

**РК №1**

**Парадигмы и конструкции языков программирования**

**19 Вариант**

**ИБМЗ-34Б**

**Примакова Александра**

```
from operator import itemgetter
```

```
class Detail:
```

```
    """Деталь"""
```

```
    def __init__(self, id, name, description, manufacturer_id, стоимость):
```

```
        self.id = id
```

```
        self.name = name
```

```
        self.description = description
```

```
        self.manufacturer_id = manufacturer_id
```

```
        self.стоимость = стоимость
```

```
class Manufacturer:
```

```
    """Производитель"""
```

```
    def __init__(self, id, name, country):
```

```
        self.id = id
```

```
        self.name = name
```

```
        self.country = country
```

```
class DetailManufacturer:
```

```
    """
```

```
    'Детали производителя' для реализации  
связи многие-ко-многим
```

```
    """
```

```
    def __init__(self, manufacturer_id, detail_id):
```

```
        self.manufacturer_id = manufacturer_id
```

```
        self.detail_id = detail_id
```

```
# Производители
```

```
manufacturers = [
```

```
    Manufacturer(1, 'Компания А', 'Россия'),
```

```
    Manufacturer(2, 'Компания В', 'Германия'),
```

```
    Manufacturer(3, 'Компания С', 'Китай'),
```

```
]
```

```
# Детали
```

```
details = [
```

```

Detail(1, 'Гайка', 'Гайка М8', 1, 10),
Detail(2, 'Шайба', 'Шайба под гайку М8', 1, 5),
Detail(3, 'Болт', 'Болт с шестигранной головкой М8', 1, 15),
Detail(4, 'Подшипник', 'Шариковый подшипник 6203', 2, 25),
Detail(5, 'Шестерня', 'Шестерня 10 зубьев', 3, 30),
]

```

# Связи деталей и производителей

```

details_manufacturers = [
    DetailManufacturer(1, 1),
    DetailManufacturer(1, 2),
    DetailManufacturer(1, 3),
    DetailManufacturer(2, 4),
    DetailManufacturer(3, 5),
]

```

def main():

"""Основная функция"""

# Задание A1: Соединение данных один-ко-многим

one\_to\_many = [(d.name, d.стоимость, m.name)

for m in manufacturers

for d in details

if d.manufacturer\_id == m.id]

print("Задание A1")

print(sorted(one\_to\_many))

# Задание A2: Суммарная стоимость деталей каждого производителя

total\_cost = {}

for d in details:

if d.manufacturer\_id not in total\_cost:

total\_cost[d.manufacturer\_id] = 0

total\_cost[d.manufacturer\_id] += d.стоимость

result\_a2 = [(m.name, total\_cost.get(m.id, 0)) for m in manufacturers]

print("Задание A2")

print(result\_a2)

# Задание A3: Список производителей с деталями

many\_to\_many\_temp = [(m.name, dm.manufacturer\_id, dm.detail\_id)

for m in manufacturers

for dm in details\_manufacturers

if m.id == dm.manufacturer\_id]

many\_to\_many = {}

for manufacturer\_name, \_, detail\_id in many\_to\_many\_temp:

detail\_name = next(d.name for d in details if d.id == detail\_id)

if manufacturer\_name not in many\_to\_many:

```

    many_to_many[manufacturer_name] = []
    many_to_many[manufacturer_name].append(detail_name)
print("Задание A3")
print(many_to_many)

```

```

if __name__ == "__main__": # Fixed!
    main()

```

#### ВЫВОД:

```

Задание A1
[('Болт', 15, 'Компания А'), ('Гайка', 10, 'Компания А'), ('Подшипник', 25, 'Компания В'), ('Шайба', 5, 'Компания А'), ('Шестерня', 30, 'Компания С')]
Задание A2
[('Компания А', 30), ('Компания В', 25), ('Компания С', 30)]
Задание A3
{'Компания А': ['Гайка', 'Шайба', 'Болт'], 'Компания В': ['Подшипник'], 'Компания С': ['Шестерня']}

```