

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

Факультет Информатика и системы управления
Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Основы информатики»

Отчет по рубежному контролю №1

Вариант №5

Запрос А.

Выполнил:

студент группы ИУ5-31Б:

Есакова О. А.

Подпись и дата:

Проверил:

преподаватель каф.ИУ5

Гапанюк Ю. Е.

Подпись и дата:

Москва, 2025 г.

Файл **classes.py**:

```
class Orchestra: #класс "оркестр"
    def __init__(self, id, name):
        self.id = id
        self.name = name

class Musician: #класс "музыкант"
    def __init__(self, id, name, salary, orchestra_id):
        self.id = id
        self.name = name
        self.salary = salary
        self.orchestra_id = orchestra_id

class MusiciansOrchestras: #для множественной связи м:м
    def __init__(self, orchestra_id, musician_id):
        self.orchestra_id = orchestra_id
        self.musician_id = musician_id
```

Файл **data.py**:

```
from classes import Orchestra, Musician, MusiciansOrchestras

orchestras = [
    #id оркестра, название оркестра
    Orchestra(1, 'симфонический оркестр'),
    Orchestra(2, 'джазовый оркестр'),
    Orchestra(3, 'камерный оркестр'),
    Orchestra(4, 'народных инструментов'),
    Orchestra(5, 'духовой оркестр'),]

musicians = [
    #id музыканта, имя, зарплата, id оркестра
    Musician(1, 'Алексей', 30000, 1),
    Musician(2, 'Николай', 35000, 2),
    Musician(3, 'Иван', 40000, 3),
    Musician(4, 'Анастасия', 27000, 1),
    Musician(5, 'Мария', 25000, 4),]

#список объектов класса MusiciansOrchestras со связью м:м
#один музыкант может играть в нескольких оркестрах, в одном оркестре
могут играть разные музыканты
musicians_orchestras = [
    #(оркестр-музыкант)
    MusiciansOrchestras(1, 1),
```

```
MusiciansOrchestras(1, 4),
MusiciansOrchestras(2, 2),
MusiciansOrchestras(2, 4),
MusiciansOrchestras(3, 3),
MusiciansOrchestras(4, 5),
MusiciansOrchestras(5, 5),]
```

Файл requests.py:

```
from operator import itemgetter
import data
```

```
def task_A1():
    # 1:м; список всех связанных музыкантов и оркестров, отсортированный
    по оркестрам, сортировка по музыкантам произвольная
    one_to_many = [(m.name, m.salary, o.name)
                    for o in data.orchestras
                    for m in data.musicians
                    if m.orchestra_id == o.id]
    sorted_result = sorted(one_to_many, key = itemgetter(2))
    print('Задание A1:')
    print(sorted_result)
    #альтернативный вывод:
    # for r in sorted_result:
        #print(r)
def task_A2():
    # 1:м; оркестры с суммарной зп музыкантов; список отсортирован по
    суммарной зарплате
    one_to_many = [(m.name, m.salary, o.name)
                    for o in data.orchestras
                    for m in data.musicians
                    if m.orchestra_id == o.id]
    res = []
    for o in data.orchestras:
        orch_musicians = list(filter(lambda i: i[2] == o.name, one_to_many))
        if len(orch_musicians)>0:
            total_salary = sum(m[1] for m in orch_musicians)
            res.append((o.name, total_salary))
    sorted_res = sorted(res, key = itemgetter(1), reverse = True)
    print('Задание A2:')
    print(sorted_res)
    # альтернативный вывод:
    # for r in sorted_res:
```

```

    #print(r)
def task_A3():
    print('Задание A3:')

    many_to_many_temp = [ #временный список. связывает оркестры и
музыкантов
        (o.name, mo.orchestra_id, mo.musician_id)
        for o in data.orchestras
        for mo in data.musicians_orchestras
        if o.id == mo.orchestra_id]

    many_to_many = [ #полный список, состоящий из имени музыканта,
зарплаты, названия оркестра
        (m.name, m.salary, o_name)
        for o_name, o_id, m_id in many_to_many_temp
        for m in data.musicians if m.id == m_id]

    filtered_orchestras = [name for name in set(d.name for d in data.orchestras) if
'оркестр' in name] #фильтруем
    res = { }
    for orchestra_name in filtered_orchestras:
        musicians_in_orchestra = list(filter(lambda i: i[2] == orchestra_name,
many_to_many))
        musician_names = [m[0] for m in musicians_in_orchestra]
        res[orchestra_name] = musician_names
    print(res)
    # альтернативный вывод:
    # for org_name, musicians_list in res.items():
    #     print(f"{org_name}: {musicians_list}")

```

Файл main.py:

```

from requests import task_A1, task_A2, task_A3

if __name__ == '__main__':
    task_A1(), print(), task_A2(), print(), task_A3()

```

Результаты выполнения программы:

Задание A1:

[('Николай', 35000, 'джазовый оркестр'), ('Иван', 40000, 'камерный оркестр'), ('Мария', 25000, 'народных инструментов'), ('Алексей', 30000, 'симфонический оркестр'), ('Анастасия', 27000, 'симфонический оркестр')]

Задание А2:

[('симфонический оркестр', 57000), ('камерный оркестр', 40000), ('джазовый оркестр', 35000), ('народных инструментов', 25000)]

Задание А3:

{'симфонический оркестр': ['Алексей', 'Анастасия'], 'камерный оркестр': ['Иван'], 'духовой оркестр': ['Мария'], 'джазовый оркестр': ['Николай', 'Анастасия']}

The screenshot shows an IDE with a Python file named `main.py`. The code defines three tasks: `task_A1`, `task_A2`, and `task_A3`. `task_A1` prints a list of tuples representing orchestra members. `task_A2` prints a list of tuples representing orchestra types and their average salaries. `task_A3` prints a dictionary mapping orchestra types to the names of their members.

```
1 from requests import task_A1, task_A2, task_A3
2
3 if __name__ == '__main__':
4     task_A1(), print(), task_A2(), print(), task_A3()
5
```

The Run window shows the execution output for three tasks:

Задание А1:
[('Николай', 35000, 'джазовый оркестр'), ('Иван', 40000, 'камерный оркестр'), ('Мария', 25000, 'народных инструментов'), ('Алексей', 30000, 'симфонический оркестр'), ('Анастасия', 27000, 'симфонический оркестр')]

Задание А2:
[(('симфонический оркестр', 57000), ('камерный оркестр', 40000), ('джазовый оркестр', 35000), ('народных инструментов', 25000))]

Задание А3:
{'симфонический оркестр': ['Алексей', 'Анастасия'], 'камерный оркестр': ['Иван'], 'духовой оркестр': ['Мария'], 'джазовый оркестр': ['Николай', 'Анастасия']}

Process finished with exit code 0