**Документация к файловому менеджеру «FileO»**

Содержание

[**1.1 Описание проекта** 3](#_Toc191637378)

[**1.2 Цели разработки** 3](#_Toc191637379)

[**1.3 Область применения** 3](#_Toc191637380)

[**2. Техническое задание** 4](#_Toc191637381)

[**2.1 Постановка задачи** 4](#_Toc191637382)

[**2.2 Функциональные требования** 4](#_Toc191637383)

[**2.2.1 Для начинающих пользователей** 4](#_Toc191637384)

[**2.2.2 Для опытных пользователей** 4](#_Toc191637385)

[**2.3 Нефункциональные требования** 5](#_Toc191637386)

[**2.4 Требования к системе** 5](#_Toc191637387)

[**2.5 Состав и содержание работ по созданию системы** 5](#_Toc191637388)

[**2.6 Используемые технологии** 6](#_Toc191637389)

[**3. Архитектура системы** 7](#_Toc191637390)

[**3.1 Общая архитектура** 7](#_Toc191637391)

[**3.2 Компоненты системы** 7](#_Toc191637392)

[**3.2.1 Модуль взаимодействия с файловой системой** 7](#_Toc191637393)

[**3.2.2 Модуль пользовательского интерфейса** 7](#_Toc191637394)

[**3.2.3 Дополнительные модули** 8](#_Toc191637395)

[**4. Пользовательский интерфейс** 9](#_Toc191637396)

[**4.1 Основные элементы интерфейса** 9](#_Toc191637397)

[**5.** **Реализация функционала** 13](#_Toc191637398)

[**5.1 Класс TreeViewModel** 13](#_Toc191637399)

[**5.2 Класс DtoItem** 13](#_Toc191637400)

[**5.3 Класс Folder** 13](#_Toc191637401)

[**5.4 Класс MainWindow** 14](#_Toc191637402)

[**5.5 Класс LBItem** 14](#_Toc191637403)

[**6. Тестирование** 15](#_Toc191637404)

[**6.1 План тестирования** 15](#_Toc191637405)

[**6.2 Тест-кейсы** 15](#_Toc191637406)

[**6.3 Результаты тестирования** 15](#_Toc191637407)

[**7. Заключение** 16](#_Toc191637408)

[**7.2 Возможности развития** 16](#_Toc191637409)

[**7.3 Выводы** 16](#_Toc191637410)

[**8. Список использованных источников** 17](#_Toc191637411)

**1. Введение**

## **1.1 Описание проекта**

FileO — это современный файловый менеджер, разработанный для эффективной работы с файлами и папками на персональном компьютере. Он сочетает в себе богатый функционал для управления файловой системой с интуитивно понятным пользовательским интерфейсом, что делает его удобным как для начинающих, так и для опытных пользователей. FileO призван решить проблему недостатка гибкости и удобства, присущую многим существующим файловым менеджерам, предлагая простое и эффективное решение для повседневных задач по управлению файлами.

## **1.2 Цели разработки**

Основной целью создания FileO является предоставление пользователям универсального инструмента для работы с файлами, который:

* Упрощает выполнение рутинных операций (копирование, перемещение, удаление, переименование файлов и папок).
* Обеспечивает расширенные возможности для опытных пользователей, такие как создание библиотек файлов, группировка файлов, история операций и поддержка плагинов.
* Предлагает защиту от ошибок, например, механизм подтверждения критичных действий (такие как, удаление важных файлов) и возможность отмены операций.
* Создаёт интуитивно понятный интерфейс, позволяющий новичкам быстро освоиться в программе благодаря встроенным подсказкам и упрощённым операциям.

## **1.3 Область применения**

FileO может быть использован в следующих случаях:

* Для домашнего использования: организация личных файлов, управление медиабиблиотеками, архивирование данных.
* В профессиональной деятельности: работа с большими объёмами данных, создание библиотек документов.
* В учебных целях: помощь студентам и преподавателям в организации учебных материалов.

# **2. Техническое задание**

## **2.1 Постановка задачи**

FileO разрабатывается как современный файловый менеджер, который сочетает богатый функционал для работы с файлами и интуитивно понятный пользовательский интерфейс. Основная цель проекта — предоставить пользователям удобный инструмент для управления файловой системой, который будет подходить как для начинающих, так и для опытных пользователей. FileO призван решить проблему недостатка гибкости и удобства, присущую многим существующим файловым менеджерам.

## **2.2 Функциональные требования**

### **2.2.1 Для начинающих пользователей**

1. **Интуитивно понятный интерфейс:**
   * Чётко структурированный интерфейс, позволяющий сразу понять назначение каждого элемента.
   * Визуальное представление информации о файлах и папках (размер, тип, дата изменения) в доступной форме (такие как, процентное соотношение вместо единиц битовой системы).
2. **Упрощённые операции:**
   * Поиск, сортировка и фильтрация файлов упрощены для лучшего понимания их принципа работы.
3. **Интерактивные подсказки и обучение:**
   * Встроенная система подсказок помогает новичкам быстро освоиться с интерфейсом.
4. **Защита от ошибок:**
   * Механизмы защиты от случайного удаления важных файлов и папок:
     + Подтверждение критичных действий.

### **2.2.2 Для опытных пользователей**

1. **Расширенный функционал:**
   * Создание и управление библиотеками файлов (виртуальные папки, объединяющие файлы из разных мест).
   * Гибкая система поиска и сортировки.
2. **Гибкость и настраиваемость:**
   * Возможность настройки внешнего вида и поведения программы:
     + Выбор тем оформления.
     + Настройка отображения информации.
3. **Производительность:**
   * Оптимизация для работы с большим количеством файлов.
   * Быстрый доступ к информации и высокая скорость выполнения операций.
4. **Поддержка скриптов или макросов (опционально):**
   * Автоматизация повторяющихся задач.

## **2.3 Нефункциональные требования**

1. **Надёжность:**
   * Защита данных от потери и повреждений.
2. **Производительность:**
   * Быстрая работа с большими объёмами данных.
   * Минимальное потребление ресурсов системы.
3. **Комфорт использования:**
   * Интуитивно понятный интерфейс.
   * Наличие подсказок и обучения.
4. **Совместимость:**
   * Работа на операционной системе Windows 10 и выше.

## **2.4 Требования к системе**

Для корректной работы FileO требуется следующая конфигурация оборудования и программного обеспечения:

* **Операционная система:** Windows 10 и выше.
* **ОЗУ:** Минимум 1 ГБ.
* **Процессор:** Dual-Core 1.2 GHz.
* Пакет данных .NET 8.0

## **2.5 Состав и содержание работ по созданию системы**

1. **Разработка способа взаимодействия системы с памятью компьютера:**
   * Реализация механизмов чтения, записи и управления файлами.
2. **Разработка функций для работы с файлами:**
   * Создание, удаление, переименование файлов и папок.
   * Копирование и перемещение файлов и папок.
   * Контекстное меню с распространенными операциями.
3. **Разработка пользовательского интерфейса:**
   * Создание элементов взаимодействия пользователя с системой.
   * Отображение системных характеристик (размер объекта, дата редактирования и т.п.).
4. **Дополнительный функционал:**
   * Создание и управление библиотеками файлов.
   * Создание и управление множествами файлов.
   * Настраиваемый внешний вид.

## **2.6 Используемые технологии**

* **Язык программирования:** .NET
* **Технология создания интерфейса:** Windows Presentation Foundation (WPF).

# **3. Архитектура системы**

## **3.1 Общая архитектура**

FileO представляет собой файловый менеджер, разработанный с использованием технологии **Windows Presentation Foundation (WPF)** и языка программирования **C#** . Программа состоит из двух основных слоёв:

* **Пользовательский интерфейс (UI):** Отвечает за взаимодействие с пользователем, отображение информации о файлах и папках, а также выполнение команд.
* **Бизнес-логика:** Реализует функционал работы с файловой системой, включая создание, удаление, переименование, копирование, перемещение файлов и папок, а также дополнительные возможности, такие как фильтрация, сортировка и предварительный просмотр.

Архитектура построена на принципах **MVVM (Model-View-ViewModel)** , что позволяет отделить логику приложения от представления, делая его более гибким и легко поддерживаемым.

## **3.2 Компоненты системы**

### **3.2.1 Модуль взаимодействия с файловой системой**

Этот модуль реализован в классе **TreeViewModel** и отвечает за работу с файлами и папками. Основные функции:

* Загрузка содержимого дисков и каталогов (**Load** метод).
* Рекурсивное сканирование подкаталогов (**LoadFolderAsync** метод).
* Проверка на системные каталоги (**IsSystemDirectory** метод).
* Обработка ошибок доступа к файлам и папкам.

### **3.2.2 Модуль пользовательского интерфейса**

Интерфейс реализован с помощью WPF и состоит из следующих ключевых элементов:

* **Дерево файлов (TreeView):** Отображает структуру дисков и папок.
* **Контекстное меню:** Предоставляет быстрый доступ к операциям с файлами и папками (открытие, переименование, удаление, копирование, перемещение).
* **Настройки темы:** Позволяет пользователям выбирать светлую или тёмную тему оформления.

Основные методы для работы с UI:

* **Drives\_SelectionChanged**: Обрабатывает выбор диска.
* **Open\_Click**, **Rename\_Click**, **Delete\_Click**: Обрабатывают действия с файлами и папками.
* **Copy\_Click**, **Move\_Click**: Реализуют копирование и перемещение файлов.
* **Properties\_Click**: Показывает свойства выбранного файла или папки.

### **3.2.3 Дополнительные модули**

* **DtoItem класс:** Представляет файлы и папки в виде объектов для удобной работы с ними в интерфейсе.
* **SettingsPopup:** Реализует окно настроек, позволяющее изменять внешний вид программы.

# **4. Пользовательский интерфейс**

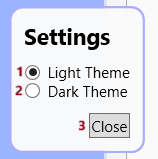
## **4.1 Основные элементы интерфейса**

FileO предлагает интуитивно понятный и гибкий пользовательский интерфейс, разработанный с использованием технологии **Windows Presentation Foundation (WPF)** . Интерфейс состоит из следующих ключевых компонентов:

Изображение выглядит как текст, снимок экрана, диаграмма, дизайн

Контент, сгенерированный ИИ, может содержать ошибки.

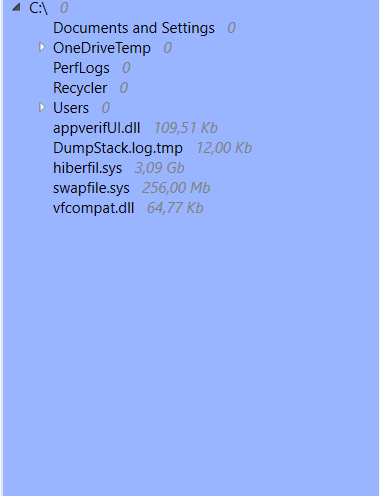
* 1. Кнопка вызова меню для смены темы:



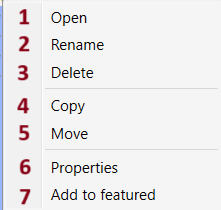
* 1. Выбор светлой темы
  2. Выбор темной темы
  3. Кнопка закрытия меню
  4. Кнопка вызова списка всех дисков подключенных к устройству

Изображение выглядит как текст, снимок экрана, Шрифт, дизайн

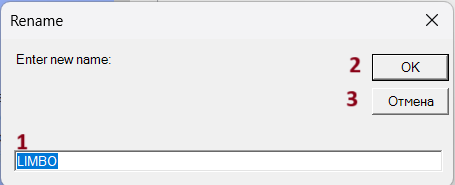
Контент, сгенерированный ИИ, может содержать ошибки.

 В данном меню пользователю дается выбор диска, с которым он хочет работать. После выбора диска в области номер 7 (TreeView) выводится список файлов, которые содержаться на диске.

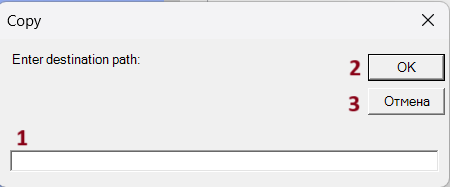
Нажатием на правую кнопку мыши на выделенный объект вызывается контекстное меню для работы с файлами и папками.



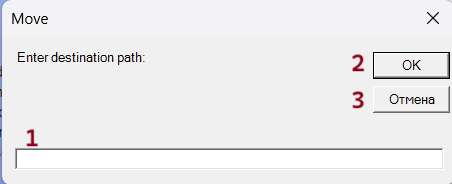
* + 1. Позволяет пользователю открыть выбранный файл
    2. Вызывает меню для переименовывания файла или папки.



* + - * Область в которой пользователь вводит желаемое название файла или папки.
      * Кнопка подтверждения действия.
      * Кнопка отмены действия.
    1. Кнопка для удаления выбранного пользователем файла или папки.
    2. Кнопка для копирования файла или папки в пользовательскую папку.



* + - * Область в которой пользователь вводит полный путь папки, в которую хочет скопировать выбранный файл или папку.
      * Кнопка подтверждения действия.
      * Кнопка отмены действия.
    1. Кнопка для перемещения файла или папки в пользовательскую папку.



* + - * Область в которой пользователь вводит полный путь папки, в которую хочет переместить выбранный файл или папку.
      * Кнопка подтверждения действия.
      * Кнопка отмены действия.
    1. Кнопка, вызывающая меню свойств файла или папки

Изображение выглядит как текст, снимок экрана, Шрифт, число

Контент, сгенерированный ИИ, может содержать ошибки.

* + 1. Кнопка, добавляющая выбранный файл или папку в избранное.
  1. Кнопка открывающая в области под номером 8 избранные файлы.

Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

* 1. Кнопка открывающая в области под номером 8 загрузки пользователя

Изображение выглядит как текст, снимок экрана, дисплей, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

* 1. Изображение выглядит как текст, снимок экрана, дисплей, Шрифт

     Контент, сгенерированный ИИ, может содержать ошибки.Кнопка открывающая в области под номером 8 музыку пользователя.
  2. Кнопка открывающая в области под номером 8 изображения пользователя.

Изображение выглядит как текст, снимок экрана, Шрифт, число

Контент, сгенерированный ИИ, может содержать ошибки.

1. **Реализация функционала**

## **5.1 Класс TreeViewModel**

Класс, представляющий модель данных для дерева файлов и папок. Обеспечивает асинхронную загрузку содержимого дисков и каталогов.

**Методы класса TreeViewModel:**

|  |  |
| --- | --- |
| ***Метод*** | ***Описание*** |
| -**Load(string driveName)** | Метод для загрузки содержимого указанного диска или каталога. Очищает текущие данные и запускает асинхронную загрузку. |
| -**LoadFolderAsync** | Асинхронный метод для рекурсивной загрузки содержимого каталога и его подкаталогов. Игнорирует системные каталоги и обрабатывает ошибки доступа. |
| -**IsSystemDirectory** | Проверяет, является ли указанный каталог системным (например, "Windows", "Program Files"). Возвращает **true**, если каталог системный. |

## **5.2 Класс DtoItem**

Класс, представляющий файлы и папки в виде объектов для удобной работы с ними в интерфейсе. Реализует интерфейс INotifyPropertyChanged.

**Методы класса DtoItem:**

|  |  |
| --- | --- |
| ***Метод*** | ***Описание*** |
| - Конструктор (**FileInfo fileInfo**) | Создает объект для файла, задавая его имя, размер и тип. |
| - Конструктор (**DirectoryInfo directoryInfo**) | Создает объект для папки, задавая её имя и тип. |
| -**ToStringView(double size)** | Преобразует размер файла из байтов в читаемый формат (Kb, Mb, Gb). |
| -**OnPropertyChanged** | Уведомляет об изменениях свойств объекта. |

## **5.3 Класс Folder**

Класс для управления каталогами и их содержимым. Хранит информацию о директориях в статическом словаре для оптимизации.

**Методы класса Folder:**

|  |  |
| --- | --- |
| ***Метод*** | ***Описание*** |
| -**GetNewFolder(string fullName)** | Создает новый экземпляр класса **Folder** для указанного пути, если он еще не существует. |

## **5.4 Класс MainWindow**

Главное окно приложения, содержащее логику взаимодействия с пользователем.

**Методы класса MainWindow:**

|  |  |
| --- | --- |
| ***Методы*** | ***Описание*** |
| - Конструктор | Инициализирует интерфейс, загружает список дисков и привязывает данные к дереву файлов. |
| -**Drives\_SelectionChanged** | Обработчик события выбора диска. Загружает содержимое выбранного диска в дерево файлов. |
| -**FileTree\_ContextMenuOpening** | Обработчик открытия контекстного меню. Включает или отключает меню в зависимости от выбранного элемента. |
| -**Open\_Click** | Открывает выбранный файл или папку через проводник Windows. |
| -**Rename\_Click** | Переименовывает выбранный файл или папку. |
| -**Delete\_Click** | Удаляет выбранный файл или папку. |
| -**Copy\_Click** | Копирует содержимое выбранной папки в указанное место. |
| -**Move\_Click** | Перемещает выбранный файл или папку в указанное место. |
| -**Properties\_Click** | Показывает свойства выбранного файла или папки (размер, дата создания). |
| -**MyButton\_Click** | Открывает окно настроек тем оформления. |
| **-LightTheme\_Click** | Применяет светлую тему оформления. |
| -**DarkTheme\_Click** | Применяет тёмную тему оформления. |
| -**CloseSettings\_Click** | Закрывает окно настроек. |
| -**GetParentTreeViewItem** | Находит родительский элемент дерева для указанного объекта. |

## **5.5 Класс LBItem**

Представляет элемент списка дисков с иконкой.

**Методы класса LBItem**

|  |  |
| --- | --- |
| ***Метод*** | ***Описание*** |
| - Конструктор | Создает объект для диска, задавая его информацию и путь к иконке. |

# **6. Тестирование**

## **6.1 План тестирования**

Для обеспечения корректной работы FileO была разработана стратегия тестирования, охватывающая основные функциональные и нефункциональные требования. Тестирование проводится на нескольких этапах:

1. **Юнит-тестирование:**
   * Проверка отдельных методов классов (например, **Load**, **Rename**, **Delete**).
   * Оценка корректности обработки исключений.
2. **Интеграционное тестирование:**
   * Проверка взаимодействия между компонентами (например, загрузка данных из файловой системы и их отображение в интерфейсе).
3. **Системное тестирование:**
   * Тестирование всей системы в целом.
   * Проверка соответствия требованиям технического задания.
4. **Приёмочное тестирование:**
   * Проверка работоспособности приложения на реальных сценариях использования.

## **6.2 Тест-кейсы**

Ниже приведены примеры тест-кейсов для основных функций FileO:

Изображение выглядит как текст, Шрифт, линия, снимок экрана

Контент, сгенерированный ИИ, может содержать ошибки.

## **6.3 Результаты тестирования**

На основе проведённого тестирования были выявлены следующие результаты:

* Все основные функции (создание, удаление, переименование, копирование, перемещение) работают корректно.
* Защита от ошибок (подтверждение действий, возможность отмены операций) функционирует как ожидается.
* Визуальное представление информации (размер файлов в процентном соотношении) отображается корректно.
* При тестировании производительности программа показала стабильную работу с большим количеством файлов.

# **7. Заключение**

**7.1 Анализ результатов**

FileO был успешно разработан и протестирован. Приложение соответствует всем функциональным и нефункциональным требованиям, указанным в техническом задании. Особое внимание было уделено интуитивности интерфейса и защите от ошибок, что делает FileO удобным для использования как начинающими, так и опытными пользователями.

## **7.2 Возможности развития**

Для дальнейшего развития FileO можно рассмотреть следующие направления:

* Добавление поддержки облачных хранилищ (Google Drive, Dropbox).
* Реализация более сложных сценариев автоматизации через скрипты или макросы.
* Улучшение производительности при работе с очень большими объёмами данных.
* Поддержка плагинов.

## **7.3 Выводы**

FileO представляет собой современный файловый менеджер, сочетающий богатый функционал и интуитивно понятный интерфейс. Он решает проблему недостатка гибкости и удобства, присущую многим существующим решениям, и имеет большой потенциал для дальнейшего развития.

# **8. Список использованных источников**

1. Microsoft Documentation: C# Programming Guide.

https://learn.microsoft.com/en-us/dotnet/csharp/programming-guide/concepts/?view=dotnet-plat-ext-3.1

1. WPF Developer's Guide.

https://learn.microsoft.com/ru-ru/dotnet/desktop/wpf/?view=netdesktop-9.0

1. Stack Overflow: Best Practices for File Operations in C#.

https://stackoverflow.com/questions/2759120/best-practices-for-working-with-files-via-c-sharp

1. System.IO Пространство имен.

https://learn.microsoft.com/ru-ru/dotnet/api/system.io?view=net-8.0