Группа: ИУ5-32Б
Студент: Подгорнова Карина

Рубежный контроль 1

17	Дирижер	Оркестр

Вариант Б.

1. «Дирижер» и «Оркестр» связаны соотношением один-ко-многим. Выведите список всех связанных дирижеров и оркестров, отсортированный по дирижерам (алфавитный порядок), сортировка по оркестрам произвольная.
2. «Дирижер» и «Оркестр» связаны соотношением один-ко-многим. Выведите список оркестром с дирижеров в каждом языке программирования, отсортированный по количеству дирижеров.
3. «Дирижер» и «Оркестр» связаны соотношением многие-ко-многим. Выведите список дирижеров, фамилии которых закачиваются на “ov”, названия оркестров, в которых они состоят.

Текст программы:

```
from operator import itemgetter

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

class Conductor:
    def __init__(self, id, fio, salary , orchestra_id):
        self.id = id
        self.fio = fio
        self.salary = salary
        self.orchesrta_id = orchestra_id

class OrchCond:
    def __init__(self, orchestra_id, cond_id):
        self.orchestra_id = orchestra_id
        self.cond_id = cond_id

orchestras = [
    Orchestra(1, "Symphony"),
```

```
    Orchestra(2, "Chamber"),
    Orchestra(3, "String"),
]

conductors = [
    Conductor(1, "Ivanov", 43,1),
    Conductor(2, "Achapkin", 45,2),
    Conductor(3, "Fundukov", 61,3),
    Conductor(4, "Shishkin", 38, 3),
    Conductor(5, "Pushkin", 47, 1),
    Conductor(5, "Levitan", 42, 1)
]

orchestras_conductors = [
    OrchCond(1, 1),
    OrchCond(2, 2),
    OrchCond(3, 3),
    OrchCond(3, 4),
    OrchCond(1, 5),
]


def first_task(cond_list):
    res1 = sorted(cond_list, key=itemgetter(0))
    return res1

def second_task(cond_list):
    res2 = []
    temp_dict = dict()
    for i in cond_list:
        if i[2] in temp_dict:
            temp_dict[i[2]] += 1
        else:
            temp_dict[i[2]] = 1
    for i in temp_dict.keys():
        res2.append((i, temp_dict[i]))

    res2.sort(key=itemgetter(1), reverse=True)
    return res2


def third_task(cond_list, end_ch):
```

```
    res3 = [(i[0], i[2]) for i in cond_list if i[0].endswith(end_ch)]
    return res3


def main():
    one_to_many = [(cond.fio, cond.salary, orch.name)
                   for orch in orchestras
                   for cond in conductors
                   if cond.orchesrta_id == orch.id]

    many_to_many_temp = [(orch.name, oc.orchestra_id, oc.cond_id)
                         for orch in orchestras
                         for oc in orchestras_conductors
                         if oc.orchestra_id == orch.id]

    many_to_many = [(cond.fio, cond.salary, orch_name)
                    for orch_name, orch_id, cond_id in many_to_many_temp
                    for cond in conductors if cond.id == cond_id]

    print('Задание Б1')
    print(first_task(one_to_many))

    print("Задание Б2")
    print(second_task(one_to_many))

    print("Задание Б3")
    print(third_task(many_to_many, 'ov'))


if __name__ == '__main__':
    main()
```

Результат выполнения программы:

```
Задание Б1
[('Achapkin', 45, 'Chamber'), ('Fundukov', 61, 'String'), ('Ivanov', 43, 'Symphony'), ('Levitan', 42, 'Symphony'), ('Pushkin', 47, 'Symphony'), ('Shishkin', 38, 'String')]
Задание Б2
[('Symphony', 3), ('String', 2), ('Chamber', 1)]
Задание Б3
[('Ivanov', 'Symphony'), ('Fundukov', 'String')]

Process finished with exit code 0
```