

**Московский государственный технический  
университет им. Н.Э. Баумана.**

Факультет «Информатика и управление»  
Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Базовые компоненты интернет-технологий»

Отчет по РК №1

Выполнил:

студент группы ИУ5-31Б  
Зелинский Даниил Михайлович

Проверил:

преподаватель каф. ИУ5  
Гапанюк Юрий Евгеньевич

Подпись и дата:

Подпись и дата:

Москва, 2021 г.

## Задание.

Класс «Улица» содержит поля:

- ID улицы;
- Название улицы.

Класс «Здание» содержит поля:

- ID здания;
- Номер здания на улице;
- Количество людей, работающих в этом здании;
- Год возведения здания;
- ID улицы.

Стоит различать ID здания и его номер на улице.

Класс «Здания на улице» (для реализации связи многие-ко-многим) содержит поля:

- ID улицы;
- ID здания.

В задании A1 «Улица» и «Здание» связаны соотношением один-ко-многим.

Вывести список всех связанных зданий и улиц, отсортированный по улицам. В списке указать, сколько людей работают в каждом здании.

В задании A2 «Улица» и «Здание» связаны соотношением один-ко-многим.

Вывести список улиц с суммарным числом людей, работающих в зданиях, расположенных на этих улицах, отсортированный по числу людей.

В задании A3 «Улица» и «Здание» связаны соотношением многие-ко-многим.

Вывести список проспектов с номерами домов, расположенными на них (номера отсортированы).

## Код программы.

```
from operator import itemgetter

class Building:
    def __init__(self, id, num, people, year, streetID):
        self.id=id
        self.num=num
        self.people=people
        self.year=year
        self.streetID=streetID

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

class BuildStreet:
    def __init__(self, streetID, buildID):
        self.streetID=streetID
        self.buildID=buildID

#список улиц
streets=[
```

```

Street(1, 'Первомайская'),
Street(2, 'Золотарёвой'),
Street(3, 'Малахитовый пр-кт'),
Street(4, 'Серебряная'),
Street(5, 'Молдавский пр-кт'),
Street(6, 'Столичная'),
Street(7, 'Матросовский пр-кт')
]
#список зданий
builds=[
    Building(2, 2, 12, 1990, 1),
    Building(3, 1, 19, 2003, 2),
    Building(4, 15, 78, 2017, 3),
    Building(5, 14, 34, 2012, 3),
    Building(6, 6, 16, 1872, 6),
    Building(7, 101, 89, 1998, 7),
    Building(8, 102, 190, 2006, 7),
    Building(9, 99, 156, 2003, 7),
    Building(10, 100, 101, 2000, 7),
    Building(1, 1, 24, 2014, 1)
]
#список "Здания на улицах" для реализации связи многие-ко-многим
buildstreets=[
    BuildStreet(1, 1),
    BuildStreet(1, 2),
    BuildStreet(2, 3),
    BuildStreet(3, 4),
    BuildStreet(3, 5),
    BuildStreet(6, 6),
    BuildStreet(7, 7),
    BuildStreet(7, 8),
    BuildStreet(7, 9),
    BuildStreet(7, 10),

    BuildStreet(5, 8),
    BuildStreet(5, 9),
    BuildStreet(3, 10)
]

if __name__ == '__main__':
    #создание связи один ко многим
    one_to_many=[(s.name, b.num, b.people)
        for b in builds
        for s in streets
        if b.streetID==s.id
        ]
    #создание связи многие-ко-многим
    many_to_many_temp=[(s.name, bs.streetID, bs.buildID)
        for s in streets
        for bs in buildstreets
        if s.id==bs.streetID
        ]
    many_to_many=[(s_name, b.num, b.people)
        for s_name, streetID, buildID in many_to_many_temp
        for b in builds if b.id==buildID
        ]

    print('Задание A1')
    #выводится название улицы, номер здания на этой улице и количество
    работающих в здании людей
    res1=sorted(one_to_many, key=itemgetter(0))
    for r in res1:
        print(r)

```

```

print('Задание A2')
res2=[]
#создание списка улиц, которые участвуют в связях со зданиями
filled_streets=set([s_name for s_name, num, people in one_to_many])
#перебор этого списка
for s in filled_streets:
    sum=0
    #поиск в списке связей зданий, расположенных на текущей улице, и подсчёт людей в них работающих
    for s_name, num, people in one_to_many:
        if s==s_name:
            sum+=people
    res2.append((s, sum))

res2=sorted(res2, key=itemgetter(1), reverse=True)
print(res2)

print('Задание A3')
#выбирается слово, искомое в названиях улиц.
str_to_find='пр-кт' #по заданию - поиск проспектов
#формируется список проспектов, участвующих в отношениях
prospects=[s_name for s_name, num, people in many_to_many if s_name.find(str_to_find)!=-1]
#создаётся словарь, ключ - название проспекта, значение - список номеров домов, расположенных на этих улицах
res3={pr:sorted([num for s_name, num, people in many_to_many if pr==s_name]) for pr in prospects }

print(res3)

```

## Результаты работы программы.

```

Задание A1
('Золотарёвой', 1, 19)
('Малахитовый пр-кт', 15, 78)
('Малахитовый пр-кт', 14, 34)
('Матросовский пр-кт', 101, 89)
('Матросовский пр-кт', 102, 190)
('Матросовский пр-кт', 99, 156)
('Матросовский пр-кт', 100, 101)
('Первомайская', 2, 12)
('Первомайская', 1, 24)
('Столичная', 6, 16)
Задание A2
[('Матросовский пр-кт', 536), ('Малахитовый пр-кт', 112), ('Первомайская', 36), ('Золотарёвой', 19), ('Столичная', 16)]
Задание A3
{'Малахитовый пр-кт': [14, 15, 100], 'Молдавский пр-кт': [99, 102], 'Матросовский пр-кт': [99, 100, 101, 102]}

Process finished with exit code 0

```