**Задание:** Консольный проводник файловой системы

**Цель:**

Создать консольное приложение, которое будет выполнять функции простого файлового проводника. Пользователь сможет просматривать содержимое каталогов, открывать файлы, создавать новые каталоги и файлы, а также удалять их.

Функциональные требования:

**Просмотр содержимого каталога:**

При запуске приложения отобразите содержимое текущего каталога.

Пользователь должен иметь возможность перемещаться по каталогам (входить в подкаталоги и возвращаться в родительский каталог).

**Открытие файлов:**

Пользователь должен иметь возможность открывать текстовые файлы для просмотра их содержимого.

**Создание нового каталога:**

Пользователь должен иметь возможность создавать новый каталог в текущем каталоге.

**Создание нового файла:**

Пользователь должен иметь возможность создавать новый текстовый файл в текущем каталоге и записывать в него текст.

**Удаление файлов и каталогов:**

Пользователь должен иметь возможность удалять файлы и каталоги (с подтверждением).

**Выход из приложения:**

Пользователь должен иметь возможность выйти из приложения.

**"Консольный проводник файловой системы"**

**Задание:** Приложение для работы с дисками

**Цель:**

Создать консольное приложение на C#, которое будет предоставлять пользователю возможность управлять дисками и их содержимым. Приложение должно позволять пользователю просматривать доступные диски, получать информацию о каждом диске, а также выполнять операции с файлами и каталогами на выбранном диске.

Функциональные требования:

**Просмотр доступных дисков:**

При запуске приложения отобразите список всех доступных дисков на компьютере.

**Получение информации о диске:**

Пользователь должен иметь возможность выбрать диск и получить информацию о нем (например, объем, доступное пространство, файловая система).

**Просмотр содержимого диска:**

Пользователь должен иметь возможность просматривать содержимое выбранного диска (каталоги и файлы).

**Создание нового каталога:**

Пользователь должен иметь возможность создавать новый каталог на выбранном диске.

**Создание нового файла:**

Пользователь должен иметь возможность создавать новый текстовый файл на выбранном диске и записывать в него текст.

**Удаление файлов и каталогов:**

Пользователь должен иметь возможность удалять файлы и каталоги (с подтверждением).

**Выход из приложения:**

Пользователь должен иметь возможность выйти из приложения.

**Задание:** Модификация консольного проводника с возможностью создания файлов

**Цель:**

Расширить функциональность консольного проводника, добавив возможность создания текстовых файлов с пользовательским содержимым. Пользователь должен иметь возможность вводить текст, который будет записан в новый файл.

**Функциональные требования:**

Просмотр доступных дисков:

При запуске приложения отобразите список всех доступных дисков на компьютере.

**Получение информации о диске:**

Пользователь должен иметь возможность выбрать диск и получить информацию о нем (например, объем, доступное пространство, файловая система).

**Просмотр содержимого диска:**

Пользователь должен иметь возможность просматривать содержимое выбранного диска (каталоги и файлы).

**Создание нового каталога:**

Пользователь должен иметь возможность создавать новый каталог на выбранном диске.

**Создание нового текстового файла:**

Пользователь должен иметь возможность создавать новый текстовый файл на выбранном диске.

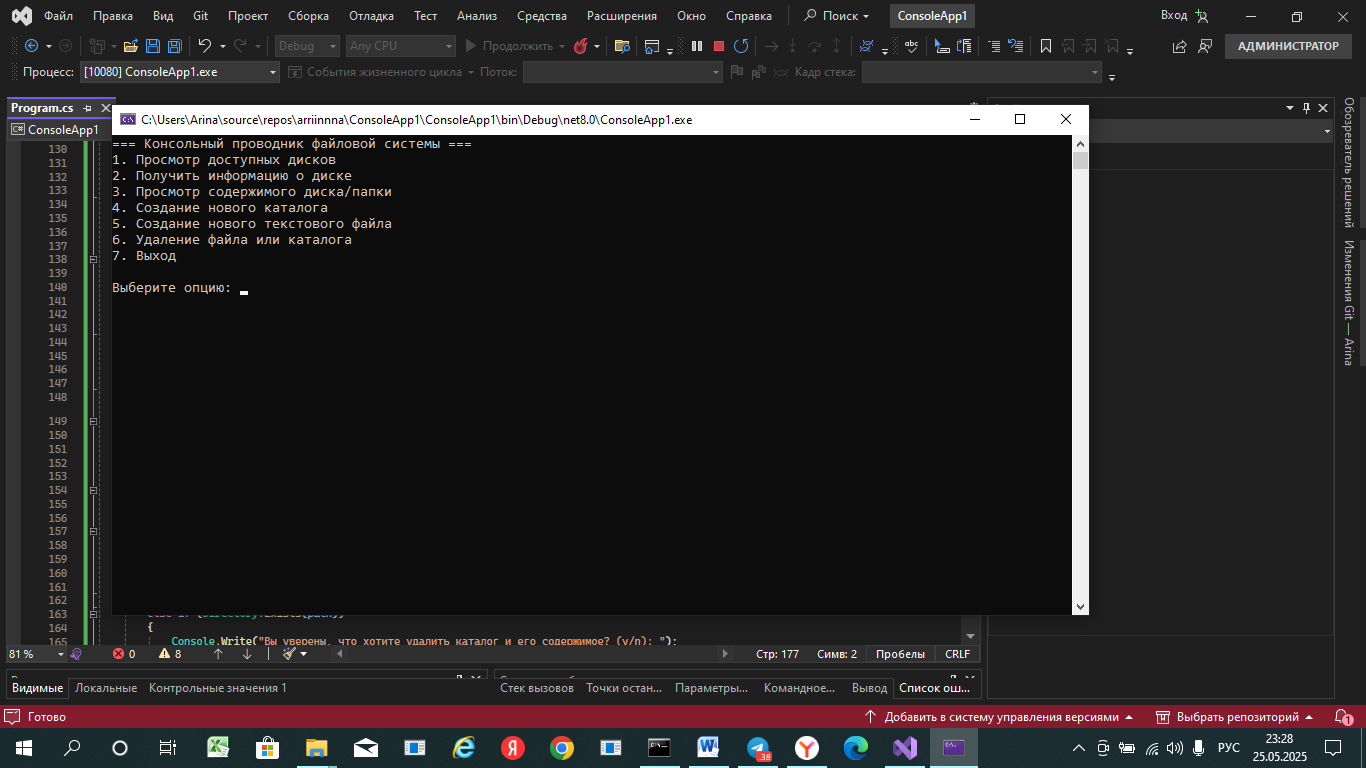
Программа должна запрашивать у пользователя текст для записи в файл.

**Удаление файлов и каталогов:**

Пользователь должен иметь возможность удалять файлы и каталоги (с подтверждением).

**Выход из приложения:**

Пользователь должен иметь возможность выйти из приложения.

Мой получившийся результат:  


using System;

using System.IO;

class FileExplorer

{

static void Main()

{

while (true)

{

Console.Clear();

Console.WriteLine("=== Консольный проводник файловой системы ===");

Console.WriteLine("1. Просмотр доступных дисков");

Console.WriteLine("2. Получить информацию о диске");

Console.WriteLine("3. Просмотр содержимого диска/папки");

Console.WriteLine("4. Создание нового каталога");

Console.WriteLine("5. Создание нового текстового файла");

Console.WriteLine("6. Удаление файла или каталога");

Console.WriteLine("7. Выход");

Console.Write("\nВыберите опцию: ");

string input = Console.ReadLine();

switch (input)

{

case "1":

ListDrives();

break;

case "2":

DiskInfo();

break;

case "3":

BrowseDirectory();

break;

case "4":

CreateDirectory();

break;

case "5":

CreateTextFile();

break;

case "6":

DeleteFileOrDirectory();

break;

case "7":

return;

default:

Console.WriteLine("Неверный выбор.");

break;

}

Console.WriteLine("\nНажмите любую клавишу для продолжения...");

Console.ReadKey();

}

}

static void ListDrives()

{

Console.WriteLine("Доступные диски:");

foreach (DriveInfo drive in DriveInfo.GetDrives())

{

Console.WriteLine($"{drive.Name} ({(drive.IsReady ? drive.VolumeLabel : "Недоступен")})");

}

}

static void DiskInfo()

{

Console.Write("Введите букву диска (например, C): ");

string driveLetter = Console.ReadLine()?.ToUpper() + ":\\";

DriveInfo drive = new DriveInfo(driveLetter);

if (!drive.IsReady)

{

Console.WriteLine("Диск недоступен.");

return;

}

Console.WriteLine($"Имя: {drive.Name}");

Console.WriteLine($"Метка тома: {drive.VolumeLabel}");

Console.WriteLine($"Файловая система: {drive.DriveFormat}");

Console.WriteLine($"Объем: {drive.TotalSize / (1024 \* 1024)} MB");

Console.WriteLine($"Свободно: {drive.AvailableFreeSpace / (1024 \* 1024)} MB");

}

static void BrowseDirectory()

{

Console.Write("Введите путь (например, C:\\): ");

string path = Console.ReadLine();

if (!Directory.Exists(path))

{

Console.WriteLine("Путь не найден.");

return;

}

Console.WriteLine("Каталоги:");

foreach (string dir in Directory.GetDirectories(path))

{

Console.WriteLine("📁 " + dir);

}

Console.WriteLine("\nФайлы:");

foreach (string file in Directory.GetFiles(path))

{

Console.WriteLine("📄 " + file);

}

}

static void CreateDirectory()

{

Console.Write("Введите путь, где создать каталог (например, C:\\Test): ");

string path = Console.ReadLine();

if (!Directory.Exists(path))

{

Directory.CreateDirectory(path);

Console.WriteLine("Каталог создан.");

}

else

{

Console.WriteLine("Каталог уже существует.");

}

}

static void CreateTextFile()

{

Console.Write("Введите путь к файлу (например, C:\\Test\\file.txt): ");

string filePath = Console.ReadLine();

if (File.Exists(filePath))

{

Console.WriteLine("Файл уже существует.");

return;

}

Console.WriteLine("Введите текст для записи в файл (для завершения введите пустую строку):");

string content = "";

string line;

do

{

line = Console.ReadLine();

if (line != "")

content += line + Environment.NewLine;

} while (line != "");

File.WriteAllText(filePath, content);

Console.WriteLine("Файл создан и записан.");

}

static void DeleteFileOrDirectory()

{

Console.Write("Введите путь к файлу или каталогу для удаления: ");

string path = Console.ReadLine();

if (File.Exists(path))

{

Console.Write("Вы уверены, что хотите удалить файл? (y/n): ");

if (Console.ReadLine()?.ToLower() == "y")

{

File.Delete(path);

Console.WriteLine("Файл удален.");

}

}

else if (Directory.Exists(path))

{

Console.Write("Вы уверены, что хотите удалить каталог и его содержимое? (y/n): ");

if (Console.ReadLine()?.ToLower() == "y")

{

Directory.Delete(path, true);

Console.WriteLine("Каталог удален.");

}

}

else

{

Console.WriteLine("Файл или каталог не найден.");

}

}

}