

```
mus_orchs = [  
    MusOrch(1, 1),  
    MusOrch(1, 2),  
    MusOrch(1, 3),  
    MusOrch(2, 1),  
    MusOrch(2, 2),  
    MusOrch(2, 3),  
    MusOrch(3, 1),  
    MusOrch(3, 2),  
    MusOrch(4, 2),  
    MusOrch(4, 3),  
    MusOrch(5, 1),  
    MusOrch(6, 1),  
    MusOrch(7, 1),  
    MusOrch(8, 1),  
    MusOrch(8, 3),  
]
```

```
def main():  
    """Основная функция"""  
  
    # Соединение данных один-ко-многим  
    one_to_many = [(m.fio, m.instrument, o.name)  
        for m in musicians  
        for o in orchs  
        if m.orch_id==o.id]
```

```

# Соединение данных многие-ко-многим
many_to_many_temp = [(o.name, mo.orch_id, mo.mus_id)

    for o in orchs

    for mo in mus_orchs

    if o.id==mo.orch_id]

many_to_many = [(m.fio, m.instrument, orch_name)

    for orch_name, orch_id, mus_id in many_to_many_temp

    for m in musicians if m.id==mus_id]

# Список всех связанных музыкантов и оркестров, отсортированный по ФИО музыканта
print('Задание A1')
res_11 = sorted(one_to_many, key=itemgetter(0))
print(res_11)

# Вывод списка оркестров с количеством музыкантов, отсортированный по количеству музыкантов
print('\nЗадание A2')
res_12_unsorted = []
for o in orchs:
    o_mus = list(filter(lambda i: i[2]==o.name, one_to_many))
    if len(o_mus) > 0:
        o_cnt = len(o_mus)
        res_12_unsorted.append((o.name, o_cnt))

res_12 = sorted(res_12_unsorted, key=itemgetter(1), reverse=True)
print(res_12)

# Вывод списка музыкантов, чья фамилия оканчивается на *ов*, для каждого - список оркестров, в которых он играет
print('\nЗадание A3')
res_13 = {}

```

for m in musicians:

if m.fio.endswith('ов'):

m\_orchs = list(filter(lambda i: i[0]==m.fio, many\_to\_many))

orch\_names = [x for \_,x in m\_orchs]

# Добавляем результат в словарь

# ключ - отдел, значение - список фамилий

res\_13[m.fio] = orch\_names

print(res\_13)

if \_\_name\_\_ == '\_\_main\_\_':

main()

Задание A1  
[('fio1 Иванов', 'guitar', 'Royal'), ('fio2 Петров', 'violin', 'Royal'), ('fio3 Сидоров', 'piano', 'Berlin'), ('fio4 Кравченко', 'vocal', 'Berlin'), ('fio5 Стацук', 'viola', 'Berlin'), ('fio6 Стриженко', 'violin', 'Vienna'), ('fio7 Мельниченко', 'violin', 'Vienna'), ('fio8 Варламов', 'piano', 'Vienna')]

Задание A2  
[('Berlin', 3), ('Vienna', 3), ('Royal', 2)]

Задание A3  
{'fio1 Иванов': ['Royal', 'Berlin', 'Vienna'], 'fio2 Петров': ['Royal', 'Berlin', 'Vienna'], 'fio3 Сидоров': ['Royal', 'Berlin'], 'fio8 Варламов': ['Royal', 'Vienna']}

---