

Вариант 19. Деталь - Производитель

```
from operator import itemgetter

class Detail:
    '''Деталь'''
    def __init__(self, id, name, price, prod_id):
        self.id = id
        self.name = name
        self.price = price
        self.prod_id = prod_id

class Producer:
    '''Производитель'''
    def __init__(self, id, name):
        self.id = id
        self.name = name

class DetailProducer:
    '''СВЯЗЬ МНОГИЕ КО МНОГИМ'''
    def __init__(self, prod_id, detail_id):
        self.prod_id = prod_id
        self.detail_id = detail_id

'''Производители'''
producers = [
    Producer(1, "Завод деталей"),
    Producer(2, "Машиностроительный завод"),
    Producer(3, "Фабрика Механическая"),
    Producer(4, "Компания ТехДеталь"),
    Producer(5, "Завод МеталлПром"),
    Producer(6, "Механический цех Деталь")
]

'''Детали'''
details = [
    Detail(1, "Болт", 10, 1),
    Detail(2, "Гайка", 8, 1),
    Detail(3, "Шайба", 7, 2),
    Detail(4, "Подшипник", 50, 3),
    Detail(5, "Поршень", 100, 3),
    Detail(6, "Втулка", 40, 4),
    Detail(7, "Шестерня", 120, 5),
    Detail(8, "Клапан", 90, 6),
    Detail(9, "Фильтр масляный", 110, 6),
]

'''связи многие ко многим'''
details_producers = [
    DetailProducer(1, 1),
    DetailProducer(1, 2),
    DetailProducer(2, 3),
    DetailProducer(3, 4),
    DetailProducer(3, 5),
    DetailProducer(4, 6),
    DetailProducer(4, 1),
    DetailProducer(2, 2),
    DetailProducer(5, 7),
    DetailProducer(6, 8),
    DetailProducer(6, 9),
    DetailProducer(2, 4),
    DetailProducer(3, 6),
]
```

```

        DetailProducer(5, 1),
    ]

def main():
    '''СВЯЗЬ ОДИН КО МНОГИМ'''
    one_to_many = [(d.name, d.price, p.name)
                    for p in producers
                    for d in details
                    if d.prod_id == p.id]

    '''СВЯЗЬ МНОГИЕ КО МНОГИМ'''
    many_to_many_temp = [(p.name, dp.prod_id, dp.detail_id)
                          for p in producers
                          for dp in details_producers
                          if p.id == dp.prod_id]

    many_to_many = [(d.name, d.price, prod_name)
                     for prod_name, prod_id, detail_id in many_to_many_temp
                     for d in details if d.id == detail_id]

    '''«Деталь» и «Производитель» связаны соотношением один-ко-многом.
    Вывести
        список всех связанных сотрудников и отделов, отсортированный по
    отделам,
        сортировка по сотрудникам произвольная'''
    print('Задание A1')
    print("Список всех связанных деталей и производителей, отсортированный по
    производителям:")
    res_1 = sorted(one_to_many, key=itemgetter(2))
    print(res_1)

    '''«Деталь» и «Производитель» связаны соотношением один-ко-многом.
    Вывести
        список производителей с суммарной стоимостью деталей,
        отсортированный по суммарной цене'''
    print('\nЗадание A2')
    print("Список производителей с суммарной стоимостью деталей,
    отсортированный по суммарной цене:")
    res_2_unsorted = []
    for p in producers:
        p_details = list(filter(lambda i: i[2] == p.name, one_to_many))
        if len(p_details) > 0:
            prices = [price for _, price, _ in p_details]
            total_price = sum(prices)
            res_2_unsorted.append((p.name, total_price))

    res_2 = sorted(res_2_unsorted, key=itemgetter(1), reverse=True)
    print(res_2)

    '''«Деталь» и «Производитель» связаны соотношением многие-ко-многом.
    Выведите
        список всех производителей, у которых в названии присутствует слово
    «завод», и их детали'''
    print('\nЗадание A3')
    print("Список производителей, у которых в названии есть слово 'завод', и
    их детали:")
    res_3 = {}
    for p in producers:
        if "завод" in p.name.lower():
            p_details = list(filter(lambda i: i[2] == p.name, many_to_many))
            p_detail_names = [x for x, _, _ in p_details]
            res_3[p.name] = p_detail_names

    print(res_3)

```

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

Вывод:

#### Задание A1

Список всех связанных деталей и производителей, отсортированный по производителям:

```
[('Шестерня', 120, 'Завод МеталлПром'), ('Болт', 10, 'Завод деталей'), ('Гайка', 8, 'Завод деталей'),  
(('Втулка', 40, 'Компания ТехДеталь'), ('Шайба', 7, 'Машиностроительный завод'), ('Клапан', 90,  
'Механический цех Деталь'), ('Фильтр масляный', 110, 'Механический цех Деталь'), ('Подшипник',  
50, 'Фабрика Механическая'), ('Поршень', 100, 'Фабрика Механическая')]
```

#### Задание A2

Список производителей с суммарной стоимостью деталей, отсортированный по суммарной цене:

```
[('Механический цех Деталь', 200), ('Фабрика Механическая', 150), ('Завод МеталлПром', 120),  
(('Компания ТехДеталь', 40), ('Завод деталей', 18), ('Машиностроительный завод', 7))]
```

#### Задание A3

Список производителей, у которых в названии есть слово 'завод', и их детали:

```
{'Завод деталей': ['Болт', 'Гайка'], 'Машиностроительный завод': ['Шайба', 'Гайка', 'Подшипник'],  
'Завод МеталлПром': ['Шестерня', 'Болт']}
```