



**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 ИУ5 «Системы обработки информации и управления»**

**Рубежный контроль №1
по дисциплине «Базовые компоненты интернет-технологий»**

**Выполнил:
студент группы ИУ5-33Б
Зорькин А.В.**

**Проверил:
Гапанюк Ю.Е.**

2021 г.

Оглавление

Постановка задачи:.....	3
Текст программы:.....	4
Результаты выполнения:.....	6

Постановка задачи:

Рубежный контроль представляет собой разработку программы на языке Python, которая выполняет следующие действия:

- 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. Запросы сформулированы в терминах классов «Сотрудник» и «Отдел», которые используются в примере. Вам нужно перенести эти требования в Ваш вариант предметной области. При разработке запросов необходимо по возможности использовать функциональные возможности языка Python (list/dict comprehensions, функции высших порядков).

Исходя из выданного задания, требуется:

1. «Отдел» и «Сотрудник» связаны соотношением один-ко-многим. Выведите список всех сотрудников, у которых фамилия начинается с буквы «А», и названия их отделов.
2. «Отдел» и «Сотрудник» связаны соотношением один-ко-многим. Выведите список отделов с минимальной зарплатой сотрудников в каждом отделе, отсортированный по минимальной зарплате.
3. «Отдел» и «Сотрудник» связаны соотношением многие-ко-многим. Выведите список всех связанных сотрудников и отделов, отсортированный по сотрудникам, сортировка по отделам произвольная.

Предметная область:

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

Текст программы:

```
import operator
from operator import itemgetter
from collections import defaultdict

class Musician:
    """Музыкант"""

    def __init__(self, id, surname, salary, orchestra_id):
        self.id = id
        self.surname = surname
        self.salary = salary
        self.orchestra_id = orchestra_id

class Orchestra:
    """Оркестр"""

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

class MusOrc:
    """
    'Музыканты оркестра' для реализации
    СВЯЗИ МНОГИЕ-КО-МНОГИМ
    """

    def __init__(self, orchestra_id, musician_id):
        self.orchestra_id = orchestra_id
        self.musician_id = musician_id

def orchestra_id():
    i = 0
    while True:
        i += 1
        yield i

def musician_id():
    i = 0
    while True:
        i += 1
        yield i

def task_1(one_to_many):
    return [(x[0], x[2]) for x in one_to_many if x[0][0] == 'A']

def task_2(one_to_many): # реализация кажется более оптимизированной, чем
    # при использовании функций высших порядков
    result = {}
    for x in one_to_many:
        if x[2] in result:
            if x[1] < result[x[2]]:
                result[x[2]] = x[1]
        else:
            result[x[2]] = x[1]
    return sorted(result.items(), key=operator.itemgetter(1))
```

```

def task_3(many_to_many):
    result = defaultdict(list)
    for x in many_to_many:
        result[x[0]].append(x[1])
    return sorted(result.items(), key=operator.itemgetter(0))

# Оркестры
orc_id = orchestra_id()
orchestras = [
    Orchestra(next(orc_id), 'Российский национальный оркестр'),
    Orchestra(next(orc_id), 'Лондонский симфонический оркестр'),
    Orchestra(next(orc_id), 'Королевский филармонический оркестр'),
    Orchestra(next(orc_id), 'Российский национальный оркестр(другой)'),
    Orchestra(next(orc_id), 'Лондонский симфонический оркестр(другой)'),
    Orchestra(next(orc_id), 'Королевский филармонический оркестр(другая)'),
]
mus_id = musician_id()
# Музыканты
musicians = [
    Musician(next(mus_id), 'Абрамов', 250000, 1),
    Musician(next(mus_id), 'Ахтамбаев', 315000, 2),
    Musician(next(mus_id), 'Некрасов', 453000, 2),
    Musician(next(mus_id), 'Зорькин', 432000, 3),
    Musician(next(mus_id), 'Прошкина', 343000, 3),
]

musorc = {
    MusOrc(1, 1),
    MusOrc(2, 2),
    MusOrc(3, 3),
    MusOrc(3, 4),
    MusOrc(3, 5),
    MusOrc(4, 1),
    MusOrc(5, 2),
    MusOrc(6, 3),
    MusOrc(6, 4),
    MusOrc(6, 5),
}

def main():
    """Основная функция"""
    # Соединение данных один-ко-многим
    one_to_many = [(m.surname, m.salary, o.name) for o in orchestras for m in
musicians if m.orchestra_id == o.id]

    # Соединение данных многие-ко-многим
    many_to_many_temp = [(o.name, r.orchestra_id, r.musician_id) for o in
orchestras for r in musorc if
        o.id == r.orchestra_id]

    many_to_many = [(m.surname, o_name) for o_name, orchestra_id, musician_id
in many_to_many_temp for m in musicians if
        m.id == musician_id]
    print("Задание №1:", '\n', task_1(one_to_many))
    print("Задание №2:", '\n', task_2(one_to_many))
    print("Задание №3:", '\n', task_3(many_to_many))

if __name__ == '__main__':
    main()

```

Результаты выполнения:

```
Задание №1:
[('Абрамов', 'Российский национальный оркестр'), ('Ахтамбаев', 'Лондонский симфонический оркестр')]

Задание №2:
[('Российский национальный оркестр', 250000), ('Лондонский симфонический оркестр', 315000), ('Королевский филармонический оркестр', 343000)]

Задание №3:
[('Абрамов', ['Российский национальный оркестр', 'Российский национальный оркестр(другой)']), ('Ахтамбаев', ['Лондонский симфонический оркестр', 'Лондонский симфонический оркестр(другой)']), ('Зорькин', ['Королевский филармонический оркестр', 'Королевский филармонический оркестр(другая)']), ('Некрасов', ['Королевский филармонический оркестр', 'Королевский филармонический оркестр(другая)']), ('Прошкина', ['Королевский филармонический оркестр', 'Королевский филармонический оркестр(другая)'])]

Process finished with exit code 0
```

Задание №1:

[('Абрамов', 'Российский национальный оркестр'), ('Ахтамбаев', 'Лондонский симфонический оркестр')]

Задание №2:

[('Российский национальный оркестр', 250000), ('Лондонский симфонический оркестр', 315000), ('Королевский филармонический оркестр', 343000)]

Задание №3:

[('Абрамов', ['Российский национальный оркестр', 'Российский национальный оркестр(другой)']), ('Ахтамбаев', ['Лондонский симфонический оркестр', 'Лондонский симфонический оркестр(другой)']), ('Зорькин', ['Королевский филармонический оркестр', 'Королевский филармонический оркестр(другая)']), ('Некрасов', ['Королевский филармонический оркестр', 'Королевский филармонический оркестр(другая)']), ('Прошкина', ['Королевский филармонический оркестр', 'Королевский филармонический оркестр(другая)'])]

Process finished with exit code 0