

Листинг кода:

```
from operator import itemgetter

class Conductor:
    "Дирижёр"
    def __init__(self, id, name, experience, orchestra_id):
        self.id = id
        self.name = name
        self.experience = experience
        self.orchestra_id = orchestra_id

    class Orchestra:
        "Оркестр"
        def __init__(self, id, name):
            self.id = id
            self.name = name

        class ConductorOrchestra:
            "Связь 'Дирижёры оркестров' для реализации многие-ко-многим"
            def __init__(self, orchestra_id, conductor_id):
                self.orchestra_id = orchestra_id
                self.conductor_id = conductor_id

    orchestras = [
        Orchestra(1, 'Арбатский оркестр'),
        Orchestra(2, 'Ленинградский оркестр'),
        Orchestra(3, 'Тверской оркестр'),
        Orchestra(4, 'Академический оркестр'),
        Orchestra(5, 'Профсоюзный оркестр'),
    ]

    conductors = [
        Conductor(1, 'Артамонов', 5, 1),
```

```
Conductor(2, 'Петров', 8, 2),
Conductor(3, 'Иваненко', 12, 3),
Conductor(4, 'Иванов', 9, 3),
Conductor(5, 'Сидоров', 4, 4),
Conductor(6, 'Алексеев', 15, 4),
]
```

```
# Связи многие-ко-многим
```

```
conductors_orchestras = [
ConductorOrchestra(1, 1),
ConductorOrchestra(2, 2),
ConductorOrchestra(3, 3),
ConductorOrchestra(3, 4),
ConductorOrchestra(4, 5),
ConductorOrchestra(4, 6),
ConductorOrchestra(5, 2),
]
```

```
def main():
```

```
    "Основная функция"
    one_to_many = [(c.name, c.experience, o.name)
        for o in orchestras
        for c in conductors
        if c.orchestra_id == o.id]
```

```
    many_to_many_temp = [(o.name, co.orchestra_id, co.conductor_id)
        for o in orchestras
        for co in conductors_orchestras
        if o.id == co.orchestra_id]
```

```
    many_to_many = [(c.name, c.experience, orchestra_name)
```

```
for orchestra_name, orchestra_id, conductor_id in many_to_many_temp
    for c in conductors if c.id == conductor_id]
```

```
print('Задание Г1')
res_1 = []
for o in orchestras:
    if o.name.startswith('A'):
        o_conductors = list(filter(lambda i: i[2] == o.name, one_to_many))
        if len(o_conductors) > 0:
            res_1.append((o.name, [name for name, _, _ in o_conductors]))
print(res_1)
```

```
print('\nЗадание Г2')
res_2_unsorted = []
for o in orchestras:
    o_conductors = list(filter(lambda i: i[2] == o.name, one_to_many))
    if len(o_conductors) > 0:
        experiences = [exp for _, exp, _ in o_conductors]
        max_exp = max(experiences)
        res_2_unsorted.append((o.name, max_exp))
res_2 = sorted(res_2_unsorted, key=itemgetter(1), reverse=True)
print(res_2)
```

```
print('\nЗадание Г3')
res_3 = sorted(many_to_many, key=itemgetter(2))
print(res_3)
if __name__ == '__main__':
    main()
```

Результаты выполнения программы

Задание Г1

Список всех оркестров, название которых начинается с буквы «А», и дирижёров, работающих в них:

[('Арбатский оркестр', ['Артамонов']),
 ('Академический оркестр', ['Сидоров', 'Алексеев'])]

Задание Г2

Список оркестров с максимальным стажем дирижёров, отсортированный по убыванию стажа:

[('Академический оркестр', 15),
 ('Тверской оркестр', 12),
 ('Ленинградский оркестр', 8),
 ('Арбатский оркестр', 5),
 ('Профсоюзный оркестр', 8)]

Задание Г3

Список всех связанных дирижёров и оркестров (многие-ко-многим), отсортированный по оркестрам:

[('Артамонов', 5, 'Арбатский оркестр'),
 ('Сидоров', 4, 'Академический оркестр'),
 ('Алексеев', 15, 'Академический оркестр'),
 ('Петров', 8, 'Ленинградский оркестр'),
 ('Петров', 8, 'Профсоюзный оркестр'),
 ('Иваненко', 12, 'Тверской оркестр'),
 ('Иванов', 9, 'Тверской оркестр')]

Пояснения к решению

1. Классы данных:

Conductor — дирижёр (id, имя, стаж, id оркестра);

Orchestra — оркестр (id, название);

ConductorOrchestra — связь многие-ко-многим.

2. Тестовые данные:

5 оркестров и 6 дирижёров;

связи реализованы корректно через идентификаторы.

3. Запросы варианта Г:

Г1: Найдены оркестры, название которых начинается с буквы «А» (Арбатский, Академический), и список их дирижёров.

Г2: Для каждого оркестра найден максимальный стаж дирижёра, затем выполнена сортировка по убыванию.

Г3: Сформирован список всех связей дирижёров и оркестров (многие-ко-многим), отсортированный по оркестрам.