Текст программы

```
from operator import itemgetter

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

class Chapter:
    """Глава"""
    def __init__(self, id, name, book_id, pages):
        self.id = id
        self.name = name
        self.book_id = book_id
        self.pages = pages

class BookChapter:
    """
    'Главы книги' для реализации
    связи многие-ко-многим
    """
    def __init__(self, book_id, chapter_id):
        self.book_id = book_id
        self.chapter_id = chapter_id

# Книги
books = [
    Book(1, 'Книга A', 'Автор A'),
    Book(2, 'Учебник B', 'Автор B'),
    Book(3, 'Книга C', 'Автор C'),
    Book(4, 'Учебник D', 'Автор D'),
    Book(5, 'Книга E', 'Автор E'),
    Book(6, 'Учебник F', 'Автор F'),
]

# Главы
chapters = [
    Chapter(1, 'Глава A', 1, 10),
    Chapter(2, 'Глава B', 2, 40),
    Chapter(3, 'Авторская Глава C', 3, 20),
    Chapter(4, 'Глава D', 4, 5),
    Chapter(5, 'Глава E', 5, 11),
```

```
    Chapter(6, 'Глава F', 6, 30),
    Chapter(7, 'Авторская Глава G', 1, 40),
    Chapter(8, 'Глава H', 2, 8),
    Chapter(9, 'Глава I', 3, 10),
    Chapter(10, 'Глава J', 4, 33),
    Chapter(11, 'Авторская Глава K', 5, 44),
    Chapter(12, 'Глава L', 6, 55),
]

book_chapters = [
    BookChapter(1, 1),
    BookChapter(1, 7),
    BookChapter(2, 2),
    BookChapter(2, 8),
    BookChapter(3, 3),
    BookChapter(3, 9),
    BookChapter(4, 4),
    BookChapter(4, 10),
    BookChapter(5, 5),
    BookChapter(5, 11),
    BookChapter(6, 6),
    BookChapter(6, 12),
]


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

    # Соединение данных один-ко-многим
    one_to_many = [(book.name, book.author, chapter.name, chapter.pages)
                   for book in books
                   for chapter in chapters
                   if chapter.book_id == book.id]

    # Соединение данных многим-ко-многим
    many_to_many_temp = [(book.name, book_chapter.book_id,
book_chapter.chapter_id)
        for book_chapter in book_chapters
        for book in books
        if book.id==book_chapter.book_id]

    many_to_many = [(chapter.name, book_name)
        for book_name, book_id, chapter_id in many_to_many_temp
        for chapter in chapters if chapter.id==chapter_id]
```

```
    print('Задание A1')
    print(list(filter(lambda i: i[0].find('Книга') != -1, one_to_many)))

    print('Задание A2')
    res_12 = []
    for book in books:
        current_book_chapters = list(filter(lambda i: i[0] == book.name,
one_to_many))
        if len(current_book_chapters) > 0:
            book_pages = [pages for _, _, _, pages in current_book_chapters]
            res_12.append(
                (
                    book.name,
                    round(sum(book_pages) / len(book_pages), 2)
                )
            )
    res_12 = sorted(res_12, key=itemgetter(1), reverse=True)
    print(res_12)

    print('Задание A3')
    res_13 = list(filter(lambda i: i[0][0] == 'A', many_to_many))
    print(res_13)

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

Результат выполнения:

Задание A1

[('Книга A', 'Автор A', 'Глава A', 10), ('Книга A', 'Автор A', 'Авторская Глава G', 40), ('Книга C', 'Автор C', 'Авторская Глава C', 20), ('Книга C', 'Автор C', 'Глава I', 10), ('Книга E', 'Автор E', 'Глава E', 11), ('Книга E', 'Автор E', 'Авторская Глава K', 44)]

Задание A2

[('Учебник F', 42.5), ('Книга E', 27.5), ('Книга A', 25.0), ('Учебник B', 24.0), ('Учебник D', 19.0), ('Книга C', 15.0)]

Задание A3

[('Авторская Глава G', 'Книга A'), ('Авторская Глава C', 'Книга C'), ('Авторская Глава K', 'Книга E')]