

**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

**Факультет прикладной математики-процессов управления**

**Программа бакалавриата**

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

**ОТЧЕТ**

**по лабораторной работе №3**

**по дисциплине «Функциональное программирование»**

**на тему**

**«Разработка рекомендательной системы книг»**

**Студент гр. 23Б15-пу  
Трофимов И.А.**

**Преподаватель  
Киямов Ж. У.**

**Санкт-Петербург  
2024 г.**

## Оглавление

1. Цель работы	3
2. Описание задачи (формализация задачи)	3
3. Теоретическая часть	4
4. Основные шаги программы	7
5. Описание программы	8
6. Рекомендации пользователя	11
7. Рекомендации программиста	11
8. Исходный код программы	11
9. Контрольный пример	12
10. Вывод	13
11. Источники	14

## **Цель работы**

Разработать программное приложение, реализующее рекомендательную систему для выбора книг на основе предпочтений пользователя, с применением принципов функционального программирования. Система должна анализировать предпочтения по жанрам, авторам и ключевым словам, предоставляя персонализированные рекомендации.

## **Описание задачи**

Рекомендательная система для выбора книг предоставляет пользователю список книг, которые соответствуют его предпочтениям по жанрам, авторам и ключевым словам. Для реализации программы необходимо создать базу данных книг, разработать алгоритм сопоставления данных пользователя с метаданными книг и обеспечить удобный способ взаимодействия с системой.

Основные этапы разработки:

### **1. Форма ввода данных пользователя:**

- Пользователь указывает любимые жанры, авторов и ключевые слова.

### **2. Создание базы данных книг:**

- Разработать набор книг с указанием названия, автора, жанра, описания и года выпуска в формате JSON или CSV.

### **3. Алгоритм рекомендаций:**

- Сопоставить предпочтения пользователя с базой данных.

- Рассчитать рейтинг соответствия для каждой книги и отсортировать результаты.

#### **4. Фильтры и сортировка:**

- Реализовать фильтры по жанрам и году выпуска, а также сортировку по рейтингу, алфавиту или году публикации.

#### **5. Интерактивность:**

- Позволить пользователю добавлять понравившиеся книги в "список прочитать".
- Сохранение рекомендаций в файл (CSV или JSON).

#### **6. Простой интерфейс (опционально):**

- Веб-приложение для ввода данных и просмотра рекомендаций.

#### **7. Соответствие нефункциональным требованиям:**

- Использование функционального программирования, кроссплатформенность и обработка ошибок.

Эти шаги обеспечат разработку рабочей и удобной системы для персонализированного подбора книг.

### **Теоретическая часть**

Рекомендательные системы — это программные приложения, которые помогают пользователям находить наиболее релевантные объекты (например, книги, фильмы или товары) на основе анализа их предпочтений. Такие системы широко применяются в электронной коммерции, онлайн-сервисах и библиотеках.

## Основные аспекты разработки рекомендательной системы:

### 1. Сбор данных о предпочтениях пользователя:

Пользователь вводит данные, такие как любимые жанры, авторы и ключевые слова. Эти данные представляют собой параметры для фильтрации и ранжирования книг в системе.

### 2. Структура базы данных книг:

База данных содержит список книг с описанием их характеристик (название, автор, жанр, описание, год выпуска). Для удобства обработки данные хранятся в формате JSON или CSV, что упрощает их чтение, обработку и модификацию.

### 3. Алгоритмы рекомендаций:

Основной задачей алгоритма является вычисление "рейтинга соответствия" каждой книги на основе введенных данных пользователя. Используются следующие критерии:

- Совпадение жанра.
- Совпадение автора.
- Наличие ключевых слов в описании книги.

Итоговый рейтинг формируется с учётом веса каждого критерия.

### 4. Фильтрация и сортировка данных:

Для повышения удобства работы с системой используются фильтры и сортировка. Пользователь может ограничить список рекомендаций по жанрам или году выпуска, а также отсортировать результаты по алфавиту, рейтингу соответствия или году публикации.

## 5. Функциональное программирование:

Применение принципов функционального программирования в системе обеспечивает лаконичность и модульность кода. Основные принципы:

- Использование чистых функций, не изменяющих состояние программы.
- Применение генераторов для работы с большими объёмами данных.
- Композиция функций для последовательной обработки данных.

## 6. Интерфейс взаимодействия с пользователем:

Для удобства работы с системой создаётся текстовый или веб-интерфейс. Пользователь вводит свои предпочтения, получает список книг и может сохранить его в файл для дальнейшего использования.

Преимущества рекомендательных систем:

- Повышение удовлетворённости пользователей за счёт персонализации рекомендаций.
- Удобство выбора книг на основе анализа предпочтений.
- Возможность масштабирования системы для добавления новых книг или параметров фильтрации.

Разработка такой системы требует сочетания алгоритмов обработки данных, модульного программирования и пользовательского интерфейса.

## Основные шаги программы

### 1. Инициализация и загрузка данных:

- Программа загружает базу данных книг из файла JSON (BOOKS\_DB\_PATH).
- Выполняется обработка данных для выделения уникальных жанров и авторов.

### 2. Графический интерфейс (GUI):

- Создается главное окно приложения с помощью библиотеки Tkinter.
- Интерфейс включает:
  - Поля ввода для авторов, ключевых слов, минимального года публикации и рейтинга.
  - Список для выбора жанров (с поддержкой множественного выбора).
  - Таблицу для отображения рекомендованных книг.
  - Таблицу для добавления избранных книг.

### 3. Функциональность интерфейса:

- **Автодополнение:** Подсказки при вводе имени автора в поле.
- **Кнопки управления:**
  - Кнопка "Сгенерировать рекомендации" вызывает функцию, которая на основе пользовательских предпочтений создает список рекомендаций.
  - Кнопка "Сохранить рекомендации" сохраняет текущий список рекомендаций в файл CSV.
  - Кнопка "Добавить в избранное" добавляет выбранные книги из таблицы рекомендаций в таблицу избранного.

### 4. Рекомендации книг:

- Пользователь задает предпочтения (авторы, ключевые слова, минимальный год, рейтинг, жанры).
- Программа рассчитывает "рейтинг" каждой книги, учитывая:
  - Совпадение жанров, авторов, ключевых слов.

■ Год публикации.

- Книги с минимальным рейтингом выше заданного значения добавляются в список рекомендаций, сортируются и отображаются в таблице.

**5. Сортировка данных:**

- Книги можно сортировать по названию, году публикации или рейтингу.

**6. Работа с избранным:**

- Пользователь может добавить книги из таблицы рекомендаций в избранное.
- В таблице избранного отображаются только выбранные книги.

**7. Сохранение рекомендаций:**

- Рекомендации можно сохранить в CSV-файл через диалоговое окно.

**8. Обработка ошибок:**

- Если данные не загружаются или возникают проблемы с сохранением файла, пользователю показывается сообщение об ошибке.

**9. Запуск основного цикла:**

- Программа запускает `root.mainloop()`, чтобы GUI оставался активным и реагировал на пользовательские действия.

## **Описание программы**

Программная реализация выполнена на языке Python 3.12.7 с использованием библиотек **asyncio**[\[1\]](#) для организации асинхронной связи между клиентом и сервером, а также **tkinter**[\[2\]](#) для создания графического интерфейса клиента. Серверная часть организована в виде асинхронного TCP-сервера, который обрабатывает подключения клиентов, управляет



чат-комнатами, пересылает сообщения и обновляет список активных пользователей в комнатах. Клиентская часть реализует подключение к серверу и обеспечивает удобный интерфейс для отправки и получения сообщений в режиме реального времени.

Таблица 1. main.py

Функция	Описание	Результат
load_books	загружает данные из файла JSON	None
extract_unique_fields	извлекает уникальные жанры и авторов из базы данных	None
recommend_books	формирует список книг, подходящих под предпочтения пользователя, и рассчитывает их рейтинг	None

update_table	обновляет таблицу рекомендаций в интерфейсе	None
save_recommendations	сохраняет текущий список рекомендаций в файл CSV	None
add_to_favorites	добавляет книгу в таблицу избранного	None
generate_recommendations	запускает процесс фильтрации и формирования рекомендаций на основе предпочтений	None
update_autocomplete	обрабатывает ввод для поля автодополнения	None

## **Рекомендации пользователя**

Перед запуском программы убедитесь, что на вашем компьютере установлен Python и необходимые библиотеки, такие как tkinter[2], которые используются для создания графического интерфейса. Программу можно запустить через файл client.py, открыв командную строку или среду разработки и выполнив команду python client.py. После запуска вам будет предложено настроить параметры, такие как жанры, авторы, ключевые слова, минимальный год публикации и минимальный рейтинг для генерации рекомендаций.

## **Рекомендации программиста**

Чтобы поддерживать программу актуальной и функциональной, важно регулярно обновлять используемые библиотеки, особенно tkinter[2], до последних версий. Следует соблюдать лучшие практики при именовании переменных и функций, чтобы улучшить читаемость и поддерживаемость кода. Важно также использовать обработку ошибок, чтобы программа не падала в случае некорректных данных или ошибок чтения/записи файлов. При увеличении объема данных следует рассмотреть оптимизацию работы с ними, например, через кэширование или создание индексов для ускорения поиска. Не забывайте документировать код и оставлять комментарии, чтобы другим разработчикам было проще разобраться в логике программы.

## **Исходный код программы:**

[https://github.com/hanglider/Jasur-Labs/tree/main/fp/books\\_system](https://github.com/hanglider/Jasur-Labs/tree/main/fp/books_system)

# Контрольный пример

Запустите main.py (Рис. 1)

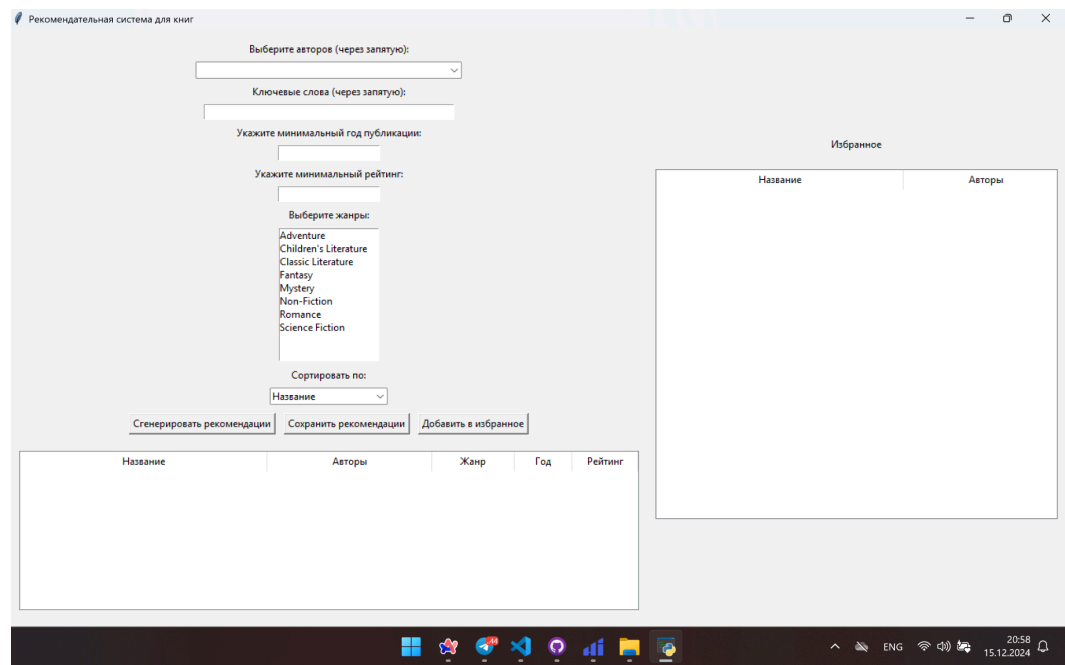


Рис. 1 Рабочая область

Выберите автора, ключевые слова, минимальный год публикации и жанр книги, затем нажмите кнопку сгенерировать рекомендации (Рис.2)

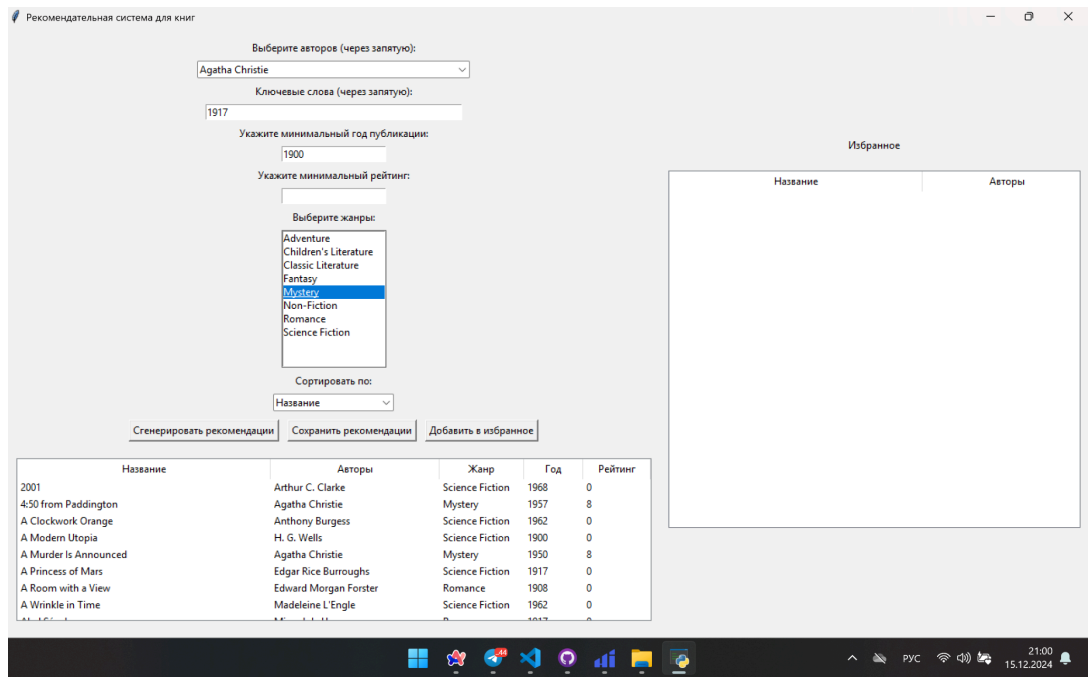


Рис.2 Рабочая область

Также вы можете выбрать книгу и сохранить ее в избранное (Рис. 3)

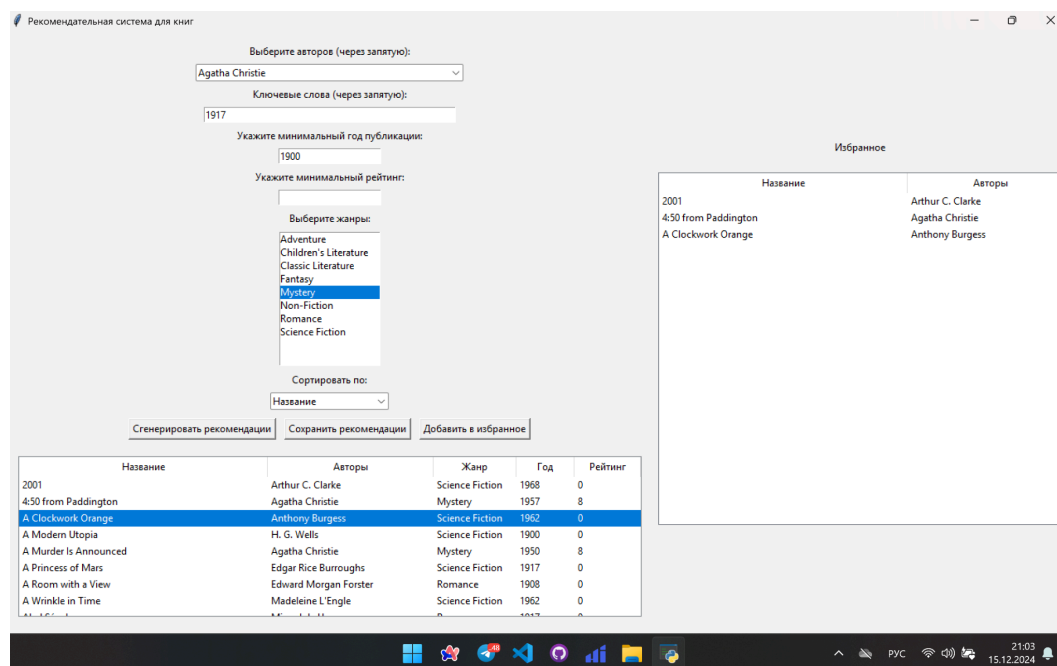


Рис.3 Сохранение в избранное

## Вывод

В ходе выполнения лабораторной работы была разработана и реализована рекомендательная система для книг с использованием Python и библиотеки tkinter для создания графического интерфейса. Программа позволяет пользователю настраивать предпочтения, такие как жанры, авторы, ключевые слова, минимальный год публикации и рейтинг книг, а затем генерировать рекомендации на основе этих данных. Система эффективно фильтрует книги из базы данных, рассчитывает для каждой книги рейтинг, а затем сортирует их по заданным критериям. Реализована возможность добавления книг в избранное и сохранения рекомендаций в CSV файл для дальнейшего использования.

Программа демонстрирует принцип работы рекомендательных систем, которые широко применяются в различных областях, включая интернет-магазины, платформы для потокового видео и онлайн-курсы. В ходе разработки были решены задачи обработки пользовательских запросов,

работы с файлами данных в формате JSON и реализации динамического интерфейса с возможностью автозаполнения.

Работа выполнена с соблюдением основных принципов разработки программного обеспечения, включая обработку ошибок, удобство использования интерфейса и эффективное взаимодействие с данными.

## **Источники**

1. **Tkinter** // <https://docs.python.org/3/library/tkinter.html> (дата обращения: 8.11.2024)
2. **asyncio** // <https://docs.python.org/3/library/asyncio.html> (дата обращения: 8.11.2024)