

Рубежный контроль

Вариант 18

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

Вариант Б:

- 1) «Оркестр» и «Музыкальное произведение» связаны соотношением один-ко-многим. Выведите список всех связанных музыкальных произведений и оркестров, отсортированных по названиям произведений, сортировка по оркестрам произвольная.
- 2) «Оркестр» и «Музыкальное произведение» связаны соотношением один-ко-многим. Выведите список оркестров с количеством музыкальных произведений у каждого оркестра, отсортированный по количеству произведений.
- 3) «Оркестр» и «Музыкальное произведение» связаны соотношением многие-ко-многим. Выведите список всех музыкальных произведений, у которых имя автора заканчивается на «н», и оркестры, которые их исполняют.

Код программы

```
from operator import itemgetter

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

class MusicalComposition:
    def __init__(self, id, name, author, orchestra_id):
        self.id = id
        self.name = name
        self.author = author
        self.orchestra_id = orchestra_id

class MusicalCompositionOrchestra:
```

```

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

orchestras = [
    Orchestra(1, "Leningrad Philharmonic"),
    Orchestra(2, "Benzogang Orchestra"),
    Orchestra(3, "Budapest Festival Orchestra"),
]

musicalCompositions = [
    MusicalComposition(1, "Morning", "Edward Grieg", 1),
    MusicalComposition(2, "Moonlight Sonata", "Ludwig van Beethoven", 2),
    MusicalComposition(3, "Saber Dance", "Aram Khachaturian", 3),
    MusicalComposition(4, "Summer Storm", "Antonio Vivaldi ", 3),
    MusicalComposition(5, "Fly", "Ludovico Einaudi", 1),
    MusicalComposition(6, "I'm swimming in the river", "Ivan Dremine", 1)
]

composition_to_orchestra = [
    MusicalCompositionOrchestra(1, 1),
    MusicalCompositionOrchestra(2, 2),
    MusicalCompositionOrchestra(3, 3),
    MusicalCompositionOrchestra(3, 4),
    MusicalCompositionOrchestra(1, 5),
    MusicalCompositionOrchestra(1, 6),
]

def first_task(composition_list):
    return sorted(composition_list, key=itemgetter(0))

def second_task(composition_list):
    res = []
    temp = dict()
    for i in composition_list:
        if i[2] in temp:
            temp[i[2]] += 1
        else:
            temp[i[2]] = 1
    for i in temp.keys():
        res.append((i, temp[i]))

    res.sort(key=itemgetter(1), reverse=True)
    return res

def third_task(composition_list, suffix):
    res = [(i[0], i[2]) for i in composition_list if str(i[1]).endswith(suffix)]
    return res

def main():

```

```

one_to_many = [(composition.name, composition.author, orchestra.name)
                for orchestra in orchestras
                for composition in musicalCompositions
                if composition.orchestra_id == orchestra.id]

many_to_many_temp = [(orchestra.name, relation.orchestra_id,
relation.composition_id)
                      for orchestra in orchestras
                      for relation in composition_to_orchestra
                      if relation.orchestra_id == orchestra.id]

many_to_many = [(composition.name, composition.author, orchestra_name)
                 for orchestra_name, orchestra_id, composition_id in
many_to_many_temp
                 for composition in musicalCompositions if composition.id ==
composition_id]

print("\nSelection 51:")
print(first_task(one_to_many))
print("\n")

print("\nSelection 52")
print(second_task(one_to_many))
print("\n")

print("\nSelection 53")
print(third_task(many_to_many, 'n'))
print("\n")

if __name__ == '__main__':
    main()

```

Результаты выполнения

```

Selection 51:
[('Fly', 'Ludovico Einaudi', 'Leningrad Philharmonic'), ('I'm swimming in the river", 'Ivan Dremin', 'Leningrad
Philharmonic'), ('Moonlight Sonata', 'Ludwig van Beethoven', 'Benzogang Orchestra'), ('Morning', 'Edward Grieg',
'Leningrad Philharmonic'), ('Saber Dance', 'Aram Khachaturian', 'Budapest Festival Orchestra'), ('Summer Storm'
, 'Antonio Vivaldi ', 'Budapest Festival Orchestra')]

Selection 52
[('Leningrad Philharmonic', 3), ('Budapest Festival Orchestra', 2), ('Benzogang Orchestra', 1)]

Selection 53
[('I'm swimming in the river", 'Leningrad Philharmonic'), ('Moonlight Sonata', 'Benzogang Orchestra'), ('Saber D
ance', 'Budapest Festival Orchestra')]

```