

Московский государственный технический университет  
им. Н.Э. Баумана

Факультет “Информатика и системы управления”  
Кафедра “Системы обработки информации и управления”



Дисциплина «Парадигмы и конструкции языков программирования»

Отчет по рубежному контролю №1  
«Основные конструкции языка Python»  
“Вариант Д-18”

**Выполнил:**  
студент группы ИУ5-36Б  
Юдин Григорий Олегович  
**Проверил:**  
Гапанюк Юрий Евгеньевич

Москва, 2025

### Вариант Д.

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 «А», и список работающих в них сотрудников.

18	Музыкальное произведение	Оркестр
----	-----------------------------	---------

```
C:\Users\grigo\PyCharmMiscProject\.venv\Scripts\python.exe C:\Users\grigo\PyCharmMiscProject\RK1.py
Задание Д1
[('Вальсов', 'Академический оркестр'), ('Концертов', 'Академический оркестр'), ('Маршов', 'Симфонический оркестр')]

Задание Д2
[('Симфонический оркестр', 6.5), ('Академический оркестр', 8.75), ('Камерный ансамбль', 12.0)]

Задание Д3
{'Академический оркестр': ['Вальсов', 'Концертов'], 'Альтернативный оркестр': ['Вальсов', 'Маршов']}
```

```
from operator import itemgetter
```

```
class MusicalWork:
```

```
    """Музыкальное произведение"""
```

```
    def __init__(self, id, title, duration, orchestra_id):
        self.id = id
        self.title = title
        self.duration = duration
        self.orchestra_id = orchestra_id
```

```
class Orchestra:
```

```
    """Оркестр"""
```

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

```
class WorkOrchestra:
```

```
    """
```

```
    Произведения оркестра
    (связь многие-ко-многим)
    """
```

```
    def __init__(self, orchestra_id, work_id):
```

```
self.orchestra_id = orchestra_id
self.work_id = work_id
```

```
# Оркестры
```

```
orchestras = [
    Orchestra(1, "Академический оркестр"),
    Orchestra(2, "Симфонический оркестр"),
    Orchestra(3, "Камерный ансамбль"),
    Orchestra(4, "Альтернативный оркестр"),
]
```

```
# Музыкальные произведения
```

```
works = [
    MusicalWork(1, "Вальсов", 7.5, 1),
    MusicalWork(2, "Маршов", 5.0, 2),
    MusicalWork(3, "Реквием", 12.0, 3),
    MusicalWork(4, "Концертов", 10.0, 1),
    MusicalWork(5, "Сюита", 8.0, 2),
]
```

```
# Связь многие-ко-многим
```

```
works_orchestras = [
    WorkOrchestra(1, 1),
    WorkOrchestra(1, 4),
    WorkOrchestra(2, 2),
    WorkOrchestra(2, 5),
    WorkOrchestra(3, 3),
    WorkOrchestra(4, 1),
    WorkOrchestra(4, 2),
]
```

```
def main():
```

```
    # Связь один-ко-многим
```

```
    one_to_many = [
        (w.title, w.duration, o.name)
        for o in orchestras
        for w in works
        if w.orchestra_id == o.id
    ]
```

```
    # Связь многие-ко-многим
```

```
    many_to_many_temp = [
        (o.name, wo.orchestra_id, wo.work_id)
        for o in orchestras
        for wo in works_orchestras
        if o.id == wo.orchestra_id
    ]
```

```
    many_to_many = [
        (w.title, w.duration, orchestra_name)
    ]
```

```

        for orchestra_name, orchestra_id, work_id in many_to_many_temp
        for w in works
        if w.id == work_id
    ]

# Задание Д1
print("Задание Д1")
res_d1 = [
    (title, orchestra)
    for title, _, orchestra in one_to_many
    if title.endswith("ов")
]
print(res_d1)

# Задание Д2
print("\nЗадание Д2")
res_d2_unsorted = []
for o in orchestras:
    o_works = list(filter(lambda x: x[2] == o.name, one_to_many))
    if len(o_works) > 0:
        durations = [dur for _, dur, _ in o_works]
        avg_duration = sum(durations) / len(durations)
        res_d2_unsorted.append((o.name, avg_duration))

res_d2 = sorted(res_d2_unsorted, key=itemgetter(1))
print(res_d2)

# Задание Д3
print("\nЗадание Д3")
res_d3 = {}
for o in orchestras:
    if o.name.startswith("A"):
        o_works = list(filter(lambda x: x[2] == o.name, many_to_many))
        work_titles = [title for title, _, _ in o_works]
        res_d3[o.name] = work_titles

print(res_d3)

if __name__ == "__main__":
    main()

```