



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

---

ФАКУЛЬТЕТ СМ  
КАФЕДРА СМ10-61Б «Колесные машины»

**Рубежный контроль №1 по курсу:**  
**«Разработка Web-приложений»**  
**Вариант №7**

Студент

Пёрушкин М.А.

(Подпись, дата)

Преподаватель

Гапанюк Ю.Е.

(Подпись, дата)

Москва, 2023 г.

### ***Условия варианта №7:***

Класс 1: “Микропроцессор”

Класс 2: “Компьютер”

Замечание: поскольку по заданию подразумевается, что классу 2 соответствует несколько значений из класса 1, будем считать, что у одного компьютера может быть несколько микропроцессоров.

### ***Задание***

1) Необходимо создать два класса данных в соответствии с Вашим вариантом предметной области, которые связаны отношениями один-ко-многим.

Классы данных для предметной области микропроцессор-компьютер:

Класс «Микропроцессор» содержит поля:

- ID записи о процессоре;
- Наименование модели;
- Стоимость (количественный признак);
- ID записи о процессоре. (для реализации связи один-ко-многим)

Класс «Компьютер» содержит поля:

- ID записи о компьютере;
- Наименование фирмы.

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 указанные далее запросы.

## Текст программы:

```
# используется для сортировки
from operator import itemgetter

class Mic:
    """Микропроцессор"""

    def __init__(self, id, name, cost, mic_id):
        self.id = id
        self.name = name
        self.cost = cost
        self.mic_id = mic_id

class comp:
    """Компьютер"""

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

# Компьютеры
comps = [
    comp(1, 'hp'),
    comp(2, 'lenovo'),
    comp(3, 'xiaomi'),
    comp(4, 'asus'),
]

# Микропроцессоры
mics = [
    Mic(1, 'Intel core i3', 5000, 1),
    Mic(2, 'Intel core i5', 8000, 2),
    Mic(3, 'Intel core i7', 24000, 3),
    Mic(4, 'AMD Ryzen 7', 20000, 3),
    Mic(5, 'AMD Ryzen 9', 37000, 4),
]

def main():
    """Основная функция"""

    # Соединение данных один-ко-многим
    one_to_many = [(e.name, e.cost, d.name)
                    for d in comps
                    for e in mics
                    if e.mic_id == d.id]

    print('Задание 1')
    res_1 = sorted(one_to_many, key=itemgetter(2))
    print(res_1)

    print('\nЗадание 2')
    res_2_unsorted = []
    # Перебираем все компьютеры
    for d in comps:
        # Список микропроцессоров у компьютера
        d_mics = list(filter(lambda i: i[2] == d.name, one_to_many))
```

```

        # Если компьютер не пустой и там есть микропроцессор из
списка
        if len(d_mics) > 0:
            # Стоимость микропроцесса компьютера
            d_sals = [sal for _, sal, _ in d_mics]
            # Суммарная стоимость микропроцессоров у компьютера
            d_sals_sum = sum(d_sals)
            res_2_unsorted.append((d.name, d_sals_sum))

        # Сортировка по суммарной стоимости от большей к меньшей
        res_2 = sorted(res_2_unsorted, key=itemgetter(1), reverse=True)
        print(res_2)

if __name__ == '__main__':
    main()

```

Результат выполнения программы:

Задание 1

```

[('AMD Ryzen 9', 37000, 'asus'), ('Intel core i3', 5000, 'hp'), ('Intel core i5',
8000, 'lenovo'), ('Intel core i7', 24000, 'xiaomi'), ('AMD Ryzen 7', 20000, 'xia-
omi')]

```

Задание 2

```

[('xiaomi', 44000), ('asus', 37000), ('lenovo', 8000), ('hp', 5000)]

```

Process finished with exit code 0