Группа: ИУ5-31Б
Студент: Болотин Ю.А.
Вариант А5

Текст программы

```
 from operator import itemgetter

class Musician:
    """Музыкант"""
    def __init__(self, id, name):
        self.id = id
        self.name = name

class Orchestra:
    """Оркестр"""
    def __init__(self, id, name, salary, mus_id):
        self.id = id
        self.name = name
        self.salary = salary
        self.mus_id = mus_id

class MusOrc:
    """Связь многие-ко-многим"""
    def __init__(self, mus_id, orc_id):
        self.mus_id = mus_id
        self.orc_id = orc_id

# Данные
musicians = [
    Musician(1, "Моцарт"),
    Musician(2, "Шостакович"),
    Musician(3, "Глинка"),
]

orcs = [
    Orchestra(1, "Королевский симфонический оркестр", 50000, 1),
    Orchestra(2, "Гвардейский военный оркестр", 40000, 2),
    Orchestra(3, "Blue Note Band", 45000, 3),
    Orchestra(4, "Русская балалайка", 30000, 3),
```

```
    Orchestra(5, "Дыхание весны", 35000, 1),
    Orchestra(6, "Филармонический оркестр Санкт-Петербурга", 60000, 2),
]

mus_orc = [
    MusOrc(1, 1),
    MusOrc(2, 2),
    MusOrc(3, 3),
    MusOrc(1, 4),
    MusOrc(2, 5),
    MusOrc(3, 6),
]

def first_task(one_to_many):
    """A1. Список всех музыкантов и оркестров, отсортированный по оркестрам."""
    sorted_data = sorted(one_to_many, key=itemgetter(0))
    return sorted_data

def second_task(orcs, musicians):
    """A2. Список музыкантов с суммарной зарплатой оркестров, отсортированный по суммарной зарплате."""
    musician_salary = {}
    for musician in musicians:
        musician_orchestras = [orc.salary for orc in orcs if orc.mus_id == musician.id]
        total_salary = sum(musician_orchestras)
        musician_salary[musician.name] = total_salary
    result = sorted(musician_salary.items(), key=itemgetter(1), reverse=True)
    return result

def third_task(many_to_many):
    """A3. Список оркестров с 'оркестр' в названии и музыкантов в них."""
    filtered_data = [(orc_name, musician) for orc_name, _, musician in many_to_many if 'оркестр' in orc_name.lower()]
    return filtered_data

def main():
    # Связь один-ко-многим
    one_to_many = [(orc.name, orc.salary, musician.name)
```

```
            for musician in musicians
            for orc in orcs
            if orc.mus_id == musician.id]

    # Связь многие-ко-многим
    many_to_many_temp = [(musician.name, mo.mus_id, mo.orc_id)
                for musician in musicians
                for mo in mus_orc
                if musician.id == mo.mus_id]
    many_to_many = [(orc.name, orc.salary, musician_name)
                for musician_name, mus_id, orc_id in many_to_many_temp
                for orc in orcs if orc.id == orc_id]

    # Задание A1
    print("Задание A1")
    for record in first_task(one_to_many):
        print(record)

    # Задание A2
    print("\nЗадание A2")
    for musician, total_salary in second_task(orcs, musicians):
        print(f"{musician}: {total_salary}")

    # Задание A3
    print("\nЗадание A3")
    for orchestra, musician in third_task(many_to_many):
        print(f"{orchestra}: {musician}")

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

Вывод:

```
Задание A1
('Blue Note Band', 45000, 'Глинка')
('Гвардейский военный оркестр', 40000, 'Шостакович')
('Дыхание весны', 35000, 'Моцарт')
('Королевский симфонический оркестр', 50000, 'Моцарт')
('Русская балалайка', 30000, 'Глинка')
('Филармонический оркестр Санкт-Петербурга', 60000, 'Шостакович')

Задание A2
Шостакович: 100000
Моцарт: 85000
Глинка: 75000

Задание A3
Королевский симфонический оркестр: Моцарт
Гвардейский военный оркестр: Шостакович
Филармонический оркестр Санкт-Петербурга: Глинка

Process finished with exit code 0
```