|  |
| --- |
| **МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ** |
| **ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ТЕХНОЛОГИЙ И УПРАВЛЕНИЯ ИМЕНИ К.Г. РАЗУМОВСКОГО (ПЕРВЫЙ КАЗАЧИЙ УНИВЕРСИТЕТ)»**  **(ФГБОУ ВО «МГУТУ ИМ. К.Г. РАЗУМОВСКОГО (ПКУ)»)** |
|  |
| **УНИВЕРСИТЕТСКИЙ КОЛЛЕДЖ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ** |

|  |
| --- |
| **КУРСОВОЙ ПРОЕКТ** |
|  |
| по междисциплинарному курсу МДК.01.02. Прикладное программирование |
|  |
| на тему: Разработка программы «Файловый менеджер» |
|  |
| студента группы ПКС-305  специальности 09.02.03 Программирование в компьютерных системах |
| Тешаева Рустама Мирзодавлатовича |

|  |  |  |
| --- | --- | --- |
| Студент |  | Р.М. Тешаев |
| Руководитель курсового проекта |  | C.Ю. Кузьменко |
| Председатель ПЦК специальности 09.02.03 Программирование в компьютерных системах |  | А.И. Глускер |

|  |  |  |
| --- | --- | --- |
| Дата защиты «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_ 2019 г. | | |
| Оценка:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | | |
| Заведующий отделением № 3 |  | Е.Л. Гусева |

|  |  |
| --- | --- |
| **МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ** | |
| **ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ТЕХНОЛОГИЙ И УПРАВЛЕНИЯ ИМЕНИ К.Г. РАЗУМОВСКОГО (ПЕРВЫЙ КАЗАЧИЙ УНИВЕРСИТЕТ)»**  **(ФГБОУ ВО «МГУТУ ИМ. К.Г. РАЗУМОВСКОГО (ПКУ)»)** | |
|  | |
| **УНИВЕРСИТЕТСКИЙ КОЛЛЕДЖ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ** | |
|  | |
| РАССМОТРЕНО | УТВЕРЖДАЮ |
| предметной (цикловой) комиссией специальности 09.02.03 Программирование в компьютерных системах | Заместитель директора по учебно-методической работе |
| Председатель ПЦК | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Е.В. Вернер |
| \_\_\_\_\_\_\_\_\_\_\_\_ А.И. Глускер | «\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2019 г. |
| Протокол № 2 от 30.09.2019 |  |

|  |  |  |
| --- | --- | --- |
| **ТЕХНИЧЕСКОЕ ЗАДАНИЕ НА КУРСОВОЙ ПРОЕКТ** | | |
|  | | |
| студенту группы ПКС-305 специальности 09.02.03 Программирование в компьютерных системах | | |
| Тешаеву Рустаму Мирзодавлатовичу | | |
|  | | |
| **Наименование МДК:** МДК.01.02. Прикладное программирование. | | |
| **Тема курсового проекта:** Разработка программы «Файловый менеджер» | | |
| **Структура курсового проекта:** | | |
| * пояснительная записка; * презентация; * работающая программа; * электронный носитель, на котором записаны: пояснительная записка, презентация, работающая программа | | |
| **Требования к пояснительной записке:** | | |
| Оформление пояснительной записки по ГОСТ 7.32-2017  Оформление блок схемы алгоритма по ГОСТ 19.701 | | |
| **Требования к проектируемому продукту:** | | |
| Отсутствие ошибок, выдаваемых программным продуктом | | |
| **Порядок сдачи курсового проекта:** | | |
| Исполнитель предъявляет:   * пояснительную записку на бумажном носителе; * электронный носитель, содержащий текст пояснительной записки, презентацию для защиты курсового проекта, работающий программный продукт; | | |
| **Дата выдачи технического задания:** «10» октября 2019 г. | | |
| **Дата сдачи курсового проекта:** «06» декабря 2019 г. | | |
|  | | |
| Руководитель курсового проекта |  | С.Ю. Кузьменко |
| Студент |  | Р.М. Тешаев |

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 5](#_Toc26716931)

[ОСНОВНАЯ ЧАСТЬ 6](#_Toc26716932)

[1 Введение в предметную область 6](#_Toc26716933)

[2 Спецификация 6](#_Toc26716935)

[2.1 Спецификация 6](#_Toc26716937)

[2.1.1 Наименование темы разработки 6](#_Toc26716938)

[2.1.2 Область применения 6](#_Toc26716940)

[2.1.3 Объект, в котором используется программа 6](#_Toc26716942)

[2.1.4 Основание для разработки 6](#_Toc26716944)

[2.1.5 Назначение разработки 6](#_Toc26716949)

[2.1.6 Требования к программе или программному изделию 7](#_Toc26716954)

[2.1.7 Требования к программной документации 8](#_Toc26716979)

[2.1.8 Стадии и этапы разработки 9](#_Toc26716991)

[2.1.9 Порядок контроля и приёмки 10](#_Toc26717003)

[3 Программа и методика испытания 10](#_Toc26717007)

[3.1 Программа и методика испытания 10](#_Toc26717009)

[3.1.1 Объект испытания 10](#_Toc26717010)

[3.1.2 Цель испытаний 11](#_Toc26717017)

[3.1.3 Требования к программе 11](#_Toc26717019)

[3.1.4 Требования программной документации 11](#_Toc26717033)

[3.1.5 Средства и порядок испытания 12](#_Toc26717038)

[3.1.6 Методы испытаний 13](#_Toc26717054)

[4 Технический проект 16](#_Toc26717071)

[4.1 Технический проект 16](#_Toc26717073)

[4.1.1 Наименование программы 16](#_Toc26717074)

[4.1.2 Условное обозначение темы разработки 16](#_Toc26717076)

[4.1.3 Документ, на основании которого ведётся разработка 16](#_Toc26717078)

[4.1.4 Назначение и область применения 16](#_Toc26717080)

[4.1.5 Назначение программы 16](#_Toc26717081)

[4.1.6 Краткая характеристика области применения 16](#_Toc26717083)

[4.1.7 Постановка задачи на разработку 16](#_Toc26717085)

[4.1.8 Описание используемых технологий 16](#_Toc26717087)

[4.1.9 Описание алгоритмов 17](#_Toc26717090)

[4.1.10 Диаграмма вариантов использования 17](#_Toc26717091)

[4.1.11 Диаграмма деятельности 18](#_Toc26717094)

[5 Реализация программного изделия на языке программирования 19](#_Toc26717096)

[5.1 Процесс реализации 19](#_Toc26717100)

[6 Тестирование программного продукта 19](#_Toc26717104)

[6.1 Протоколы прохождения тестов 19](#_Toc26717106)

[6.2 Протоколы проверки требований к исходным кодам 20](#_Toc26717109)

[6.3 Протоколы проверки требований к программе 21](#_Toc26717111)

[6.4 Протоколы проверки требований к пояснительной записке 22](#_Toc26717113)

[6.5 Протоколы проверки требований к компиляции исходных кодов 22](#_Toc26717114)

[ЗАКЛЮЧЕНИЕ 23](#_Toc26717116)

[СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ 24](#_Toc26717117)

[ПРИЛОЖЕНИЕ А Исходный код 25](#_Toc26717118)

# ВВЕДЕНИЕ

Файловые менеджеры - это класс программ, служащих для всевозможной работы с файлами, включая поддержку таких операций, как создание, редактирование, копирование и удаление, для обеспечения гибкого и удобного запуска других программ, предназначенных для работы с этими файлами. Нередко они сопровождаются дополнительными утилитами, облегчающими жизнь пользователя. Для многих пользователей любимый файловый менеджер зачастую выступает в роли оболочки, заменяя часть стандартных средств работы с файлами, имеющихся в операционной системе. Некоторые из них могут подключать к file transfer protocol (ftp). Менее функциональные помогут вам сразу запустить программу с нужным параметром.

Необходимо отметить, что в развитии файловых менеджеров наметилась четкая тенденция. Созданные изначально для того, чтобы заменить командную строку, они на данный момент превратились в весьма сложные программные комплексы, обладающие функционалом, расширяемым посредством плагинов и дополнительных модулей. Можно с уверенностью утверждать, что файловые менеджеры сегодня превратились в важный инструмент профессиональных пользователей, обеспечивающий высокую эффективность работы.

У файловых менеджеров есть общие черты:

. Возможность просмотра скрытых файлов, когда проводник не показывает скрытые файлы в результате поражения вирусом;

. Две рабочих «зоны», благодаря которым можно не тратить время на перемещение между папками для копирования;

. Возможность поиска файлов по дате, имени, или другому параметру которые позволяют быстро находить нужные вам файл.

# ОСНОВНАЯ ЧАСТЬ

## Введение в предметную область

Файловый менеджер (англ. file manager) - компьютерная программа, предоставляющая интерфейс пользователя для работы с файловой системой и файлами. Файловый менеджер позволяет выполнять наиболее частые операции:

* копирования;
* переноса;
* удаления;
* редактирования;
* запуска программы для работы с этими файлами

Существует два вида файловых менеджеров - навигационные и ортодоксальные. Основное отличие - в последних имеется две панели, реализована соответствующая модель работы.

Наиболее известные ортодоксальные файловые менеджеры: Norton Commander, Dos Navigator, Volkov Commander, PIE Commander, FAR Manager, Total Commander, POSIX (Linux, BSD и т.д.), Midnight Commander.

Навигационные файловые менеджеры: проводник Windows (англ. Windows Explorer) - встроен в Windows, Mac OS X, Finder, Path Finder, POSIX (Linux, BSD и т.д.), Konqueror - поставляется с KDE, Nautilus (файловый менеджер) - поставляется с GNOME.

Файловый менеджер должен: обеспечивать удобную возможность работы с файлами, копировать, удалять, перемещать, создавать файлы и папки, запускать внешние программы для работы с разными типами файлов.

## Спецификация

## В данном разделе описана основная информация о программном продукте, требования по эксплуатации и разработки, а также выполняемый им функционал.

## Спецификация

## Определение требований

## Наименование темы разработки

## Разработка программы «Файловый менеджер».

## Область применения

## Для рабочих целей.

## Объект, в котором используется программа

## Программа «Файловый менеджер» может быть использована в любых объектах.

## Основание для разработки

## Программа должна работать без сбоев.

## Документ, на основании которого ведётся разработка

## Техническое задание на курсовой проект.

## Наименование и (или) условное обозначение темы для разработки

## Программа «Файловый менеджер».

## Назначение разработки

## Программа предназначена под ОС Windows.

## Функциональное назначение

## Программа должна позволять пользователю использовать возможности файлового менеджера.

## Эксплуатационное назначение

## Программа предназначена для использования неограниченным кругом лиц.

## Требования к программе или программному изделию

## Программа должна выполнять поставленную при разработке задачу.

## Требования к функциональным характеристикам

Исходные данные и требуемые результаты.

## Требования к составу выполняемых функций

Программа должна обеспечивать выполнение следующих функций:

работа с папками, просмотр папок, просмотр содержимого папок, поиск файлов, удаление, переименование, копирование файлов.

## Требования к временным характеристикам

## Время между нажатием кнопок пользователем и получением результата не должно превышать 2 секунды.

## Требования к интерфейсу

## Для программы «Файловый менеджер» нужно:

## чтобы у пользователя были на экране самые необходимые элементы, которые нужны для использования программы «Файловый менеджер», а именно, элемент управления listbox, для отображения папок, файлов, т.д., элемент управления textbox, где пользователь, может ввести путь папки либо файла, элемент управления button кнопка «Перейти и «Назад», функции из контекстного меню, такие как копировать, вставить, удалить, переименовать.

## Требования к надежности

## Для программы «Файловый менеджер» пользователю нужно:

## При нажатии пользователем на какую-либо клавишу, не являющейся горящей,то ничего не должно произойти. При выполнении всех условий, необходимых для каких-либо действий с файлами и папками, все будет успешно работать.

## Другие требования к надёжности не предъявляются.

## Условия эксплуатации

## Минимальное количество персонала - один человек. Другие специальные требования к условиям эксплуатации не предъявляются.

## Требования к составу и параметрам технических средств

## В состав технических средств должен входить IBM-совместимый компьютер, включающий:

- процессор Pentium или совместимый с ним,

­- оперативную память не менее 500 Мб,

- видеопамять не менее 256 Мб,

- мышь,

- видеокарту,

- монитор,

- жесткий диск.

## Требования к информационной и программной совместимости

Данное программное обеспечение разрабатывается под платформу Windows.

## Требования к информационным структурам на входе и выходе

## Не предъявляются.

## Требования к методам решения

## Все вычисления могут быть выполнены с помощью стандартных библиотек, разработанных в Microsoft Visual Studio 2019 на языке C#.

## Требования к программной документации

Стандарт требований разработан с учетом всех действующих стандартов на документацию.

## Состав программной документации

## Состав программной документации должен включать

## - описание программы,

## - программа и методика испытаний,

## - текст программы.

## Специальные требования к пояснительной записке

## Пояснительная записка должна содержать UML - диаграмму деятельности и UML - диаграмму вариантов использования.

## Требования к исходным кодам

## Требования к исходным кодам изложены в документе: А. И. Глускер

## «Сборник задач по курсу “Основы алгоритмизации и программирования” [Электронный ресурс] - 2011 /раздел 3.1/ (п. 4.5.2 технического задания).

## Программа должна быть написана на языке C# и обязана компилироваться транслятором JIT (Just-In-Time) (п. 4.5.3 технического задания).

## Стадии и этапы разработки

Стадии разработки устанавливает стадии разработки программ, программной документации, а также этапы и содержание работ

## Стадии разработки

## Разработка осуществляется в четыре стадии:

## разработка пояснительной записки,

## проектирование,

## реализация проекта,

## тестирование.

## Этапы разработки

## На стадии разработки пояснительной записки осуществляется разработка, согласование и утверждение пояснительной записки.

## На стадии проектирование осуществляется проектирование, согласование и утверждение программного продукта.

## На стадии реализации проекта осуществляется разработка текста программы.

## На стадии тестирование осуществляется тестирование программного продукта.

## Порядок контроля и приёмки

Порядок контроля и приемки системы должен быть доведен до необходимого и достаточного объема, способного удовлетворить заказчика.

## Приёмосдаточные испытания

## Приемосдаточные испытания должны проводится в соответствии с программой и методикой испытаний, разработанной, согласованной и утвержденной не позднее 6 декабря 2019 года.

## В данном разделе были изложены все основные особенности программного продукта и требования к нему.

## Программа и методика испытания

## Данный раздел описывает основные методы тестирования для проверки соответствия программного продукта функциональным требованиям.

## Программа и методика испытания

Программа методики испытаний призвана установить технические данные, которые подлежат проверке во время испытаний всей системы в целом или ее отдельных компонентов.

## Объект испытания

## Программа «Файловый менеджер».

## Наименование темы разработки

## Разработка программы «Файловый менеджер».

## Область применения

## Для рабочих целей.

## Обозначение программы

## Разработка программы «Файловый менеджер».

## Цель испытаний

## Проверка соответствия программного продукта требованиям технического задания.

## Требования к программе

## Требования могут выражаться в виде текстовых утверждений.

## Требования к функциональным характеристикам

## В подразделе должны быть указаны требования к составу выполняемых функций.

## Требования к составу выполняемых функций

## Программа должна обеспечивать выполнение следующих функций:

## переходить по директориям, папкам и файлам программы «Файловый менеджерa»

## возвращаться назад

## использовать функции контекстного меню, такие как копировать, вставить, удалить и переименовать.

## Требования к интерфейсу

## Пункт 2.1.6.1.3 Пояснительной записки.

## Требования к надежности

## Пункт 2.1.6.2 Пояснительной записки.

## Требования к информационной и программной совместимости

## Пункт 2.1.6.5 Пояснительной записки.

## Требования программной документации

Специальные требования к программной документации не предъявляются.

## Состав программной документации

## Пункт 2.1.7.1 Пояснительной записки.

## Специальные требования к пояснительной записке

## Пункт 2.1.7.2 Пояснительной записки.

## Средства и порядок испытания

Системные программные средства, используемые программой, должны быть представлены лицензионной локализованной версией операционной системы.

## Технические средства, используемые при проведении испытаний

## В состав технических средств, используемых при проведении испытаний входит:

- процессор Pentium или совместимый с ним,

- оперативную память не менее 500 Мб,

- видеопамять не менее 256 Мб,

- Мышь,

- видеокарту,

- монитор,

- жесткий диск.

## Программные средства, используемые при проведении испытаний

## В состав программных средств должна входить готовая, к запуску программа

## Порядок проведения испытаний

## Проведение испытаний должно осуществляться поэтапно.

## Подготовка к проведению испытаний

## Подготовка к проведению испытаний заключается в обеспечении наличия компьютера, соответствующего требованиям в пункте 2.1.6.4, и программных средств, описанных в пункте 2.1.6.5.

## Ход проведения испытаний

## Ход проведения испытаний документируются в протоколе, где указывается перечень проводимых испытаний, результат каждого испытания и, возможно, замечания.

## Состав испытания

## Проверка состава программной документации в соответствии с методом, описанным в пункте 2.1.7.1.

## Проверка требований к программе

Программа должна легко запускаться. По нажатии кнопок, программа должна выполнять соответствующие игровые действия.

## Проверка обеспечений требования к программе (п. 3)

## Проверка обеспечения требований к программе в соответствии с методом, описанным в пункте 2.1.6.

## Методы испытаний

## Испытания осуществляются тестированием функций программы.

## Метод проверки требований к программе

## Проверка осуществляется путём запуска программы и сравнения ожидаемых результатов, приведённых в пункте 3.1.6.6.

## Метод проверки требований к составу пояснительной записки

## Проверка состава пояснительной записки осуществляется визуально путем набора предъявленных документов (в форме распечатки или в рукописной форме) списку, приведенному в пункте 2.1.7.1. При этом исходные тексты программ должны быть предоставлены так же и в электронной форме.

## В случае, если состав пояснительной записки соответствует списку, а исходные тексты предоставлены также и в электронной форме, то в протокол заносится запись: «Состав пояснительной записки» - соответствует, в противном случае: «Состав пояснительной записки» - не соответствует.

## Метод проверки специальных требований к пояснительной записке

## Проверка состоит из проверки наличия и соответствия всех UML - диаграмм.

## Проверка соответствия каждой UML - диаграммы алгоритму, закодированному в программе, осуществляется путем экспертной оценки лицом, осуществляющим проведение испытаний. В случае, если все вышеприведенные проверки прошли успешно, в протокол заносится запись: «Специальные требования к пояснительной записке» - соответствуют, в противном случае «Специальные требования к пояснительной записке» - не соответствуют.

## Метод проверки требований к исходным кодам

## Изложенный ниже метод применяется ко всем файлам, содержащим исходный текст, и входящим в состав программной документации по отдельности. Для каждого файла вносится в протокол запись: «Требования к исходным кодам для файла #####» -­ соответствует / не соответствует (где вместо ##### указывается название файла).

## Метод проверки требований к исходным кодам в части компиляции в Visual Studio

## Изложенный ниже метод применяется ко всем файлам, содержащим исходный текст, и входящим в состав программной документации по отдельности. Для каждого файла вносится в протокол запись: «Требования к исходным кодам в части компиляции в Visual Studio для файла #####» – соответствует / не соответствует (где вместо ##### указывается название файла).

## Проверка осуществляется путем открытия файла с исходным кодом в среде Visual Studio, осуществления компиляции (обе работы делаются в соответствии с документацией к Visual Studio). В случае, если компиляция завершилась успешно, в протокол вносится запись о соответствии требованиям, в противном случае – о несоответствии.

## Ожидаемые результаты работы программы

## Ниже описаны ожидаемые результаты работы программы (Таблица 1). В итоге все ожидаемые результаты программного продукта должны соответствовать ожидаемому результату, изложенному в (Таблица 1). Также возможны замечания по ожидаемому результату.

Таблица 1 - Ожидаемые результаты работы программы

|  |  |  |
| --- | --- | --- |
| **Название функции** | **Описание функции** | **Результат** |
| Кнопка «Перейти» | Нужно указать путь папки, файла, директории, а после этого уже нажимать на кнопку перейти | Перейдет к указанному папке, файлу, директории |
| Кнопка «Назад» | Возвращает назад | Предназначено для возвращения, к прошлому действию |
| Функция «Копировать» | Копирует папку и файлы | Произойдет копирование, если пользователь кликнет правой кнопкой мыши и выберет из контекстного меню функцию «Копировать» |
| Функция «Вставить» | Вставляет папки и файлы, которые находятся в буфере обмена | Произойдет вставка, если пользователь кликнет правой кнопкой мыши и выберет из контекстного меню функцию «Вставить» |
| Функция «Удалить | Удаляет папки и файлы | Произойдет удаление, если пользователь кликнет правой кнопкой мыши и выберет из контекстного меню функцию «Удалить» |
| Функция «Переименовать» | Переименовывает папки и файлы | Произойдет переименование, если пользователь кликнет правой кнопкой мыши и выберет из контекстного меню функцию «Переименовать» |
| Текстовое поле textBox | Для ввода текста в специальное поле | Ввод пути файла или папки в текстовое поле, после чего пользователь кликает на кнопку «Перейти» и происходит переход по указанному пути |

## В этом разделе были изложены методы, которые будут применяться к программному продукту для проверки соответствия требованиям технического задания.

# 2.ПРАКТИЧЕСКАЯ ЧАСТЬ

## Технический проект

## В данном разделе изложены все архитектурные решения и алгоритмы, используемые в проектировании и разработке программного продукта «Файловый менеджер».

## Технический проект

Технический проект содержит технические решения и данные, достаточные для полного представления об устройстве и принципе работы программы.

## Наименование программы

## Разработка программы «Файловый менеджер».

## Условное обозначение темы разработки

## Разработка программы «Файловый менеджер».

## Документ, на основании которого ведётся разработка

## Техническое задание на курсовой проект.

## Назначение и область применения

## В рабочих целях.

## Назначение программы

## Программа должна позволять использовать возможности Файлового менеджера.

## Краткая характеристика области применения

## Программа может использоваться любыми лицами.

## Постановка задачи на разработку

## Создание развлекательной программы «Файловый менеджер».

## Описание используемых технологий

## Игра разрабатывается с использованием Microsoft Visual Studio 2019.

## Visual Studio включает в себя редактор исходного кода с поддержкой технологии IntelliSense и возможностью простейшего рефакторинга кода. Встроенный отладчик может работать как отладчик уровня исходного кода, так и отладчик машинного уровня. Остальные встраиваемые инструменты включают в себя редактор форм для упрощения создания графического интерфейса приложения, веб-редактор, дизайнер классов и дизайнер схемы базы данных.

## Описание алгоритмов

## Алгоритмы представлены в UML схеме вариантов использования и в UML диаграмме деятельности.

## Диаграмма вариантов использования

## 

Рисунок 1 – UML – диаграмма вариантов использования

## Диаграмма деятельности

## 

Рисунок 2 - UML - диаграмма деятельности

## Реализация программного изделия на языке программирования

## Этот этап нужен для создания работающей программы на выбранном языке программирования, в ходе которого осуществляется тестирование и отладка продукта. Здесь описаны ошибки и трудности, с которыми я столкнулся и их решение.

## Была разработана программа «Файловый менеджер». Исходный код (Приложение А).

## При разработке программного изделия встречались ошибки и трудности.

## Процесс реализации

## Разработка игры началась в ноябре 2019 года и завершилась в декабре 2019 года.

## В ходе разработки были применены ранее приобретенные навыки. На данный момент, это готовая версия программы «Файловый Менеджер», только некоторые функции из контекстного меню не работают, основные работают замечательно.

## Не работают такие функции, как создание папки, создание текстового документа, переименование.

## Тестирование программного продукта

## Тестирование исходного кода путем написания юнит–тестов не производилось. Сдача программного продукта происходит путем приёмосдаточных испытаний.

## Протоколы прохождения тестов

## Все тестовые примеры были пройдены успешно.

## Ниже на (Рисунок 3, 4) приведены результаты прохождения некоторых тестов.

## 

Рисунок 3 - Результат прохождения теста 1

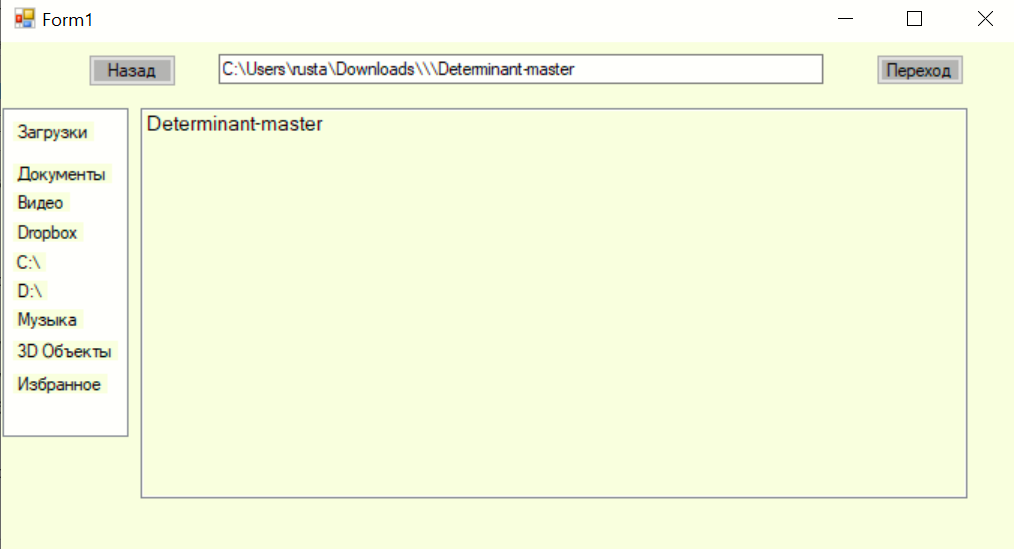


Рисунок 4 - Результат прохождения теста 2

## Протоколы проверки требований к исходным кодам

## Все файлы были успешно откомпилированы и соответствуют требованиям к исходным кодам.

## Протоколы проверки требований к программе

## Программа была полностью протестирована по таблице, приведенной в методах испытаний в пункте 3.1.6.6 (Рисунок 5 - ).

## Все тесты были успешно завершены. Программа соответствует требованиям.

## 

Рисунок 5 - Результат прохождения теста 3

## 

Рисунок 6 - Результат прохождения теста 4

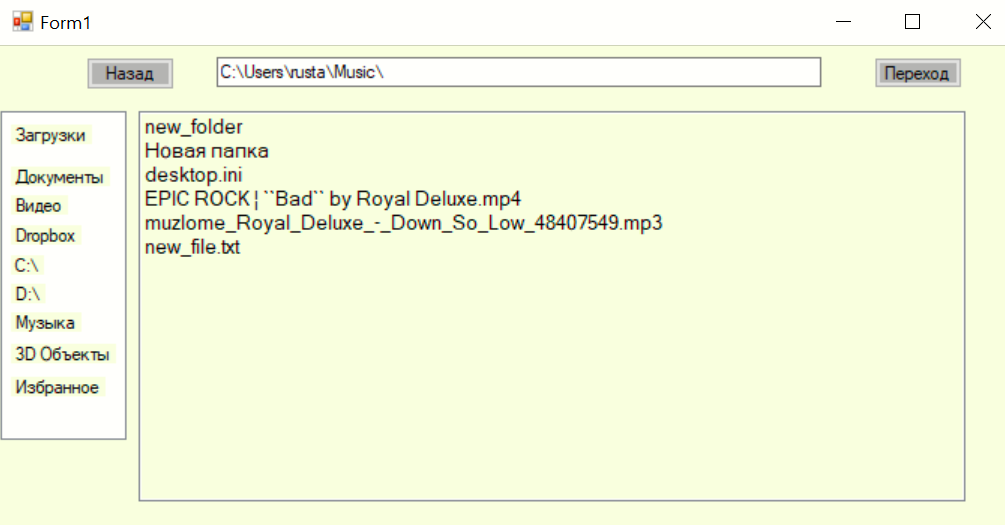


Рисунок 7 - Результат прохождения теста 5

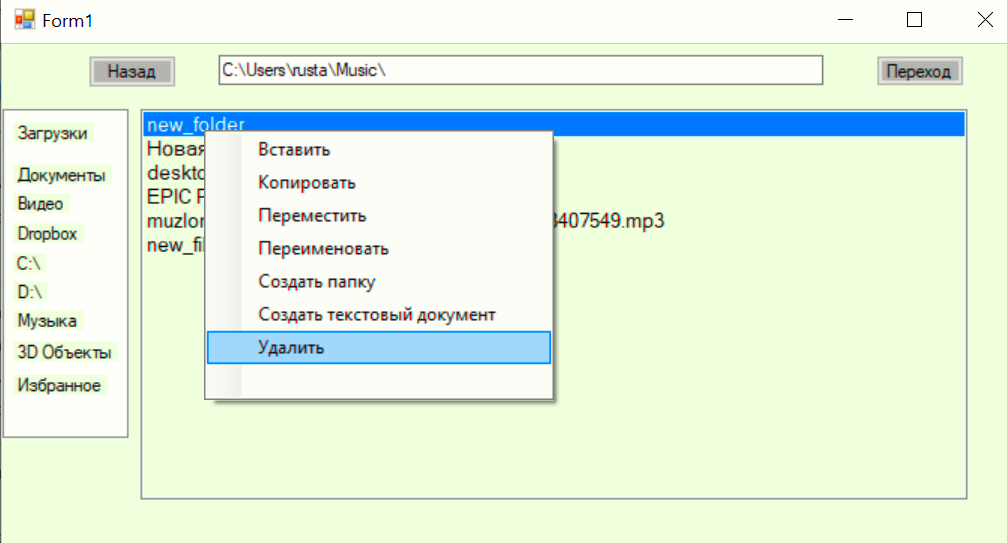


Рисунок 8 - Результат прохождения теста 6



Рисунок 9 - Результат прохождения теста 7

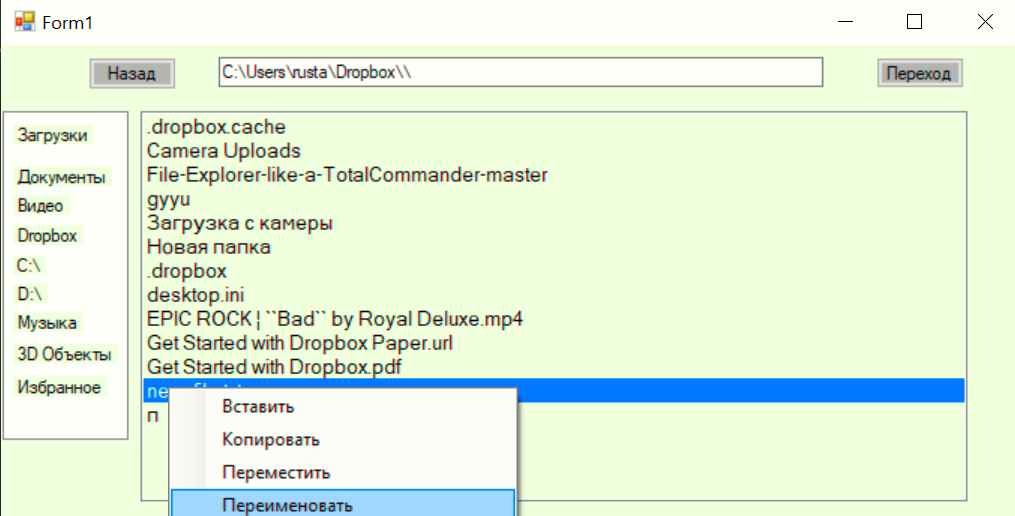


Рисунок 10 - Результат прохождения теста 8

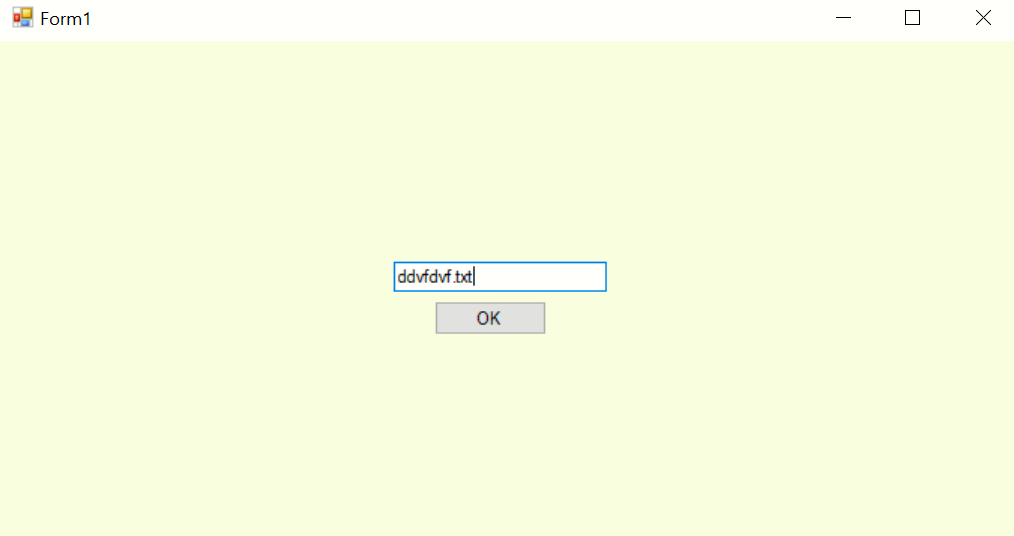


Рисунок 11 - Результат прохождения теста 9

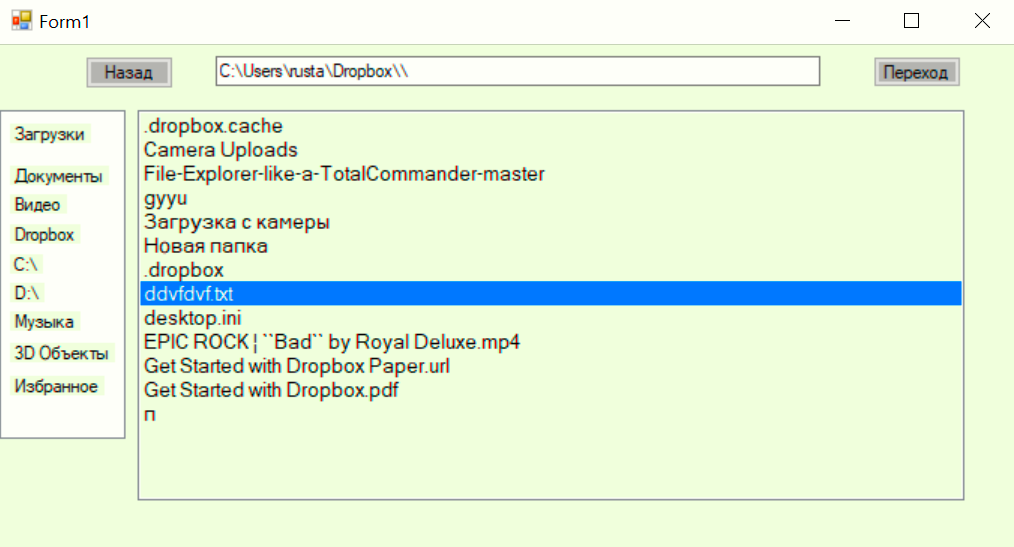


Рисунок 12 - Результат прохождения теста 10

## Протоколы проверки требований к пояснительной записке

Пояснительная записка была проверена и полностью соответствует требованиям.

## Протоколы проверки требований к компиляции исходных кодов

## Все файлы, содержащие исходный код были успешно откомпилированы и соответствуют всем требованиям к компиляции в Visual Studio.

# ЗАКЛЮЧЕНИЕ

В заключение хочется сказать, что существует очень много файловых менеджеров, один похожи друг на друга, другие уникальны. Моей целью было сделать понятный файловый менеджер, чтобы обычный пользователь мог без проблем его использовать. Без сомнения файловые менеджеры очень облегчают жизнь пользователя. Выбирать файловый менеджер лучше по своему предпочтению и необходимому функционалу, который требуется пользователю.

# СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

1. ГОСТ 19.301-79 «Программа и методика испытаний» (12.01.2018)
2. <http://skachate.ru/informatika/208683/index.html> «Файловый менеджер»
3. <https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%B2%D0%BE%D0%B4%D0%BD%D0%B8%D0%BA_Windows> **«**Проводник Windows»
4. <https://works.doklad.ru/view/vLiz-1WUp90.html> «Файловые менеджеры»
5. <https://studopedia.ru/7_107699_faylovie-menedzheri-naznachenie-i-osnovnie-vozmozhnosti-faylovie-menedzheri-Total-Commander-Far-provodnik-Windows.html>

# 

# 

# ПРИЛОЖЕНИЕ А

## Исходный код

# Form1

using System;

using System.IO;

using System.Diagnostics;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace WindowsFormsApp1

{

public partial class Form1 : Form

{

List<string> \_Buffer = new List<string>();

public Form1()

{

InitializeComponent();

}

private void Form1\_Load(object sender, EventArgs e)

{

}

public void update()

{

listBox1.Items.Clear();

DirectoryInfo dir = new DirectoryInfo(textBox1.Text);

DirectoryInfo[] dirs = dir.GetDirectories();

foreach (DirectoryInfo crrDir in dirs)

{

listBox1.Items.Add(crrDir);

}

FileInfo[] files = dir.GetFiles();

foreach (FileInfo crrfile in files)

{

listBox1.Items.Add(crrfile);

}

}

//Загрузка файлов и папок - кнопка "Перейти"

private void button1\_Click(object sender, EventArgs e)

{

textBox1.Text += '\\';

listBox1.Items.Clear();

DirectoryInfo dir = new DirectoryInfo(textBox1.Text);

DirectoryInfo[] dirs = dir.GetDirectories();

foreach (DirectoryInfo crrDir in dirs)

{

listBox1.Items.Add(crrDir);

}

FileInfo[] files = dir.GetFiles();

foreach (FileInfo crrfile in files)

{

listBox1.Items.Add(crrfile);

}

}

//двойной щелчок по элементу listBox

private void listBox1\_MouseDoubleClick(object sender, MouseEventArgs e)

{

if (Path.GetExtension(Path.Combine(textBox1.Text, listBox1.SelectedItem.ToString())) == "")

{

textBox1.Text = Path.Combine(textBox1.Text + '\\' + listBox1.SelectedItem.ToString());

listBox1.Items.Clear();

DirectoryInfo dir = new DirectoryInfo(textBox1.Text);

foreach (DirectoryInfo crrDir in dir.GetDirectories())

{

listBox1.Items.Add(crrDir);

}

FileInfo[] files = dir.GetFiles();

foreach (FileInfo crrfile in files)

{

listBox1.Items.Add(crrfile);

}

}

else

{

try

{

Process.Start(Path.Combine(textBox1.Text, listBox1.SelectedItem.ToString()));

}

catch (System.ComponentModel.Win32Exception)

{

MessageBox.Show("Формат не поддерживается");

}

}

}

//загрузка файлов и папок - кнопка "Назад"

private void button2\_Click(object sender, EventArgs e)

{

if (textBox1.Text[textBox1.Text.Length - 1] == '\\')

{

textBox1.Text = textBox1.Text.Remove(textBox1.Text.Length - 1, 1);

while (textBox1.Text[textBox1.Text.Length - 1] != '\\')

{

textBox1.Text = textBox1.Text.Remove(textBox1.Text.Length - 1, 1);

}

}

else if (textBox1.Text[textBox1.Text.Length - 1] != '\\')

{

while (textBox1.Text[textBox1.Text.Length - 1] != '\\')

{

textBox1.Text = textBox1.Text.Remove(textBox1.Text.Length - 1, 1);

}

}

listBox1.Items.Clear();

DirectoryInfo dir = new DirectoryInfo(textBox1.Text);

DirectoryInfo[] dirs = dir.GetDirectories();

foreach (DirectoryInfo crrDir in dirs)

{

listBox1.Items.Add(crrDir);

}

FileInfo[] files = dir.GetFiles();

foreach (FileInfo crrfile in files)

{

listBox1.Items.Add(crrfile);

}

}

private void copyToolStripMenuItem\_Click(object sender, EventArgs e)

{

textBox1.Copy();

}

private void cutToolStripMenuItem\_Click(object sender, EventArgs e)

{

textBox1.Cut();

}

private void pastToolStripMenuItem\_Click(object sender, EventArgs e)

{

textBox1.Paste();

}

private void clearMenuItem3\_Click(object sender, EventArgs e)

{

listBox1.Items.Clear();

}

private void label1\_Click(object sender, EventArgs e)//Документы

{

textBox1.Text = "C:\\Users\\rusta\\Documents\\"; button1\_Click(sender, e);

}

private void label2\_Click(object sender, EventArgs e)//Видео

{

textBox1.Text = "C:\\Users\\rusta\\Videos"; button1\_Click(sender, e);

}

private void label3\_Click(object sender, EventArgs e)//Dropbox

{

textBox1.Text = "C:\\Users\\rusta\\Dropbox\\"; button1\_Click(sender, e);

}

private void label4\_Click(object sender, EventArgs e)

{

textBox1.Text = "C:\\Users\\"; button1\_Click(sender, e);

}

private void label4\_Click\_1(object sender, EventArgs e)

{

textBox1.Text = "C:"; button1\_Click(sender, e);

}

private void label6\_Click(object sender, EventArgs e)

{

textBox1.Text = "D:"; button1\_Click(sender, e);

}

private void label7\_Click(object sender, EventArgs e)

{

textBox1.Text = "C:\\Users\\rusta\\Music"; button1\_Click(sender, e);

}

private void label8\_Click(object sender, EventArgs e)

{

textBox1.Text = "C:\\Users\\rusta\\3D Objects\\"; button1\_Click(sender, e);

}

private void label9\_Click(object sender, EventArgs e)

{

textBox1.Text = "C:\\Users\\rusta\\Favorites"; button1\_Click(sender, e);

}

private void label10\_Click(object sender, EventArgs e)

{

textBox1.Text = "C:\\Users\\rusta\\Downloads"; button1\_Click(sender, e);

}

private void toolStripMenuItem8\_Click(object sender, EventArgs e)//создать текстовый документ

{

File.Create("C:\\Users\\rusta\\Documents\\");

File.Create("C:\\Users\\rusta\\Favorites\\");

File.Create("C:\\Users\\rusta\\Downloads\\");

File.Create("C:\\Users\\rusta\\Music\\");

File.Create("C:\\Users\\rusta\\3D Objects\\");

File.Create("C:\\Users\\rusta\\Dropbox\\");

File.Create("C:\\Users\\");

File.Create("C:\\Users\\rusta\\Videos\\");

string file = null;

File.SetAttributes(file, FileAttributes.Normal);

File.Create("new\_file.txt");

}

private void toolStripMenuItem9\_Click(object sender, EventArgs e)//удалить текстовый документ

{

if (listBox1.SelectedItem.ToString() != null)

{ remove(textBox1.Text + '\\' + listBox1.SelectedItem.ToString()); }

else

{

remove(textBox1.Text);

}

update();

}

private void toolStripMenuItem7\_Click(object sender, EventArgs e)

{

var Form2 = new Form2();

Form2.Show();

Directory.CreateDirectory("C:\\Users\\rusta\\Documents\\");

Directory.CreateDirectory("C:\\Users\\rusta\\Documents\\");

Directory.CreateDirectory("C:\\Users\\rusta\\Favorites\\");

Directory.CreateDirectory("C:\\Users\\rusta\\Downloads\\");

Directory.CreateDirectory("C:\\Users\\rusta\\Music\\");

Directory.CreateDirectory("C:\\Users\\rusta\\3D Objects\\");

Directory.CreateDirectory("C:\\Users\\rusta\\Dropbox\\");

Directory.CreateDirectory("C:\\Users\\");

Directory.CreateDirectory("C:\\Users\\rusta\\Videos\\");

}

public void remove\_file(string path)

{

if (System.IO.Directory.Exists(path))

{

MessageBox.Show("This is a not file");

}

System.IO.File.Delete(path);

}

private void remove\_folder(string path)

{

DirectoryInfo dir = new DirectoryInfo(path);

foreach (FileInfo fi in dir.GetFiles())

{

fi.Delete();

}

foreach (DirectoryInfo di in dir.GetDirectories())

{

remove\_folder(di.FullName);

di.Delete();

}

dir.Delete(true);

}

public void remove(string path)

{

if (System.IO.File.Exists(path))

{

remove\_file(path);

}

else if (System.IO.Directory.Exists(path))

{

remove\_folder(path);

}

}

private void rename(string path, string name, string new\_name)

{

if (System.IO.File.Exists(path))

{

System.IO.File.Move(Path.Combine(path, name), Path.Combine(path, new\_name));

}

else if (System.IO.Directory.Exists(path))

{

System.IO.Directory.Move(Path.Combine(path, name), Path.Combine(path, new\_name));

}

}

public void buffer\_clear()

{

\_Buffer.Clear();

}

public void copy(string path)

{

buffer\_clear();

if (System.IO.File.Exists(path))

{

\_Buffer.Add(path);

}

else if (System.IO.Directory.Exists(path))

{

foreach (var file in Directory.EnumerateFiles(path, "\*", SearchOption.AllDirectories))

{

\_Buffer.Add(file.ToString());

}

}

}

public void paste(string path)

{

if (System.IO.File.Exists(path))

{

return;

}

else if (System.IO.Directory.Exists(path))

{

foreach (string \_ in \_Buffer)

{

string[] \_\_ = \_.Split('\\');

System.IO.File.Copy(\_, path + "\\" + \_\_[\_\_.Length - 1]);

}

}

update();

}

private void ListBox1\_SelectedIndexChanged(object sender, EventArgs e){}

private void ToolStripMenuItem6\_Click(object sender, EventArgs e)

{

textBox2.Visible = true;

button3.Visible = true;

listBox1.Visible = false;

treeView1.Visible = false;

textBox1.Visible = false;

button2.Visible = false;

button1.Visible = false;

label1.Visible = false;

label2.Visible = false;

label3.Visible = false;

label4.Visible = false;

label5.Visible = false;

label6.Visible = false;

label7.Visible = false;

label8.Visible = false;

label9.Visible = false;

label10.Visible = false;

}

private void ToolStripMenuItem4\_Click(object sender, EventArgs e)

{

if (listBox1.SelectedItem != null)

{ copy(textBox1.Text + '\\' + listBox1.SelectedItem.ToString()); }

else

{

copy(textBox1.Text);

}

update();

}

private void ToolStripMenuItem3\_Click(object sender, EventArgs e)

{

if (listBox1.SelectedItem != null)

{ paste(textBox1.Text + '\\' + listBox1.SelectedItem.ToString());}

else

{

paste(textBox1.Text);

}

update();

}

private void Button3\_Click(object sender, EventArgs e)

{

textBox2.Visible = false;

button3.Visible = false;

listBox1.Visible = true;

treeView1.Visible = true;

textBox1.Visible = true;

button2.Visible = true;

button1.Visible = true;

label1.Visible = true;

label2.Visible = true;

label3.Visible = true;

label4.Visible = true;

label5.Visible = true;

label6.Visible = true;

label7.Visible = true;

label8.Visible = true;

label9.Visible = true;

label10.Visible = true;

rename(textBox1.Text, listBox1.SelectedItem.ToString(), textBox2.Text);

update();

}

}

}