Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
«Московский государственный технический университет имени Н.Э. Баумана»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ	**«Информатики и систем управления»**
КАФЕДРА	ИУ5

Дисциплина «Разработка интернет-приложений»

Отчет по рубежному контролю №1
Вариант Б-7

Студент	группы ИУ5-52Б	Гришин Илья
Преподаватель		Гапанюк Ю.Е.

Классы для предметной области

1. Класс «Микропроцессор», содержащий поля:
   - ID записи о микропроцессоре;
   - Название микропроцессора;
   - Цена (количественный признак);
   - ID записи о компьютере. (для реализации связи один-ко-многим)
2. Класс «Компьютер», содержащий поля:
   - ID записи о компьютере;
   - Наименование компьютера.
3. (Для реализации связи многие-ко-многим) Класс «Микропроцессоры компьютера», содержащий поля:
   - ID записи о микропроцессоре;
   - ID записи о компьютере.

Задание

1. «Компьютер» и «Микропроцессор» связаны соотношением один-ко-многим. Выведите список всех связанных микропроцессоров и компьютеров, отсортированный по микропроцессорам, сортировка по компьютерам произвольная.
2. «Компьютер» и «Микропроцессор» связаны соотношением один-ко-многим. Выведите список компьютеров с количеством микропроцессоров в каждом компьютере, отсортированный по количеству микропроцессоров.
3. «Компьютер» и «Микропроцессор» связаны соотношением многие-ко-многим. Выведите список всех микропроцессоров, у которых название заканчивается на «К», и названия компьютеров, в которые они установлены.

Код

```
#Вариант Б 7; Гришин Илья Алексеевич ИУ5-52Б
# используется для сортировки
from operator import itemgetter

class Cpu:
    """Микропроцессор"""
    def __init__(self, id, name, price, comp_id):
        self.id = id
        self.name = name
        self.price = price
        self.comp_id = comp_id

class Comp:
    """Компьютер"""
```

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

class CpuComp:
    """
    'Микропроцессоры компьютера' для реализации
    связи многие-ко-многим
    """
    def __init__(self, comp_id, cpu_id):
        self.comp_id = comp_id
        self.cpu_id = cpu_id

# Компьютеры
comps = [
    Comp(1, 'Acer'),
    Comp(2, 'ASUS'),
    Comp(3, 'HP'),

    Comp(11, 'GIGABYTE'),
    Comp(22, 'Lenovo'),
    Comp(33, 'Dell'),
]

# Микропроцессоры
cpus = [
    Cpu(1, 'Intel i5-9400F', 13000, 1),
    Cpu(2, 'Intel i5-10400F', 15000, 1),
    Cpu(3, 'Intel i7-10700K', 37000, 3),
    Cpu(4, 'Intel i9-10900K', 50000, 11),
    Cpu(5, 'Intel i9-9900', 35000, 11),
    Cpu(6, 'AMD Ryzen 5 2600', 13000, 33),
    Cpu(7, 'AMD Ryzen 9 3900X', 45000, 11),
]

cpus_comps = [
    CpuComp(1,1),
    CpuComp(2,2),
    CpuComp(3,3),
    CpuComp(22,4),
    CpuComp(3,5),

    CpuComp(11,1),
    CpuComp(22,2),
    CpuComp(33,3),
    CpuComp(33,4),
    CpuComp(33,5),
]
```

```python
def main():
    """Основная функция"""
    # Соединение данных один-ко-многим
    one_to_many = [(m.name, m.price, c.name)
        for c in comps
        for m in cpus
        if m.comp_id==c.id]

    # Соединение данных многие-ко-многим
    many_to_many_temp = [(c.name, mc.comp_id, mc.cpu_id)
        for c in comps
        for mc in cpus_comps
        if c.id==mc.comp_id]

    many_to_many = [(m.name, m.price, comp_name)
        for comp_name, comp_id, cpu_id in many_to_many_temp
        for m in cpus if m.id==cpu_id]

    print('Задание Б1')
    res_11 = sorted(one_to_many, key=itemgetter(0))
    print(res_11)

    print('\nЗадание Б2')
    res_12_unsorted = []
    # Перебираем все компьютеры
    for c in comps:
        # Список микропроцессоров компьютера
        c_cpus = list(filter(lambda i: i[2]==c.name, one_to_many))
        # Если компьютер не пустой
        if len(c_cpus) > 0:
            res_12_unsorted.append((c.name, len(c_cpus)))

    # Сортировка по количеству микропроцессоров
    res_12 = sorted(res_12_unsorted, key=itemgetter(1), reverse=True)
    print(res_12)

    print('\nЗадание Б3')
    res_13 = {}
    # Перебираем все микропроцессоры
    for c in cpus:
        if c.name[-1]=="К":
            # Список микропроцессоров компьютера
            c_cpus = list(filter(lambda i: i[0]==c.name, many_to_many))
            # Только наименования компьютеров
            d_cpus_names = [x for _,_,x in c_cpus]
            # Добавляем результат в словарь
            # ключ - микропроцессор, значение - список названий компьютеров
            res_13[c.name] = d_cpus_names
    print(res_13)
if __name__ == '__main__':
    main()
```

Результат выполнения программы:

Задание Б1

[('AMD Ryzen 5 2600', 13000, 'Dell'), ('AMD Ryzen 9 3900X', 45000, 'GIGABYTE'), ('Intel i5-10400F', 15000, 'Acer'), ('Intel i5-9400F', 13000, 'Acer'), ('Intel i7-10700K', 37000, 'HP'), ('Intel i9-10900K', 50000, 'GIGABYTE'), ('Intel i9-9900', 35000, 'GIGABYTE')]

Задание Б2

[('GIGABYTE', 3), ('Acer', 2), ('HP', 1), ('Dell', 1)]

Задание Б3

{'Intel i7-10700K': ['HP', 'Dell'], 'Intel i9-10900K': ['Lenovo', 'Dell']}