**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 системы для выбора книг»**

**Студент гр. 23Б15-пу**

**Черевко М. Е.**

**Преподаватель**

**Киямов Ж. У.**

**Санкт-Петербург**

**2024 г.**

Оглавление

1. [Цель работы 3](#_Toc179492837)
2. [Описание задачи (формализация задачи) 4](#_Toc179492838)
3. [Теоретическая часть 6](#_Toc179492839)

[**1.** **Обработка изображений** 6](#_Toc179492840)

[**2. Классификация космических объектов** 6](#_Toc179492841)

[**3.** **Параллельные вычисления** 7](#_Toc179492842)

[**4.** **Методы анализа объектов на изображениях** 8](#_Toc179492843)

[**5.** **Хранение и визуализация результатов** 8](#_Toc179492844)

1. [Основные шаги программы 9](#_Toc179492845)
2. [Описание программы 12](#_Toc179492846)
3. [Рекомендации пользователя 14](#_Toc179492847)

[**Примечания:** 14](#_Toc179492848)

1. [Рекомендации программиста 15](#_Toc179492849)
2. [Исходный код программы 15](#_Toc179492850)
3. [Контрольный пример 16](#_Toc179492851)
4. [Вывод 18](#_Toc179492852)

# 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)

Задача заключается в разработке рекомендательной системы для выбора книг, которая будет учитывать предпочтения пользователей и предлагать книги на основе введенных данных. Система должна:

1. **Принимать входные данные от пользователя**, включая:
   * **Любимые жанры** — список жанров, которые наиболее интересуют пользователя (например, "фантастика", "детектив").
   * **Любимых авторов** — список авторов, чьи книги пользователь предпочитает (например, "Айзек Азимов", "Агата Кристи").
   * **Ключевые слова** — слова, связанные с интересующими темами (например, "искусственный интеллект", "расследование").
2. **Иметь базу данных книг**, представленную в формате JSON или CSV, с данными о каждой книге, включающими:
   * **Название книги**.
   * **Автор книги**.
   * **Жанр книги**.
   * **Краткое описание**.
   * **Год публикации**.
3. **Осуществлять обработку предпочтений пользователя**:
   * Формировать словарь предпочтений, основанный на введенных пользователем данных (жанры, авторы, ключевые слова).
   * Сравнивать введенные предпочтения с метаданными в базе данных и оценивать степень соответствия каждого элемента.
4. **Применять алгоритм рекомендаций**:
   * Для каждой книги из базы данных вычислять **рейтинг соответствия** на основе совпадений по жанрам, авторам и ключевым словам.
   * **Сортировать книги по убыванию рейтинга**, чтобы на первом месте выводились наиболее соответствующие предпочтениям пользователя.
5. **Реализовать фильтры и сортировку**:
   * Пользователь может фильтровать рекомендации по:
     + Жанрам.
     + Году публикации (например, только книги, выпущенные после 2000 года).
   * Пользователь может сортировать список книг по:
     + Рейтингу соответствия.
     + Алфавиту.
     + Году публикации.
6. **Обеспечить интерактивность интерфейса**:
   * Возможность добавления книг в **список "прочитать"**.
   * Возможность **сохранения списка рекомендаций** в файл в формате JSON или CSV.
7. **Реализовать модульное программирование**:
   * Все ключевые функциональные блоки должны быть реализованы через **модульные функции**, такие как:
     + Функции для **фильтрации** данных.
     + Функции для **оценки рейтинга** книг.
     + Функции для **сортировки** списка книг.

# Теоретическая часть

В современном мире персонализированные рекомендации становятся неотъемлемой частью пользовательского опыта, особенно в таких областях, как книги, фильмы, музыка и многие другие. Система рекомендаций позволяет пользователям находить новые объекты, которые соответствуют их интересам, на основе предыдущего поведения, предпочтений и характеристик объектов.

#### **1.** **Система рекомендаций: общий подход**

Системы рекомендаций делятся на несколько типов в зависимости от подхода к обработке данных:

* **Коллаборативная фильтрация** – основывается на анализе поведения пользователей, например, анализирует предпочтения пользователей, схожих с текущим. Этот подход широко используется в таких сервисах, как Netflix, Amazon и Spotify.
* **Контентная фильтрация** – анализирует характеристики объектов, такие как жанры, авторы, ключевые слова, и рекомендует объекты, которые схожи с теми, которые пользователь уже оценил положительно.
* **Гибридные системы** – комбинируют оба подхода для получения более точных рекомендаций. Например, Netflix использует как коллаборативную фильтрацию, так и контентную для предложений фильмов.

В данной задаче мы используем **контентную фильтрацию**, где рекомендации основаны на характеристиках книг, таких как жанр, автор, ключевые слова и другие метаданные.

#### **2. Представление предпочтений пользователя**

Основной задачей в рекомендательных системах является правильное представление предпочтений пользователя. В нашей системе предпочтения могут включать:

* **Жанры книг**, которые пользователю интересны (например, фантастика, детективы, исторические романы).
* **Авторы**, чьи книги пользователь уже читал или предпочитает.
* **Ключевые слова** из описаний книг, которые соответствуют интересам пользователя.

Эти предпочтения обрабатываются для создания **оценки соответствия**, которая показывает, насколько книга подходит под интересы пользователя. Важно отметить, что эта оценка не является бинарной (нравится/не нравится), а количественной, что позволяет гибко учитывать разные аспекты книги.

#### **3. Метод расчёта оценки соответствия**

Оценка соответствия книги предпочтениям пользователя рассчитывается с использованием весовых коэффициентов для различных характеристик книги. Например:

* **Жанр** книги может иметь высокий вес, так как часто жанр является основным критерием при выборе книги.
* **Автор** имеет средний вес, поскольку некоторые пользователи предпочитают книги определённых авторов, но могут быть открыты к новому.
* **Ключевые слова** – это более тонкая настройка, которая позволяет учитывать более специфические интересы пользователя.

Каждое из этих полей анализируется, и, если оно совпадает с предпочтениями пользователя, добавляется соответствующая сумма баллов в общий результат.

# 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

#### **2**. **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** (например, фантастика, детективы, исторические романы).
* **Авторы** книг, которых пользователь уже читал или хотел бы прочитать.
* **Ключевые слова** из описаний книг, связанные с интересами пользователя (например, "приключения", "любовь", "путешествия").

#### **3.** **Оценка соответствия книг предпочтениям**

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

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

#### **4.** **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.

#### **5.** **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.

#### **6.** **Сохранение результатов**

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

# Описание программы

Программная реализация выполнена на языке **Python 3.12.7** с использованием библиотеки **tkinter** для создания графического интерфейса и **openpyxl** для работы с **Excel-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 | Параметры функции | Описание параметров |
| load\_books | Загружает базу данных книг из JSON-файла | filename | Путь к файлу JSON с данными о книгах. По умолчанию "books.json". |
| process\_preferences | Генерирует словарь предпочтений пользователя | genres, authors, keywords | Списки жанров, авторов и ключевых слов, которые определяют предпочтения пользователя. |
| filter\_books | Фильтрует книги по жанру и году публикации | books, genre, year | Список книг и критерии фильтрации: жанр и год публикации. |
| calculate\_match | Рассчитывает оценку соответствия книги предпочтениям пользователя | book, preferences | Книга и словарь предпочтений пользователя. |
| recommend\_books | Генерирует список книг с оценками соответствия | books, preferences | Список книг и словарь предпочтений пользователя. |
| update\_selected\_authors | Обновляет строку выбранных авторов | selected\_authors, selected\_authors\_text | |  | | --- | |  |  |  | | --- | | Множество выбранных авторов и текстовая переменная для отображения на интерфейсе. | |
| update\_author\_suggestions | Обновляет список предложений авторов | authors, author\_search\_entry, suggestions\_frame, select\_author | Список всех авторов, поле поиска, фрейм для отображения предложений и функция выбора автора. |
| select\_author | Добавляет или удаляет автора из списка выбранных | author, selected\_authors, selected\_authors\_text, author\_search\_entry, update\_author\_suggestions, authors, suggestions\_frame | Автор, множества выбранных авторов, текстовая переменная, поле поиска, функции обновления. |
| show\_recommendations | Отображает список рекомендаций в виде таблицы | recommendations, results\_frame | Список книг с оценками и фрейм для отображения таблицы. |
| get\_recommendations | Собирает данные из формы, фильтрует книги и показывает рекомендации | books, genre\_vars, selected\_authors, keywords\_entry, year\_from\_entry, year\_to\_entry, sort\_option, sort\_order, only\_selected\_genres\_var, show\_recommendations, results\_frame | Список книг, словарь жанров, множество авторов, текстовое поле для ключевых слов, поля для годов, опции сортировки, флаг фильтрации по жанрам, функции обновления и отображения результатов. |
| save\_to\_read\_list | Сохраняет выбранные книги в Excel-файл | tree, max\_col\_width | Дерево с данными о книгах и максимальная ширина столбцов в файле. |

# Рекомендации пользователя

Для использования этой программы и получения результатов, следуйте следующим рекомендациям:

1. **Выбор жанров**:
   * Перейдите в раздел "Жанры" и установите флажки для жанров, которые вас интересуют. Вы можете выбрать несколько жанров.
2. **Выбор годов публикации**:
   * В разделе "Года" укажите диапазон лет, в котором должны быть опубликованы книги (начало и конец). Введите эти данные в поля "Начиная с" и "До".
3. **Выбор авторов**:
   * В разделе "Авторы" начните вводить имя автора в поле поиска. Программа предложит вам список совпадающих авторов.
   * Выберите автора из предложенного списка, кликнув по соответствующей кнопке. Выбранные авторы будут отображены в поле "Выбранные авторы".
   * Если вы хотите удалить автора из списка, выберите его снова, и программа предложит подтверждение удаления.
4. **Указание ключевых слов**:
   * Введите ключевые слова, которые должны быть учтены в описании книг, в поле "Ключевые слова", разделяя их запятой.
5. **Фильтрация книг**:
   * Если активирован флажок "Рекомендовать только указанные жанры", программа будет учитывать только книги из выбранных жанров при фильтрации.
6. **Сортировка результатов**:
   * Выберите параметр сортировки (по алфавиту или по году) и порядок сортировки (по возрастанию или убыванию).
7. **Получение рекомендаций**:
   * Нажмите кнопку для получения рекомендаций, и программа отобразит результаты в виде таблицы с книгами, отсортированными по рейтингу соответствия.
8. **Сохранение списка книг**:
   * Чтобы сохранить список выбранных книг в файл Excel, выберите книги в таблице и нажмите кнопку для сохранения. Укажите путь и имя файла для сохранения.

Следуя этим шагам, вы сможете получить список рекомендованных книг, соответствующих вашим предпочтениям.

# Рекомендации программиста

* **Установка необходимых библиотек**:
* Программа использует библиотеки Python такие как: **tkinter** и **openpyxl**. **tkinter** устанавливается автоматически с Python и не требуют дополнительной установки. А **openpyxl** необходимо устанавливать:



* Если вы работаете в окружении, где эти библиотеки отсутствуют (например, минимальная установка Python), убедитесь, что tkinter доступен. На **Ubuntu** и подобных системах его можно установить с помощью команды:



* **Проверка версии Python**:
* Код протестирован и совместим с Python 3.7 и выше, но рекомендуется использовать Python 3.10 или новее для наилучшей поддержки **asyncio**.
* Убедитесь, что версия Python установлена корректно, запустив:



* **Проверка базы данных**:
* 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/Kliooo/Functional-programming>

# Контрольный пример

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

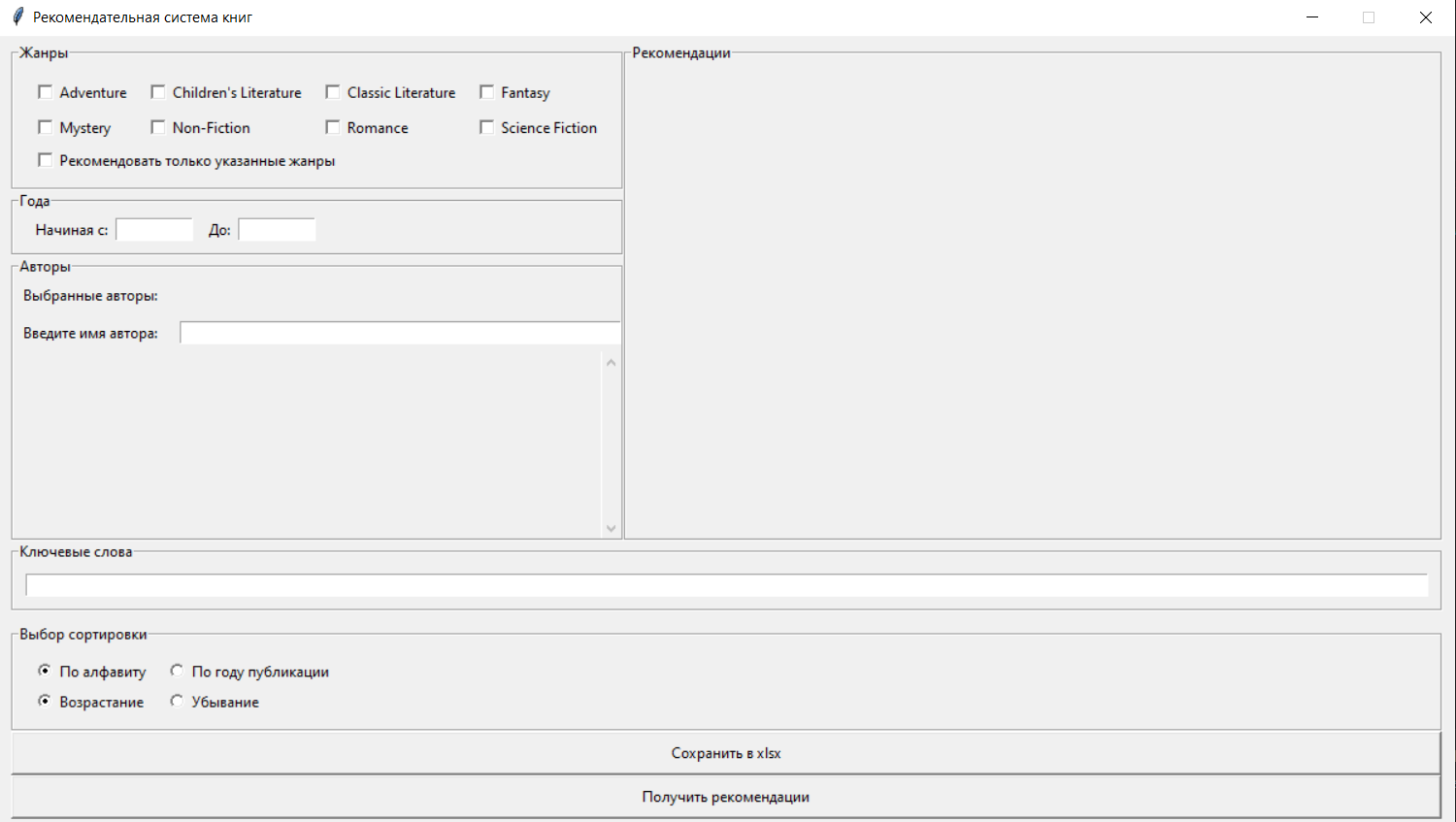


Рис. 1 Интерфейс программы

2. Выбор параметров: Выберете подходящие для вас жанры книг, укажите надо ли показывать книги других жанров в блоке. Введите года первой публикации, найдите подходящих авторов. Так же есть возможность ввести ключевые слова. (Рис.2)

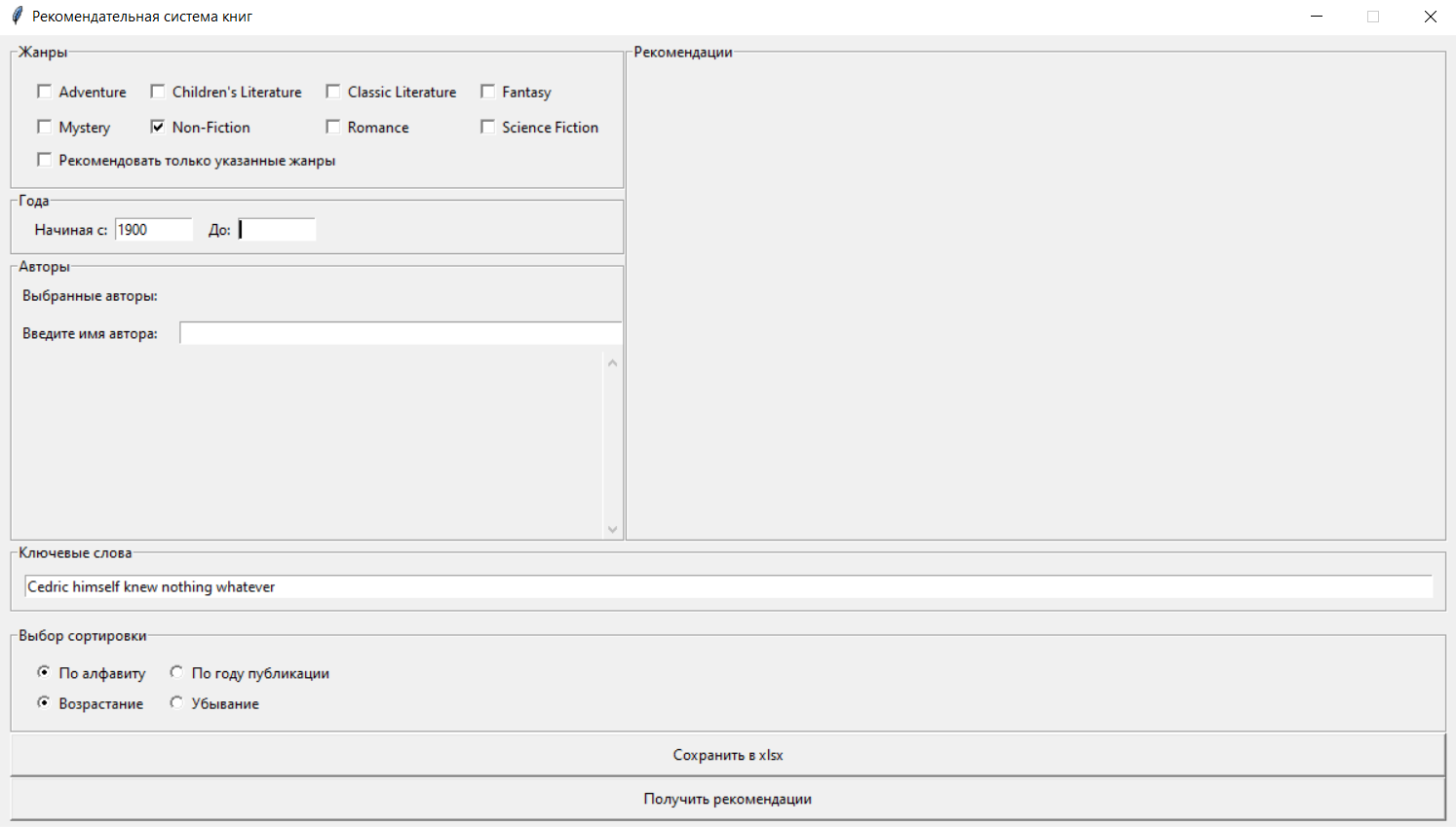


Рис. 2 Выбор параметров поиска

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

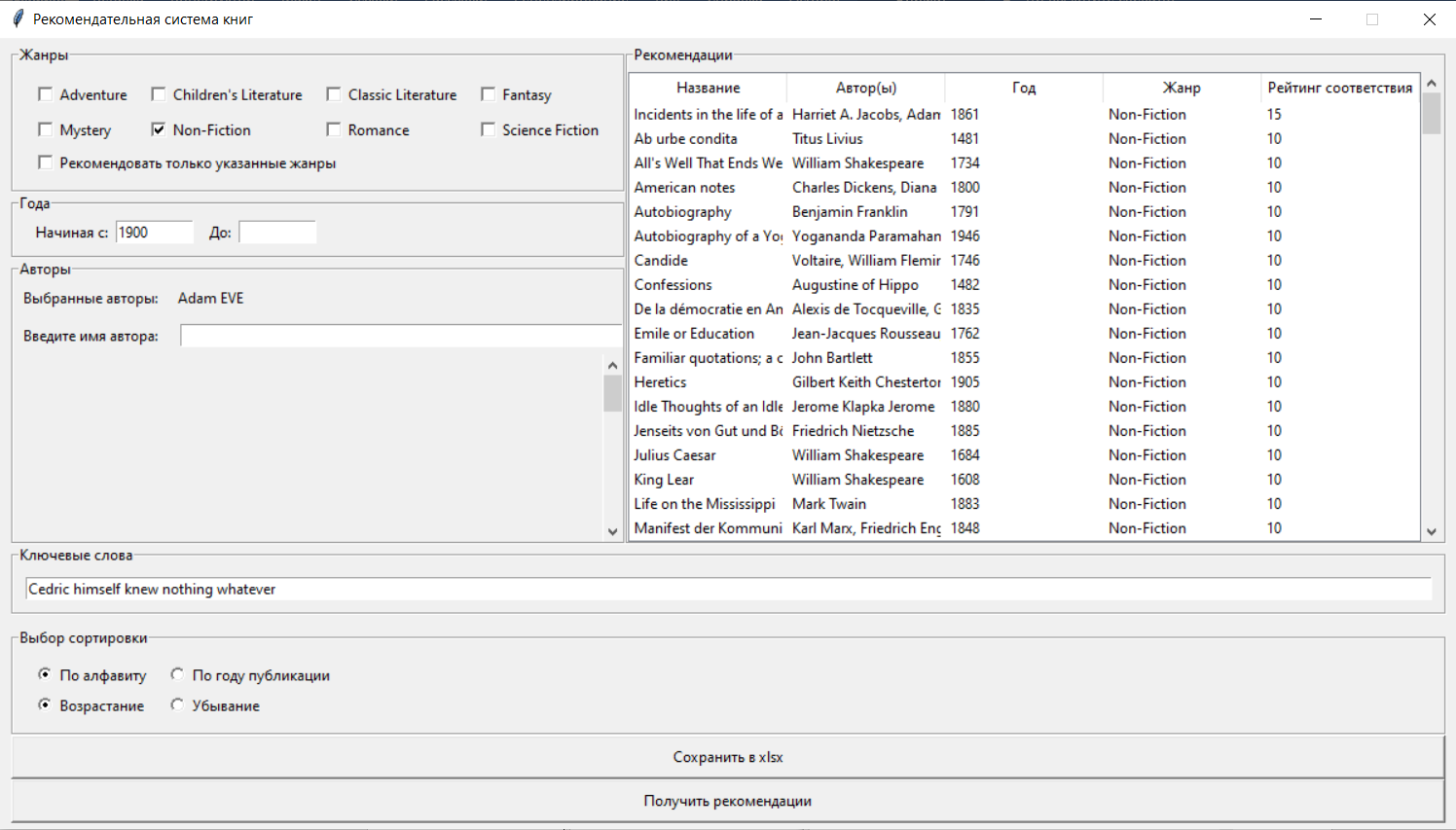


Рис. 3 Получение рекомендаций

4. Сохранение понравившихся: В таблице можно выбрать понравившиеся вам книги и нажатием на: “Сохранить в xlsx” сохранить их в удобный табличный формат. (Рис. 4)

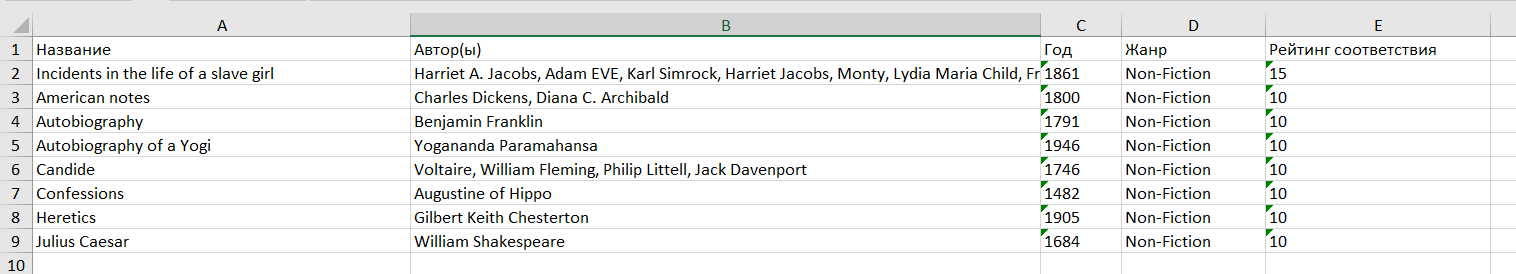


Рис. 4 Сохранение

# 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)*

* **OpenPyXL**

[*https://openpyxl.readthedocs.io/en/stable/*](https://openpyxl.readthedocs.io/en/stable/%20)

*дата обращения: (29.11.2024)*