

РК1, вариант 20, вариант запроса В

```
from operator import itemgetter

class Detail:
    def __init__(self, id, name, price, supplier_id):
        self.id = id
        self.name = name
        self.price = price
        self.supplier_id = supplier_id

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

class Supply:
    def __init__(self, supplier_id, detail_id):
        self.supplier_id = supplier_id
        self.detail_id = detail_id

suppliers = [
    Supplier(1, 'Авиатехно'),
    Supplier(2, 'Автодеталь'),
    Supplier(3, 'Металлопрокат'),
    Supplier(4, 'Станкоимпорт'),
    Supplier(5, 'Техносила'),
]

details = [
    Detail(1, 'Анкерный болт', 28.90, 1),
    Detail(2, 'Гайка М10', 8.30, 2),
    Detail(3, 'Шайба 10мм', 5.20, 2),
    Detail(4, 'Винт М6', 12.10, 3),
    Detail(5, 'Подшипник 6305', 245.00, 4),
    Detail(6, 'Амортизатор', 320.75, 4),
    Detail(7, 'Арматура А1', 45.80, 1),
]

supplies = [
    Supply(1, 1),
    Supply(1, 7),
    Supply(2, 2),
    Supply(2, 3),
    Supply(3, 4),
    Supply(4, 5),
```

```
Supply(4, 6),
Supply(5, 1),
Supply(5, 4),
]
```

```
def main():
```

```
    # Соединение данных один-ко-многим
    one_to_many = [(d.name, d.price, s.name)
                    for s in suppliers
                    for d in details
                    if d.supplier_id == s.id]
```

```
    # Соединение данных многие-ко-многим
    many_to_many_temp = [(s.name, sup.supplier_id, sup.detail_id)
                          for s in suppliers
                          for sup in supplies
                          if s.id == sup.supplier_id]
```

```
    many_to_many = [(d.name, d.price, supplier_name)
                     for supplier_name, supplier_id, detail_id in many_to_many_temp
                     for d in details if d.id == detail_id]
```

```
    print("Задание B1")
```

```
    # Список всех деталей, у которых название начинается с буквы «А», и названия их поставщиков
    res1 = list(filter(lambda i: i[0].startswith('A'), one_to_many))
    print(res1)
```

```
    print("\nЗадание B2")
```

```
    # Список поставщиков с минимальной ценой детали в каждом отделе, отсортированный по
    минимальной цене
```

```
    res2_unsorted = []
    for s in suppliers:
        s_details = list(filter(lambda i: i[2] == s.name, one_to_many))
        if len(s_details) > 0:
            s_prices = [price for _, price, _ in s_details]
            min_price = min(s_prices)
            res2_unsorted.append((s.name, min_price))
```

```
    res2 = sorted(res2_unsorted, key=itemgetter(1))
    print(res2)
```

```
    print("\nЗадание B3")
```

```
    res3 = sorted(many_to_many, key=itemgetter(0))
    print(res3)
```

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

## Результат выполнения:

Задание В1

```
[('Анкерный болт', 28.9, 'Авиатехно'), ('Арматура А1', 45.8, 'Авиатехно'), ('Амортизатор', 320.75, 'Станкоимпорт')]
```

Задание В2

```
[('Автодеталь', 5.2), ('Металлопрокат', 12.1), ('Авиатехно', 28.9), ('Станкоимпорт', 245.0)]
```

Задание В3

```
[('Амортизатор', 320.75, 'Станкоимпорт'), ('Анкерный болт', 28.9, 'Авиатехно'), ('Анкерный болт', 28.9, 'Техносила'), ('Арматура А1', 45.8, 'Авиатехно'), ('Винт М6', 12.1, 'Металлопрокат'), ('Винт М6', 12.1, 'Техносила'), ('Гайка М10', 8.3, 'Автодеталь'), ('Подшипник 6305', 245.0, 'Станкоимпорт'), ('Шайба 10мм', 5.2, 'Автодеталь')]
```