

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

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

Курс «Разработка интернет приложений»

Рубежный контроль №1

Вариант №7Б

Выполнил:

студент группы ИУ5-52Б
Васильченко Дарья

Проверил:

преподаватель каф. ИУ5
Гапанюк Ю.Е.

Москва, 2021 г.

Описание задания:

- 1) Необходимо создать два класса данных в соответствии с Вашим вариантом предметной области, которые связаны отношениями один-ко-многим и многие-ко-многим.
- 2) Необходимо создать списки объектов классов, содержащих тестовые данные (3-5 записей), таким образом, чтобы первичные и вторичные ключи соответствующих записей были связаны по идентификаторам.
- 3) Необходимо разработать запросы в соответствии с Вашим вариантом. Запросы сформулированы в терминах классов «Сотрудник» и «Отдел», которые используются в примере. Вам нужно перенести эти требования в Ваш вариант предметной области. При разработке запросов необходимо по возможности использовать функциональные возможности языка Python (list/dict comprehensions, функции высших порядков).

Вариант Б.

1. «Микропроцессор» и «Компьютер» связаны соотношением один-ко-многим. Выведите список всех связанных компьютеров и микропроцессоров, отсортированный по компьютерам, сортировка по микропроцессорам произвольная.
2. «Микропроцессор» и «Компьютер» связаны соотношением один-ко-многим. Выведите список микропроцессоров с количеством компьютеров с каждым микропроцессором, отсортированный по количеству компьютеров.
3. «Микропроцессор» и «Компьютер» связаны соотношением многие-ко-многим. Выведите список всех компьютеров, у которых название заканчивается на «НР», и названия их микропроцессоров.

Текст программы

```
class Processor:
```

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

```
class PC:
```

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

```

        self.price = price
        self.processor_id = processor_id

class PC_Processor:

    def __init__(self, processor_id, pc_id):
        self.processor_id = processor_id
        self.pc_id = pc_id

def main():
    list_of_processors = [Processor(1, "Intel i5-9600", 3600, 128),
                          Processor(2, "Intel i7-10600", 3700, 128),
                          Processor(3, "Intel i5-6500", 3600, 12),
                          Processor(4, "Intel i5-6600", 3700, 32),
                          Processor(5, "Intel i7-8600", 3500, 64),
                          Processor(6, "Intel i7-10600", 3800, 256),
                          Processor(7, "Intel i3-5600", 3500, 128),
                          Processor(8, "Intel i7-4790", 3200, 128),
                          Processor(9, "Ryzen 5 5600X", 3800, 32),
                          Processor(10, "Ryzen 7 5800X", 3700, 32),
                          Processor(11, "Ryzen 9 5900X", 3700, 32),
                          Processor(12, "Ryzen 9 5950X", 3400, 64)]

    list_of_pc = [PC(1, "Lenovo IdeaCentre G5 14IMB05", 88000, 8),
                  PC(2, "HP Pavilion Gaming TG01", 110000, 1),
                  PC(3, "ZOTAC MAGNUS ONE", 187000, 7),
                  PC(4, "Apple Mac mini 2020 M1", 173000, 6),
                  PC(5, "Acer Aspire TC-895", 34000, 7),
                  PC(6, "Apple iMac 24\" 2021", 248000, 11),
                  PC(7, "Gigabyte GB-BR", 93000, 4),
                  PC(8, "HP M01", 61000, 3)]

    list_of_pc_processor = [PC_Processor(5, 2),
                            PC_Processor(10, 1),
                            PC_Processor(4, 1),
                            PC_Processor(4, 2),
                            PC_Processor(7, 5),
                            PC_Processor(2, 1),
                            PC_Processor(10, 3),
                            PC_Processor(4, 2),
                            PC_Processor(10, 1),
                            PC_Processor(2, 5),
                            PC_Processor(8, 8)]

    res1 = sorted(
        [(pc.name, processor.name) for pc in list_of_pc for processor in
list_of_processors if pc.processor_id == processor.id],
        key=lambda x: x[0])
    res2 = sorted(
        {processor.name: len(list(filter(lambda x: x.processor_id ==
processor.id, list_of_pc))) for processor in list_of_processors}.items(),
        key=lambda x: x[1], reverse=True)
    res3 = {pc.name: [processor.name for processor in list_of_processors if
processor.id in [pc_processor.processor_id for
pc_processor in list_of_pc_processor if
pc_processor.pc_id == pc.id]] for pc in
list_of_pc if
str(pc.name).startswith('HP')}

```

```
print(res1)
print(res2)
print(res3)

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

Результаты выполнения программы:

Задание Б1

```
[('Acer Aspire TC-895', 'Intel i3-5600'), ('Apple Mac mini 2020 M1',
'Intel i7-10600'), ('Apple iMac 24" 2021', 'Ryzen 9 5900X'), ('Gigabyte
GB-BR', 'Intel i5-6600'), ('HP M01', 'Intel i5-6500'), ('HP Pavilion
Gaming TG01', 'Intel i5-9600'), ('Lenovo IdeaCentre G5 14IMB05', 'Intel
i7-4790'), ('ZOTAC MAGNUS ONE', 'Intel i3-5600')]
```

Задание Б2

```
[('Intel i3-5600', 2), ('Intel i5-9600', 1), ('Intel i7-10600', 1),
('Intel i5-6500', 1), ('Intel i5-6600', 1), ('Intel i7-4790', 1), ('Ryzen
9 5900X', 1), ('Intel i7-8600', 0), ('Ryzen 5 5600X', 0), ('Ryzen 7
5800X', 0), ('Ryzen 9 5950X', 0)]
```

Задание Б3

```
{'HP Pavilion Gaming TG01': ['Intel i5-6600', 'Intel i7-8600'], 'HP M01':
['Intel i7-4790']}
```