**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 выбора книг на основе предпочтений пользователя»**

**Студент гр. 23Б15-пу**

**Абрахин Е.Д.**

**Преподаватель**

**Киямов Ж. У.**

**Санкт-Петербург**

**2024 г.**

Оглавление

1. [Цель работы 3](#_Toc183725671)
2. [Описание задачи (формализация задачи) 4](#_Toc183725672)
3. [Теоретическая часть 6](#_Toc183725673)
4. [Основные шаги программы 8](#_Toc183725674)
5. [Описание программы 10](#_Toc183725675)
6. [Рекомендации пользователя 11](#_Toc183725676)
7. [Рекомендации программиста 13](#_Toc183725677)
8. [Исходный код программы 13](#_Toc183725678)
9. [Контрольный пример 14](#_Toc183725679)
10. [Вывод 17](#_Toc183725680)
11. [Литература 18](#_Toc183725681)

# Цель работы

Целью данной работы является создание графического приложения для рекомендации книг на основе предпочтений пользователя. Приложение должно быть интерактивным, удобным в использовании и предоставлять функционал для ввода пользовательских данных, обработки рекомендаций и сохранения результатов.

# Описание задачи (формализация задачи)

Необходимо разработать приложение на Python с использованием библиотеки tkinter, которое будет предоставлять пользователю список рекомендованных книг на основе его предпочтений. Основные требования к разработке включают:

1. Создать графический интерфейс, позволяющий пользователю вводить:

* Любимые жанры.
* Любимых авторов.
* Ключевые слова.
* Год публикации (фильтр).

1. Добавить выпадающий список для выбора параметра сортировки (по рейтингу, алфавиту, году).
2. Разработать алгоритм оценки релевантности книг на основе:

* Жанров.
* Авторов.
* Ключевых слов в описании.

1. Предусмотреть фильтрацию по году публикации.
2. Реализовать сортировку результатов в зависимости от выбранного пользователем параметра.
3. Отображать список рекомендованных книг в виде таблицы с указанием:

* Названия книги.
* Автора.
* Жанра.
* Года публикации.
* Рейтинга (оценки релевантности).

1. Добавить функционал для сохранения списка рекомендаций в файл формата JSON.
2. Реализовать проверку корректности введенных данных (например, числовой формат для года).
3. Уведомлять пользователя о проблемах (например, отсутствие данных в файле или пустой результат рекомендаций).
4. Загрузить данные из файла books.json. Если файл отсутствует, уведомить пользователя об ошибке.
5. Использовать данные книг, содержащие поля:

* Название.
* Авторы.
* Жанр.
* Год публикации.
* Описание.

# Теоретическая часть

Асинхронный Программное обеспечение для рекомендаций книг представляет собой приложение с графическим интерфейсом, которое позволяет пользователям получать рекомендации на основе их предпочтений. В основе приложения лежат алгоритмы фильтрации и ранжирования данных, а также удобный инструмент для взаимодействия с пользователем через библиотеку tkinter. Основная цель заключается в создании эффективной и интуитивно понятной системы, которая предоставляет релевантные рекомендации книг и позволяет сохранять результаты в формате JSON.

Основы работы с рекомендационными системами

Рекомендационные системы — это программные инструменты, которые анализируют предпочтения пользователей для предоставления наиболее релевантных предложений. В данной системе пользователь вводит такие параметры, как:

* Предпочитаемые жанры.
* Любимые авторы.
* Ключевые слова, связанные с книгами.
* Ограничения по году публикации.

На основе введенных данных система оценивает релевантность каждой книги и предлагает список с учетом выбранного пользователем критерия сортировки (рейтинг, алфавитный порядок, год).

Алгоритмы фильтрации и оценки релевантности

Для оценки релевантности используется система подсчета очков (score), основанная на совпадениях:

* Жанры имеют наибольший вес, так как они ключевым образом определяют интерес пользователя.
* Совпадения по авторам добавляют меньшее количество очков.
* Ключевые слова в описании книг повышают вероятность их включения в список рекомендаций.

Итоговая оценка книги формируется как сумма очков за все совпадения.

Архитектура приложения

Приложение состоит из трех основных компонентов:

1. Графический интерфейс пользователя (GUI)

* Создается с использованием библиотеки tkinter.
* Содержит поля для ввода предпочтений пользователя, выбора параметров сортировки и фильтров.
* Предоставляет таблицу для отображения результатов.

1. Обработка данных

* Загрузка данных о книгах из файла books.json.
* Алгоритмы фильтрации и ранжирования, реализованные на основе пользовательских предпочтений.
* Сортировка списка рекомендаций по выбранному критерию.

Сохранение результатов

* Реализована возможность сохранения рекомендаций в формате JSON для последующего использования.

Принципы взаимодействия приложения с пользователем

* Пользователь вводит свои предпочтения (жанры, авторы, ключевые слова, год) и выбирает способ сортировки.
* Алгоритм анализирует данные книг и формирует список с учетом предпочтений и фильтров.
* Результаты отображаются в виде таблицы. Пользователь может сохранить их в файл для последующего использования.

# Основные шаги программы

Загрузка данных

* При запуске программы загружается файл books.json, содержащий данные о книгах.
* Если файл отсутствует, выводится сообщение об ошибке, а список книг остается пустым.

Графический интерфейс

1. Поля ввода:

* Жанры, авторы, ключевые слова.
* Фильтр по году публикации.
* Выбор метода сортировки (рейтинг, алфавитный порядок, год).

1. Кнопки:

* "Показать рекомендации" — запускает расчет и отображение списка книг.
* "Сохранить рекомендации" — сохраняет текущие рекомендации в файл JSON.

1. Таблица:

* Отображает список книг с информацией о названии, авторе, жанре, годе и рейтинге.

Расчет рекомендаций

1. Пользователь вводит предпочтения: жанры, авторов, ключевые слова, год публикации и выбирает метод сортировки.
2. Программа фильтрует книги:

* Исключает книги, опубликованные раньше указанного года (если установлен фильтр).
* Вычисляет "оценку релевантности" для каждой книги:
* +3 очка за совпадение по жанру.
* +2 очка за совпадение по автору.
* +1 очко за совпадение ключевого слова в описании книги.

1. После формирования списка книги сортируются:

* По убыванию оценки (по умолчанию).
* По алфавиту названия книги.
* По году публикации.

Отображение результатов

* Список рекомендаций отображается в таблице.
* Таблица автоматически обновляется после каждого расчета.

Сохранение рекомендаций

* Пользователь может сохранить текущий список рекомендаций в файл JSON:
* При выборе файла сохраняются все книги из таблицы.
* Если список пуст, программа выводит предупреждение.

# Описание программы

Программа представляет собой графическое приложение для формирования рекомендаций книг, основанное на пользовательских предпочтениях. Приложение реализовано на языке Python 3.12.6 с использованием библиотек json, tkinter, и ttk. Основная задача программы — предложить список книг, наиболее соответствующих заданным критериям: жанрам, авторам, ключевым словам и году публикации. Программа поддерживает сортировку результатов и возможность сохранения рекомендаций в файл.

Таблица 1. BookRecommender.py

|  |  |  |  |
| --- | --- | --- | --- |
| Функция | Описание | Аргументы | Возвращаемое значение |
| calculateScore | Вычисляет оценку релевантности книги на основе жанра, авторов и ключевых слов. | book (словарь с данными о книге), genres (список жанров), authors (список авторов), keywords (список ключевых слов) | Целое число (оценка) |
| recommendBooks | Формирует список рекомендаций на основе пользовательских предпочтений и фильтров. | genres , authors, keywords, yearFilter, sortBy (параметр сортировки) | Список словарей (рекомендации) |
| showRecommendations | Формирует и отображает список рекомендаций книг в таблице интерфейса. | Без аргументов | None |
| saveRecommendations | Сохраняет текущий список рекомендаций из таблицы в файл формата JSON. | Без аргументов | None |

# Рекомендации пользователя

Программа представляет собой графический интерфейс для формирования рекомендаций книг на основе предпочтений пользователя. Ниже приведены инструкции по работе с программой:

1. Загрузка данных о книгах

Перед началом работы убедитесь, что файл books.json находится в том же каталоге, где запускается программа. Этот файл должен содержать данные о книгах в формате JSON.

Если файл отсутствует, программа выдаст сообщение об ошибке и продолжит работу с пустым списком.

1. Ввод предпочтений пользователя

После запуска программы вы сможете ввести ваши предпочтения в соответствующие поля:

* Жанры: укажите ваши любимые жанры книг через запятую. Например, Фантастика, Детектив.
* Авторы: перечислите предпочитаемых авторов через запятую. Например, Толстой, Достоевский.
* Ключевые слова: введите слова или фразы, которые должны содержаться в описании книг, через запятую.
* Год публикации: укажите минимальный год публикации книги для фильтрации. Например, 2000. Если оставить поле пустым, фильтр по году применяться не будет.

Сортировка: выберите метод сортировки результатов:

* Рейтинг: по убыванию релевантности.
* Алфавит: по названию книги.
* Год: по году публикации.

3. Получение рекомендаций

После ввода предпочтений нажмите кнопку "Показать рекомендации".

Программа отобразит список книг, соответствующих заданным критериям, в таблице.

* Каждая книга будет содержать следующую информацию:
* Название.
* Автора(ов).
* Жанр.
* Год публикации.
* Рейтинг (оценка релевантности).

4. Сохранение рекомендаций

Нажмите кнопку "Сохранить рекомендации", укажите имя и путь для сохранения файла в появившемся диалоговом окне и программа сохранит файл с вашими рекомендациями.

Если список рекомендаций пуст, программа сообщит об этом.

5. Ошибки и их обработка

Если файл books.json отсутствует, программа отобразит сообщение: "Файл books.json не найден".

Если в поле "Год публикации" введено некорректное значение, программа выдаст сообщение: "Год должен быть числом".

# Рекомендации программиста

Для успешного запуска программы убедитесь, что у вас установлен Python версии не ниже 3.12.0 и необходимые библиотеки, такие как json и tkinter.

Поддерживайте актуальные версии используемых библиотек и Python для обеспечения работоспособности программы на современных системах. Следуйте передовым практикам разработки, уделяя внимание четкому именованию переменных и функций, чтобы улучшить читаемость и понимание кода.

Регулярно проводите тестирование программы на различных входных данных, чтобы убедиться в ее надежности и корректности работы. Рассмотрите возможность добавления логирования для отслеживания процесса выполнения и отладки. Используйте комментарии для пояснения сложных участков кода и логики программы.

# Исходный код программы

<https://github.com/FasterXaos/Functional_programming>

# Контрольный пример

1. Запуск программы

Для начала запустите файл main.py. Откроется графическое окно, где вам будет предложено ввести ваши предпочтения (жанры, авторы, ключевые слова, год публикации). После ввода предпочтений нажмите кнопку "Показать рекомендации".

Программа отобразит список книг, соответствующих вашим критериям, в таблице с колонками: название, автор, жанр, год публикации и рейтинг (Рис. 1).

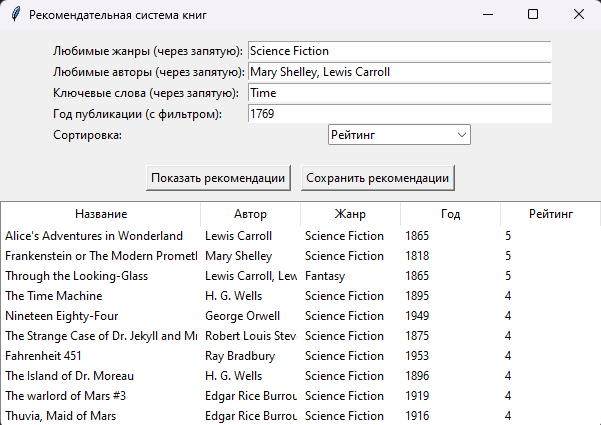


Рис. 1. Пример отображения рекомендаций в таблице

1. Настройка предпочтений

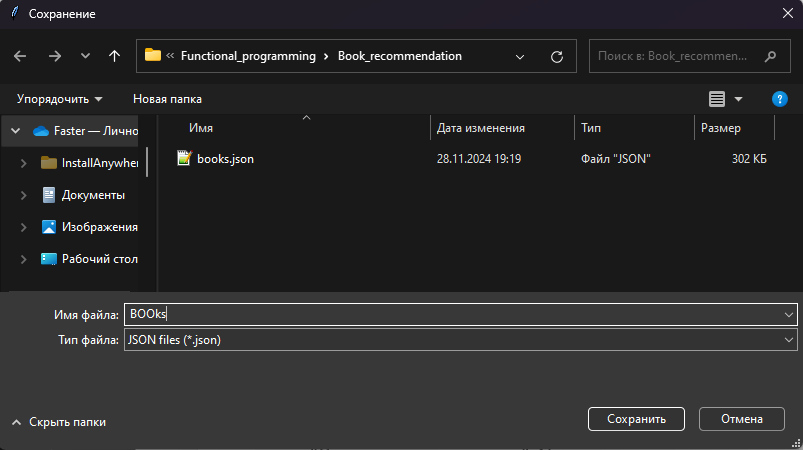
Введите ваши предпочтения в соответствующие поля:

* Жанры: перечислите любимые жанры через запятую.
* Авторы: укажите авторов.
* Ключевые слова: добавьте ключевые слова для поиска.
* Год публикации: укажите год фильтрации.
* Сортировка: выберите метод сортировки (по рейтингу, по алфавиту или по году).

После заполнения полей нажмите "Показать рекомендации" для получения списка книг.

1. Сохранение рекомендаций

Для сохранения списка рекомендаций в файл нажмите кнопку "Сохранить рекомендации". Укажите путь для сохранения и выберите имя файла (Рис. 2).

Рис. 2. Пример диалогового окна для сохранения файла

# Вывод

В ходе работы была разработана система рекомендаций книг, которая позволяет пользователям получать персонализированные рекомендации на основе их предпочтений, таких как жанры, авторы и ключевые слова. Основной целью было создание эффективного инструмента для подбора книг с возможностью фильтрации по году публикации и сортировки по различным критериям.

В процессе разработки была реализована функция подсчета рейтинга каждой книги на основе введенных пользователем предпочтений, что позволило формировать список книг с наивысшими баллами. Также была предусмотрена возможность фильтрации книг по году и сортировки по рейтингам, алфавиту или году публикации. Программа использует библиотеки Python, такие как tkinter для графического интерфейса и json для работы с данными, что обеспечило удобство взаимодействия с пользователем и гибкость в обработке данных.

Созданная система не только помогает пользователям находить книги, соответствующие их интересам, но и позволяет сохранить рекомендации в отдельный файл для дальнейшего использования.

# Литература

1. json — JSON encoder and decoder // docs.python.org URL: https://docs.python.org/3/library/asyncio.html (дата обращения: 28.11.2024).
2. tkinter — Python interface to Tcl/Tk // docs.python.org URL: <https://docs.python.org/3/library/tkinter.html> (дата обращения: 28.11.2024).