

МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ  
им. Н.Э. Баумана

Факультет “Информатика и системы управления”  
Кафедра “Системы обработки информации и управления”



Дисциплина “Парадигмы и конструкции языков программирования”

Отчет по рубежному контролю №1  
Вариант Д7



**Выполнила:**  
Студентка группы ИУ5-36Б  
Крахмальникова А. И.

**Проверил:**  
Гапанюк Ю. Е.

Москва 2025

## Листинг кода

### RK1.py

```
from operator import itemgetter

class Microprocessor:
    """Микропроцессор"""

    def __init__(self, id, model, frequency, computer_id):
        self.id = id
        self.model = model
        self.frequency = frequency
        self.computer_id = computer_id

class Computer:
    """Компьютер"""

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

class MicroprocessorComputer:
    """Связь многие-ко-многим между микропроцессорами и компьютерами"""

    def __init__(self, computer_id, microprocessor_id):
        self.computer_id = computer_id
        self.microprocessor_id = microprocessor_id

# Компьютеры
computers = [
    Computer(1, 'Acer'),
    Computer(2, 'Компьютер1'),
    Computer(3, 'A server'),
    Computer(11, 'Asus'),
    Computer(22, 'Ноутбук'),
    Computer(33, 'Компьютер2'),
]

# Микропроцессоры
microprocessors = [
    Microprocessor(1, 'Intel Xeon', 3500, 1),
    Microprocessor(2, 'AMD', 3200, 2),
    Microprocessor(3, 'Intel', 2800, 3),
    Microprocessor(4, 'AMD Ryzen', 3000, 3),
    Microprocessor(5, 'Intel Pentium', 2400, 3),
]

# Связи многие-ко-многим
microprocessors_computers = [
```

```

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

```

```

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

    # Соединение данных один-ко-многим
    one_to_many = [(m.model, m.frequency, c.name)
                    for c in computers
                    for m in microprocessors
                    if m.computer_id == c.id]

    # Соединение данных многие-ко-многим
    many_to_many_temp = [(c.name, mc.computer_id, mc.microprocessor_id)
                          for c in computers
                          for mc in microprocessors_computers
                          if c.id == mc.computer_id]

    many_to_many = [(m.model, m.frequency, comp_name)
                     for comp_name, comp_id, mproc_id in many_to_many_temp
                     for m in microprocessors if m.id == mproc_id]

    print('Задание Д1')
    res1 = list(filter(lambda i: i[0].endswith('\n'), one_to_many))
    print(res1)

    print('\nЗадание Д2')
    res2_unsorted = []
    for c in computers:
        # Список микропроцессоров компьютера
        c_microprocessors = list(filter(lambda i: i[2] == c.name, one_to_many))
        if len(c_microprocessors) > 0:
            # Частоты микропроцессоров
            c_freqs = [freq for _, freq, _ in c_microprocessors]
            # Средняя частота
            avg_freq = sum(c_freqs) / len(c_freqs)
            res2_unsorted.append((c.name, round(avg_freq, 2)))

    res2 = sorted(res2_unsorted, key=itemgetter(1))
    print(res2)

    print('\nЗадание Д3')
    res3 = {}
    for c in computers:

```

```
if c.name.startswith('A'):
    # Список микропроцессоров компьютера (многие-ко-многим)
    c_microprocessors = list(filter(lambda i: i[2] == c.name, many_to_many))
    # Модели микропроцессоров
    c_models = [model for model, _, _ in c_microprocessors]
    res3[c.name] = c_models
```

```
print(res3)
```

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

## Скриншоты работы программы

```
/Users/anastasiakrahmalnikova/PycharmProjects/PCPL_2025/venv/bin/python /Users/anastasiakrahmalnikova/PycharmProjects/PCPL_2025/RK1.py
Задание Д1
[('Intel Xeon', 3500, 'Acer'), ('AMD Ryzen', 3000, 'A server')]

Задание Д2
[('A server', 2733.33), ('Компьютер1', 3200.0), ('Acer', 3500.0)]

Задание Д3
{'Acer': ['Intel Xeon'], 'A server': ['Intel', 'AMD Ryzen', 'Intel Pentium'], 'Asus': ['Intel Xeon']}

Process finished with exit code 0
```