

## Текст программы

```
from operator import itemgetter

class Emp:
    """Деталь"""
    def __init__(self, id, name, price, dep_id):
        self.id = id
        self.name = name
        self.price = price
        self.dep_id = dep_id

class Dep:
    """Производитель"""
    def __init__(self, id, name):
        self.id = id
        self.name = name

class EmpDep:
    """
    'Детали от данного производителя' для реализации
    СВЯЗИ МНОГИЕ-КО-МНОГИМ
    """
    def __init__(self, dep_id, emp_id):
        self.dep_id = dep_id
        self.emp_id = emp_id

# Производители
deps = [
    Dep(1, 'AZ Spa'),
    Dep(2, 'Magido'),
    Dep(3, 'Carmec'),
    Dep(11, 'VTM Group'),
    Dep(22, 'Carmec Pro'),
    Dep(33, 'Newen'),
]

# Детали
emps = [
    Emp(1, 'Фреза', 2500, 1),
    Emp(2, 'Головка', 3000, 2),
    Emp(3, 'Форсунка', 4500, 3),
    Emp(4, 'Цилиндр', 5000, 3),
    Emp(5, 'Кардан', 2500, 22),
]

emps_deps = [
    EmpDep(1, 1),
    EmpDep(2, 2),
    EmpDep(3, 3),
    EmpDep(3, 4),
    EmpDep(3, 5),
    EmpDep(11, 1),
    EmpDep(22, 2),
    EmpDep(33, 3),
    EmpDep(33, 4),
    EmpDep(33, 5),
]

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

    # Соединение данных один-ко-многим
    one_to_many = [(e.name, e.price, d.name)
```

```

        for d in deps
        for e in emps
        if e.dep_id == d.id]

# Соединение данных многие-ко-многим
many_to_many_temp = [(d.name, ed.dep_id, ed.emp_id)
                      for d in deps
                      for ed in emps_deps
                      if d.id == ed.dep_id]

many_to_many = [(e.name, e.price, dep_name)
                 for dep_name, dep_id, emp_id in many_to_many_temp
                 for e in emps if e.id == emp_id]

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

print('\nЗадание A2')
res_12_unsorted = []
# Перебираем всех производителей
for d in deps:
    # Список деталей производителя
    d_emps = list(filter(lambda i: i[2] == d.name, one_to_many))
    # Если производитель пустой
    if len(d_emps) > 0:
        # Цены деталей производителя
        d_sals = [price for _, price, _ in d_emps]
        # Суммарная цена деталей производителя
        d_sals_sum = sum(d_sals)
        res_12_unsorted.append((d.name, d_sals_sum))

# Сортировка по суммарной цене
res_12 = sorted(res_12_unsorted, key=itemgetter(1), reverse=True)
print(res_12)

print('\nЗадание A3')
res_13 = {}
# Перебираем всех производителей
for d in deps:
    if 'Carmec' in d.name:
        # Список деталей производителя
        d_emps = list(filter(lambda i: i[2] == d.name, many_to_many))
        # Только названия деталей производителя
        d_emps_names = [x for x, _, _ in d_emps]
        # Добавляем результат в словарь
        # ключ - производитель, значение - список названий
        res_13[d.name] = d_emps_names

print(res_13)

if __name__ == '__main__':
    main()

```

## Результаты выполнения

### Задание A1

```
[('Фреза', 2500, 'AZ Spa'), ('Форсунка', 4500, 'Carmec'), ('Цилиндр', 5000, 'Carmec'),
('Кардан', 2500, 'Carmec Pro'), ('Головка', 3000, 'Magido')]
```

### Задание A2

```
[('Carmec', 9500), ('Magido', 3000), ('AZ Spa', 2500), ('Carmec Pro', 2500)]
```

### Задание A3

```
{'Carmec': ['Форсунка', 'Цилиндр', 'Кардан'], 'Carmec Pro': ['Головка']}
```