

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

Факультет информатика и системы управления
Кафедра системы обработки информации и
управления

Курс «Парадигмы и конструкции языков программирования» Отчет по
рубежному контролю №1
Вариант Б18

Выполнил:
студент группы ИУ5-32Б:
Щепнов Ф.В.
Подпись и дата: 03.11.25

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

Москва, 2025 г.

Постановка задачи

В рамках рубежного контроля №1 по курсу «Парадигмы и конструкции языков программирования» требовалось разработать программу на языке Python.

Цель работы:

1. Создать два класса данных, соответствующие предметной области «Музыкальное произведение» (Класс 1) и «Оркестр» (Класс 2) (Вариант №18).
2. Реализовать связь «один-ко-многим» между классами (одно музыкальное произведение принадлежит одному оркестру, но один оркестр может исполнять несколько произведений).
3. Создать дополнительный класс для реализации связи «многие-ко-многим» (одно произведение может исполняться разными оркестрами, и один оркестр может исполнять разные произведения).
4. Создать списки объектов классов, содержащие тестовые данные (3-5 записей).
5. Реализовать три запроса в соответствии с Вариантом Б, адаптировав их под выбранную предметную область.
6. При реализации запросов использовать функциональные возможности Python (list/dict comprehensions, функции высших порядков).

Задачи (Вариант Б):

1. «Оркестр» и «Музыкальное произведение» связаны соотношением «один-ко-многим». Вывести список всех связанных произведений и оркестров, отсортированный по произведениям (по названию).
2. «Оркестр» и «Музыкальное произведение» связаны соотношением «один-ко-многим». Вывести список оркестров с количеством произведений в каждом, отсортированный по количеству произведений.
3. «Оркестр» и «Музыкальное произведение» связаны соотношением «многие-ко-многим». Вывести список всех произведений, у которых название заканчивается на «ов», и названия их оркестров.

Текст программы

```

        if o.id == co.orchestra_id]

    many_to_many = [(c.title, c.duration, orchestra_name)
                     for orchestra_name, orchestra_id, composition_id in
many_to_many_temp
                     for c in compositions if c.id == composition_id]

    print('Задание B1')

    res_1 = sorted(one_to_many, key=itemgetter(0))
    [print(f"{title} - {orchestra}") for title, duration, orchestra in res_1]

    print('\nЗадание B2')

    res_2_unsorted = []
    for o in orchestras:
        o_compositions = list(filter(lambda i: i[2] == o.name, one_to_many))
        o_compositions_count = len(o_compositions)
        res_2_unsorted.append((o.name, o_compositions_count))

    res_2 = sorted(res_2_unsorted, key=itemgetter(1), reverse=True)
    [print(f"{orchestra}: {count} произведений") for orchestra, count in res_2]

    print('\nЗадание B3')

    res_3 = []
    for title, duration, orchestra in many_to_many:
        if title.endswith('об'):
            res_3.append((title, orchestra))

    res_3 = list(set(res_3))
    [print(f"{title} - {orchestra}") for title, orchestra in res_3]

if __name__ == '__main__':
    main()

```

Скриншот работы программы

```

Run  main (2) ×
Задание B1
Концерт для скрипки - Камерный оркестр
Равсодия в стиле блюз - Джазовый оркестр
Симфониетта - Камерный оркестр
Симфония №5 - Симфонический оркестр
Симфония №9 - Симфонический оркестр
Увертюра 1812 - Симфонический оркестр

Задание B2
Симфонический оркестр: 3 произведений
Камерный оркестр: 2 произведений
Джазовый оркестр: 1 произведений
Народный оркестр: 0 произведений

Задание B3

Process finished with exit code 0

```