

**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 №2

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:

Файл main.py

```
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 ConductorOfOrchestra:
    def __init__(self, Id_Conductor, Id_Orchestra):
        self.Id_Conductor = Id_Conductor
        self.Id_Orchestra = Id_Orchestra
```

```
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"),
]
```

```
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 = [
    ConductorOfOrchestra(1, 1),
    ConductorOfOrchestra(2, 2),
    ConductorOfOrchestra(3, 3),
    ConductorOfOrchestra(4, 4),
    ConductorOfOrchestra(5, 1),
    ConductorOfOrchestra(6, 3),
    ConductorOfOrchestra(7, 3),
```

```

    ConductorOfOrchestra(8, 2),
    ConductorOfOrchestra(9, 4),
    ConductorOfOrchestra(1, 5),
    ConductorOfOrchestra(2, 6),
    ConductorOfOrchestra(3, 7),
    ConductorOfOrchestra(4, 8),
]

```

```

def get_orchestras_starting_with(letter):
    result = []
    for o in orchestras:
        if o.Name.startswith(letter):
            surnames = [c.Surname for c in conductors if c.Id_Orchestra == o.Id_Orchestra]
            result.append((o.Name, surnames))
    return result

def get_max_payment_by_orchestra():
    result = {}
    for c in conductors:
        if c.Id_Orchestra not in result:
            orchestra_name = next(o.Name for o in orchestras if o.Id_Orchestra == c.Id_Orchestra)
            result[c.Id_Orchestra] = (orchestra_name, c.Payment)
        else:
            result[c.Id_Orchestra] = (
                result[c.Id_Orchestra][0],
                max(result[c.Id_Orchestra][1], c.Payment)
            )
    return sorted(result.values(), key=lambda x: x[1], reverse=True)

def get_conductor_orchestra_links():
    links = []
    for co in cond_orch:
        orchestra = next(o.Name for o in orchestras if o.Id_Orchestra == co.Id_Orchestra)
        conductor = next(c.Surname for c in conductors if c.Id_Conductor == co.Id_Conductor)
        links.append((orchestra, conductor))
    return sorted(links, key=lambda x: x[0])

# def main():
#     print(get_orchestras_starting_with("A"))
#     print(get_max_payment_by_orchestra())
#     print(get_conductor_orchestra_links())

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

```

Файл test\_orchestra.py

```

from main import (
    get_orchestras_starting_with,
    get_max_payment_by_orchestra,
    get_conductor_orchestra_links
)

```

```

def test_orchestras_starting_with_A():
    result = get_orchestras_starting_with("A")
    assert len(result) == 1
    assert result[0][0] == "Академический симфонический оркестр"
    assert "Аверьянов" in result[0][1]
    assert "Лосев" in result[0][1]

def test_max_payment_sorted_desc():
    result = get_max_payment_by_orchestra()
    assert result[0][1] == 300000
    assert result[0][0] == "Академический симфонический оркестр"

def test_conductor_orchestra_links():
    result = get_conductor_orchestra_links()
    assert ("Большой симфонический оркестр", "Аверьянов") in result
    assert ("Каменный оркестр", "Филатов") in result

```

## Результат выполнения:

```

===== test session starts =====
platform darwin -- Python 3.9.6, pytest-8.4.2, pluggy-1.6.0 -- /Applications/Xcode.app/Contents/Developer/usr/bin/python3
cachedir: .pytest_cache
rootdir: /Users/macbook/pikyap_lab
collected 3 items

rk1 + rk2/test_orchestra.py::test_orchestras_starting_with_A PASSED [ 33%]
rk1 + rk2/test_orchestra.py::test_max_payment_sorted_desc PASSED [ 66%]
rk1 + rk2/test_orchestra.py::test_conductor_orchestra_links PASSED [100%]

===== 3 passed in 0.03s =====

```