**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

**Факультет прикладной математики-процессов управления**

**Программа бакалавриата**

**“Большие данные и распределенная цифровая платформа”**

**ОТЧЕТ**

**по лабораторной работе №4**

**по дисциплине «Функциональное программирование»**

**на тему «****Разработка рекомендательной системы для подбора книг»**

**Студент гр. 23Б16-пу**

**Аникин Р. С.**

**Преподаватель**

**Киямов Ж. У.**

**Санкт-Петербург**

**2024 г.**

**Оглавление**

[**Цель работы** 2](#_Toc185666014)

[**Описание задачи (формализация задачи)** 2](#_Toc185666015)

[**Теоретическая часть** 5](#_Toc185666016)

[1. Основы рекомендательных систем 5](#_Toc185666017)

[2. Представление пользовательских предпочтений 6](#_Toc185666018)

[3. Методика расчёта оценки соответствия 6](#_Toc185666019)

[**Основные шаги программы** 7](#_Toc185666020)

[**Описание программы** 9](#_Toc185666021)

[**Рекомендации пользователю** 10](#_Toc185666022)

[**Исходный код программы** 11](#_Toc185666023)

[**Контрольный пример** 12](#_Toc185666024)

[**Вывод** 13](#_Toc185666025)

[**Источники** 14](#_Toc185666026)

# **Цель работы**

Целью данной работы является разработка программного приложения, представляющего собой рекомендательную систему для выбора книг на основе предпочтений пользователя. Система должна анализировать введенные пользователем данные, такие как любимые жанры, авторы и ключевые слова, и на основе этих предпочтений предлагать пользователю список книг, которые могут быть ему интересны. Основное внимание уделяется использованию принципов функционального программирования, что позволит создать модульный, читаемый и легко расширяемый код.

# **Описание задачи (формализация задачи)**

Цель заключается в создании системы, которая учитывает предпочтения пользователя и предлагает подходящие книги на основе введённых данных. Система должна обладать следующими возможностями:

1. Ввод данных от пользователя

Система должна принимать от пользователя следующую информацию:

**Любимые жанры**: список интересующих жанров, например, "фентези", "комедия".

**Любимые авторы**: список предпочтительных авторов, например, "Джордж Оруэлл", "Джоан Роулинг".

**Ключевые слова**: термины, описывающие интересующие темы, например, "магия", "антиутопия".

2. База данных книг

Книги должны быть представлены в базе данных в формате JSON и включать следующую информацию:

Название книги.

Автор книги.

Жанр книги.

Краткое описание.

Год публикации.

3. Обработка пользовательских предпочтений

Система должна:

Формировать словарь предпочтений на основе введённых данных (жанры, авторы, ключевые слова).

Сравнивать предпочтения пользователя с метаданными книг в базе данных.

Оценивать степень соответствия каждой книги предпочтениям.

4. Алгоритм рекомендаций

Для генерации рекомендаций система должна:

Рассчитывать рейтинг соответствия каждой книги, учитывая совпадения по жанрам, авторам и ключевым словам.

Сортировать книги по убыванию рейтинга, чтобы наиболее подходящие варианты отображались первыми.

5. Фильтры и сортировка

Система должна предоставлять пользователю возможность:

**Фильтрации рекомендаций** по:

Жанрам.

Году публикации (например, только книги после 2000 года).

**Сортировки списка книг** по:

1. Рейтингу соответствия.
2. Алфавиту.
3. Году публикации.

6. Интерактивный интерфейс

1. Система должна поддерживать:
2. Добавление книг в список "Прочитать".
3. Сохранение рекомендаций в файл в формате JSON или CSV.

7. Модульное программирование

Ключевые функции системы должны быть реализованы в виде модулей, таких как:

1. **Фильтрация данных**.
2. **Оценка рейтинга книг**.
3. **Сортировка списка книг**.

Этот подход позволит улучшить читаемость и повторное использование кода, а также упростить его поддержку и расширение.

# **Теоретическая часть**

Персонализированные рекомендации играют ключевую роль в улучшении пользовательского опыта, особенно в таких сферах, как литература, кино, музыка и другие. Такие системы помогают пользователям открывать новые объекты, соответствующие их интересам, на основе анализа их поведения, предпочтений и характеристик предлагаемых объектов.

**1. Основы рекомендательных систем**

Рекомендательные системы можно разделить на несколько типов в зависимости от подхода к обработке данных:

* **Коллаборативная фильтрация**:  
  Основана на анализе поведения пользователей. Например, система определяет предпочтения пользователя, сравнивая их с предпочтениями других пользователей с похожими интересами. Этот подход применяется в сервисах вроде Netflix, Amazon и Spotify.
* **Контентная фильтрация**:  
  Анализируются характеристики объектов (например, жанры, авторы, ключевые слова), чтобы рекомендовать объекты, похожие на те, которые пользователь уже оценил положительно.
* **Гибридные системы**:  
  Объединяют оба подхода для повышения точности рекомендаций. Например, Netflix сочетает коллаборативную и контентную фильтрацию, чтобы предлагать фильмы и сериалы.

В данной задаче используется **контентная фильтрация**, которая основывается на характеристиках книг, таких как жанры, авторы и ключевые слова.

**2. Представление пользовательских предпочтений**

Одним из ключевых этапов в создании рекомендательной системы является корректное представление предпочтений пользователя. В контексте системы рекомендаций книг предпочтения пользователя могут включать:

* **Жанры книг**:  
  Например, фантастика, детективы, исторические романы.
* **Авторы**:  
  Писатели, чьи работы пользователь уже читал или предпочитает.
* **Ключевые слова**:  
  Термины из описаний книг, соответствующие интересам пользователя.

Эти предпочтения используются для расчёта оценки соответствия книги интересам пользователя. Оценка является количественной, что позволяет учитывать разные аспекты книги и их влияние на выбор.

**3. Методика расчёта оценки соответствия**

Оценка соответствия рассчитывается с использованием весовых коэффициентов для различных характеристик книги:

* **Жанр**:  
  Имеет высокий вес, так как жанр обычно является главным фактором при выборе книги.
* **Автор**:  
  Имеет средний вес, поскольку пользователи часто выбирают книги известных им авторов, но также готовы открывать для себя новых писателей.
* **Ключевые слова**:  
  Позволяют учитывать более специфические интересы пользователя, добавляя нюансы в рекомендацию.

Каждая из этих характеристик анализируется, и при совпадении с предпочтениями пользователя добавляет определённое количество баллов в общий результат. Итоговая оценка позволяет ранжировать книги по степени соответствия интересам пользователя.

# **Основные шаги программы**

1. **Загрузка и обработка данных о книгах**

Программа начинается с загрузки информации о книгах. Данные могут быть получены из различных источников (например, базы данных, текстовые файлы или API). Для каждой книги собираются следующие характеристики:

Название книги

Автор

Жанр

Ключевые слова

Год публикации

1. **Получение пользовательских предпочтений**

На следующем этапе программа собирает информацию о предпочтениях пользователя. Пользователь может выбрать из предложенного списка жанры, авторов, ввести ключевые слова, которые ему интересны. Эти предпочтения будут служить критериями для фильтрации книг.

Предпочтения пользователя могут включать:

**Жанры книг**

**Авторы**

**Ключевые слова**

1. **Оценка соответствия книг предпочтениям**

Программа вычисляет оценку соответствия каждой книги предпочтениям пользователя. Этот этап включает:

1. **Проверку жанра**: Если жанр книги совпадает с одним из предпочтений пользователя, то книга получает положительную оценку.
2. **Проверку автора**: Если автор книги присутствует в списке предпочтений пользователя, это также увеличивает оценку книги.
3. **Проверку ключевых слов**: Сравнение ключевых слов книги с интересами пользователя.

Оценка для каждой книги вычисляется с учётом весов различных факторов (например, жанр может иметь больший вес, чем ключевые слова). Это позволяет гибко настраивать рекомендации и учитывать приоритеты пользователя.

1. **Фильтрация и сортировка книг**

После того как для всех книг рассчитаны оценки, программа переходит к этапу фильтрации и сортировки:

**Фильтрация**: Книги, которые не соответствуют предпочтениям пользователя (или имеют низкую оценку), исключаются из списка.

**Сортировка**: Оставшиеся книги сортируются по убыванию оценки соответствия, чтобы вначале списка были те книги, которые наилучшим образом соответствуют предпочтениям пользователя.

Сортировка может включать дополнительные критерии, такие как год публикации или алфавитный порядок.

1. **Предоставление результатов пользователю**

После того как список книг отсортирован, программа выводит результаты пользователю. Это может быть выполнено через интерфейс, в котором отображаются названия книг, их авторы, жанры, а также оценка соответствия.

1. **Сохранение результатов**

Если пользователь хочет сохранить результат для дальнейшего использования, программа предоставляет возможность экспортировать список рекомендованных книг в формат json. Это удобно для тех пользователей, которые хотят запомнить или поделиться списком книг с другими.

# **Описание программы**

Программная реализация выполнена на языке Python 3.10 с использованием библиотеки tkinter для создания графического интерфейса и json для работы с json-файлами. Программа представляет собой систему рекомендаций книг, позволяющую пользователю выбирать предпочтительные жанры, авторов и ключевые слова для поиска. Она фильтрует и сортирует книги на основе заданных условий, а затем выводит список рекомендаций в виде таблицы, с возможностью сохранения в файл Excel.

Таблица 1. **book\_recommender.py**

|  |  |  |
| --- | --- | --- |
| **Функция** | **Описание** | **Возвращаемое значение** |
| Load\_books() | Загрузка книг | Не возвращает |
| load\_books\_from\_json() | Загружает базу данных книг из JSON-файла (вспомогательная) | Список словарей |
| process\_preferences() | Генерирует словарь предпочтений пользователя | Словарь |
| create\_widgets() | Отрисовка интерфейса | Не возвращает |
| Calculate\_rating() | Рассчитывает оценку соответствия книги предпочтениям пользователя | Целое число |
| recommend\_books() | Генерирует список книг с оценками соответствия | Список |
| update\_selected\_authors() | Обновляет строку выбранных авторов | Не возвращает |
| update\_author\_suggestions() | Обновляет список предложений авторов | Не возвращает |
| add\_to\_read\_list () | Добавление в список для чтения | Не возвращает |
| display\_recommendations() | Отображает список рекомендаций в виде таблицы | Не возвращает |
| apply\_filters\_and\_sort() | Применение фильтров к списку книг | Не возвращает |
| save\_recommendations\_to\_json () | Сохраняет выбранные книги в Excel-файл | Не возвращает |

## **Рекомендации пользователю**

Чтобы воспользоваться программой и получить рекомендации, выполните следующие действия:

1. **Выбор жанров**:

Перейдите в раздел "Жанры" и отметьте флажками интересующие вас категории. Допускается выбор нескольких жанров.

1. **Указание диапазона годов публикации**:

В разделе "Года" введите начальный и конечный годы, чтобы задать диапазон публикации книг. Используйте поля "Начиная с" и "До".

1. **Выбор авторов**:

В разделе "Авторы" начните вводить имя в поле поиска. Программа автоматически предложит подходящие варианты.

Выберите автора из предложенного списка, кликнув по его имени. Добавленные авторы отобразятся в поле "Выбранные авторы".

Чтобы удалить автора из списка, повторно кликните на его имя и подтвердите удаление.

1. **Добавление ключевых слов**:

Введите интересующие вас ключевые слова в поле "Ключевые слова", разделяя их запятыми. Программа будет учитывать их при анализе описаний книг.

1. **Фильтрация книг**:

Если вы введёте фильтр жанра, программа отберёт только книги из выбранного жанра.

1. **Настройка сортировки**:

Укажите критерий сортировки (например, по алфавиту или году публикации) и выберите порядок (по возрастанию или убыванию).

1. **Получение рекомендаций**:

Нажмите кнопку для генерации рекомендаций. Программа отобразит список книг в виде таблицы, упорядоченных по рейтингу соответствия вашим предпочтениям.

1. **Сохранение списка книг**:

Для сохранения результата в файл JSON выделите книги из таблицы и нажмите кнопку "Сохранить". Укажите место и имя файла для сохранения.

Следуя этим рекомендациям, вы сможете сформировать персонализированный список книг, максимально соответствующих вашим интересам.

# **Рекомендации программисту**

* Убедитесь, что установлены все необходимые (tkinter, json) библиотеки установлены.
* Проверьте, что JSON-файл с данными (books.json) существует и имеет корректный формат, подходящий для загрузки.
* Убедитесь, что все необходимые поля (например, "genre", "author", "first\_publish\_year", "description") присутствуют в данных JSON.

# **Исходный код программы**

[**https://github.com/RomanMuteki/func\_prog**](https://github.com/RomanMuteki/func_prog)

# **Контрольный пример**

1. Запуск программы: для запуска сервера используйте файл **recommender1.py.** Программа должна запустить интерфейс. (Рис. 1)

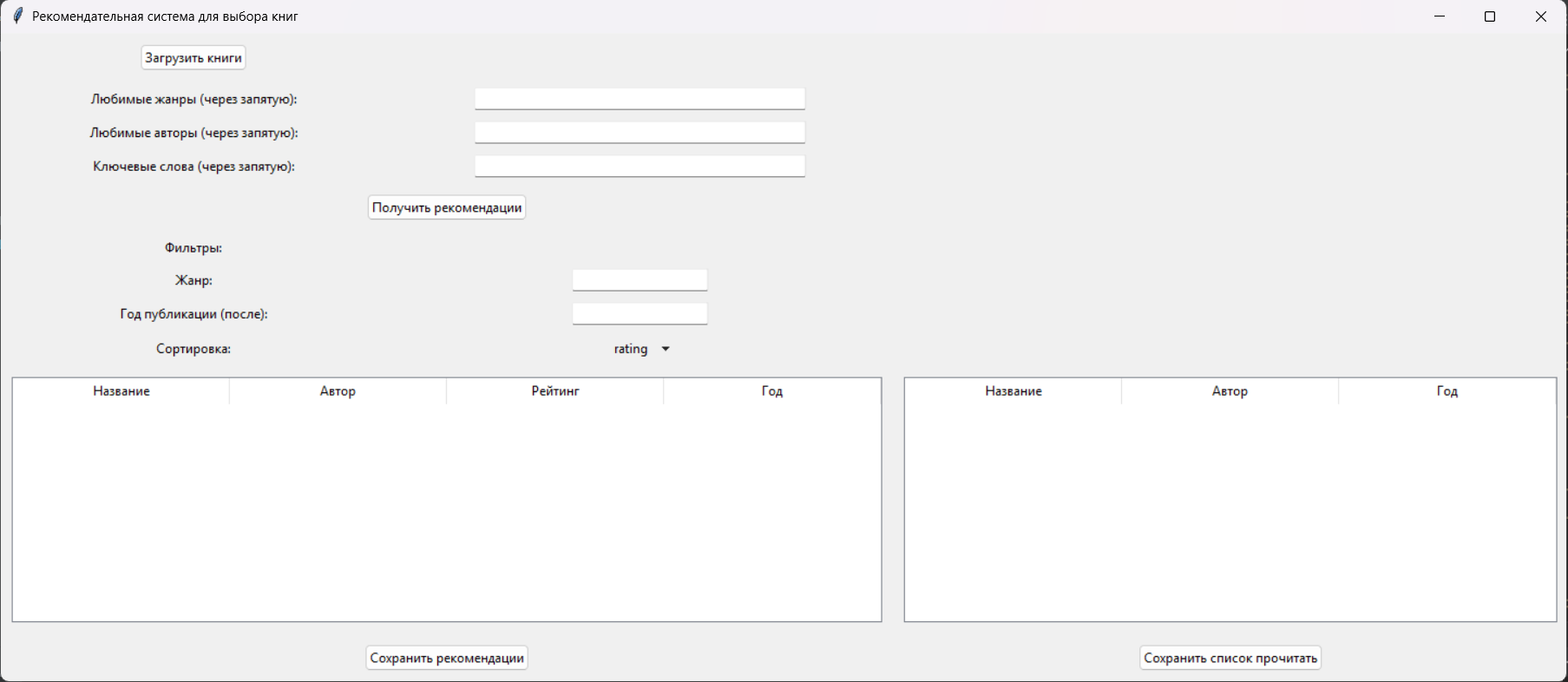


Рис. 1 Интерфейс программы

2. Выбор параметров: выберете подходящие для вас жанры книг. Введите года первой публикации, найдите подходящих авторов. Так же есть возможность ввести ключевые слова. (Рис.2)

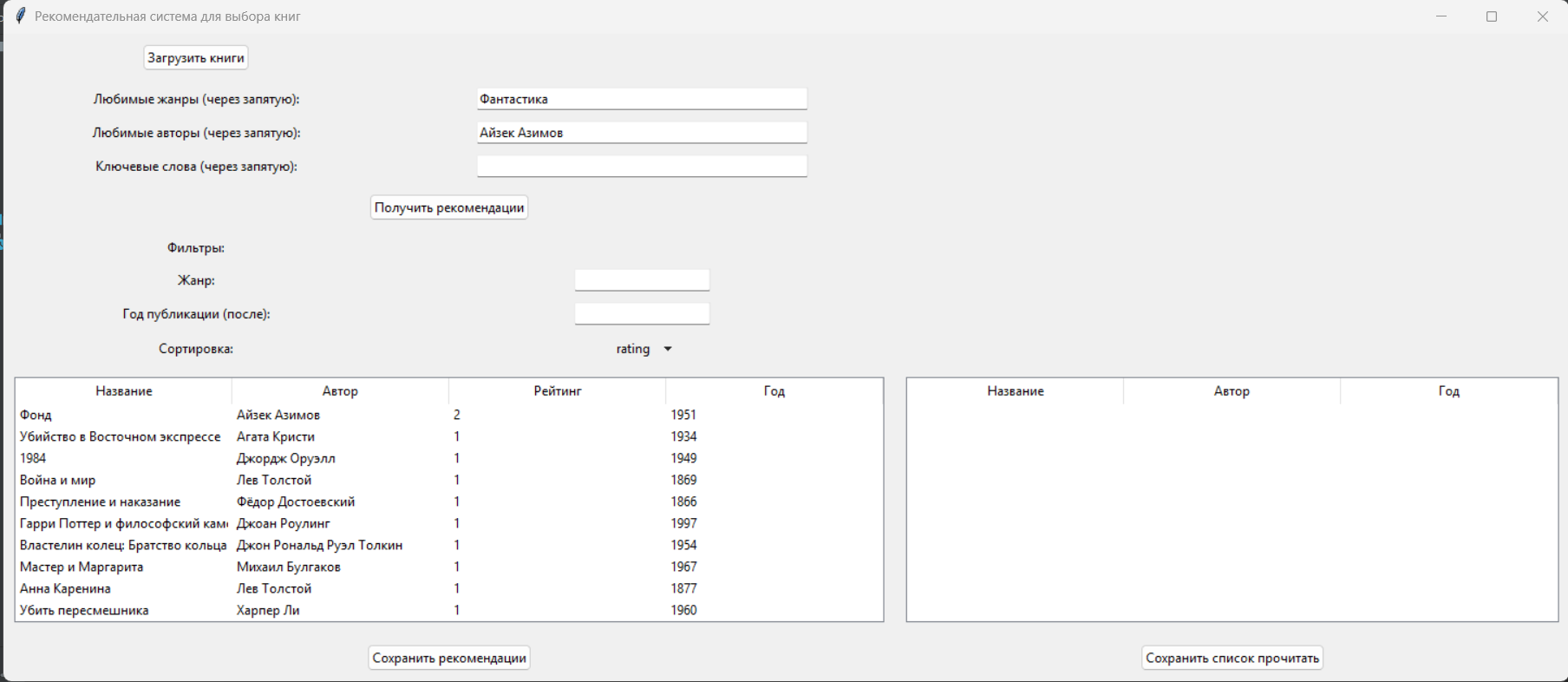


Рис. 2 Выбор параметров поиска

3. Получение рекомендаций: после нажатия: “Получить рекомендации” программа составит рейтинг соответствия, но основе введённых параметров и покажет результаты в таблице.

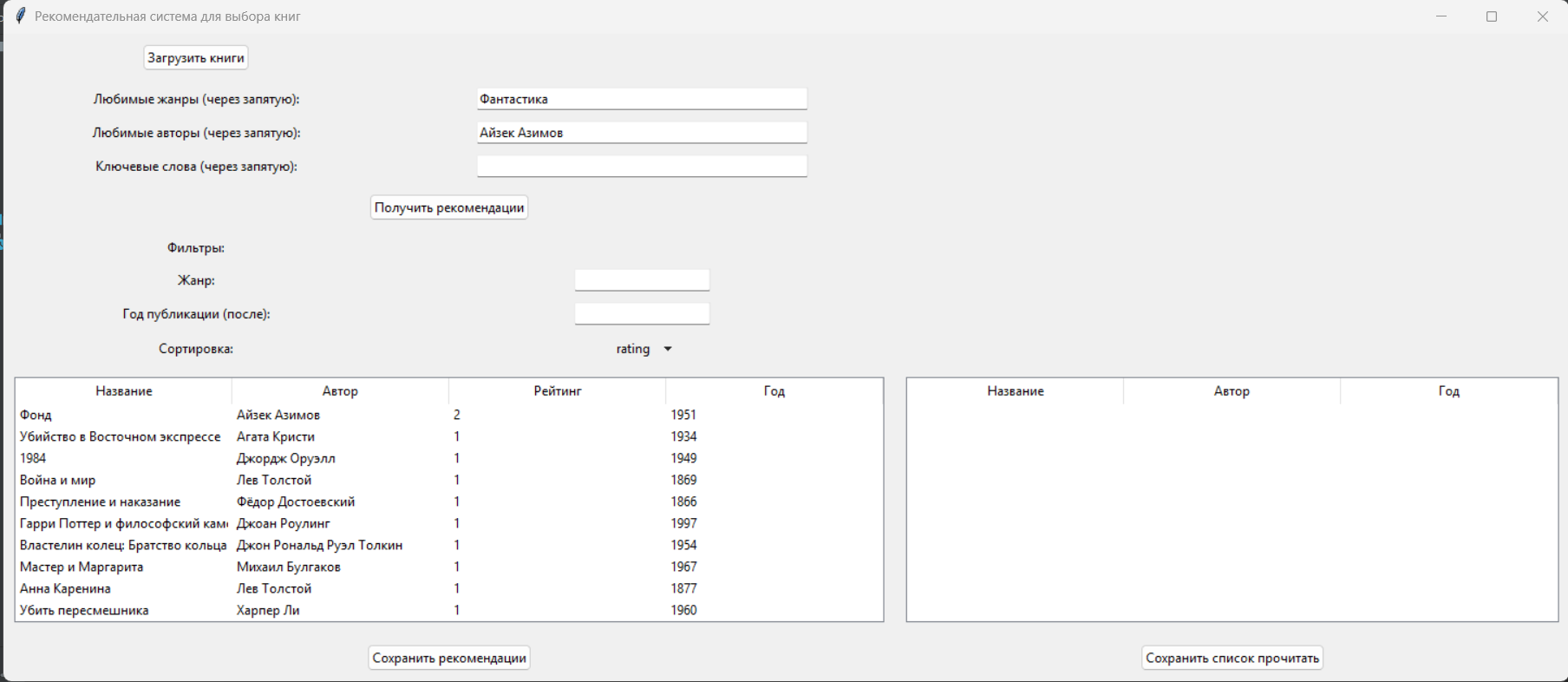


Рис. 3 Получение рекомендаций

4. Сохранение понравившихся: В таблице можно выбрать понравившиеся вам книги и нажатием на: “Сохранить список прочитать” сохранить их в удобном месте. (Рис. 4)

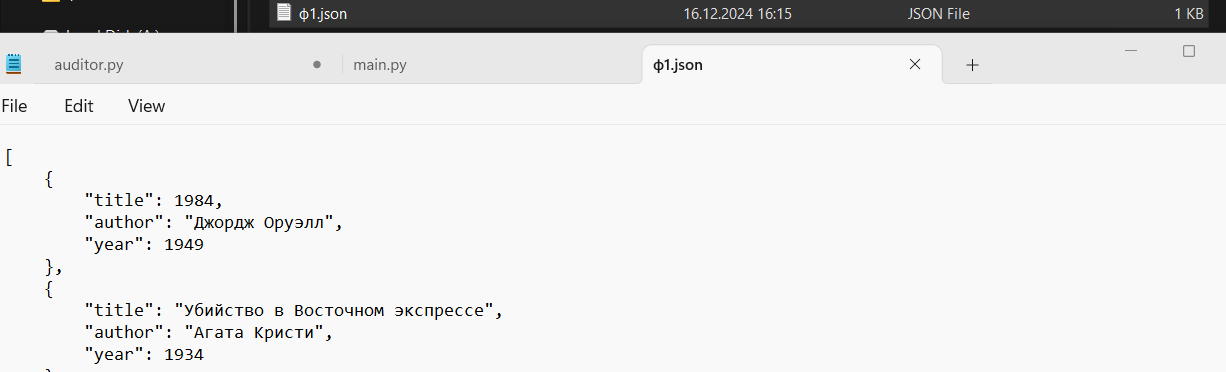


Рис. 4 Сохранение

# **Вывод**

В результате работы с программой пользователи могут удобно управлять данными о книгах, включая их загрузку, просмотр, фильтрацию, получение рекомендаций и сохранение новых записей. Все действия поддерживаются простым и интуитивно понятным интерфейсом, что облегчает работу и экономит время. Программа обеспечивает удобный способ поиска книг по заданным критериям и помогает находить новые произведения, которые могут быть интересны пользователю, на основе уже имеющихся данных.

В результате использования программы пользователи смогут:

* Быстро загружать и отображать книги.
* Эффективно фильтровать и находить книги по нужным параметрам.
* Получать рекомендации для расширения своей коллекции.
* Сохранять и загружать данные для дальнейшего использования.
* Обновлять данные и интерфейс для актуальности информации.

Эти возможности делают программу полезным инструментом для всех любителей книг и специалистов, работающих с библиотечными системами и базами данных.

# **Источники**

* **tkinter**

[*https://docs.python.org/3/library/tkinter.html/*](https://docs.python.org/3/library/tkinter.html/)

*дата обращения: (29.11.2024)*