

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

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

Курс «Базовые компоненты интернет-технологий»

Отчет по рубежному контролю №1 по курсу БКИТ

Выполнил:

студент группы ИУ5-34Б
Жданова Яна

Подпись и дата:

Проверил:

преподаватель каф. ИУ5
Гапанюк Ю.Е.

Подпись и дата:

Задание:

- 1) Необходимо создать два класса данных в соответствии с Вашим вариантом предметной области, которые связаны отношениями один-ко-многим и многие-ко-многим.
- 2) Необходимо создать списки объектов классов, содержащих тестовые данные (3-5 записей), таким образом, чтобы первичные и вторичные ключи соответствующих записей были связаны по идентификаторам.
- 3) Необходимо разработать запросы в соответствии с Вашим вариантом.

Вариант Г.

1. «Оркестр» и «Музыкант» связаны соотношением один-ко-многим. Выведите список всех Оркестров, у которых название начинается с буквы «А», и список работающих в них Музыкантов.
2. «Оркестр» и «Музыкант» связаны соотношением один-ко-многим. Выведите список Оркестров с максимальной зарплатой Музыкантов в каждом Оркестре, отсортированный по максимальной зарплате.
3. «Оркестр» и «Музыкант» связаны соотношением многие-ко-многим. Выведите список всех связанных Музыкантов и Оркестров, отсортированный по Оркестрам, сортировка по Музыкантам произвольная.

Текст программы:

```
from operator import itemgetter

class Musician:
    """Музыкант"""
    def __init__(self, id, surname, salary, orch_id):
        self.id = id
        self.surname = surname
        self.salary = salary
        self.orch_id = orch_id

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

class MusOrch:
    """
    'Музыканты оркестра' для реализации
    связи многие-ко-многим
    """
    def __init__(self, mus_id, orch_id):
        self.mus_id = mus_id
        self.orch_id = orch_id

# Оркестры
orchestras = [
    Orchestra(1, 'Дуделки и свистелки'),
    Orchestra(2, 'Стучалки и бренчалки'),
    Orchestra(3, 'Баян или плохая шутка'),
    Orchestra(4, 'Ансамбль песнопелки и плясалки')
]

# Музыканты
```

```

musicians = [
    Musician(1, 'Соколов', 500000, 2),
    Musician(2, 'Жданова', 250000, 4),
    Musician(3, 'Тумановский', 300000, 3),
    Musician(4, 'Савицкая', 150000, 4),
    Musician(5, 'Теряева', 4050, 1)
]

mus_orches = [
    MusOrch(1, 1),
    MusOrch(2, 2),
    MusOrch(3, 3),
    MusOrch(4, 4),
    MusOrch(5, 1),

    MusOrch(1, 4),
    MusOrch(2, 3),
    MusOrch(3, 2),
    MusOrch(4, 2),
    MusOrch(5, 4),
]

def main():
    """Основная функция"""

    # Соединение данных один-ко-многим
    one_to_many = [(mus.surname, mus.salary, orch.name)
                    for orch in orchestras
                    for mus in musicians
                    if mus.orch_id == orch.id]

    # Соединение данных многие-ко-многим
    many_to_many_temp = [(orch.name, mu_or.orch_id, mu_or.mus_id)
                          for orch in orchestras
                          for mu_or in mus_orches
                          if orch.id == mu_or.orch_id]

    many_to_many = [(mus.surname, mus.salary, orch_name)
                    for orch_name, orch_id, mus_id in many_to_many_temp
                    for mus in musicians if mus.id == mus_id]

    print ('Задание Г1')
    res_1 = {}
    for orch in orchestras:
        if orch.name[0] == 'А':
            orch_muses = list(filter(lambda i: i[2] == orch.name,
one_to_many))
            orch_muses_names = [x for x, _, _ in orch_muses]
            res_1[orch.name] = orch_muses_names
    print (res_1)

    print ('Задание Г2')
    res_2 = []
    for orch in orchestras:
        orch_muses = list(filter(lambda i: i[2] == orch.name, one_to_many))
        if len(orch_muses) > 0:
            orch_salaries = [sal for _, sal, _ in orch_muses]
            orch_salaries_max = max(orch_salaries)
            res_2.append((orch.name, orch_salaries_max))
    res_2_sort = sorted(res_2, key=itemgetter(1))
    for i in res_2_sort:
        print(i)

```

```

print ('Задание Г3')
res_3_sort = sorted (many_to_many, key=itemgetter(2))
for i in res_3_sort:
    print(i)

if __name__ == '__main__':
    main()

```

Результат выполнения программы:

```

main x
"C:\Users\user\Desktop\PK1 БКИТ\venv\Scripts\python.exe" "C:\Users\user\Desktop\PK1 БКИТ\main.py"
Задание Г1
{'Ансамбль песнопелки и плясалки': ['Жданова', 'Савицкая']}
Задание Г2
('Дуделки и свистелки', 4050)
('Ансамбль песнопелки и плясалки', 250000)
('Баян или плохая шутка', 300000)
('Стучалки и бренчалки', 500000)
Задание Г3
('Савицкая', 150000, 'Ансамбль песнопелки и плясалки')
('Соколов', 500000, 'Ансамбль песнопелки и плясалки')
('Теряева', 4050, 'Ансамбль песнопелки и плясалки')
('Тумановский', 300000, 'Баян или плохая шутка')
('Жданова', 250000, 'Баян или плохая шутка')
('Соколов', 500000, 'Дуделки и свистелки')
('Теряева', 4050, 'Дуделки и свистелки')
('Жданова', 250000, 'Стучалки и бренчалки')
('Тумановский', 300000, 'Стучалки и бренчалки')
('Савицкая', 150000, 'Стучалки и бренчалки')

Process finished with exit code 0

```