Название проекта

Это консольное приложение - "Файловый менеджер", которое содержит набор нумерованных текстовых меню и может выполнять следующие операции:

1) просмотр списка дисков компьютера и выбор диска;

2) переход в другую директорию (выбор папки);

3) просмотр списка файлов в директории;

4) вывод содержимого текстового файла в консоль в выбранной пользователем кодировке;

5) копирование файла;

6) перемещение файла в выбранную пользователем директорию;

7) удаление файла;

8) создание простого текстового файла в выбранной пользователем кодировке;

9) конкатенация содержимого двух или более текстовых файлов и вывод результата в консоль в кодировке UTF-8.

Данное консольное приложение разработано для C# .NET 5.0. Тип проекта - Console Application .NET 5.0.

Работа с проектом

Это инструкция по запуску консольного приложения на вашем локальном компьютере с целью его тестирования и проверки корректности работы.

Для получения подробной информации о требованиях к программному обеспечению, распаковке zip-архива, операциях с файлами и тестах прочитайте следующие параграфы.

Требования ПО

Одна из следующих операционных систем:

-Apple macOS;

-Microsoft Windows;

-Linux Operating System.

Программа для выполнения консольного приложения:

-Microsoft Visual Studio.

Распаковка zip-архива

Скачайте zip-архив на свой компьютер и откройте его. Там вы найдете этот файл "README.docx" и каталог "FileManager", содержащий "FileManager.sln", "FileManager.csproj", "Program.cs". Загрузите все на свой компьютер, прочитайте этот файл и затем запустите консольное приложение.

Выполнение «Файлового менеджера»

Для начала Вас поприветствуют и расскажут про доступные операции с файлами, а затем попросят выбрать операцию. Прежде чем выполнять операции, давайте разберемся, как я понял каждую операцию и как ее реализовал.

1) просмотр списка дисков компьютера и выбор диска;

При выборе данной операции вам на экран выводится список дисков компьютера, доступных для работы. Затем предлагается выбрать диск. Вы печатаете путь к диску или же просто копируете один из предложенных вариантов. Теперь вы выбрали конкретный диск для работы, при дальнейшем взаимодействии с программой он будет местом по умолчанию (то есть текущим местом, в котором вы находитесь). В случае, если вы указали неверный путь к диску или же диск не готов к работе, вы получите соответствующее сообщение, а затем вас попросят повторить свой выбор до тех пор, пока вы не выберете верный диск для работы.

2) переход в другую директорию (выбор папки);

Сначала вам выводится информация о той директории, в которой вы находитесь в данный момент, и предлагается указать путь к новой директории. Вы должны прописать полный путь к той директории, в которую вы хотите попасть. Если вы укажете неверную директорию, то выведется сообщение об ошибке и вас попросят повторить свой выбор до тех пор, пока вы не укажете верную директорию для работы.

3) просмотр списка файлов в директории;

Сначала вам выводится информация о той директории, в которой вы находитесь на данный момент. Чтобы просмотреть список ее файлов, необходимо нажать "ENTER", а чтобы просмотреть список желаемой директории, необходимо нажать "BACKSPACE" указать полный путь к ней вручную. Если вы укажете неверную директорию, то выведется сообщение об ошибке и вас попросят повторить свой выбор до тех пор, пока вы не укажете верную директорию для просмотра списка ее файлов.

4) вывод содержимого текстового файла в консоль в выбранной пользователем кодировке (UTF-8, ASCII, UTF-16, UTF-32, Latin1 (ISO-8859-1));

Сначала вам предлагается указать путь к текстовому файлу, внимание! название файла должно быть полным, то есть содержать как имя файла, так и его расширение, а именно “example.txt”, а затем предлагается выбрать одну из пяти доступных кодировок. Текст файла выводится в консоль.

Важные примечания!

1. Если ваш текстовый файл содержит русские слова, а вы выбрали кодировку, в таблице которой отсутствуют русские буквы (например, ASCII), и на экране у вас появилось вопросительных знаков (?) или иных символов, это не значит, что программа неверно работает, это значит, что при декодировании исходного файла в указанную кодировку неопознанные символы были изменены на символы или коды по умолчанию (все это зависит от кодировки). В этом плане я могу уверенно ссылаться на документацию Майкрософт (тут пример про кодировку ASCII: <https://docs.microsoft.com/ru-ru/dotnet/api/system.text.encoding.ascii?view=netframework-4.8> , с другими кодировками аналогично).
2. В основной директории "FileManager" по пути "FileManager -> bin -> Debug -> net5.0" лежит папка “ex” с текстовыми файлами для проверки и использования их в качестве инструментов для тестирования.

5) копирование файла;

Вам предлагается указать путь к исходному файлу (название файла должно быть полным, то есть содержать как имя файла, так и его расширение, а именно “example.txt”). Затем вам предлагается указать путь к директории, в которую вы хотите скопировать данный файл. Если вы указываете неверный путь или такой файл уже существует в указанной директории, то вам необходимо повторить выбор. Затем файл копируется в указанную директорию.

6) перемещение файла в выбранную пользователем директорию;

Данный пункт аналогичен предыдущему. Указываете полный путь к исходному файлу (файл содержит название + расширение), затем путь к новой директории, и файл перемещается в указанную директорию.

7) удаление файла;

Вам предлагается указать путь к файлу (файл содержит название + расширение), который вы собираетесь удалить, и файл удаляется.

8) создание текстового файла в консоль в выбранной пользователем кодировке (UTF-8, ASCII, UTF-16, UTF-32, Latin1 (ISO-8859-1));

Вам предлагается указать путь к директории вместе с названием нового файла (название файла должно быть полным, то есть содержать как имя файла, так и его расширение, а именно “example.txt”). Затем предлагается выбрать одну из пяти доступных кодировок. Файл в указанной кодировке создается.

9) конкатенация содержимого двух или более текстовых файлов и вывод

результата в консоль в кодировке UTF-8.

Вам предлагается ввести количество файлов для конкатенации (от 1 до 10). Затем вы поочередно указываете полный путь к каждому файлу (название файла должно быть полным, то есть содержать как имя файла, так и его расширение, а именно “example.txt”). Содержимое файлов построчно сливается и выводится на экран через пробел (например, первая строка первого файла “hello”, а первая строка второго файла “world”, результат конкатенации выводится на консоль -> “hello world”).

Общие примечания:

1. Метод GetDestEncoding() класса Operations содержит 31 строку вместо нужных 30, но там блок switch, прошу опустить этот момент при проверке.
2. Метод PrintFileData() класса Operations тоже больше нормы, он содержит 35 строк, прошу также опустить этот момент, потому что я не стал делать его меньше, чтобы не разбивать логику метода. Более того, в нем есть 3 строки с комментариями, также один вывод разбивается в две строки.
3. Метод MergeTextFiles() класса Operations содержит 31 строку вместо нужных 30, но там есть две строки с комментариями, прошу опустить этот момент при проверке.

Тесты

Приложение протестировано разработчиком на абсолютно разных тестах, все успешно работает и выводит на экран нужный результат.

Написано на

Программа написана с использованием C# .NET 5 и JetBrains IDE «Rider». Там она и была протестирована.

Тем не менее, основные требования для тестирования были для Visual Studio. Поэтому моя программа была также успешно запущена и протестирована в Visual Studio.

Автор

Анонимный аноним! Желаю удачной проверки)

Изображение выглядит как внешний, жираф, млекопитающее, закрыть

Автоматически созданное описание