



**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 «Информатика и системы управления»**  
**Кафедра «Системы обработки информации и управления»**

**Рубежный контроль 1**  
**по дисциплине «Базовые компоненты интернет-технологий»**

**Выполнил:**  
**Студент группы ИУ5-34Б М.С. Хатин**

**2021 г.**

## Текст программы (main.py)

```
import pandas as pd

class Det:
    """Деталь"""

    def __init__(self, id, name, price, prod_id):
        self.id = id
        self.name = name # название детали
        self.price = price # стоимость [$]
        self.prod_id = prod_id

class Prod:
    """Производитель"""

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

class DetProd:
    """
    'Детали производителя' для реализации
    связи многие-ко-многим
    """

    def __init__(self, prod_id, det_id):
        self.prod_id = prod_id
        self.det_id = det_id

# Производители
prods = [
```

```
Prod(1, 'Samsung'),
Prod(2, 'LG'),
Prod(3, 'Apple'),

Prod(11, 'Acronis'),
Prod(22, 'Microsoft'),
Prod(33, 'Panasonic'),
]
```

# Детали

```
dets = [
    Det(1, 'Кабель type-c', 10, 1),
    Det(2, 'Шнур usb 2.0', 15, 2),
    Det(3, 'Шнур usb 3.0', 11, 3),
    Det(4, 'Блок питания 12 v', 12, 3),
    Det(5, 'Блок питания 14 v', 16, 3),
]
```

```
dets_prods = [
    DetProd(1, 1),
    DetProd(2, 2),
    DetProd(3, 3),
    DetProd(3, 4),
    DetProd(3, 5),

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

```
def main():
```

```
    one_to_many = [(det.name, det.price, prod.name)
                    for prod in prods
```

```

        for det in dets
        if det.prod_id == prod.id]

# Соединение данных многие-ко-многим
many_to_many_temp = [(p.name, dp.prod_id, dp.det_id)
                      for p in prods
                      for dp in dets_prods
                      if p.id == dp.prod_id]

many_to_many = [(d.name, d.price, prod_name)
                 for prod_name, _, det_id in many_to_many_temp
                 for d in dets if d.id == det_id]

print('Задание Г1')
res_11 = filter(lambda entry: entry[2][0] == 'A', one_to_many)
print(pd.DataFrame(res_11, columns=['Название детали', 'Стоимость ($)',
'Производитель']))

print('\nЗадание Г2')
res = [(prod.name, max([price for _, price, prod_name in one_to_many if prod_name ==
prod.name])) for prod in prods
       if len(list(filter(lambda entry: entry[2] == prod.name, one_to_many))) > 0]
print(pd.DataFrame(sorted(res, key=lambda x: x[1], reverse=True),
columns=['Производитель', 'Макс. стоимость']))

print('\nЗадание Г3')
res = sorted(many_to_many, key=lambda entry: entry[2])
print(pd.DataFrame(res, columns=['Деталь', 'Стоимость', 'Производитель']))

if __name__ == '__main__':
    main()

```

## Результаты выполнения

```
(base) maxim@maxim-HLYL-WXX9:~/PycharmProjects/rk1[BKIT]$ python main.py
```

Задание Г1

	Название детали	Стоимость (\$)	Производитель
0	Шнур usb 3.0	11	Apple
1	Блок питания 12 v	12	Apple
2	Блок питания 14 v	16	Apple

Задание Г2

	Производитель	Макс. стоимость
0	Apple	16
1	LG	15
2	Samsung	10

Задание Г3

	Деталь	Стоимость	Производитель
0	Кабель type-c	10	Acronis
1	Шнур usb 3.0	11	Apple
2	Блок питания 12 v	12	Apple
3	Блок питания 14 v	16	Apple
4	Шнур usb 2.0	15	LG
5	Шнур usb 2.0	15	Microsoft
6	Шнур usb 3.0	11	Panasonic
7	Блок питания 12 v	12	Panasonic
8	Блок питания 14 v	16	Panasonic
9	Кабель type-c	10	Samsung