

"""Вариант 16-Г"""

```
class Book:
    def __init__(self, id, title, price, shop_id):
        self.id = id
        self.title = title
        self.price = price
        self.shop_id = shop_id
```

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

```
class BookShop:
    def __init__(self, book_id, shop_id):
        self.book_id = book_id
        self.shop_id = shop_id
```

```
shops = [
    Shop(1, "Books and Humans"),
    Shop(2, "Book World"),
    Shop(3, "Read and enjoy"),
    Shop(4, "The reading owl")
]
```

```
books = [
    Book(1, "Пикник на обочине", 950, 1),
    Book(2, "Три товарища", 1100, 1),
    Book(3, "Зеленая миля", 1200, 2),
    Book(4, "Десять негритят", 850, 3),
    Book(5, "Турецкий гамбит", 1050, 4)
]
```

```
books_shops = [
    BookShop(1, 1),
    BookShop(1, 2),
    BookShop(2, 3),
    BookShop(2, 2),
    BookShop(2, 1),
    BookShop(3, 4),
    BookShop(3, 1),
    BookShop(4, 2),
    BookShop(4, 4),
    BookShop(5, 3),
]
```

```

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

    # Соединение данных один-ко-многим
    one_to_many = [(b.title, b.price, s.name)
                    for b in books
                    for s in shops
                    if b.shop_id == s.id]

    # Соединение данных многие-ко-многим
    many_to_many_temp = [(s.name, sh.shop_id, sh.book_id)
                           for s in shops
                           for sh in books_shops
                           if s.id == sh.shop_id]

    many_to_many = [(b.title, b.price, shop_name)
                     for shop_name, shop_id, book_id in
many_to_many_temp
                     for b in books if b.id == book_id]

    print('Задание Г1:\n"Магазин" и "Книга" связаны отношением один-
ко-многим. Выведите список всех магазинов, '
          'в названии которых присутствует слово "book", и список книг
в этих магазинах\n')

    correct_shop_names = list(filter(lambda x: 'book' in x.lower(),
[sh.name for sh in shops]))
    res1 = {name: [(a[0], a[1]) for a in one_to_many if a[2] == name]
            for name in correct_shop_names}
    print(res1)

    print('\nЗадание Г2:\n"Магазин" и "Книга" связаны отношением один-
ко-многим. Выведите список магазинов с '
          'максимальной ценой одной книги в каждом магазине,
отсортированный по максимальной цене')

    shop_names = [sh.name for sh in shops]
    res2 = sorted([(name, max([a[1] for a in one_to_many if a[2] ==
name])) for name in shop_names], key=lambda x: -x[1])
    print(res2)

    print('\nЗадание Г3:\n"Магазин" и "Книга" связаны отношением
многие-ко-многим. Выведите список всех '
          'связанных книг и магазинов, отсортированный по магазинам,
сортировка по книгам произвольная')

    shop_names = [sh.name for sh in shops]
    for name in sorted(shop_names):
        print('\n', name)
        [print(f'\t{a[0]} {a[1]}') for a in many_to_many if a[2] ==

```

```
name]
```

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

Задание Г1:

"Магазин" и "Книга" связаны отношением один-ко-многим. Выведите список всех магазинов, в названии которых присутствует слово "book", и список книг в этих магазинах

```
{'Books and Humans': [('Пикник на обочине', 950), ('Три товарища', 1100)], 'Book World': [('Зеленая миля', 1200)]}
```

Задание Г2:

"Магазин" и "Книга" связаны отношением один-ко-многим. Выведите список магазинов с максимальной ценой одной книги в каждом магазине, отсортированный по максимальной цене

```
[('Book World', 1200), ('Books and Humans', 1100), ('The reading owl', 1050), ('Read and enjoy', 850)]
```

Задание Г3:

"Магазин" и "Книга" связаны отношением многие-ко-многим. Выведите список всех связанных книг и магазинов, отсортированный по магазинам, сортировка по книгам произвольная

Book World

Пикник на обочине 950

Три товарища 1100

Десять негритят 850

Books and Humans

Пикник на обочине 950

Три товарища 1100

Зеленая миля 1200

Read and enjoy

Три товарища 1100

Турецкий гамбит 1050

The reading owl

Зеленая миля 1200

Десять негритят 850