

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

**Факультет «Радиотехнический»  
Кафедра «Системы обработки информации и управления»**

**Курс «Парадигмы и конструкции языков программирования»**

**Отчет по рубежному контролю №1  
Вариант 19**

Выполнил:  
студент группы РТ5-31Б:  
Стенин В. В.

Подпись и дата:

Проверил:  
преподаватель кафедры ИУ5  
Гапанюк Ю. Е.

Подпись и дата:

Москва, 2024 г.

## Текст программы:

```
from operator import itemgetter

class Part:
    def __init__(self, id, name, price, man_id):
        self.id = id
        self.name = name
        self.price = price
        self.man_id = man_id

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

class PartMan:
    def __init__(self, man_id, part_id):
        self.man_id = man_id
        self.part_id = part_id

mans = [
    Manufacturer(1, 'Производитель 1'),
    Manufacturer(2, 'Производитель 2'),
    Manufacturer(3, 'Производитель 3'),
    Manufacturer(11, 'Производитель (другой) 1'),
    Manufacturer(22, 'Производитель (другой) 2'),
    Manufacturer(33, '(другой) Производитель 3'),
]

parts = [
    Part(1, 'Деталь 1', 25000, 1),
    Part(2, 'Деталь 2', 35000, 2),
    Part(3, 'Деталь 3', 45000, 3),
    Part(4, 'Деталь 4', 35000, 3),
    Part(5, 'Деталь 5', 25000, 3),
]

parts_mans = [
    PartMan(1, 1),
    PartMan(2, 2),
    PartMan(3, 3),
    PartMan(3, 4),
    PartMan(3, 5),
    PartMan(11, 1),
    PartMan(22, 2),
    PartMan(33, 3),
    PartMan(33, 4),
    PartMan(33, 5),
]

def main():
    one_to_many = [(p.name, p.price, m.name)
                    for m in mans
                    for p in parts
                    if p.man_id == m.id]

    many_to_many_temp = [(m.name, pm.man_id, pm.part_id)
                          for m in mans
```

```

        for pm in parts_mans
        if m.id == pm.man_id]

many_to_many = [(p.name, p.price, man_name)
                 for man_name, man_id, part_id in many_to_many_temp
                 for p in parts if p.id == part_id]

print('Задание A1')
res_11 = sorted(one_to_many, key=itemgetter(2))
print(res_11)

print('\nЗадание A2')
res_12_unsorted = []
for m in mans:
    m_parts = list(filter(lambda i: i[2] == m.name, one_to_many))
    if len(m_parts) > 0:
        m_prices = [price for _, price, _ in m_parts]
        m_prices_sum = sum(m_prices)
        res_12_unsorted.append((m.name, m_prices_sum))

res_12 = sorted(res_12_unsorted, key=itemgetter(1), reverse=True)
print(res_12)

print('\nЗадание A3')
res_13 = {}
for m in mans:
    if 'Производитель' in m.name:
        m_parts = list(filter(lambda i: i[2] == m.name, many_to_many))
        m_parts_names = [x for x, _, _ in m_parts]
        res_13[m.name] = m_parts_names

print(res_13)

if __name__ == '__main__':
    main()

```

Результаты:

Задание A1

[('Деталь 1', 25000, 'Производитель 1'), ('Деталь 2', 35000, 'Производитель 2'), ('Деталь 3', 45000, 'Производитель 3'), ('Деталь 4', 35000, 'Производитель 3'), ('Деталь 5', 25000, 'Производитель 3')]

Задание A2

[('Производитель 3', 105000), ('Производитель 2', 35000), ('Производитель 1', 25000)]

Задание A3

{'Производитель 1': ['Деталь 1'], 'Производитель 2': ['Деталь 2'], 'Производитель 3': ['Деталь 3', 'Деталь 4', 'Деталь 5'], 'Производитель (другой) 1': ['Деталь 1'], 'Производитель (другой) 2': ['Деталь 2'], '(другой) Производитель 3': ['Деталь 3', 'Деталь 4', 'Деталь 5']}

Process finished with exit code 0