

## Листинг кода программы:

```
from operator import itemgetter

class House:
    """Дом"""
    def __init__(self, id, number, floors, street_id):
        self.id = id
        self.number = number # номер дома
        self.floors = floors # количество этажей (количественный
признак)
        self.street_id = street_id

class Street:
    """Улица"""
    def __init__(self, id, name):
        self.id = id
        self.name = name

class HouseStreet:
    """
    'Дома на улицах' для реализации
связи многие-ко-многим
    """
    def __init__(self, street_id, house_id):
        self.street_id = street_id
        self.house_id = house_id

# Улицы
streets = [
    Street(1, 'Арбат'),
    Street(2, 'Ленинский проспект'),
    Street(3, 'Тверская'),
    Street(4, 'Авиамоторная'),
    Street(5, 'Профсоюзная'),
]

# Дома
houses = [
    House(1, '15', 5, 1),
    House(2, '25', 9, 2),
    House(3, '10', 12, 3),
    House(4, '8', 3, 4),
    House(5, '42', 16, 2),
    House(6, '7', 7, 3),
    House(7, '3', 2, 4),
]

houses_streets = [
    HouseStreet(1, 1),
    HouseStreet(2, 2),
    HouseStreet(2, 5),
    HouseStreet(3, 3),
    HouseStreet(3, 6),
```

```

        HouseStreet(4, 4),
        HouseStreet(4, 7),
        HouseStreet(5, 1),
        HouseStreet(5, 3),
    ]

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

    # Соединение данных один-ко-многим
    one_to_many = [(h.number, h.floors, s.name)
                   for s in streets
                   for h in houses
                   if h.street_id == s.id]

    # Соединение данных многие-ко-многим
    many_to_many_temp = [(s.name, hs.street_id, hs.house_id)
                          for s in streets
                          for hs in houses_streets
                          if s.id == hs.street_id]

    many_to_many = [(h.number, h.floors, street_name)
                    for street_name, street_id, house_id in many_to_many_temp
                    for h in houses if h.id == house_id]

    print('Задание Г1')
    # Список всех улиц, начинающихся на "А", и домов на них
    res_1 = []
    for s in streets:
        if s.name.startswith('A'):
            # Список домов на этой улице
            s_houses = list(filter(lambda i: i[2] == s.name,
                                   one_to_many))
            if len(s_houses) > 0:
                res_1.append((s.name, [house_num for house_num, _, _
                                      in s_houses]))

    print(res_1)

    print('\nЗадание Г2')
    # Список улиц с максимальной этажностью домов на каждой улице
    res_2_unsorted = []
    for s in streets:
        # Список домов на улице
        s_houses = list(filter(lambda i: i[2] == s.name, one_to_many))
        if len(s_houses) > 0:
            # Этажность домов на улице
            s_floors = [floors for _, floors, _ in s_houses]
            # Максимальная этажность
            max_floors = max(s_floors)
            res_2_unsorted.append((s.name, max_floors))

    # Сортировка по максимальной этажности (по убыванию)
    res_2 = sorted(res_2_unsorted, key=itemgetter(1), reverse=True)

```

```

print(res_2)

print('\nЗадание Г3')
# Список всех связанных домов и улиц, отсортированный по улицам
res_3 = sorted(many_to_many, key=itemgetter(2))
print(res_3)

if __name__ == '__main__':
    main()

```

## **Результаты выполнения программы:**

Задание Г1

[('Арбат', [15]), ('Авиамоторная', [8, 3])]

Задание Г2

[('Ленинский проспект', 16), ('Тверская', 12), ('Авиамоторная', 3), ('Арбат', 5), ('Профсоюзная', 12)]

Задание Г3

[('15', 5, 'Арбат'), ('8', 3, 'Авиамоторная'), ('3', 2, 'Авиамоторная'), ('25', 9, 'Ленинский проспект'),
 ('42', 16, 'Ленинский проспект'), ('10', 12, 'Тверская'), ('7', 7, 'Тверская'), ('15', 5, 'Профсоюзная'), ('10',
 12, 'Профсоюзная')]

## **Пояснения к решению:**

### **1. Классы данных:**

- House (Дом) - содержит ID, номер дома, количество этажей (количественный признак) и ID улицы
- Street (Улица) - содержит ID и название улицы
- HouseStreet - для связи многие-ко-многим

### **2. Тестовые данные:**

- 5 улиц и 7 домов
- Связи организованы корректно через идентификаторы

### **3. Запросы варианта Г:**

- **Г1:** Найдены все улицы, начинающиеся на "А" (Арбат, Авиамоторная) и список домов на них
- **Г2:** Для каждой улицы найдена максимальная этажность домов, результат отсортирован по убыванию этажности
- **Г3:** Выведены все связи домов и улиц (многие-ко-многим), отсортированные по названиям улиц