

**Министерство науки и высшего образования Российской Федерации
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 системы управления»
Кафедра «Системы обработки информации и управления»
Курс «Базовые компоненты интернет-технологий»**

Рубежный контроль №1

Вариант 17

(Вариант В, классы – Дирижёр, Оркестр)

Выполнил:

студент группы ИУ5-33Б

Александр Саргсян

Проверил:

к.т.н., доц., Ю. Е. Гапанюк

2022 г.

Текст программы

```
from operator import itemgetter

class Cond:
    """Дирижёр"""

    def __init__(self, id, fio, sal, orch_id):
        self.id = id
        self.fio = fio
        self.sal = sal
        self.orch_id = orch_id

class Orch:
    """Оркестр"""

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

class CondOrch:
    """
    'Дирижёры оркестра' для реализации
    связи многие-ко-многим
    """

    def __init__(self, orch_id, cond_id):
        self.orch_id = orch_id
        self.cond_id = cond_id

# Оркестры
orchs = [
    Orch(1, 'симфонический оркестр'),
    Orch(2, 'струнный оркестр'),
    Orch(3, 'духовой оркестр'),
    Orch(4, 'джазовый оркестр'),
    Orch(5, 'эстрадный оркестр'),
    Orch(6, 'камерный оркестр'),
]

# Дирижёры
conds = [
    Cond(1, 'Андреев', 45000, 1),
    Cond(2, 'Иванов', 35000, 2),
    Cond(3, 'Станков', 45000, 3),
    Cond(4, 'Ноткин', 30000, 3),
    Cond(5, 'Алексеев', 50000, 3),
]

conds_orchs = [
    CondOrch(1, 1),
    CondOrch(2, 2),
    CondOrch(3, 3),
    CondOrch(3, 4),
    CondOrch(3, 5),
]
```

```

CondOrch(4, 1),
CondOrch(5, 2),
CondOrch(5, 3),
CondOrch(6, 4),
CondOrch(6, 5),
]

def main():
    # Соединение данных один-ко-многим
    one_to_many = [(c.fio, c.sal, o.name)
                    for o in orchs
                    for c in conds
                    if c.orch_id == o.id]

    # Соединение данных многие-ко-многим
    many_to_many_temp = [(o.name, co.orch_id, co.cond_id)
                           for o in orchs
                           for co in conds_orchs
                           if o.id == co.orch_id]

    many_to_many = [(c.fio, c.sal, orch_name)
                     for orch_name, orch_id, cond_id in many_to_many_temp
                     for c in conds if c.id == cond_id]

    print('Задание B1')
    res_11 = {}
    for c in conds:
        if c.fio[0] == 'A':
            c_orchs = list(filter(lambda i: i[0] == c.fio, one_to_many))
            c_orchs_names = c_orchs[0][2]
            res_11[c.fio] = c_orchs_names
    print(res_11)

    print('\nЗадание B2')
    res_12_unsorted = []
    for o in orchs:
        o_conds = list(filter(lambda i: i[2] == o.name, one_to_many))
        if len(o_conds) > 0:
            o_sals = [sal for _, sal, _ in o_conds]
            min_sal = sorted(o_sals)
            res_12_unsorted.append((o.name, min_sal[0]))
    res_12 = sorted(res_12_unsorted, key=itemgetter(1), reverse=True)
    print(res_12)

    print('\nЗадание B3')
    res_13 = sorted(many_to_many, key=itemgetter(0))
    print(res_13)

if __name__ == '__main__':
    main()

```

Результаты выполнения

Задание В1

{'Андреев': 'симфонический оркестр', 'Алексеев': 'духовой оркестр'}

Задание В2

[('симфонический оркестр', 45000), ('струнный оркестр', 35000),
(духовой оркестр', 30000)]

Задание В3

[('Алексеев', 50000, 'духовой оркестр'), ('Алексеев', 50000, 'камерный оркестр'), ('Андреев', 45000, 'симфонический оркестр'), ('Андреев', 45000, 'джазовый оркестр'), ('Иванов', 35000, 'струнный оркестр'), ('Иванов', 35000, 'эстрадный оркестр'), ('Ноткин', 30000, 'духовой оркестр'), ('Ноткин', 30000, 'камерный оркестр'), ('Станков', 45000, 'духовой оркестр'), ('Станков', 45000, 'эстрадный оркестр')]