



Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Московский государственный технический университет  
имени Н.Э. Баумана  
(национальный исследовательский университет)»  
(МГТУ им. Н.Э. Баумана)

---

ФАКУЛЬТЕТ

Информатика и системы управления

КАФЕДРА

Системы обработки информации и управления

Курс «Парадигмы и конструкции языков программирования»

**Отчет по РК №1**  
**Вариант запросов: Г**  
**Вариант предметной области: 24**

Подготовила:

Студентка группы ИУ5Ц-54Б

Тихонова Д.Д.

20.10.2023

(подпись, дата)

Проверил:

Гапанюк Ю.Е.

\_\_\_\_\_  
(подпись, дата)

*МГТУ им. Н.Э. Баумана*

*Москва, 2023 г.*

## Вариант запросов - Г.

### Предметная область – «Глава-Книга».

1. «Книга» и «Глава» связаны соотношением один-ко-многим.  
Выведите список всех книг, у которых название начинается с буквы «П», и список их глав.
2. «Книга» и «Глава» связаны соотношением один-ко-многим.  
Выведите список книг с максимальным количеством знаков в каждой книге, отсортированный по максимальному количеству знаков.
3. «Книга» и «Глава» связаны соотношением многие-ко-многим.  
Выведите список всех связанных глав и книг, отсортированный по книгам, сортировка по главам произвольная.

### Листинг программы.

```
# Классы данных для предметной области "Глава-Книга"
class Chapter:
    """Глава"""
    def __init__(self, id, title, book_id, num_of_chars):
        self.id = id
        self.title = title
        self.book_id = book_id
        self.num_of_chars = num_of_chars

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

# Создадим класс для связующей таблицы между книгами и главами
class BookChapter:
    """Книги и главы"""
    def __init__(self, book_id, chapter_id):
        self.book_id = book_id
        self.chapter_id = chapter_id

# Списки объектов классов с тестовыми данными
books = [
    Book(1, 'Война и мир', 'Лев Толстой'),
    Book(2, 'Преступление и наказание', 'Федор Достоевский'),
    Book(3, 'Мастер и Маргарита', 'Михаил Булгаков'),
    Book(4, '1984', 'Джордж Оруэлл'),
    Book(5, 'Улисс', 'Джеймс Джойс'),
]

chapters = [
    Chapter(1, 'Часть первая. Война', 1, 100),
    Chapter(2, 'Часть вторая. Мир', 1, 120),
    Chapter(3, 'Часть первая. Преступление', 2, 80),
    Chapter(4, 'Часть вторая. Наказание', 2, 90),
    Chapter(5, 'Часть первая. Москва', 3, 110),
    Chapter(6, 'Часть вторая. Маргарита', 3, 130),
    Chapter(7, 'Часть первая. Новый мир', 4, 150),
```

```

Chapter(8, 'Часть вторая. Глава 8', 4, 110),
Chapter(9, 'Часть первая. Лестница', 5, 120),
Chapter(10, 'Часть вторая. Пенелопа', 5, 140),
]

# Связь многие-ко-многим между книгами и главами
book_chapters = [
    BookChapter(1, 1),
    BookChapter(1, 2),
    BookChapter(2, 3),
    BookChapter(2, 4),
    BookChapter(3, 5),
    BookChapter(3, 6),
    BookChapter(4, 7),
    BookChapter(4, 8),
    BookChapter(5, 9),
    BookChapter(5, 10),
]

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

    # Соединение данных один-ко-многим
    one_to_many = [(c.title, b.title) for b in books for c in chapters if c.book_id
== b.id]

    # Соединение данных многие-ко-многим
    many_to_many_temp = [(b.title, c.title) for bc in book_chapters for b in books
for c in chapters if bc.book_id == b.id and bc.chapter_id == c.id]

    many_to_many = sorted(many_to_many_temp, key=lambda x: x[0])

    # Задание Г1
    print('Задание Г1 - выводит книги, у которых название начинается с буквы "П" и
список их глав.')
    for book in books:
        if book.title.startswith('П'):
            chapters_list = [c for c, b in one_to_many if b == book.title]
            print(f'{book.title}: {chapters_list}')

    # Задание Г2
    print('\nЗадание Г2 - выводит книги с максимальным количеством знаков в каждой
книге, отсортированные по максимальному количеству знаков.')
    # Создайте список, в котором каждый элемент - кортеж (название книги,
максимальное количество знаков)
    book_max_chars = [(b.title, max(c.num_of_chars for c in chapters if c.book_id ==
b.id)) for b in books]
    # Отсортируйте список по максимальному количеству знаков в убывающем порядке
    sorted_books = sorted(book_max_chars, key=lambda x: x[1], reverse=True)
    for book_title, max_chars in sorted_books:
        print(f'{book_title}: {max_chars}')

    # Задание Г3
    print('\nЗадание Г3 - выводит список связанных глав и книг, отсортированный по
книгам.')
    for i in many_to_many:
        print(i)

if __name__ == '__main__':
    main()

```

## Результаты выполнения программы

```
C:\Program Files (x86)\Microsoft Visual Studio\Shared\Python39_64\python.exe
Задание Г1 - выводит книги, у которых название начинается с буквы "П" и список их глав.
Преступление и наказание: ['Часть первая. Преступление', 'Часть вторая. Наказание']

Задание Г2 - выводит книги с максимальным количеством знаков в каждой книге, отсортированные по максимальному количеству
знаков.
1984: 150
Улисс: 140
Мастер и Маргарита: 130
Война и мир: 120
Преступление и наказание: 90

Задание Г3 - выводит список связанных глав и книг, отсортированный по книгам.
('1984', 'Часть первая. Новый мир')
('1984', 'Часть вторая. Глава 8')
('Война и мир', 'Часть первая. Война')
('Война и мир', 'Часть вторая. Мир')
('Мастер и Маргарита', 'Часть первая. Москва')
('Мастер и Маргарита', 'Часть вторая. Маргарита')
('Преступление и наказание', 'Часть первая. Преступление')
('Преступление и наказание', 'Часть вторая. Наказание')
('Улисс', 'Часть первая. Лестница')
('Улисс', 'Часть вторая. Пенелопа')
```