

Вариант 18 А

Музыкальное произведение Оркестр

```
from operator import itemgetter
```

```
class MusicalPiece:
```

```
    """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 MusicOrch:
```

```
    """Музыкальные произведения оркестра для реализации связи  
многие-ко-многим"""
```

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

```
        self.orchestra_id = orchestra_id
```

```
        self.music_id = music_id
```

```
    """Оркестр"""
```

```
orch = [
```

```
    Orchestra(1, 'Оркестр имени Чайковского'),
```

```
    Orchestra(2, 'Народный Оркестр'),
```

```
    Orchestra(3, 'Камерный оркестр Карелии'),
```

```
    Orchestra(11, 'ГосОркестр им. Сталина'),
```

```
    Orchestra(22, 'Могучая кучка'),
```

```
    Orchestra(33, 'Штутградский Оркестр'),
```

```
]
```

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

```

mus = [
    MusicalPiece(1, 'Симфония №1', 120, 1),
    MusicalPiece(2, 'Симфония №3', 40, 2),
    MusicalPiece(3, 'Увертюра', 15, 3),
    MusicalPiece(4, 'Половецкие пляски', 30, 3),
    MusicalPiece(5, 'Концерт для скрипки', 21, 3),
]

```

```

music_orch = [
    MusicOrch(1, 1),
    MusicOrch(2, 2),
    MusicOrch(3, 3),
    MusicOrch(3, 4),
    MusicOrch(3, 5),

    MusicOrch(11, 1),
    MusicOrch(22, 2),
    MusicOrch(33, 3),
    MusicOrch(33, 4),
    MusicOrch(33, 5),
]

```

```

def main():
    # Соединение данных один-ко-многим
    one_to_many = [(m.title, m.duration, o.name)
                    for o in orch
                    for m in mus
                    if m.orchestra_id == o.id]

    # Соединение данных многие-ко-многим

    many_to_many_temp = [(o.name, mo.orchestra_id, mo.music_id)
                          for o in orch
                          for mo in music_orch
                          if o.id == mo.orchestra_id]

    many_to_many = [(m.title, m.duration, orchestra_name)
                    for orchestra_name, orchestra_id, music_id in
many_to_many_temp
                    for m in mus if m.id == music_id]

    print('Задание A1')
    res_1 = sorted(one_to_many, key=itemgetter(2))
    print(res_1)

```

```

print('\nЗадание A2')
res_2 = []
for orc in orch:
    temp_orc = list(filter(lambda i: i[2] == orc.name, one_to_many))
    total_duration=0
    if len(temp_orc) > 0:
        for i in temp_orc:
            total_duration+=i[1]
        res_2.append((orc.name, total_duration))

res_2 = sorted(res_2, key=itemgetter(1), reverse=True)
print(res_2)

print('\nЗадание A3')
res_3 = {}
for o in orch:
    if 'Оркестр' in o.name:
        this_orc_music=list(filter(lambda i: i[2] == o.name,
many_to_many))
        title_list = [i[0] for i in this_orc_music]
        res_3[o.name]=title_list

print(res_3)

if __name__ == '__main__':
    main()

```

## Результаты выполнения:

### Задание A1

[('Увертюра', 15, 'Камерный оркестр Карелии'), ('Половецкие пляски', 30, 'Камерный оркестр Карелии'), ('Концерт для скрипки', 21, 'Камерный оркестр Карелии'), ('Симфония №3', 40, 'Народный Оркестр'), ('Симфония №1', 120, 'Оркестр имени Чайковского')]

### Задание A2

[('Оркестр имени Чайковского', 120), ('Камерный оркестр Карелии', 66), ('Народный Оркестр', 40)]

### Задание A3

{'Оркестр имени Чайковского': ['Симфония №1'], 'Народный Оркестр': ['Симфония №3'], 'ГосОркестр им. Сталина': ['Симфония №1'], 'Штутгартский Оркестр': ['Увертюра', 'Половецкие пляски', 'Концерт для скрипки']}