

Московский государственный технический университет им. Н.Э. Баумана

Факультет «Информатика и системы управления»

Кафедра «Системы обработки информации и управления»



Домашнее задание по дисциплине “Сети и телекоммуникации”

Студент группы ИУ5 -53 _____ Бахрамов Н.А.

Преподаватель _____ Ревунков Г.И.

Москва 2021

Задача

Вариант 5:

класс1: Музыкант

класс2: Оркестр

Запросы:

1. «Оркестр» и «Музыкант» связаны соотношением один-ко-многим. Выведите список всех музыкантов, у которых фамилия начинается с буквы «А», и названия их оркестров.
2. «Оркестр» и «Музыкант» связаны соотношением один-ко-многим. Выведите список оркестров с минимальным возрастом музыкантов в каждом оркестре, отсортированный по минимальному возрасту.
3. «Оркестр» и «Музыкант» связаны соотношением многие-ко-многим. Выведите список всех связанных музыкантов и оркестров, отсортированный по музыкантам, сортировка по оркестрам произвольная.

Текст программы

```
from operator import itemgetter

class Mus:
    """Музыкант"""

    def __init__(self, id, fio, age, ork_id):
        self.id = id
        self.fio = fio
        self.age = age
        self.ork_id = ork_id

class Ork:
    """Оркестр"""

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

class MusOrk:
    """
    'Сотрудники отдела' для реализации
    связи многие-ко-многим
    """

    def __init__(self, ork_id, mus_id):
        self.ork_id = ork_id
        self.mus_id = mus_id
```

```

# Отделы
Orks = [
    Ork(1, 'Чайковского'),
    Ork(2, 'Бетховена'),
    Ork(3, 'По кайфу'),
]

# Сотрудники
mus_d = [
    Mus(1, 'Артамонов', 25, 1),
    Mus(2, 'Петров', 23, 1),
    Mus(3, 'Иваненко', 40, 2),
    Mus(4, 'Иванов', 35, 2),
    Mus(5, 'Иванин', 25, 3),
]

musokr_link = [
    MusOrk(1, 1),
    MusOrk(2, 2),
    MusOrk(3, 3),
    MusOrk(3, 4),
    MusOrk(3, 5),
]

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

    # Соединение данных один-ко-многим
    one_to_many = [(e.fio, e.age, d.name)
                    for d in Orks
                    for e in mus_d
                    if e.ork_id == d.id]

    # Соединение данных многие-ко-многим
    many_to_many_temp = [(d.name, ed.ork_id, ed.mus_id)
                           for d in Orks
                           for ed in musokr_link
                           if d.id == ed.ork_id]

    many_to_many = [(e.fio, e.age, ork_name)
                     for ork_name, ork_id, mus_id in many_to_many_temp
                     for e in mus_d if e.id == mus_id]

    print('Задание B1')
    res_11 = {}
    for d in Orks:
        d_mus = list(filter(lambda i: i[2] == d.name, one_to_many))
        if len(d_mus) > 0:
            d_mus_names = [x for x, _, _ in d_mus]
            d_mus_names_a = []
            for el in d_mus_names:
                if el[0] == 'A':
                    d_mus_names_a.append(el)
            res_11[d.name] = d_mus_names_a
    print(res_11)

    print('\nЗадание B2')
    res_12_unsorted = []
    for d in Orks:
        d_mus = list(filter(lambda i: i[2] == d.name, one_to_many))
        if len(d_mus) > 0:
            d_mus_age = [age for _, age, _ in d_mus]
            d_mus_age_min = min(d_mus_age)

```

```

        res_12_unsorted.append((d.name, d_mus_age_min))

res_12 = sorted(res_12_unsorted, key=itemgetter(1), reverse=True)
print(res_12)

print('\nЗадание В3')
res_13 = sorted(many_to_many, key=itemgetter(2));
print(res_13);

if __name__ == '__main__':
    main()

```

Результаты работы программы

Задание В1

```
{'Чайковского': ['Артамонов'], 'Бетховена': [], 'По кайфу': []}
```

Задание В2

```
[('Бетховена', 35), ('По кайфу', 25), ('Чайковского', 23)]
```

Задание В3

```
[('Петров', 23, 'Бетховена'), ('Иваненко', 40, 'По кайфу'), ('Иванов', 35, 'По кайфу'), ('Иванин', 25, 'По кайфу'), ('Артамонов', 25, 'Чайковского')]
```