Московский государственный технический

университет им. Н.Э. Баумана

Факультет Радиотехнический
Кафедра “Системы обработки информации и управления”

Курс «Парадигмы и конструкции языков программирования»

Отчет по рубежному контролю №1

Выполнил:

студент группы РТ5-31Б:

Стукановский Максим Владимирович

Подпись и дата:

Проверил:

преподаватель каф. ИУ5

Гапанюк Юрий Евгеньевич

Подпись и дата:

Москва, 2025 г.

Код программы:

```
# вариант 17 Дирижер - Оркестр
# Вариант Г

class Conductor:
    def __init__(self, Id_Conductor, Surname, Payment, Id_Orchestra):
        self.Id_Conductor = Id_Conductor
        self.Surname = Surname
        self.Payment = Payment
        self.Id_Orchestra = Id_Orchestra


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


class Conductor_of_Orchestra:
    def __init__(self, Id_Conductor, Id_Orchestra):
        self.Id_Conductor = Id_Conductor
        self.Id_Orchestra = Id_Orchestra


# оркестры
orchestras = [
    Orchestra(1, "Академический симфонический оркестр"),
    Orchestra(2, "Струнный оркестр"),
    Orchestra(3, "Духовный оркестр"),
    Orchestra(4, "Народный окрестр"),

    Orchestra(5, "Большой симфонический оркестр"),
    Orchestra(6, "Гусарский оркестр"),
    Orchestra(7, "Молодежный оркестр"),
    Orchestra(8, "Каменный оркестр"),
]

# дирижеры
conductors = [
    Conductor(1, "Аверьянов", 300000, 1),
    Conductor(2, "Баранов", 200000, 2),
    Conductor(3, "Ставровский", 50000, 3),
    Conductor(4, "Филатов", 250000, 4),
    Conductor(5, "Лосев", 280000, 1),
    Conductor(6, "Булыга", 120000, 3),
    Conductor(7, "Корпачев", 175000, 3),
    Conductor(8, "Куцай", 235000, 2),
    Conductor(9, "Скидан", 276000, 4),
]

cond_orch = [
    Conductor_of_Orchestra(1, 1),
    Conductor_of_Orchestra(2, 2),
    Conductor_of_Orchestra(3, 3),
    Conductor_of_Orchestra(4, 4),
```

```
    Conductor_of_Orchestra(5, 1),
    Conductor_of_Orchestra(6, 3),
    Conductor_of_Orchestra(7, 3),
    Conductor_of_Orchestra(8, 2),
    Conductor_of_Orchestra(9, 4),

    Conductor_of_Orchestra(1, 5),
    Conductor_of_Orchestra(2, 6),
    Conductor_of_Orchestra(3, 7),
    Conductor_of_Orchestra(4, 8),
]

# 1. «Отдел» и «Сотрудник» связаны соотношением один-ко-многим.
# Выведите спиисок всех отделов, у которых название начинается с буквы «А», и список работающих в них сотрудников.
def task1():
    for o in orchestras:
        if o.Name[0].upper() == 'A':
            print(f"Оркестр: {o.Name}")
            surnames = [c.Surname for c in conductors if c.Id_Orchestra == o.Id_Orchestra]
            print(f"Дирижеры: {', '.join(surnames)}")

# 2. «Отдел» и «Сотрудник» связаны соотношением один-ко-многим.
# Выведите список отделов с максимальной зарплатой сотрудников в каждом отделе, отсортированный по максимальной зарплате.
def sort_by_payment(way):
    return sorted(conductors, key = lambda c: c.Payment, reverse=way)

def task2():
    results = {}
    for c in sort_by_payment(True):
        if c.Id_Orchestra not in results:
            orchestra_name = next(o.Name for o in orchestras if o.Id_Orchestra == c.Id_Orchestra)
            results[c.Id_Orchestra] = (orchestra_name, c.Payment)

    for name, payment in sorted(results.values(), key=lambda x: x[1], reverse=True):
        print(f"Оркестр: {name}, Макс. зарплата: {payment}")

# 3. «Отдел» и «Сотрудник» связаны соотношением многие-ко-многим.
# Выведите список всех связанных сотрудников и отделов, отсортированный по отделам, сортировка по сотрудникам произвольная.
def task3():
    connections = []
    for conn in cond_orch:
        orchestra = next(o for o in orchestras if o.Id_Orchestra == conn.Id_Orchestra)
        conductor = next(c for c in conductors if c.Id_Conductor == conn.Id_Conductor)
        connections.append((orchestra.Name, conductor.Surname, orchestra.Id_Orchestra))
    connections.sort(key=lambda x: x[2])

    for orchestra_name, surname, _ in connections:
        print(f"Оркестр: {orchestra_name}, Дирижер: {surname}")


def main():
    # один ко многим, задание Г1
    print("="*80)
    print("Задание Г1:")
    task1()
```

```
    print("="*80)

    # один ко многим, задание Г2
    print("Задание Г2:")
    task2()
    print("="*80)

    # много ко многим, задание Г3
    print("Задание Г3:")
    task3()
    print("="*80)

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

Результаты выполнения:

```
================================================================================
Задание Г1:
Оркестр: Академический симфонический оркестр
Дирижеры: Аверьянов, Лосев
================================================================================
Задание Г2:
Оркестр: Академический симфонический оркестр, Макс. зарплата: 300000
Оркестр: Народный окрестр, Макс. зарплата: 276000
Оркестр: Струнный оркестр, Макс. зарплата: 235000
Оркестр: Духовный оркестр, Макс. зарплата: 175000
================================================================================
Задание Г3:
Оркестр: Академический симфонический оркестр, Дирижер: Аверьянов
Оркестр: Академический симфонический оркестр, Дирижер: Лосев
Оркестр: Струнный оркестр, Дирижер: Баранов
Оркестр: Струнный оркестр, Дирижер: Куцай
Оркестр: Духовный оркестр, Дирижер: Ставровский
Оркестр: Духовный оркестр, Дирижер: Булыга
Оркестр: Духовный оркестр, Дирижер: Корпачев
Оркестр: Народный окрестр, Дирижер: Филатов
Оркестр: Народный окрестр, Дирижер: Скидан
Оркестр: Большой симфонический оркестр, Дирижер: Аверьянов
Оркестр: Гусарский оркестр, Дирижер: Баранов
Оркестр: Молодежный оркестр, Дирижер: Ставровский
Оркестр: Каменный оркестр, Дирижер: Филатов
================================================================================
```