| Московский | государственный | технический | университет | имени Н.Э. | Баумана |
|------------|-----------------|-------------|-------------|------------|---------|
| | | | | | |

Факультет РТ Радиотехнический

Кафедра ИУ5 Системы обработки информации и управления

Отчет по рубежному контролю № 1 по курсу Базовые компоненты

5 (количество листов)

| Исполнитель | | | |
|------------------------|---------------|----------------------|--|
| студент группы РТ5-31Б | | Викулин И.С. | |
| | | "27" октября 2022 г. | |
| Проверил | | | |
| Доцент кафедры ИУ5 | | Гапанюк Ю.Е. | |
| | | ""2022 г. | |
| | Москва – 2022 | | |

Вариант Е. Предметная область 6

- 1. «Улица» и «Дом» связаны соотношением один-ко-многим. Выведите список всех улиц, у которых в названии присутствует слово «Улица», и список расположенных на ней домов.
- 2. «Улица» и «Дом» связаны соотношением один-ко-многим. Выведите список улиц со средней площадью домов на каждой улице, отсортированный по средней площади. Средняя площадь должна быть округлена до 2 знака после запятой.
- 3. «Улица» и «Дом» связаны соотношением многие-ко-многим. Выведите список всех домов, у которых номер начинается с цифры «5», и названия их улиц.

Листинг программы

```
class Street:
   def __init__(self, id, name):
     self.id = id
     self.name = name
class House:
   def <u>init</u> (self, id, name, square, id street):
     self.name = name
     self.id = id
     self.square = square
     self.id street = id street
class HouseStreet:
   def __init__(self, id_house, id_street):
     self.id_house = id_house
     self.id street = id street
streets = [Street(1, 'Lenina street'),
   Street(2, 'Kirova'),
  Street(3, 'Sovetskaya street')
   1
houses = [House(1, 12, 1000, 1),
  House(2, 21, 2000, 1),
  House(3, 33, 1500, 1),
   House(4, 54, 4000, 2),
   House(5, 5, 1500, 2),
  House(6, 65, 200, 3),
  House(7, 53, 900, 3),
  House(8, 81, 200, 3)
   1
houses_streets = [HouseStreet(1, 1),
   HouseStreet(2, 1),
  HouseStreet(3, 1),
  HouseStreet(4, 2),
   HouseStreet(5, 2).
  HouseStreet(6, 3),
  HouseStreet(7, 3),
  HouseStreet(8, 3)
```

```
def task1():
  result = \{\}
  for street in streets:
     if 'street' in street.name:
        house data = \Pi
        for house in houses:
          if house.id street == street.id:
             cur_house_data = [house.name, house.square]
             house data.append(cur house data)
        result[street.name] = house data
  return result
def task2():
  result = \{\}
  # Будем сортировать по площади домов на улице
  for street in streets:
     count = 0
     for house in houses:
        if house.id_street == street.id:
          \overline{\text{count}} = 1
          if street.name in result:
             result[street.name] += house.square
             result[street.name] = house.square
     if (result[street.name] != 0):
        result[street.name] /= count
        result[street.name] = round(result[street.name], 2)
  return sorted(result.items(),reverse=False)
def task3():
  # Свойства класса дом не имеют полей, которые содержат буквы.
  # Поэтому будем вытаскивать не по первой букве, а по первой цифре
(например, 5)
  result = \{\}
  for house in houses:
     if str(abs(house.name))[0] == '5':
        for house_street in houses_streets:
          if house_street.id_house == house.id:
             for street in streets:
               if street.id == house_street.id_street:
                  result[house.name] = street.name
  return result
def main():
  print('\tTask 1')
```

```
print(task1())

print('\n\tTask 2')
print(task2())

print('\n\tTask 3')
print(task3())

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

Результаты выполнения программы

```
Task 1
{'Lenina street': [[12, 1000], [21, 2000], [33, 1500]], 'Sovetskaya street': [[65, 200], [53, 900], [81, 200]]}

Task 2
[('Kirova', 2750.0), ('Lenina street', 1500.0), ('Sovetskaya street', 433.33)]

Task 3
{54: 'Kirova', 5: 'Kirova', 53: 'Sovetskaya street'}
```