

Отчет по РК № 1 по курсу
"Разработка Интернет-Приложений"

Выполнила:

Студентка группы

ИУ5-55Б

Зубарева А. М.

Задание:

1. «Музыкант» и «Оркестр» связаны соотношением один-ко-многим. Выведите список всех Музыкантов, у которых фамилия заканчивается на «ов», и названия их Оркестров.
2. «Музыкант» и «Оркестр» связаны соотношением один-ко-многим. Выведите список Оркестр со средней зарплатой Музыкантов в каждом Оркестре, отсортированный по средней зарплате.
3. «Музыкант» и «Оркестр» связаны соотношением многие-ко-многим. Выведите список всех Оркестров, у которых название начинается с буквы «L», и список работающих в них Музыкантов.

№ варианта	Класс 1	Класс 2
5	Музыкант	Оркестр

Текст программы:

```
# используется для сортировки
from operator import itemgetter

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 Orchestra:
    """ Класс "Оркестр" """

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

class MusicianOrch:
    """
    класс "Музыканты оркестра" для реализации
    связи многие-ко-многим
    """

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

# Оркестры
orchestras = [
    Orchestra(1, 'London Symphony Orchestra'),
    Orchestra(2, 'Berlin Philharmonic Orchestra'),
    Orchestra(3, 'Leipzig Gewandhaus Orchestra'),
]
```

```

# Музыканты
musicians = [
    Musician(1, 'Антипова', 25000, 1),
    Musician(2, 'Иванов', 35000, 2),
    Musician(3, 'Фёдорова', 45000, 3),
    Musician(4, 'Сидоров', 35000, 1),
    Musician(5, 'Покровский', 25000, 2),
    Musician(6, 'Бахман', 51000, 3),
    Musician(7, 'Кожиев', 25000, 1),
    Musician(8, 'Уралова', 25000, 2),
    Musician(9, 'Богданова', 45000, 3),
    Musician(10, 'Смыслов', 45000, 1),
    Musician(11, 'Хижняков', 55000, 3),
    Musician(12, 'Зубарева', 25000, 2),
]

musician_orch = [
    MusicianOrch(1, 1),
    MusicianOrch(2, 2),
    MusicianOrch(3, 3),
    MusicianOrch(3, 2),
    MusicianOrch(3, 1),
    MusicianOrch(4, 1),
    MusicianOrch(5, 2),
    MusicianOrch(6, 1),
    MusicianOrch(7, 2),
    MusicianOrch(8, 3),
]

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

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

    # Соединение данных многие-ко-многим
    many_to_many_temp = [(orch.name, or_m.musician_id, or_m.orchestra_id)
                          for orch in orchestras
                          for or_m in musician_orch
                          if orch.id == or_m.orchestra_id]

    many_to_many = [(m.name, m.salary, orch_name)
                     for orch_name, music_id, orch_id in many_to_many_temp
                     for m in musicians if m.id == music_id]

    print('Задание Д1')
    res_1 = [(d.name, d.salary, p.name)
              for p in orchestras
              for d in musicians
              if d.orchestra_id == p.id and d.name.endswith('ов')]
    for item in res_1:
        print("Музыкант: {}, Оркестр: {}".format(item[0], item[2]))

    print('\nЗадание Д2')
    res_2_unsorted = []
    # Перебираем все оркестры
    for orch in orchestras:
        # Список музыкантов оркестра
        orch_musics = list(filter(lambda i: i[2] == orch.name, one_to_many))

```

```

# Если оркестр не пустой
if len(orch_musics) > 0:
    # Зарплаты музыкантов оркестра
    m_sals = [sal for _, sal, _ in orch_musics]
    # Средняя зарплата музыкантов оркестра
    average_sal = sum(m_sals) / len(m_sals)
    res_2_unsorted.append((orch.name, average_sal))

# Сортировка по средней зарплате
res_2 = sorted(res_2_unsorted, key=itemgetter(1), reverse=True)
for item in res_2:
    print("Оркестр: {}, средняя зарплата: {}".format(item[0], item[1]))

print('\nЗадание Д3')
res_3 = {}
# Перебираем все оркестры
for orch in orchestras:
    if orch.name.lower().startswith('l'):
        # Список музыкантов оркестра
        musics_of_orch = list(filter(lambda i: i[2] == orch.name,
many_to_many))
        # Только имена музыкантов
        music_names = [x for x, _, _ in musics_of_orch]
        # Добавляем результат в словарь
        # ключ - оркестр, значение - список музыкантов
        res_3[orch.name] = music_names

for item in res_3:
    print("Оркестр: {}".format(item))
    print("Музыканты:")
    for val in res_3[item]:
        print(val)

if __name__ == '__main__':
    main()

```

Результат работы:

Задание Д1

Музыкант: Сидоров, Оркестр: London Symphony Orchestra

Музыкант: Смыслов, Оркестр: London Symphony Orchestra

Музыкант: Иванов, Оркестр: Berlin Philharmonic Orchestra

Музыкант: Хижняков, Оркестр: Leipzig Gewandhaus Orchestra

Задание Д2

Оркестр: Leipzig Gewandhaus Orchestra, средняя зарплата: 49000.0

Оркестр: London Symphony Orchestra, средняя зарплата: 32500.0

Оркестр: Berlin Philharmonic Orchestra, средняя зарплата: 27500.0

Задание Д3

Оркестр: London Symphony Orchestra

Музыканты:

Антипова

Фёдорова

Сидоров

Бахман

Оркестр: Leipzig Gewandhaus Orchestra

Музыканты:

Фёдорова

Уралова