



**Министерство науки и высшего образования
Российской Федерации Федеральное государственное
бюджетное образовательное учреждение высшего
образования**

**«Московский государственный технический
университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)**

**Факультет «Информатика, искусственный интеллект и
системы управления»**

**Кафедра «Системы обработки информации и
управления»**

**Рубежный контроль №1 по курсу
«Базовые компоненты интернет-технологий»**

**Выполнил:
студент группы ИУ5-33Б
Иванченко Максим**

**Проверил:
Доцент кафедры ИУ5
Гапанюк Юрий Евгеньевич**

2022 г.

Текст программы

```
from operator import itemgetter
```

```
class House:
```

```
    """Дом"""
```

```
    def __init__(self, id, house_num, building_num, resident_cnt, street_id):
```

```
        self.id = id
```

```
        self.house_num = house_num
```

```
        self.building_num = building_num
```

```
        self.resident_cnt = resident_cnt
```

```
        self.street_id = street_id
```

```
class Street:
```

```
    """Улица"""
```

```
    def __init__(self, id, name):
```

```
        self.id = id
```

```
        self.name = name
```

```
class HouseStreet:
```

```
    """Дома на улице"""
```

```
    def __init__(self, street_id, house_id):
```

```
        self.street_id = street_id
```

```
        self.house_id = house_id
```

```
# Улицы
```

```
streets = [
```

```
    Street(1, 'Веселая'),
```

```
    Street(2, 'Замечательная'),
```

```
    Street(3, 'Великолепная'),
```

```
    # Для связи М-М
```

```
    Street(11, 'Потрясная (старое название)'),
```

```
    Street(22, 'Неповторимая (старое название)'),
```

```
    Street(33, 'Отличная (старое название)')
```

```
]
```

```
# Дома
```

```
houses = [
```

```
    House(1, 5, 2, 100, 1),
```

```
    House(2, 2, 1, 125, 2),
```

```
    House(3, 10, 3, 232, 2),
```

```
    House(4, 7, 1, 260, 3),
```

```
    House(5, 4, 1, 170, 3),
```

```
    House(6, 3, 4, 200, 3)
```

```
]
```

```
# Дома на улицах (связь М-М)
```

```
houses_streets = [
```

```
    HouseStreet(1, 1),
```

```
    HouseStreet(2, 2),
```

```
    HouseStreet(2, 3),
```

```
    HouseStreet(3, 4),
```

```
    HouseStreet(3, 5),
```

```
    HouseStreet(3, 6),
```

```
    HouseStreet(11, 1),
```

```
    HouseStreet(22, 2),
```

```
    HouseStreet(22, 3),
```

```
    HouseStreet(33, 4),
```

```
    HouseStreet(33, 5),
```

```
    HouseStreet(33, 6)
```

```
]
```

```
def main():
    """Функция main()"""
    # Связь 1-М
    one_to_many = [
        (h.house_num, h.building_num, h.resident_cnt, s.name)
        for h in houses
        for s in streets
        if h.street_id == s.id
    ]

    # Связь М-М
    many_to_many = [
        (h.house_num, h.building_num, h.resident_cnt, [s.name for s in streets
if s.id == h_s.street_id][0])
        for h in houses
        for h_s in houses_streets
        if h.id == h_s.house_id
    ]

    # Задание 1: список домов, номер которых больше 3, и название их улиц
    print('Задание №1')
    result_1 = list(filter(lambda x: x[0] > 3, one_to_many))
    print(result_1)
    for h_n, b_n, r_c, s_n in result_1:
        print(f'Дом {h_n} строение {b_n}, {r_c} жителей - улица {s_n}')

    # Задание 2: отсортированный список улиц по минимальному числу жителей в
    доме
    print('\nЗадание №2')
    streets_min_house_number = []
    for s in streets:
        street_house_numbers = [(s_n, r_c) for h_n, b_n, r_c, s_n in one_to_many
if s.name == s_n]
        # Если на улице есть дома
        if len(street_house_numbers) > 0:
            streets_min_house_number.append(min(street_house_numbers))
    result_2 = sorted(streets_min_house_number, key=itemgetter(1), reverse=True)
    print(result_2)
    for s_name, r_cnt in result_2:
        print(f'На улице {s_name} - в доме минимум {r_cnt} жителей')

    # Задание 3: список связанных домов и улиц, отсортированный по номерам домов
    print('\nЗадание №3')
    result_3 = sorted(many_to_many, key=itemgetter(0, 1))
    print(result_3)
    for h_num, b_num, r_cnt, s_name in result_3:
        print(f'Дом {h_num} строение {b_num}, {r_cnt} жителей - улица {s_name}')

if __name__ == '__main__':
    main()
```

Результаты

Задание В1

[(5, 2, 100, 'Веселая'), (10, 3, 232, 'Замечательная'), (7, 1, 260, 'Великолепная'), (4, 1, 170, 'Великолепная')]

Дом 5 строение 2, 100 жителей - улица Веселая

Дом 10 строение 3, 232 жителей - улица Замечательная

Дом 7 строение 1, 260 жителей - улица Великолепная

Дом 4 строение 1, 170 жителей - улица Великолепная

Задание В2

[('Великолепная', 170), ('Замечательная', 125), ('Веселая', 100)]

На улице Великолепная - в доме минимум 170 жителей

На улице Замечательная - в доме минимум 125 жителей

На улице Веселая - в доме минимум 100 жителей

Задание В3

[(2, 1, 125, 'Замечательная'), (2, 1, 125, 'Неповторимая (старое название)'), (3, 4, 200, 'Великолепная'), (3, 4, 200, 'Отличная (старое название)'), (4, 1, 170, 'Великолепная'), (4, 1, 170, 'Отличная (старое название)'), (5, 2, 100, 'Веселая'), (5, 2, 100, 'Потрясная (старое название)'), (7, 1, 260, 'Великолепная'), (7, 1, 260, 'Отличная (старое название)'), (10, 3, 232, 'Замечательная'), (10, 3, 232, 'Неповторимая (старое название)')]

Дом 2 строение 1, 125 жителей - улица Замечательная

Дом 2 строение 1, 125 жителей - улица Неповторимая (старое название)

Дом 3 строение 4, 200 жителей - улица Великолепная

Дом 3 строение 4, 200 жителей - улица Отличная (старое название)

Дом 4 строение 1, 170 жителей - улица Великолепная

Дом 4 строение 1, 170 жителей - улица Отличная (старое название)

Дом 5 строение 2, 100 жителей - улица Веселая

Дом 5 строение 2, 100 жителей - улица Потрясная (старое название)

Дом 7 строение 1, 260 жителей - улица Великолепная

Дом 7 строение 1, 260 жителей - улица Отличная (старое название)

Дом 10 строение 3, 232 жителей - улица Замечательная

Дом 10 строение 3, 232 жителей - улица Неповторимая (старое название)