



**Федеральное государственное бюджетное  
образовательное учреждение  
высшего образования**

---

**«Московский государственный технический  
университет  
имени Н.Э. Баумана  
(национальный исследовательский университет)»  
(МГТУ им. Н.Э. Баумана)**

**Факультет «Информатика и вычислительная техника»  
Кафедра ИУ5 «Системы обработки информации и управления»**

**Курс «Разработка интернет-приложений»**

**Рубежный контроль №1  
Вариант 5Б**

**Выполнил:  
студент группы ИУ5-52Б**

**Ванина П.В.**

**Подпись и дата:**

**Проверил:  
преподаватель каф.  
ИУ5**

**Гапанюк Ю.Е.**

**Подпись и дата:**

**Москва, 2021 г.**

## Описание задания

- 1) Необходимо создать два класса данных в соответствии с Вашим вариантом предметной области, которые связаны отношениями один-ко-многим и многие-ко-многим.
- 2) Необходимо создать списки объектов классов, содержащих тестовые данные (3-5 записей), таким образом, чтобы первичные и вторичные ключи соответствующих записей были связаны по идентификаторам.
- 3) Необходимо разработать запросы в соответствии с Вашим вариантом. При разработке запросов необходимо по возможности использовать функциональные возможности языка Python (list/dict comprehensions, функции высших порядков).

## Вариант Б.

1. «Оркестр» и «Музыкант» связаны соотношением один-ко-многим. Выведите список всех связанных сотрудников и отделов, отсортированный по сотрудникам, сортировка по отделам произвольная.
2. «Оркестр» и «Музыкант» связаны соотношением один-ко-многим. Выведите список отделов с количеством сотрудников в каждом отделе, отсортированный по количеству сотрудников.
3. «Оркестр» и «Музыкант» связаны соотношением многие-ко-многим. Выведите список всех сотрудников, у которых фамилия заканчивается на «ов», и названия их отделов.

## Текст программы

```
class Orchestra:
    def __init__(self, id, type, name):
        self.id=id
        self.type=type
        self.name=name

class Musician:
    def __init__(self, id, fio, instr, id_orch): #self, id, name, id_orch
        self.id=id
        self.fio=fio
        self.instr=instr
        self.id_orch=id_orch

class Conductor:
    def __init__(self, id, fio):
        self.id = id
        self.fio = fio

class OrchCond:
    def __init__(self, id, id_orch, id_cond): #self, id, id_orchestra,
id_conductor
```

```

        self.id = id
        self.id_orch = id_orch
        self.id_cond = id_cond

def main():
    orch=[
        Orchestra(0, 'Symphonic', 'Glenn Miller'),
        Orchestra(1, 'Stringed', 'Paul Mauriat'),
        Orchestra(2, 'Cuivres', 'André Rieu')
    ]

    musician=[
        Musician(0, 'Antonov', 'flute', 0),
        Musician(1, 'Vlasov', 'violin', 1),
        Musician(2, 'Kostochkin', 'tuba', 2),
        Musician(3, 'Leonov', 'cello', 0)
    ]

    conductors=[
        Conductor(0, 'Dudin'),
        Conductor(1, 'Rastrov'),
        Conductor(2, 'Eremin')
    ]

    orch_con=[
        OrchCond(0, 0, 0),
        OrchCond(1, 1, 1),
        OrchCond(2, 2, 2),
        OrchCond(3, 3, 3),
        OrchCond(4, 4, 4)
    ]

    res1 = sorted([(mus.fio, orc.type, orc.name) for mus in musician for orc in
orch if mus.id_orch == orc.id], key=lambda x: x[0])
    print(res1)
    print()

    res2 = sorted({orc.name: len(list(filter(lambda x: x.id_orch == orc.id,
musician))) for orc in orch}.items(), key=lambda x: x[0], reverse=True)
    print(res2)
    print()

    res3 = {cond.fio: [orc.name for orc in orch if orc.id in [orc_cond.id_orch
for orc_cond in orch_con if orc_cond.id_cond == cond.id]] for cond in conductors
if str(cond.fio).endswith('ov')}
    print(res3)

if __name__ == '__main__':
    main()

```

## Экранные формы с примерами выполнения программы

```

[('Antonov', 'Symphonic', 'Glenn Miller'), ('Kostochkin', 'Cuivres', 'André Rieu'), ('Leonov', 'Symphonic', 'Glenn Miller'),
('Vlasov', 'Stringed', 'Paul Mauriat')]

[('Paul Mauriat', 1), ('Glenn Miller', 2), ('André Rieu', 1)]

{'Rastrov': ['Paul Mauriat']}

```