



**Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)**

ФАКУЛЬТЕТ «ИНЖЕНЕРНЫЙ БИЗНЕС И МЕНЕДЖМЕНТ»

КАФЕДРА «ПРОМЫШЛЕННАЯ ЛОГИСТИКА» (ИБМ-3)

Отчёт

По рубежному контролю №1

По дисциплине «Парадигмы и конструкции языков программирования»

Вариант 18

Студент ИБМ3-34Б:

Нургалиева А.А.

Преподаватель:

Гапанюк Ю.Е.

2024 г.

Необходимо составить 3 запроса для предметной области, состоящей из двух классов: «Музыкальное произведение», «Оркестр».

Варианты запросов:

1. «Отдел» и «Сотрудник» связаны соотношением один-ко-многим. Выведите список всех связанных сотрудников и отделов, отсортированный по отделам, сортировка по сотрудникам произвольная.
2. «Отдел» и «Сотрудник» связаны соотношением один-ко-многим. Выведите список отделов с суммарной зарплатой сотрудников в каждом отделе, отсортированный по суммарной зарплате.
3. «Отдел» и «Сотрудник» связаны соотношением многие-ко-многим. Выведите список всех отделов, у которых в названии присутствует слово «отдел», и список работающих в них сотрудников.

Код программы:

```
class MusicalPiece:
    def __init__(self, id, title, composer, duration, genre):
        self.id = id
        self.title = title
        self.composer = composer
        self.duration = duration
        self.genre = genre
        self.orchestral_ids = [] # Список ID оркестров, исполняющих
произведение

class Orchestra:
    def __init__(self, id, name, location):
        self.id = id
        self.name = name
        self.location = location
        self.piece_ids = [] # Список ID музыкальных произведений, исполняемых
оркестром

class Performance:
    def __init__(self, piece_id, orchestra_id):
        self.piece_id = piece_id
        self.orchestra_id = orchestra_id

# Музыкальные произведения
pieces = [
    MusicalPiece(1, 'Симфония №5', 'Бетховен', 45, 'Классика'),
    MusicalPiece(2, 'Кармен', 'Бизе', 60, 'Опера'),
    MusicalPiece(3, 'Болеро', 'Равель', 15, 'Классика'),
    MusicalPiece(4, 'Времена года', 'Вивальди', 30, 'Классика')
]
```

```

# Оркестры
orchestras = [
    Orchestra(1, 'Филармонический оркестр', 'Москва'),
    Orchestra(2, 'Симфонический оркестр', 'Санкт-Петербург'),
    Orchestra(3, 'Камерный оркестр', 'Мюнхен')
]

# Выступления
performances = [
    Performance(1, 1),
    Performance(1, 2),
    Performance(2, 1),
    Performance(3, 3),
    Performance(4, 2)
]

# Добавим связи многие-ко-многим в объекты классов
for p in pieces:
    for perf in performances:
        if perf.piece_id == p.id:
            p.orchestral_ids.append(perf.orchestra_id)

for o in orchestras:
    for perf in performances:
        if perf.orchestra_id == o.id:
            o.piece_ids.append(perf.piece_id)

from operator import itemgetter

def main():

    # Задание A1: Вывод списка произведений и оркестров, отсортированный по оркестрам
    one_to_many = [(p.title, p.duration, o.name)
                    for o in orchestras
                    for p in pieces
                    if p.id in o.piece_ids]

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

    # Задание A2: Вывод списка оркестров с суммарной длительностью произведений
    res_12_unsorted = []
    for o in orchestras:
        # Список произведений оркестра
        o_pieces = list(filter(lambda i: i[2] == o.name, one_to_many))
        # Если оркестр не пустой
        if len(o_pieces) > 0:
            # Длительность произведений оркестра

```

```

        o_durations = [duration for _, duration, _ in o_pieces]
        res_12_unsorted.append((o.name, sum(o_durations)))

    res_12 = sorted(res_12_unsorted, key=itemgetter(1))
    print('\nЗадание A2')
    print(res_12)

    # Задание A3: Вывод списка оркестров с названием "оркестр" и их произведений
    many_to_many = [(p.title, p.duration, o.name)
                     for o in orchestras
                     for p in pieces
                     if p.id in o.piece_ids]

    res_13 = [(o.name, [p.title for p in pieces if p.id in o.piece_ids])
              for o in orchestras if 'оркестр' in o.name]

    print('\nЗадание A3')
    print(res_13)

if __name__ == "__main__":
    main()

```

Результат выполнения:

```

Задание A1
[('Болеро', 15, 'Камерный оркестр'), ('Симфония №5', 45, 'Симфонический оркестр'), ('Времена года', 30, 'Симфонический оркестр'), ('Симфония №5', 45, 'Филармонический оркестр')]

Задание A2
[('Камерный оркестр', 15), ('Симфонический оркестр', 75), ('Филармонический оркестр', 105)]

Задание A3
[('Филармонический оркестр', ['Симфония №5', 'Кармен']), ('Симфонический оркестр', ['Симфония №5', 'Времена года']), ('Камерный оркестр', ['Болеро'])]

```