

## РК 1

### Текст программы:

```
from operator import itemgetter

class MusicalPiece:
    """Музыкальное произведение"""

    def __init__(self, piece_id, title, composer, orchestra_id):
        self.piece_id = piece_id
        self.title = title
        self.composer = composer
        self.orchestra_id = orchestra_id

class Orchestra:
    """Оркестр"""

    def __init__(self, orchestra_id, name):
        self.orchestra_id = orchestra_id
        self.name = name

class Performers:
    """Исполнители произведения"""

    def __init__(self, piece_id, orchestra_id, performers_count):
        self.piece_id = piece_id
        self.orchestra_id = orchestra_id
        self.performers_count = performers_count

# Оркестры
orchestras = [
    Orchestra(1, "Филармонический оркестр"),
    Orchestra(2, "Государственный оркестр"),
    Orchestra(3, "Молодежный оркестр"),
    Orchestra(4, "Главный оркестр Москвы"),
]

# Музыкальные произведения
musical_pieces = [
    MusicalPiece(1, "Симфония №5", "Бетховен", 1),
    MusicalPiece(2, "Концерт для фортепиано", "Григ", 1),
    MusicalPiece(3, "Симфония №9", "Дворжак", 2),
    MusicalPiece(4, "Рапсодия на тему Паганини", "Рахманинов", 1),
    MusicalPiece(5, "Импровизация", "Шостакович", 3),
    MusicalPiece(6, "Императорский марш для оркестра", "Рихард Вагнер", 4),
]

# Исполнители
performers = [
    Performers(1, 1, 25),
```

```

Performers(2, 1, 20),
Performers(3, 2, 30),
Performers(4, 1, 22),
Performers(5, 3, 15),
Performers(6, 4, 45),
]

```

```
def main():
```

```
    """Основная функция для обработки данных"""
```

```
    # Запрос 1: Список всех музыкальных произведений с заглавием, заканчивающимся на 'а', и
    # названиями их оркестров.
```

```
    pieces_with_a = [
        (piece.title, next((orch.name for orch in orchestras if orch.orchestra_id == piece.orchestra_id),
        None))
        for piece in musical_pieces if piece.title.endswith('а')
    ]
    print('Задание 1:', sorted(pieces_with_a, key=itemgetter(1)))
```

```
    # Запрос 2: Список оркестров со средней численностью исполнителей.
```

```
    orchestra_performers_count = {}
```

```
    for performer in performers:
```

```
        if performer.orchestra_id not in orchestra_performers_count:
```

```
            orchestra_performers_count[performer.orchestra_id] = []
```

```
            orchestra_performers_count[performer.orchestra_id].append(performer.performers_count)
```

```
    averages = {
```

```
        orch.name: sum(counts) / len(counts)
```

```
        for orch in orchestras
```

```
        if (counts := orchestra_performers_count.get(orch.orchestra_id)) is not None
```

```
    }
```

```
    print('Задание 2:', sorted(averages.items(), key=itemgetter(1), reverse=True))
```

```
    # Запрос 3: Все оркестры, название которых начинается с "Г", и список исполняемых
    # музыкальных произведений.
```

```
    orchestras_starting_with_G = {
```

```
        orch.name: [piece.title for piece in musical_pieces if piece.orchestra_id == orch.orchestra_id]
```

```
        for orch in orchestras if orch.name.startswith("Г")
```

```
    }
```

```
    print('Задание 3:', orchestras_starting_with_G)
```

```
if __name__ == '__main__':
```

```
    main()
```

## Результат выполнения программы:

```

C:\Users\DNS\PycharmProjects\pythonProject2\venv\Scripts\python.exe C:\Users\DNS\PycharmProjects\pythonProject2\main.py
Задание 1: [('Императорский марш для оркестра', 'Главный оркестр Москвы')]
Задание 2: [('Главный оркестр Москвы', 45.0), ('Государственный оркестр', 30.0), ('Филармонический оркестр', 22.333333333333332), ('Молодежный оркестр', 15.0)]
Задание 3: {'Государственный оркестр': ['Симфония №9'], 'Главный оркестр Москвы': ['Императорский марш для оркестра']}

```