

**Московский государственный технический
университет им. Н.Э. Баумана.**

Факультет «Информатика и управление»
Кафедра «Системы обработки информации и управления»

Курс «Разработка интернет-приложений»
Отчет по рубежному контролю №1

Выполнил:
студент группы ИУ5-51Б
Рогозин Данила
Подпись и дата:

Проверил:
преподаватель каф. ИУ5
Гапанюк Ю.Е.
Подпись и дата:

г. Москва, 2020 г.

Рогозин Данила ИУ5-51Б Вариант А18

1. «Музыкальное произведение» и «Оркестр» связаны соотношением один-ко-многим. Выведите список всех оркестров и произведений, которые они исполняют и в каком количестве, отсортированный по оркестрам, сортировка по дирижерам произвольная.
2. «Музыкальное произведение» и «Оркестр» связаны соотношением один-ко-многим. Выведите список оркестров с суммарным количеством сыгранных произведений, отсортированный по суммарной количеству.
3. «Музыкальное произведение» и «Оркестр» связаны соотношением многие-ко-многим. Выведите список всех оркестров, у которых в названии присутствует слово «симфонический», и список работающих в них дирижеров.

Текст программы

main.py

```
from operator import itemgetter

class MusicalComposition:
    """Музыкальное произведение"""

    def __init__(self, id, title, quantity, Orchestra_id):
        self.id = id
        self.title = title
        self.quantity = quantity
        self.Orchestra_id = Orchestra_id

class Orchestra:
    """Оркестр"""

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

class MusicalCompositionOrchestra:
    """
    'Репертуар оркестров' для реализации
    связи многие-ко-многим
    """

    def __init__(self, Orchestra_id, MusicalComposition_id):
        self.Orchestra_id = Orchestra_id
        self.MusicalComposition_id = MusicalComposition_id

# Оркестры
Orchestras = [
    Orchestra(1, 'Российский национальный оркестр'),
    Orchestra(2, 'Симфонический оркестр Большого театра'),
    Orchestra(3, 'Большой симфонический оркестр им. Чайковского'),
]

# Музыкальные произведения
MusicalCompositions = [
    MusicalComposition(1, 'Вторая (богатырская) симфония Александра Бородина',
    15, 1),
    MusicalComposition(2, 'Фрагменты из оперы «Руслан и Людмила», 5, 2),
    MusicalComposition(3, 'Марш Черномора', 40, 3),
    MusicalComposition(4, '«Баба-яга», фантазия-скерцо', 35, 3),
```

```

    MusicalComposition(5, 'Цыганский танец из оперы «Русалка», 20, 3),
]

MusicalCompositions_Orchestras = [
    MusicalCompositionOrchestra(1, 1),
    MusicalCompositionOrchestra(2, 2),
    MusicalCompositionOrchestra(3, 3),
    MusicalCompositionOrchestra(3, 4),
    MusicalCompositionOrchestra(3, 5),

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

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

    # Соединение данных один-ко-многим
    one_to_many = [(e.title, e.quantity, d.name)
                   for d in Orchestras
                   for e in MusicalCompositions
                   if e.Orchestra_id == d.id]

    # Соединение данных многие-ко-многим
    many_to_many_temp = [(d.name, ed.Orchestra_id, ed.MusicalComposition_id)
                          for d in Orchestras
                          for ed in MusicalCompositions_Orchestras
                          if d.id == ed.Orchestra_id]

    many_to_many = [(e.title, e.quantity, Orchestra_name)
                    for Orchestra_name, Orchestra_id, MusicalComposition_id in
many_to_many_temp
                    for e in MusicalCompositions if e.id ==
MusicalComposition_id]

    print('Задание A1')
    res_11 = sorted(one_to_many, key=itemgetter(2))
    print(res_11)

    print('\nЗадание A2')
    res_12_unsorted = []
    # Перебираем все оркестры
    for d in Orchestras:
        # Список оркестров
        d_MusicalCompositions = list(filter(lambda i: i[2] == d.name,
one_to_many))
        if len(d_MusicalCompositions) > 0:
            # Количество сыгранных раз
            d_quantitys = [quantity for _, quantity, _ in
d_MusicalCompositions]
            # Суммарная количество сыгранных композиций
            d_quantitys_sum = sum(d_quantitys)
            res_12_unsorted.append((d.name, d_quantitys_sum))

    # Сортировка по суммарной зарплате
    res_12 = sorted(res_12_unsorted, key=itemgetter(1), reverse=True)
    print(res_12)

    print('\nЗадание A3')
    res_13 = {}
    # Перебираем все оркестры
    for d in Orchestras:
        if 'симфонический' in d.name:

```

```

        # Список оркестров
        d_MusicalCompositions = list(filter(lambda i: i[2] == d.name,
many_to_many))
        d_MusicalCompositions_names = [x for x, _, _ in
d_MusicalCompositions]
        # Добавляем результат в словарь
        res_13[d.name] = d_MusicalCompositions_names

    print(res_13)

if __name__ == '__main__':
    main()

```

Результат

```

Задание A1
[('Марш Черномора', 40, 'Большой симфонический оркестр им. Чайковского'), ('«Баба-яга», фантазия-скерцо', 35, 'Большой симфонический оркестр им. Чайковского'),

Задание A2
[('Большой симфонический оркестр им. Чайковского', 95), ('Российский национальный оркестр', 15), ('Симфонический оркестр Большого театра', 5)]

Задание A3
{'Большой симфонический оркестр им. Чайковского': ['Марш Черномора', '«Баба-яга», фантазия-скерцо', 'Цыганский танец из оперы «Русалка»']}

```