Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования

«Московский государственный технический университет имени Н.Э. Баумана»

(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ Информатика и системы управления

КАФЕДРА Системы обработки информации и управления (ИУ5)

О т ч е т

по рубежному контролю №1

Дисциплина: Разработка Интернет-Приложений

Студент гр. ИУ5-54Б _____ Самойлов А.М.

(Подпись, дата) (И.О. Фамилия)

Преподаватель _____ Гапанюк Ю.Е.

(Подпись, дата) (И.О. Фамилия)

Москва, 2020

Задание

Вариант Г

Вариант предметной области

18	Музыкальное произведение	Оркестр

1. «Оркестр» и «Музыкальное произведение» связаны соотношением один-ко-многим. Выведите список всех отделов, у которых название начинается с буквы «А», и список работающих в них сотрудников.
2. «Оркестр» и «Музыкальное произведение» связаны соотношением один-ко-многим. Выведите список отделов с максимальной зарплатой сотрудников в каждом отделе, отсортированный по максимальной зарплате.
3. «Оркестр» и «Музыкальное произведение» связаны соотношением многие-ко-многим. Выведите список всех связанных сотрудников и отделов, отсортированный по отделам, сортировка по сотрудникам произвольная.

Текст программы

```
from operator import itemgetter

# Самойлов Алексей Михайлович, группа ИУ5-54Б, Вариант Г-18

tasks = [
    "– Список оркестров на букву 'А' и их композиций",                          # Г1
    "– Список оркестров с максимальной длиной композиции, сортировка по длине", # Г2
    "– Список всех связанных композиций и оркестров, сортировка по оркестрам"   # Г3
]


class composition:
    #Музыкальная композиция

    def __init__(self, id, title, author, length, orchestra_id):
        self.id = id
        self.title = title
        self.author = author
        self.length = length
        self.orchestra_id = orchestra_id


class orchestra:
    #Оркестр
    def __init__(self, id, name):
        self.id = id
```

```
        self.name = name


class CompOrch:
    """
    'Музыкальное произведение оркестров' для реализации
    связи многие-ко-многим
    """
    def __init__(self, orchestra_id, composition_id):
        self.orchestra_id = orchestra_id
        self.composition_id = composition_id


# Оркестры
orchestras = [
    orchestra(1, 'Большой симфонический оркестр'),
    orchestra(2, 'Симфонический оркестр Большого театра'),
    orchestra(3, 'Академический симфонический оркестр'),
]

# Композиции
compositions = [
    composition(1, 'К Элизе', 'Людвиг ван Бетховен', 170, 1),
    composition(2, 'Турецкое рондо', 'Вольфганг Амадей Моцарт', 258, 2),
    composition(3, 'Аве Мария', 'Франц Шуберт', 324, 3),
    composition(4, 'Утро', 'Эдвард Григ', 563, 1),
    composition(5, 'Лунный свет', 'Клод Дебюсси', 434, 2),
    composition(6, 'Лебедь', 'Камиль Сен-Санс', 453, 3),
]

# Оркестры - музыкальные произведения
orchestra_composition = [
    CompOrch(1, 1),
    CompOrch(2, 2),
    CompOrch(3, 3),
    CompOrch(1, 4),
    CompOrch(2, 5),
    CompOrch(3, 6)
]


def main():
    """Основная функция"""

    #один ко многим
    one_to_many = [(b.title, b.length, s.name)
                   for s in orchestras
                   for b in compositions
                   if b.orchestra_id == s.id]

    #многие ко многим
```

```
    many_to_many_temp = [(s.name, bs.orchestra_id, bs.composition_id)
                         for s in orchestras
                         for bs in orchestra_composition
                         if s.id == bs.orchestra_id]

    many_to_many = [(b.title, orchestra_name)
                    for orchestra_name, orchestra_id, composition_id in many_to_many_
temp
                    for b in compositions if b.id == composition_id]

    print('— Задание Г1'  + "\n" + tasks[0])
    res_11 = list(filter(lambda x: x[2].startswith('A'), one_to_many))
    for i in res_11:
        print(i)


    print('\n\n— Задание Г2' + "\n" + tasks[1])
    res_12_unsorted = []

    for s in orchestras:
        s_compositions = list(filter(lambda i: i[2] == s.name, one_to_many))
        if len(s_compositions) > 0:
            s_length = [length for _, length, _ in s_compositions]
            s_length_max = max(s_length)
            res_12_unsorted.append((s.name, s_length_max))
    res_12 = sorted(res_12_unsorted, key=itemgetter(1), reverse=True)
    print(res_12)


    print('\n\n— Задание Г3'  + "\n" + tasks[2])
    res_13 = sorted(many_to_many, key=itemgetter(1))
    print(res_13)
if __name__ == '__main__':
    main()
```

Результат работы программы

```
— Задание Г1
— Список оркестров на букву 'А' и их композиций
('Аве Мария', 324, 'Академический симфонический оркестр')
('Лебедь', 453, 'Академический симфонический оркестр')


— Задание Г2
— Список оркестров с максимальной длиной композиции, сортировка по длине
[('Большой симфонический оркестр', 563), ('Академический симфонический оркестр', 453), ('Симфонический оркестр Б
ольшого театра', 434)]


— Задание Г3
— Список всех связанных композиций и оркестров, сортировка по оркестрам
[('Аве Мария', 'Академический симфонический оркестр'), ('Лебедь', 'Академический симфонический оркестр'), ('К Эл
изе', 'Большой симфонический оркестр'), ('Утро', 'Большой симфонический оркестр'), ('Турецкое рондо', 'Симфониче
ский оркестр Большого театра'), ('Лунный свет', 'Симфонический оркестр Большого театра')]
```