

Рубежный контроль 1
Вариант 5

| | | |
|---|----------|---------|
| 5 | Музыкант | Оркестр |
|---|----------|---------|

Вариант Б:

1. «Музыкант» и «Оркестр» связаны соотношением один-ко-многим. Выведите список всех связанных сотрудников и отделов, отсортированный по оркестрам, сортировка по отделам произвольная.
2. «Музыкант» и «Оркестр» связаны соотношением один-ко-многим. Выведите список музыкантов с количеством оркестров в каждом отделе, отсортированный по количеству оркестров.
3. «Музыкант» и «Оркестр» связаны соотношением многие-ко-многим. Выведите список всех оркестров, у которых название заканчивается на «а», и имена их музыкантов.

Текст программы

```
from operator import itemgetter

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

class Orchestra:
    def __init__(self, id, name, orchestra_type, mus_id):
        self.id = id
        self.name = name
        self.orchestra_type = orchestra_type
        self.mus_id = mus_id

class MusOrc:
    def __init__(self, mus_id, orc_id):
        self.mus_id = mus_id
        self.orc_id = orc_id

musicians = [
    Musician(1, "Бетховен"),
    Musician(2, "Чайковский"),
    Musician(3, "Прокофьев"),
```

```

]

orcs = [
    Orchestra(1, "Симфонический оркестр города Гусь Хрустальный",
    "Симфонический оркестр", 1),
    Orchestra(2, "Военный оркестр Александровского полка", "Военный
оркестр", 2),
    Orchestra(3, "Bad Apple", "Джазовый оркестр", 3),
    Orchestra(4, "Масленичное чучело", "Оркестр народных инструментов", 3),
    Orchestra(5, "Дудочка", "Духовой оркестр", 1),
    Orchestra(5, "Оркестр имени Петра Ильича Чайковского", "Симфонический
оркестр", 2)
]

mus_orc = [
    MusOrc(1, 1),
    MusOrc(2, 2),
    MusOrc(3, 3),
    MusOrc(1, 4),
    MusOrc(2, 5),
]

def first_task(musicians):
    res_1 = sorted(musicians, key=itemgetter(0))
    return res_1

def second_task(orcs):
    res_2 = []
    temp_dict = dict()
    for i in orcs:
        if i[2] in temp_dict:
            temp_dict[i[2]] += 1
        else:
            temp_dict[i[2]] = 1
    for i in temp_dict.keys():
        res_2.append((i, temp_dict[i]))

    res_2.sort(key=itemgetter(1), reverse=True)
    return res_2

def third_task(op_list, end_ch):
    res_3 = [(i[0], i[2]) for i in op_list if i[0].endswith(end_ch)]
    return res_3

def main():
    one_to_many = [(orc.name, orc.orchestra_type, m.name)
                    for m in musicians
                    for orc in orcs
                    if orc.mus_id == m.id]

    many_to_many_temp = [(m.name, mo.mus_id, mo.orc_id)
                          for m in musicians
                          for mo in mus_orc
                          if mo.mus_id == m.id]

    many_to_many = [(orc.name, orc.orchestra_type, mus_name)
                    for mus_name, mus_id, orc_id in many_to_many_temp

```

```

        for orc in orcs if orc.id == orc_id]

print('Задание Б1')
print(first_task(one_to_many))

print("\nЗадание Б2")
print(second_task(one_to_many))

print("\nЗадание Б3")
print(third_task(many_to_many, 'a'))

if __name__ == '__main__':
    main()

```

Результат

Задание Б1

[('Bad Apple', 'Джазовый оркестр', 'Прокофьев'), ('Военный оркестр Александровского полка', 'Военный оркестр', 'Чайковский'), ('Дудочка', 'Духовой оркестр', 'Бетховен'), ('Масленичное чучело', 'Оркестр народных инструментов', 'Прокофьев'), ('Оркестр имени Петра Ильича Чайковского', 'Симфонический оркестр', 'Чайковский'), ('Симфонический оркестр города Гусь Хрустальный', 'Симфонический оркестр', 'Бетховен')]

Задание Б2

[('Бетховен', 2), ('Чайковский', 2), ('Прокофьев', 2)]

Задание Б3

[('Военный оркестр Александровского полка', 'Чайковский'), ('Дудочка', 'Чайковский')]