Код:

```python
# вариант запроса Д
# вариант предметной области 24 : Глава - Книга
from operator import itemgetter
class Chapter:
    # Глава
    def __init__(self, id, title, wordcount, Book_id):
        self.id = id
        self.title = title
        self.wordcount = wordcount
        self.Book_id = Book_id

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


class BookChapter:
    # главы для реализации связи
    # многие-ко-многим
    def __init__(self, books_id, chapters_id):
        self.books_id = books_id
        self.chapters_id = chapters_id
books = [
```

```python
    Book(1, "Сталкер"),
    Book(2, "Метро 2033"),
    Book(3, "Пикник у обочины"),
    Book(4, "Бегущий в лабиринте"),
    Book(5, "Преступление и наказание"),
]
chapters = [
    Chapter(1, "Слово автора", 350, 1),
    Chapter(2, "Примечания", 210, 2),
    Chapter(3, "Автобиография автора", 1100, 2),
    Chapter(4, "Реклама", 120, 3),
    Chapter(5, "Карта местности", 500, 3),
    Chapter(6, "Краткое описание", 2600, 3),
    Chapter(7, "Рекомендуемые книги к прочтению", 1200, 3),
    Chapter(8, "Описание редакции", 480, 3),
]
```

```python
chapters_books = [
    BookChapter(1, 1),
    BookChapter(1, 4),
    BookChapter(1, 5),
    BookChapter(1, 7),
    BookChapter(2, 1),
    BookChapter(2, 2),
    BookChapter(2, 3),
    BookChapter(2, 4),
    BookChapter(3, 2),
    BookChapter(3, 3),
    BookChapter(4, 2),
    BookChapter(4, 6),
    BookChapter(4, 8),
    BookChapter(5, 3),
    BookChapter(5, 8),
]
def main():
    one_to_many = [(c.title, c.wordcount, b.name)
                   for b in books
                   for c in chapters
                   if c.Book_id == b.id]
    many_to_many_temp = [(b.name, cb.books_id, cb.chapters_id)
                         for b in books
                         for cb in chapters_books
                         if b.id == cb.books_id]

    many_to_many = [(c.title, c.wordcount, books_name)
                    for books_name, books_id, chapters_id in many_to_many_temp
                    for c in chapters if c.id == chapters_id]
    print('Задание Д1')
    res1 = []
    for i in one_to_many:
        if i[0][-2:] == "pa":
            res1.append(i[0:3:2])
    print(res1)

    print('\nЗадание Д2')
    res2_unsorted = []
    for b in books:
        b_chapters = list(filter(lambda i: i[2] == b.name, one_to_many))
        if len(b_chapters) > 0:
            b_wordcount = [wordcount for _, wordcount, _ in b_chapters]
            b_wordcount_sum = sum(b_wordcount)
            b_wordcount_count = len(b_wordcount)
            b_wordcount_average = b_wordcount_sum / b_wordcount_count
```

```python
            res2_unsorted.append((b.name, int(b_wordcount_average)))
    res2 = sorted(res2_unsorted, key=itemgetter(1), reverse=True)
    print(res2)


    print('\nЗадание Д3')
    res3 = {}
    for b in books:
        if b.name[0] == "C":
            b_chapters = list(filter(lambda i: i[2] == b.name, many_to_many))
            b_chapters_wordcount = [x for x, _, _ in b_chapters]
            res3[b.name] = b_chapters_wordcount
    print(res3)


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

Результат:

C:\Users\USER\PycharmProjects\pythonProject1\venv\Scripts\python.exe
C:/Users/USER/PycharmProjects/pythonProject1/main.py

Задание Д1

[('Слово автора', 'Сталкер'), ('Автобиография автора', 'Метро 2033')]

Задание Д2

[('Пикник у обочины', 980), ('Метро 2033', 655), ('Сталкер', 350)]

Задание Д3

{'Сталкер': ['Слово автора', 'Реклама', 'Карта местности', 'Рекомендуемые книги к прочтению']}

Process finished with exit code 0

Run: main

C:\Users\USER\PycharmProjects\pythonProject1\venv\Scripts\python.exe C:/Users/USER/PycharmProjects/pythonProject1/main.py
Задание Д1
[('Слово автора', 'Сталкер'), ('Автобиография автора', 'Метро 2033')]

Задание Д2
[('Пикник у обочины', 980), ('Метро 2033', 655), ('Сталкер', 350)]

Задание Д3
{'Сталкер': ['Слово автора', 'Реклама', 'Карта местности', 'Рекомендуемые книги к прочтению']}

Process finished with exit code 0