**Разработка "Рекомендательной системы для выбора книг на основе предпочтений пользователя**"

---

Описание проекта

Необходимо разработать программное приложение, реализующее функциональную рекомендательную систему для выбора книг. Приложение должно анализировать предпочтения пользователей (например, любимые жанры, авторы, темы) и на основе введенных данных предлагать книги, которые могут быть интересны пользователю. Система должна быть построена с использованием принципов функционального программирования.

---

Функциональные требования

1. Форма ввода данных пользователя:

- Пользователь указывает:

- Любимые жанры (например, "фантастика", "детектив").

- Любимых авторов (например, "Айзек Азимов", "Агата Кристи").

- Предпочитаемые ключевые слова (например, "искусственный интеллект", "расследование").

2. База данных книг:

- Должен быть предопределенный набор книг с метаданными:

- Название.

- Автор.

- Жанр.

- Краткое описание.

- База должна быть представлена в формате JSON или CSV.

3. Алгоритм рекомендаций:

- Сравнивать введенные пользователем данные с базой книг.

- Присваивать каждому элементу базы "рейтинг соответствия", основанный на совпадении по жанрам, авторам и ключевым словам.

- Выдавать список книг, отсортированных по убыванию рейтинга.

4. Фильтры и сортировка:

- Пользователь может отфильтровать рекомендации по следующим критериям:

- Только указанные жанры.

- Книги, выпущенные после определенного года.

- Сортировка:

- По рейтингу соответствия.

- По алфавиту.

- По году публикации.

5. Интерактивность:

- Интерфейс должен позволять пользователю:

- Выбрать понравившиеся книги для добавления в "список прочитать".

- Сохранить список рекомендаций в файл (CSV или JSON).

---

Интерфейс

1. Веб-интерфейс (опционально):

- Простой интерфейс с формой для ввода предпочтений и таблицей для просмотра и сортировки рекомендаций.

- Реализуется с использованием Flask/Django (Python) или других технологий.

---

Нефункциональные требования

1. Использование функционального программирования:

- Модульные функции для фильтрации, оценки рейтинга и сортировки.

- Использование генераторов для работы с большими объемами данных.

- Композиция функций для обработки данных.

2. Кроссплатформенность:

- Код должен работать на всех основных операционных системах (Windows, Linux, macOS).

3. Обработка ошибок:

- Валидация пользовательского ввода (например, проверка формата данных).

---

Архитектура

1. Модуль загрузки данных:

- Чтение базы книг из файла JSON/CSV.

- Пример структуры книги:

| {  "title": "Фонд",  "author": "Айзек Азимов",  "genre": "фантастика",  "description": "Будущее человечества под угрозой...",  "year": 1951  } |
| --- |

2. Модуль обработки предпочтений:

- Обрабатывает введенные пользователем данные.

- Генерирует словарь предпочтений.

3. Модуль рекомендаций:

- Сравнивает предпочтения пользователя с базой данных.

- Возвращает отсортированный список книг.

4. Модуль интерфейса:

- Взаимодействует с пользователем.

- Обеспечивает вывод рекомендаций и сохранение результатов.

---

Сроки выполнения

1. Разработка архитектуры и базы данных: 3 дня.

2. Реализация основных модулей: 5 дней.

3. Тестирование функциональности: 2 дня.

4. Финальная отладка и документация: 2 дня.

---

Ожидаемый результат

1. Рабочее приложение, позволяющее пользователю получить рекомендации по книгам на основе предпочтений.

2. Возможность сохранить рекомендации в файл.

3. Модульный и расширяемый код, соответствующий принципам функционального программирования.