

**Московский государственный технический университет
им. Н. Э. Баумана**

**Факультет «Информатика и системы управления»
Кафедра ИУ5 «Системы обработки информации и управления»**

**Курс «Парадигмы и конструкции языков программирования»
Отчет по рубежному контролю №1
«Вариант Г, 16»**

**Выполнил:
Студент группы ИУ5-35Б
Корольков Игорь**

**Проверил:
Гапанюк Ю. Е.**

2025 г.

Листинг программы:

```
from operator import itemgetter

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

class Shop:
    """Книжный магазин"""
    def __init__(self, id, name):
        self.id = id
        self.name = name

class BookShop:
    """Связь книги и магазина (многие-ко-многим)"""
    def __init__(self, shop_id, book_id):
        self.shop_id = shop_id
        self.book_id = book_id

# --- Данные ---
shops = [
    Shop(1, 'Альфа-Книги'),
    Shop(2, 'Арбатская книга'),
    Shop(3, 'Город книг'),
    Shop(4, 'Мир литературы'),
    Shop(5, 'Астра-Книга'),
]

books = [
    Book(1, 'Алые паруса', 450, 1),
    Book(2, 'Преступление и наказание', 550, 1),
    Book(3, 'Мастер и Маргарита', 700, 3),
    Book(4, 'Война и мир', 600, 3),
    Book(5, 'Анна Каренина', 500, 2),
    Book(6, '1984', 400, 4),
    Book(7, 'А зори здесь тихие', 480, 5),
]

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

```

BookShop(3, 4),
BookShop(4, 6),
BookShop(5, 7),
# СВЯЗИ МНОГИЕ-КО-МНОГИМ
BookShop(1, 3),
BookShop(2, 4),
BookShop(5, 1)
]

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

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

    # --- Многие-ко-многим ---
    many_to_many_temp = [(s.name, bs.shop_id, bs.book_id)
                          for s in shops
                          for bs in books_shops
                          if s.id == bs.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]

    # === Задание Г1 ===
    print('Задание Г1')
    res_1 = {}
    for s in shops:
        if s.name.startswith('A'):
            # Список книг магазина
            s_books = list(filter(lambda i: i[2] == s.name, one_to_many))
            book_titles = [x for x, _, _ in s_books]
            res_1[s.name] = book_titles
    print(res_1)

    # === Задание Г2 ===
    print('\nЗадание Г2')
    res_2_unsorted = []
    for s in shops:
        s_books = list(filter(lambda i: i[2] == s.name, one_to_many))
        if len(s_books) > 0:
            prices = [price for _, price, _ in s_books]
            max_price = max(prices)
            res_2_unsorted.append((s.name, max_price))

```

```

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

# === Задание Г3 ===
print('\nЗадание Г3')
res_3 = sorted(many_to_many, key=itemgetter(2))
print(res_3)

if __name__ == '__main__':
    main()

```

Результаты выполнения программы:

```

Задание Г1
{'Альфа-Книги': ['Алые паруса', 'Преступление и наказание'], 'Арбатская книга': ['Анна Каренина'], 'Астра-Книга': ['А зори здесь тихие']}

Задание Г2
[('Город книг', 700), ('Альфа-Книги', 550), ('Арбатская книга', 500), ('Астра-Книга', 480), ('Мир литературы', 400)]

Задание Г3
[('Алые паруса', 450, 'Альфа-Книги'), ('Преступление и наказание', 550, 'Альфа-Книги'), ('Мастер и Маргарита', 700, 'Альфа-Книги'), ('Анна Каренина', 500, 'Арбатская книга'), ('Война и мир', 600, 'Арбатская книга'), ('А зори здесь тихие', 480, 'Астра-Книга'), ('Алые паруса', 450, 'Астра-Книга'), ('Мастер и Маргарита', 700, 'Город книг'), ('Война и мир', 600, 'Город книг'), ('1984', 400, 'Мир литературы')]

```