



**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»**

Рубежный контроль №1

Выполнил студент группы ИУ5-35Б

Сулайманов Р. Б.

Москва

2022 г.

Полученное задание:

- 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).

Для реализации запроса №2 введите в класс, находящийся на стороне связи «много», произвольный количественный признак, например, «зарплата сотрудника».

Вариант Д 17

| | | |
|----|---------|---------|
| 17 | Дирижер | Оркестр |
|----|---------|---------|

1. «Отдел» и «Сотрудник» связаны соотношением один-ко-многим. Выведите список всех сотрудников, у которых фамилия заканчивается на «ов», и названия их отделов.
2. «Отдел» и «Сотрудник» связаны соотношением один-ко-многим. Выведите список отделов со средней зарплатой сотрудников в каждом отделе, отсортированный по средней зарплате (отдельной функции вычисления среднего значения в Python нет, нужно использовать комбинацию функций вычисления суммы и количества значений).
3. «Отдел» и «Сотрудник» связаны соотношением многие-ко-многим. Выведите список всех отделов, у которых название начинается с буквы «Р», и список работающих в них сотрудников.

Текст программы

Программа разделена на несколько файлов.

- 1) Файл classes.py содержит классы для реализации задания:

```
class conductor:  
    """дирижер"""
```

```

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

class orchestra:
    """оркестр"""

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

class condOrch:
    def __init__(self, id_cond, id_orch):
        self.id_cond = id_cond
        self.id_orch = id_orch

```

2) Файл connections.py содержит функции для создания связей один-ко-многим и многие-ко-многим:

```

def one_to_many(conductors, orchestras):
    return [(cond.surname, cond.salary, orch.name)
            for cond in conductors
            for orch in orchestras
            if cond.orch_id == orch.id]

def many_to_many(conductors, orchestras, condOrch):
    return [(cond.surname, cond.salary, orch.name)
            for oc in condOrch
            for cond in conductors
            for orch in orchestras
            if cond.id == oc.id_cond and orch.id == oc.id_orch]

```

3) Файл processing.py содержит функции для реализации требуемых заданий:

```

def task1(one_to_many):
    return [el for el in one_to_many if el[0].endswith('ов')]

def task2(one_to_many):
    orch = []
    salary = []
    for el in one_to_many:
        if el[2] not in orch:
            orch.append(el[2])
            salary.append((el[1], 1))
        else:
            idx = orch.index(el[2])
            salary[idx] = ((salary[idx][0] + el[1]) / (salary[idx][1] + 1), salary[idx][1] + 1)
    salary = [i[0] for i in salary]
    return sorted(list(zip(orch, salary)), key=lambda p: p[1], reverse=True)

```

```
def task3(many_to_many):
    orch_R = []
    cond_orch_R = []
    for el in many_to_many:
        if el[2].startswith('P'):
            if el[2] not in orch_R:
                orch_R.append(el[2])
                cond_orch_R.append((el[0],))
            else:
                cond_orch_R[orch_R.index(el[2])] += (el[0],)
    return list(zip(orch_R, cond_orch_R))
```

4) Файл main.py реализует работу программы:

```
from processing.classes import conductor, orchestra, condOrch
from processing.connections import one_to_many, many_to_many
from processing.processing import task1, task2, task3

# Оркестры
orchestras = [
    orchestra(1, 'Российский национальный оркестр'),
    orchestra(2, 'Ансамбль песни и пляски'),
    orchestra(3, 'Большой симфонический оркестр'),
    orchestra(4, 'Русская филармония'),
    orchestra(5, 'Виртуозы Москвы'),
    orchestra(6, 'Местные ребята')
]

# Дирижеры
conductors = [
    conductor(1, 'Гончаренко', 15000, 1),
    conductor(2, 'Иванов', 38000, 2),
    conductor(3, 'Самоян', 12500, 4),
    conductor(4, 'Улепетов', 20000, 6),
    conductor(5, 'Вобер', 28500, 3),
    conductor(6, 'Мамонтов', 17500, 5),
    conductor(7, 'Галыгин', 6000, 6),
    conductor(8, 'Кенчур', 30000, 4),
    conductor(9, 'Сергеев', 10000, 2),
    conductor(10, 'Тополев', 18000, 1),
    conductor(11, 'Мамик', 37000, 5),
    conductor(12, 'Везучий', 22000, 3),
    conductor(13, 'Концов', 20500, 3)
]

# Для связи многие-ко-многим
condOrchs = [
    condOrch(1, 1),
    condOrch(2, 1),
    condOrch(3, 2),
    condOrch(4, 2),
    condOrch(5, 1),
    condOrch(6, 3),
    condOrch(7, 4),
    condOrch(8, 4),
    condOrch(9, 5),
    condOrch(10, 5),
    condOrch(11, 5),
    condOrch(12, 6),
    condOrch(13, 6),
]
```

```

    condOrch(7, 1),
    condOrch(8, 2),
    condOrch(2, 3),
    condOrch(3, 4),
    condOrch(4, 5),
    condOrch(5, 6)
]

# Установка связей
one_to_many = one_to_many(conductors, orchestras)
many_to_many = many_to_many(conductors, orchestras, condOrchs)

# print(*one_to_many, sep='\n', end='\n\n')
# print(*many_to_many, sep='\n')

def main():
    print('Task 1', task1(one_to_many), sep='\n', end='\n\n')
    print('Task 2', task2(one_to_many), sep='\n', end='\n\n')
    print('Task 3', task3(many_to_many), sep='\n', end='\n\n')

if __name__ == '__main__':
    main()

```

Работа программы

1) После связывания один-ко-многим имеем:

('Гончаренко', 15000, 'Российский национальный оркестр')

('Иванов', 38000, 'Ансамбль песни и пляски')

('Самонян', 12500, 'Русская филармония')

('Улепетов', 20000, 'Местные ребята')

('Бобер', 28500, 'Большой симфонический оркестр')

('Мамонтов', 17500, 'Виртуозы Москвы')

('Галыгин', 6000, 'Местные ребята')

('Кенчур', 30000, 'Русская филармония')

('Сергеев', 10000, 'Ансамбль песни и пляски')

('Тополев', 18000, 'Российский национальный оркестр')

('Мамик', 37000, 'Виртуозы Москвы')

('Везучий', 22000, 'Большой симфонический оркестр')

('Концов', 20500, 'Большой симфонический оркестр')

2) После связывания многие-ко-многим имеем:

('Гончаренко', 15000, 'Российский национальный оркестр')

('Иванов', 38000, 'Российский национальный оркестр')

('Самонян', 12500, 'Ансамбль песни и пляски')

('Улепетов', 20000, 'Ансамбль песни и пляски')

('Бобер', 28500, 'Российский национальный оркестр')

('Мамонтов', 17500, 'Большой симфонический оркестр')

('Галыгин', 6000, 'Русская филармония')

('Кенчур', 30000, 'Русская филармония')

('Сергеев', 10000, 'Виртуозы Москвы')

('Тополев', 18000, 'Виртуозы Москвы')

('Мамик', 37000, 'Виртуозы Москвы')

('Везучий', 22000, 'Местные ребята')

('Концов', 20500, 'Местные ребята')

('Галыгин', 6000, 'Российский национальный оркестр')

('Кенчур', 30000, 'Ансамбль песни и пляски')

('Иванов', 38000, 'Большой симфонический оркестр')

('Самонян', 12500, 'Русская филармония')

('Улепетов', 20000, 'Виртуозы Москвы')

('Бобер', 28500, 'Местные ребята')

3) Результаты выполнения заданий:

Task 1

[('Иванов', 38000, 'Ансамбль песни и пляски'), ('Улепетов', 20000, 'Местные ребята'), ('Мамонтов', 17500, 'Виртуозы Москвы'), ('Концов', 20500, 'Большой симфонический оркестр')]

Task 2

[('Виртуозы Москвы', 27250.0), ('Ансамбль песни и пляски', 24000.0), ('Русская филармония', 21250.0), ('Российский национальный оркестр', 16500.0), ('Большой симфонический оркестр', 15250.0), ('Местные ребята', 13000.0)]

Task 3

[('Российский национальный оркестр', ('Гончаренко', 'Иванов', 'Бобер', 'Галыгин')), ('Русская филармония', ('Галыгин', 'Кенчур', 'Самонян'))]