ИУ5-32Б,

Жуков Никита

***Рубежный контроль***

***Вариант 6***

******

Вариант Б:

1. «Производитель» и «Деталь» связаны соотношением один-ко-многим. Выведите список всех связанных деталей и производителей, отсортированных по деталям, сортировка по производителям произвольная.
2. «Производитель» и «Деталь» связаны соотношением один-ко-многим. Выведите список производителей с количеством деталей у каждого производителя, отсортированный по количеству деталей.
3. «Производитель» и «Деталь» связаны соотношением многие-ко-многим. Выведите список всех деталей, у которых id госта заканчивается на «1», и названия их производителей.

**Код программы**

*from operator import itemgetter  
  
  
class Home:  
 def \_\_init\_\_(self, id, home\_number):  
 self.id = id  
 self.home\_number=home\_number  
  
  
class Street:  
 def \_\_init\_\_(self, id, streetName , ppl\_count,Home\_id):  
 self.id = id  
 self.streetName = streetName  
 self.ppl\_count = ppl\_count  
 self.Home\_id = Home\_id  
  
  
class HomeStreet:  
 def \_\_init\_\_(self, Home\_id, Street\_id):  
 self.Home\_id = Home\_id  
 self.Street\_id = Street\_id  
  
  
Homes = [  
 Home(1, 1),  
 Home(2, 25),  
 Home(3, 30),  
]  
  
Streets = [  
 Street(1, "Bataskiy Proezd", 43, 1),  
 Street(2, "Lugovoi proezd", 21, 2),  
 Street(3, "Maryinskiy bulvar", 61, 3),  
 Street(4, "Rahovo", 38, 3),  
 Street(5, "Leninskiy Prospekt", 47, 1),  
 Street(6, "Ulica Lenina", 42, 1)  
]  
  
Home\_to\_Street = [  
 HomeStreet(1, 1),  
 HomeStreet(2, 2),  
 HomeStreet(3, 3),  
 HomeStreet(3, 4),  
 HomeStreet(1, 5),  
]  
  
  
def first\_task(Streets\_list):  
 res\_1 = sorted(Streets\_list, key=itemgetter(2))  
 return res\_1  
  
  
def second\_task(Streets\_list):  
 res\_2 = []  
 temp\_dict = dict()  
 for i in Streets\_list:  
 if i[2] in temp\_dict:  
 temp\_dict[i[2]] += 1  
 else:  
 temp\_dict[i[2]] = 1  
 for i in temp\_dict.keys():  
 res\_2.append((i, temp\_dict[i]))  
  
 res\_2.sort(key=itemgetter(1), reverse=True)  
 return res\_2  
  
  
def third\_task(Streets\_list, end\_ch):  
 res\_3 = [(i[0], i[2]) for i in Streets\_list if str(i[1]).endswith(end\_ch)]  
 return res\_3  
  
  
def main():  
 one\_to\_many = [(Street.streetName, Street.ppl\_count, Home.home\_number)  
 for Home in Homes  
 for Street in Streets  
 if Street.Home\_id == Home.id]  
  
 many\_to\_many\_temp = [(Home.home\_number, connection.Home\_id, connection.Street\_id)  
 for Home in Homes  
 for connection in Home\_to\_Street  
 if connection.Home\_id == Home.id]  
  
 many\_to\_many = [(Street.streetName, Street.ppl\_count, Home\_adress)  
 for Home\_adress, Home\_id, Street\_id in many\_to\_many\_temp  
 for Street in Streets if Street.id == Street\_id]  
  
 print('Задание Б1')  
 print(first\_task(one\_to\_many))  
  
 print("\nЗадание Б2")  
 print(second\_task(one\_to\_many))  
  
 print("\nЗадание Б3")  
 print(third\_task(many\_to\_many, '1'))  
  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 main()*

**Результаты выполнения**

