



**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 №2 по курсу
«Базовые компоненты интернет-технологий»**

Выполнил:
студент группы ИУ5-33Б
Иванченко Максим

Проверил:
Доцент кафедры ИУ5
Гапанюк Юрий Евгеньевич

2022 г.

Текст программы

Файл classes.py (классы)

```
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
```

Файл data_connections.py (данные и связи)

```
from classes import *
# 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 (для связи 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)
    ]

# 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
]

```

Файл task_1.py (задание №1)

```

# one_to_many - СВЯЗЬ 1-М
def house_num_more_3(one_to_many):
    return list(filter(lambda x: x[0] > 3, one_to_many))

```

Файл task_2.py (задание №2)

```

from operator import itemgetter
from data_connections import streets

# one_to_many - СВЯЗЬ 1-М
def streets_with_min_resident_cnt(one_to_many):
    streets_min_resident_cnt = []
    for s in streets:
        street_resident_cnt = [(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_resident_cnt) > 0:
            streets_min_resident_cnt.append(min(street_resident_cnt))
    return sorted(streets_min_resident_cnt, key=itemgetter(1), reverse=True)

```

Файл task_3.py (задание №3)

```

from operator import itemgetter

# many_to_many - СВЯЗЬ М-М
def data_sorted_by_house_num(many_to_many):
    return sorted(many_to_many, key=itemgetter(0, 1))

```

Файл test.py (тестирование)

```

import unittest
from data_connections import *
from task_1 import house_num_more_3
from task_2 import streets_with_min_resident_cnt
from task_3 import data_sorted_by_house_num

class MyTestCase(unittest.TestCase):

```

```

def test_task_1(self):
    res = [(5, 2, 100, 'Веселая'), (10, 3, 232, 'Замечательная'), (7, 1,
260, 'Великолепная'), (4, 1, 170,
'Великолепная')]
    self.assertEqual(house_num_more_3(one_to_many), res)

def test_task_2(self):
    res = [('Великолепная', 170), ('Замечательная', 125), ('Веселая', 100)]
    self.assertEqual(streets_with_min_resident_cnt(one_to_many), res)

def test_task_3(self):
    res = [(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)')]
    self.assertEqual(data_sorted_by_house_num(many_to_many), res)

if __name__ == '__main__':
    unittest.main()

```

Результат

