

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ВІДОКРЕМЛЕНИЙ СТРУКТУРНИЙ ПІДРОЗДІЛ
«ПОЛТАВСЬКИЙ ПОЛІТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ
НАЦІОНАЛЬНОГО ТЕХНІЧНОГО УНІВЕРСИТЕТУ
«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

Циклова комісія дисциплін програмної інженерії

КУРСОВА РОБОТА

з дисципліни «Основи програмування та АМ»

на тему Розробка файлового менеджера

Виконав: здобувач освіти 2 курсу,
групи 24
спеціалізації «Розробка програмного
забезпечення»

Чорний Р.Д.

(прізвище та ініціали)

Керівник _____

(підпис)

Олійник В.В.

(прізвище та ініціали)

Полтава – 2023

ЗМІСТ

ВСТУП	3
1. ПОСТАНОВКА ЗАДАЧІ.....	4
1.1. Основні вимоги до програми	4
2. ОПИС ПРОЕКТНИХ РІШЕНЬ.....	5
2.1. Алгоритм програми.....	5
2.2. Опис програмних засобів та програмного забезпечення	7
2.3. Тестування	11
3. ІНСТРУКЦІЯ КОРИСТУВАЧА	15
ВИСНОВКИ.....	28
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ	29
ДОДАТОК А. ЛІСТИНГ ПРОГРАМИ	31

ВСТУП

Метою цієї курсової роботи є розробка консольного файлового менеджера з використанням мови програмування C++. Ця робота спрямована на закріплення, поглиблення та узагальнення фундаментальних теоретичних знань з програмування та їх застосування для вирішення конкретного завдання.

Мова програмування C++ є ефективною та потужною мовою, яка широко використовується в розробці операційних систем, компіляторів і вбудованих систем завдяки її функціям низькорівневого керування пам'яттю і в той же час підтримкою конструкцій програмування високого рівня.

Менеджери файлів є основними інструментами, які дозволяють користувачам керувати файлами та каталогами на своєму комп'ютері. Вони забезпечують графічний інтерфейс користувача, який дозволяє користувачам виконувати такі завдання, як копіювання, переміщення, видалення та перейменування файлів і каталогів. Однак іноді консольний файловий менеджер може бути навіть більш ефективним і легким, ніж графічний, особливо у випадках, коли користувач має обмежені ресурси або потребує виконання завдань віддалено.

Розробка такої програми на сьогоднішній день є досить актуальною через її швидкість роботи, простоту і зручність використання. Враховуючи швидкий темп зростання обсягу цифрових даних, користувачам стає дедалі складніше підтримувати свої файли впорядкованими та швидкодоступними. Менеджери файлів допомагають спростити процес керування файлами, полегшуючи користувачам пошук і швидкий доступ до своїх даних.

1. ПОСТАНОВКА ЗАДАЧІ

Основні функції будь-якого файлового менеджера націлені на вирішення наступних завдань:

- перейменування файлів;
- забезпечення програмного інтерфейсу роботи з файлами для додатків;
- відображення логічної моделі файлової системи на фізичну організацію сховища даних;
- організація стійкості файлової системи до збоїв живлення, апаратних помилок і програмних засобів;
- зміна параметрів файла, необхідних для його правильної роботи з іншими компонентами системи;

Окрім цього завданням файлового менеджера є також захист файлів одного користувача від несанкціонованого доступу іншого користувача, і навіть забезпечення спільної роботи з файлами, приміром, під час відкриття файла одним з користувачів, іншому користувачеві той самий файл тимчасово буде доступний лише як «для читання».

1.1. Основні вимоги до програми

Програма яка буде розроблена, повинна забезпечити користувачеві можливість керування файлами та каталогами в консольному середовищі. Програма дозволить користувачеві створювати, видаляти, переміщувати та копіювати файли та каталоги, змінювати права доступу та відображати інформацію про окремі файли і їх атрибути.

Інтерфейс програми повинен бути простий і зручний, програма повинна мати повноцінне і функціональне меню.

Для розробки програми було вирішено використовувати мову програмування C++ і середовище розробки Microsoft Visual Studio.

2. ОПИС ПРОЕКТНИХ РІШЕНЬ

2.1. Алгоритм програми

Запустивши програму, спочатку з'являється меню з двох пунктів.

При виборі першого пункту з'явиться рядок з підказкою ввести шлях (директорію) в файловій системі користувача, в якій він бажає зараз працювати.

При виборі другого пункту меню програма запропонує вибрати директорію в якій бажає працювати користувач і розпочати роботу за допомогою інтерфейсу самого файлового менеджера, відображаючи спочатку список логічних дисків доступних в системі, звідки користувач зможе переміститися до потрібної йому папки.

Вибравши шлях, з яким буде працювати користувач, програма зверху відобразить меню в якому можна вибрати опції для загального керування файловою системою і перегляду файлів та папок (перехід до іншої папки, повернення до папки рівнем вище, зміна робочого шляху тощо), після цього відобразить список усіх файлів і папок у даній директорії.

Після відображення списку, користувач зможе вводити спеціальні команди файлому менеджеру, а також проводити різноматні дії з файлами та папками.

Тепер, при вводі імені будь-якого файлу або папки, програма відобразить невелике меню з опціями дій, які можна виконати над даним файлом або папкою (скопіювати, вирізати, перейменувати, видалити, відобразити, змінити властивості і т. д). При виборі відповідної опції меню, програма виконає задані дії над файлом чи папкою обраною користувачем.

Алгоритм програми роботи в консолі представлений блок-схемою на рисунку 2.1.

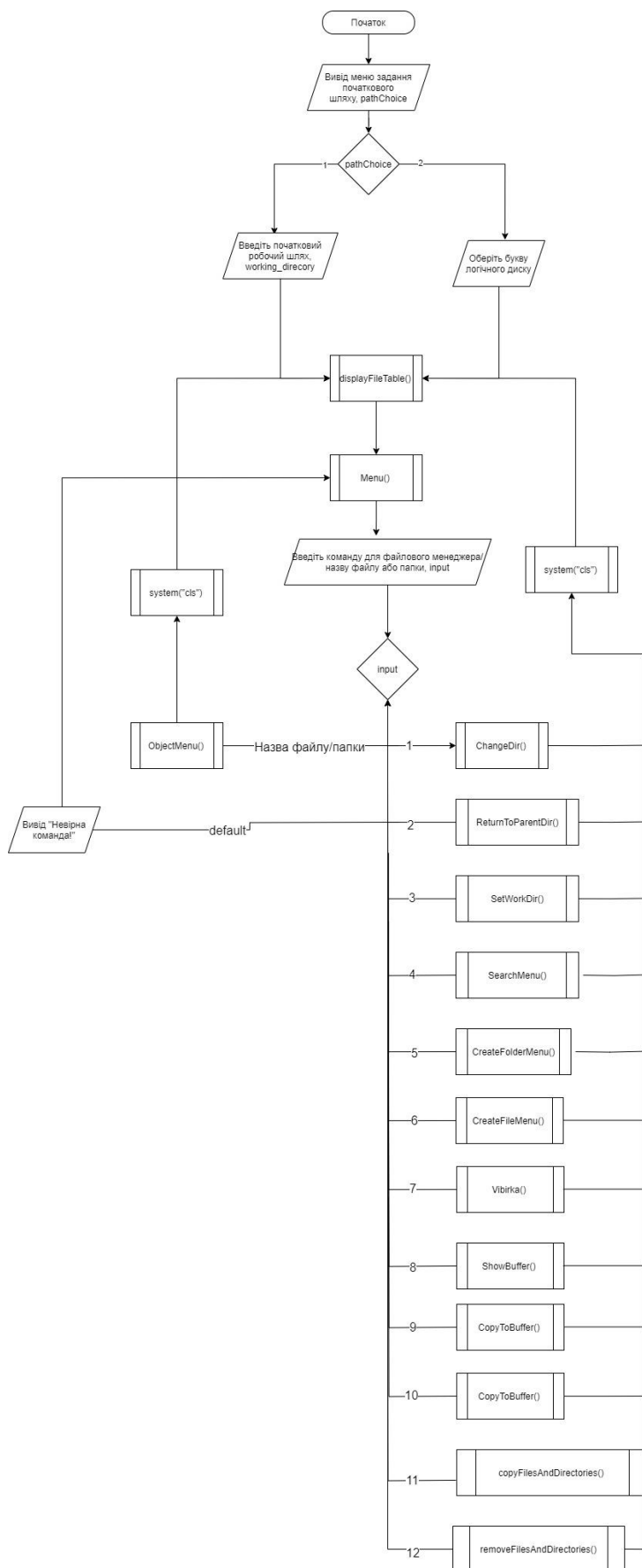


Рисунок 2.1 – Алгоритм роботи програми

2.2. Опис програмних засобів та програмного забезпечення

Мова C++ — це універсальна мова програмування високого рівня з підтримкою кількох парадигм програмування, таких як об'єктно-орієнтоване, загальне та процедурне програмування. Розроблена Б'ярном Страуструпом в AT&T Bell Laboratories у 1979 році, C++ є прямим нащадком мови C, що пропонує програмістам велику гнучкість і велику стандартну бібліотеку. За останні десятиліття C++ зарекомендувала себе як одна з переважаючих мов програмування загального призначення, яка використовується для системного програмування, розробки програмного забезпечення, драйверів, вбудованих систем, серверних і клієнтських програм, а також відеоігор та іншого програмного забезпечення для розваг.

Основні елементи, які використовувались для написання програми наведені у таблиці 2.1.

Таблиця 2.1 Опис специфікації функцій.

№ п/п	Назва функції	Призначення функції	Заголовочний файл
1	time_point_cast()	Конвертація заданої мітки часу до іншого типу	<filesystem>
2	TP::clock::now()	Отримання поточного часу	<filesystem>
3	system_clock::now()	Отримання поточного часу за допомогою класу	<filesystem>
4	system_clock::to_time_t()	Конвертація заданої мітки часу з типу system_clock у тип time_t	<filesystem>
5	fs::status()	Перевірка статусу (існування) шляху	<filesystem>
6	fs::is_regular_file()	Визначення чи є заданий шлях файлом	<filesystem>
7	fs::file_size()	Отримання розміру файлу в байтах	<filesystem>

Продовження таблиці 2.2 – Опис стандартних функцій

8	<code>fs::is_directory()</code>	Визначення чи є заданий шлях директорією	<code><filesystem></code>
9	<code>std::setw()</code>	Встановлення ширини поля для відображення наступного форматowanego виводу	<code><iomanip></code>
10	<code>fs::last_write_time()</code>	Отримання часу останньої зміни файлу	<code><filesystem></code>
11	<code>std::localtime()</code>	Конвертація значення мітки часу до локального формату часу	<code><filesystem></code>
12	<code>GetFileAttributes()</code>	Отримання атрибутів файлу або директорії	<code><windows.h></code>
13	<code>std::filesystem::remove_all()</code>	Видалення директорії та її вмісту або окремого файлу	<code><filesystem></code>
14	<code>std::filesystem::create_directory()</code>	Створення нової директорії	<code><filesystem></code>
15	<code>std::filesystem::copy()</code>	Копіювання вмісту директорії або файлу до іншого шляху	<code><filesystem></code>
16	<code>GetLogicalDrives()</code>	Отримання бітової маски доступних в системі логічних дисків	<code><windows.h></code>
17	<code>GetVolumeInformationA()</code>	Отримання інформації про том	<code><windows.h></code>
18	<code>path.find_last_of()</code>	Знаходження позиції останнього входження символу у рядку	<code><string></code>
19	<code>std::filesystem::create_directory()</code>	Створення нової директорії	<code><filesystem></code>
20	<code>FindFirstFile()</code>	Пошук файлу або піддиректорії з заданим ім'ям у директорії	<code><windows.h></code>
21	<code>SHGetFileInfo()</code>	Отримання інформації про об'єкт, зазвичай файл або папку	<code>Shell32.lib</code>

Продовження таблиці 2.2 – Опис стандартних функцій

22	FileTimeToSystemTime()	Конвертація часу зміни файлу в формат системного часу	<windows.h>
23	file_path.extension().string()	Отримання розширення файлу	<filesystem>
24	SetFileAttributes()	Встановлення атрибутів файлу або директорії	<windows.h>
25	to_time_t()	Конвертація мітки часу останньої модифікації файлу до типу std::time_t	Створена функція
26	IsFolderOrFile()	Перевіряє чи є файлом чи папкою заданий шлях	Створена функція
27	pathExists()	Перевіряє чи існує заданий шлях	Створена функція
28	listLogicalDrives()	Повертає список букв логічних дисків та відображає їх список	Створена функція
29	removeFilesAndDirectories()	Видаляє файли і папки зі списку шляхів переданому в якості параметру	Створена функція
30	copyFilesAndDirectories()	Копіює файли і папки у задане розташування зі списку шляхів переданому в якості параметру	Створена функція
31	getParentDirectory()	Повертає батьківську директорію для заданого шляху (об'єкту)	Створена функція
32	DiskSelectMenu()	Виводить меню вибору логічних дисків	Створена функція
33	SearchFolderAndFile()	Відображає список (таблицю) об'єктів знайдених у заданій директорії по їх назві	Створена функція

Продовження таблиці 2.2 – Опис стандартних функцій

34	SearchMenu()	Відображає меню пошуку об'єктів	Створена функція
35	Create_File()	Створює пустий файл у заданій директорії	Створена функція
36	hasInvalidChars()	Перевіряє, чи не містить переданий рядок символів які не повинні містити імена файлів та папок	Створена функція
37	createFolder()	Створює пусту папку у заданій директорії	Створена функція
38	File_Properties()	Відображає властивості заданого файлу	Створена функція
39	Dir_Properties()	Відображає властивості заданої директорії	Створена функція
40	ShowAttr()	Відображає атрибути заданого об'єкту	Створена функція
41	ObjectMenu()	Відображає меню дій над об'єктом	Створена функція
42	displayFileTable()	Виводить список (таблицю) файлів та папок у заданій директорії	Створена функція

Visual Studio — це широко поширене інтегроване середовище розробки (IDE) для програмування на C++ на платформах Windows, яке нещодавно було розширено для підтримки платформ macOS і Linux. Visual Studio пропонує корисні інструменти, які полегшують розробку програм, наприклад підсвічування синтаксису, доповнення коду, рефакторинг і підтримку налагодження. Крім того, вона інтегрує кілька ключових компонентів, включаючи компілятор Microsoft Visual C++ (MSVC) і платформу .NET, які сприяють спрощеному та ефективному програмуванню.

Компілятор MSVC, що входить до складу Visual Studio, відповідає найновішим мовним стандартам, водночас надаючи різноманітні розширення й оптимізації, характерні для екосистеми Microsoft. Завдяки

високопродуктивній генерації коду та відповідності останнім стандартам C++, MSVC дозволяє розробникам отримати доступ до найсучасніших функцій мови.

2.3. Тестування

На сьогоднішній день існує багато різноманітних методів тестування ПЗ, які використовуються для забезпечення якості програмного продукту перед його випуском на ринок. Кожен метод має свої переваги та недоліки та відповідає певному етапу життєвого циклу програмного забезпечення. За знанням будови системи розрізняють такі методи тестування як:

- Тестування чорної скриньки (black box);
- Тестування білої скриньки (white box);
- Тестування сірої скриньки (gray box).

Кожен з цих методів має свої відмінні особливості, які дозволяють виявляти різноманітні помилки та недоліки в ПЗ.

Тестування чорної скриньки (black box), яке іноді називають тестуванням на основі специфікацій або поведінковим тестуванням — це метод тестування, який зосереджується виключно на зовнішніх інтерфейсах системи, що тестується. Цей тип тестування охоплює як функціональне, так і нефункціональне тестування без будь-яких припущень щодо внутрішньої роботи компонента чи системи. Тестові приклади для такого роду тестування розробляються або вибираються на основі аналізу функціональних або нефункціональних специфікацій компонента або системи без будь-яких знань про їх внутрішню структуру.

Тестування білої скриньки (white box), також: прозорого, відкритого, скляного ящика; засноване на коді або структурне тестування) – метод тестування програмного забезпечення, який передбачає, що внутрішня структура/пристрій/реалізація системи відомі тестувальнику. Ми вибираємо вхідні значення, ґрунтуючись на знанні коду, який буде їх обробляти. Точно так само ми знаємо, яким повинен бути результат цієї обробки. Знання всіх

особливостей тестованої програми та її реалізації – обов’язкові для цієї техніки. Тестування білого ящика – поглиблення у код системи, за межі її зовнішніх інтерфейсів.

Тестування сірої скриньки (gray box) – це метод тестування програмного забезпечення, який передбачає, комбінацію White Box і Black Box підходів. Тобто, внутрішній устрій програми нам відомо лише частково. Передбачається, наприклад, доступ до внутрішньої структури та алгоритмів роботи ПЗ для написання максимально ефективних тест-кейсів, але саме тестування проводиться за допомогою техніки чорного ящика, тобто, з позиції користувача. Цю техніку тестування також називають методом напівпрозорого ящика, коли щось ми бачимо, а щось – ні.

Також, часто використовують довільне тестування (ad-hoc – тестування), яке виконується без підготовки до тестування продукту, без визначення очікуваних результатів, проектування тестових сценаріїв, тобто неформальне, імпровізаційне тестування.

Є такі види довільного, неформального тестування:

- Buddy testing;
- Pair testing;
- Monkey testing;

Buddy testing - процес, коли 2 людини, як правило, розробник і тестувальник, працюють паралельно і знаходять дефекти в одному і тому ж модулі продукту, що тестується. Відразу після того, як розробник завершує модульне тестування, тестувальник та розробник разом працюють над модулем. Цей вид тестування дозволяє обом сторонам розглядати цю функцію у ширшому масштабі.

Pair testing – у цьому тестуванні два тестувальники (краще з різним досвідом) працюють разом над одним модулем. Ідея, що лежить в основі цієї форми тестування, полягає в тому, щоб змусити двох тестувальників провести мозковий штурм ідей і методів, щоб виявити ряд дефектів.

Monkey testing – довільне тестування продукту з метою якнайшвидше, використовуючи різні варіації вхідних даних, порушити роботу програми або викликати її зупинку (простими словами – зламати).

Для проведення тестування розробленої програми «Файловий менеджер» я використув метод тестування White Box, Black Box і Monkey testing.

Для проведення тестування цього програмного продукту методом Monkey testing, була запрошена група людей, в якій деякі з них не мали уявлення про внутрішню структуру та роботу програми, а деякі мали лише коротке уявлення про можливості та функції програми. Після проведення тестування, кожен учасник поділився своїми результатами та досвідом роботи з програмою, назвав можливі недоліки та вразливості програми. Загалом, під час тестування учасниками не було виявлено помилок або серйозних недопрацювань.

Перед проведенням тестування методом Whitw Box, я виявив які «проблемні» ділянки може мати розроблений програмний код, в яких місцях можуть виникати потенційні проблеми або «баги». На основі цього, обрав вхідні дані для тестування, ґрунтуючись на знанні структури програми, її функцій і коду який буде обробляти ці вхідні дані. В процесі тестування та аналізу роботи програми, було виявлено, що у меню вибірки (при попередньому виборі пункту 7 «Відобразити/редагувати список вибірки» меню керування) якщо ввести будь-яку комбінацію символів окрім цифр 0-9, програма вводить меню у безкінечний цикл з постійним виводом напису «Невірний пункт!» (Рисунок 2.1).



Рисунок 2.1 – безкінечний цикл у меню вибірки при вводу символічних значень

Під час перегляду коду програми, була з'ясована причина помилки, яка полягала у некоректній перевірці вхідних даних при виборі пункту списку, а саме перевірці вводу користувача тільки на цілочислові значення, внаслідок чого символічні значення проходили повз перевірку.

Для проведення тестування методом Black Box була запрошена інша людина – тестер, яка не має сильних навичок програмування, не ознайоmlена із загальною структурою, архітектурою та засобами роботи програми. Під час тестування, тестер використовував у якості вхідних значень довільні дані, а також дані, які користувач вірогідніше всього використовував би при звичайній роботі з програмою. В результаті проведення тестування тестером не було виявлено серйозних помилок або недоліків у роботі програми.

В процесі подальшого тестування та налагодження програми, усі виявлені в процесі тестування помилки були виправлені, робота програмного продукту була налагоджена.

3. ІНСТРУКЦІЯ КОРИСТУВАЧА

Програма «Фацловий менеджер» дозволяє користувачу переглядати, змінювати та керувати різноманітними об'єктами файлової системи на встановлених у комп'ютер накопичувачах інформації.

Вимоги програми до системи користувача мінімальні.

Запуск програми відбувається за допомогою файлу FileManager.exe. Одразу після запуску, програма відображає початкове меню, де пропонує вибрати метод задання початкового робочого шляху, тобто вручну або за допомогою інтерфесу програми (Рисунок 3.1).

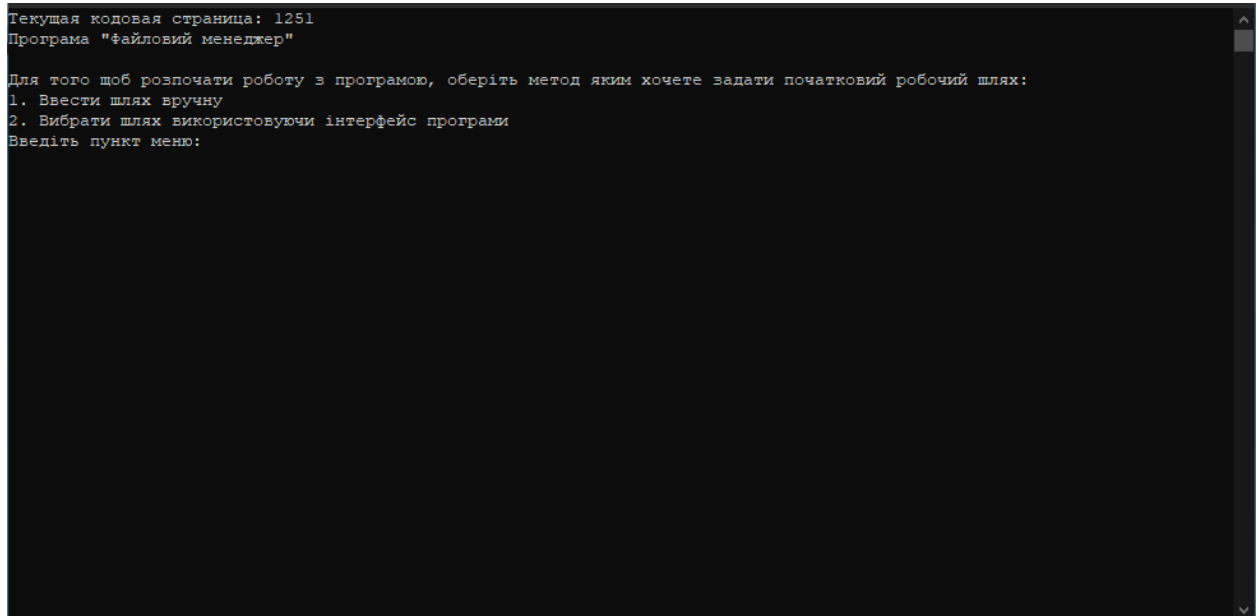


Рисунок 3.1 – початкове меню програми

Оберемо пункт меню «Вибрати шлях використовуючи інтерфейс програми», програма виведе список логічних дисків встановлених на комп'ютері і запропонує обрати один з них, щоб перейти до його вмісту і розпочати роботу (Рисунок 3.2).

```

Текущая кодовая страница: 1251
Програма "файловий менеджер"

Для того щоб розпочати роботу з програмою, оберіть метод яким хочете задати початковий робочий шлях:
1. Ввести шлях вручну
2. Вибрати шлях використовуючи інтерфейс програми
Введіть пункт меню: 2
Список логічних дисків на комп'ютері:
Локальний диск (C:)
Локальний диск (D:)
Новий том (F:)
Введіть букву диску щоб перейти до нього:

```

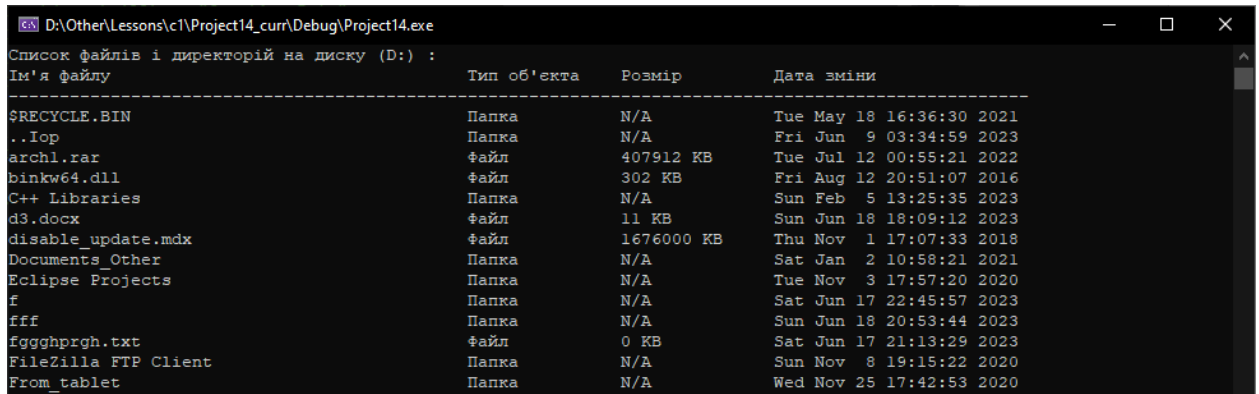
Рисунок 3.2 – відображення списку логічних дисків

При виборі пункту меню «Ввести шлях вручну» програма запропонує ввести повний шлях до директорії у якій ви бажаєте працювати.

Після того як користувач обрав диск на якому бажає працювати, він може використовувати усі базові функції файлового менеджера у будь-якій після цього обраній директорії.

Після виконання будь-якої команди (використання пункту меню) і повернення до головного меню керування, програма запропонує користувачу натиснути клавішу «ENTER» для оновлення списку файлів та директорій і продовження роботи.

Оберемо букву диску «D», програма одразу виведе меню керування файловою системою і під ним список усіх файлів і директорій на цьому диску(Рисунок 3.3).



Ім'я файлу	Тип об'єкта	Розмір	Дата зміни
\$RECYCLE.BIN	Папка	N/A	Tue May 18 16:36:30 2021
..Iop	Папка	N/A	Fri Jun 9 03:34:59 2023
arch1.rar	файл	407912 KB	Tue Jul 12 00:55:21 2022
binkw64.dll	файл	302 KB	Fri Aug 12 20:51:07 2016
C++ Libraries	Папка	N/A	Sun Feb 5 13:25:35 2023
d3.docx	файл	11 KB	Sun Jun 18 18:09:12 2023
disable_update.mdx	файл	1676000 KB	Thu Nov 1 17:07:33 2018
Documents_Other	Папка	N/A	Sat Jan 2 10:58:21 2021
Eclipse Projects	Папка	N/A	Tue Nov 3 17:57:20 2020
f	Папка	N/A	Sat Jun 17 22:45:57 2023
fff	Папка	N/A	Sun Jun 18 20:53:44 2023
fggghprgh.txt	файл	0 KB	Sat Jun 17 21:13:29 2023
FileZilla FTP Client	Папка	N/A	Sun Nov 8 19:15:22 2020
From_tablet	Папка	N/A	Wed Nov 25 17:42:53 2020

Рисунок 3.3 - вміст диску «D»

Щоб переміститися до іншої папки на диску користувач може використати меню, або ж ввести комбінацію символів «./» після чого, не ставлячи пробіл після останнього символу, потрібно ввести назву бажаної папки. Щоб переміститися назад, тобто до директорії рівнем вище, потрібно ввести комбінацію символів «..».

Перемістимося до папки «Documents_Other» яка є у списку виведених об'єктів, для цього введемо комбінацію символів «./» і назву папки після цього (Рисунок 3.4). Тепер, після натискання клавіші «ENTER» програма оновить список файлів та директорій і відобразить вміст папки «Other_Documents» (Рисунок 3.5).

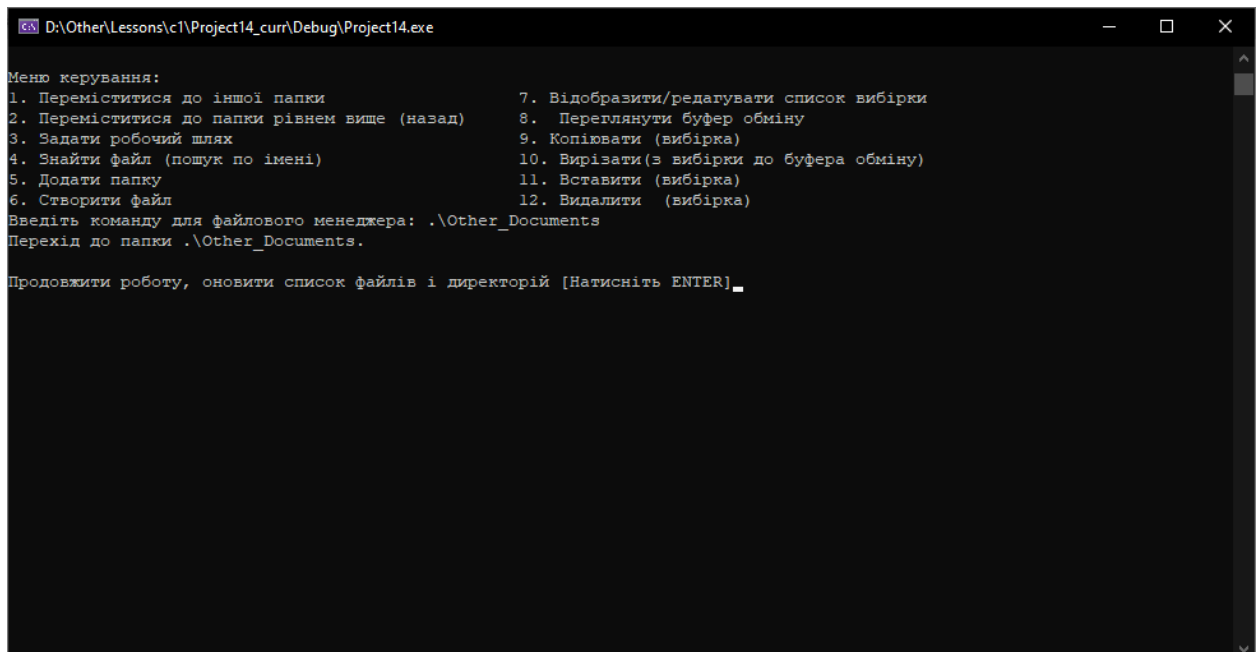


Рисунок 3.4 – перехід до папки «Other_Documents»

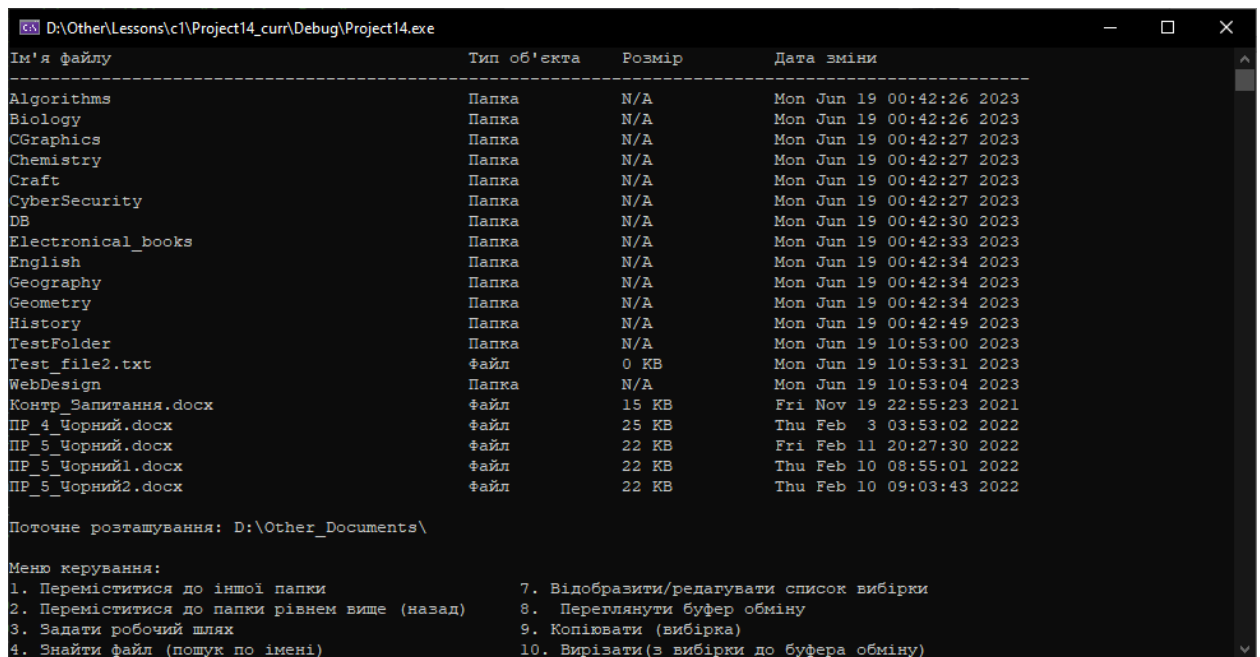


Рисунок 3.5 – вміст папки «Other_Documents»

При виборі пункту 3 («Знайти файл (пошук по імені)»), програма запропонує ввести повне ім'я або частину ім'я певного файлу чи папки яку користувач хоче знайти, після цього відобразить список об'єктів у директорії які мають у своїй назві комбінацію символів яку увів користувач (Рисунок 3.6).

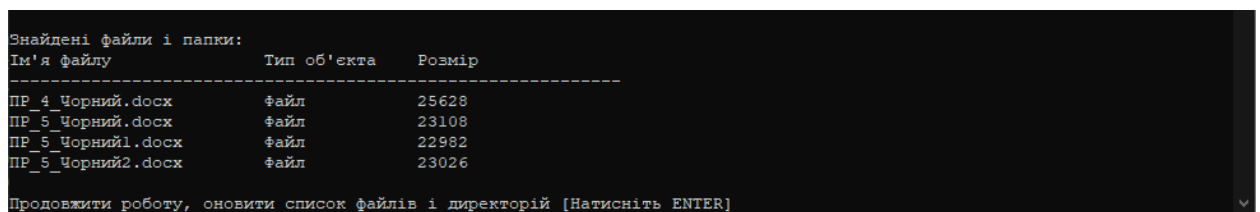


Рисунок 3.6 – список знайдених об'єктів у папці за комбінацією «PR_»

При виборі пункту 5 і 6 користувач може створити папку або файл, для цього програма запропонує ввести довільне ім'я нового об'єкту, на рисунку червоною рамкою виділена тільки що створена папка «TestFolder» (Рисунок 3.7).

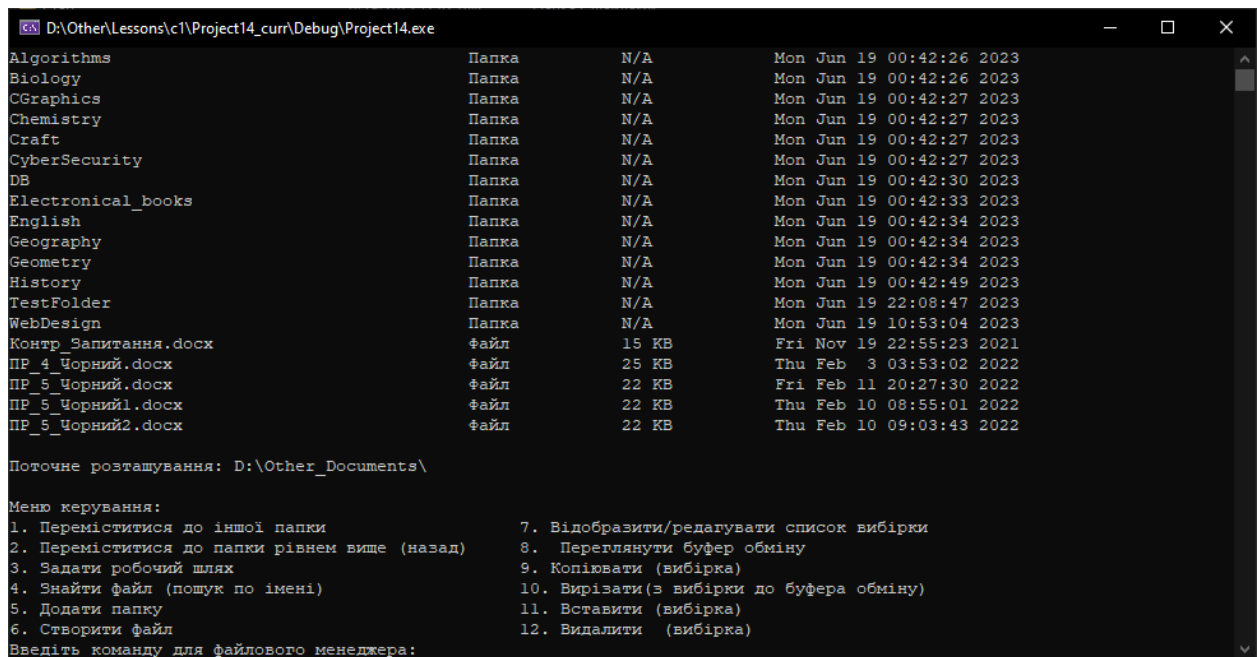


Рисунок 3.7 – створено папку «TestFolder»

Аналогічно відбувається створення нового файлу.

Файловий менеджер також має список вибірки, який дозволяє користувачу проводити операції одночасно над кількома файлами або папками. Щоб переглянути або додати елемент до списку вибірки, необхідно обрати пункт меню 7, після чого програма відобразить невелике меню списку вибірки. При виборі пункту 1, програма запропонує ввести назву файлу або папки в поточній директорії який потрібно додати до списку вибірки (Рисунок 3.8), при виборі пункту 2, користувач також може видалити елемент зі списку. Обравши пункт 3, вміст вибірки буде очищений, пункт 4 дозволяє відобразити поточний вміст списку. Для того щоб повернутися до попереднього меню, користувачу необхідно обрати пункт 5.

```

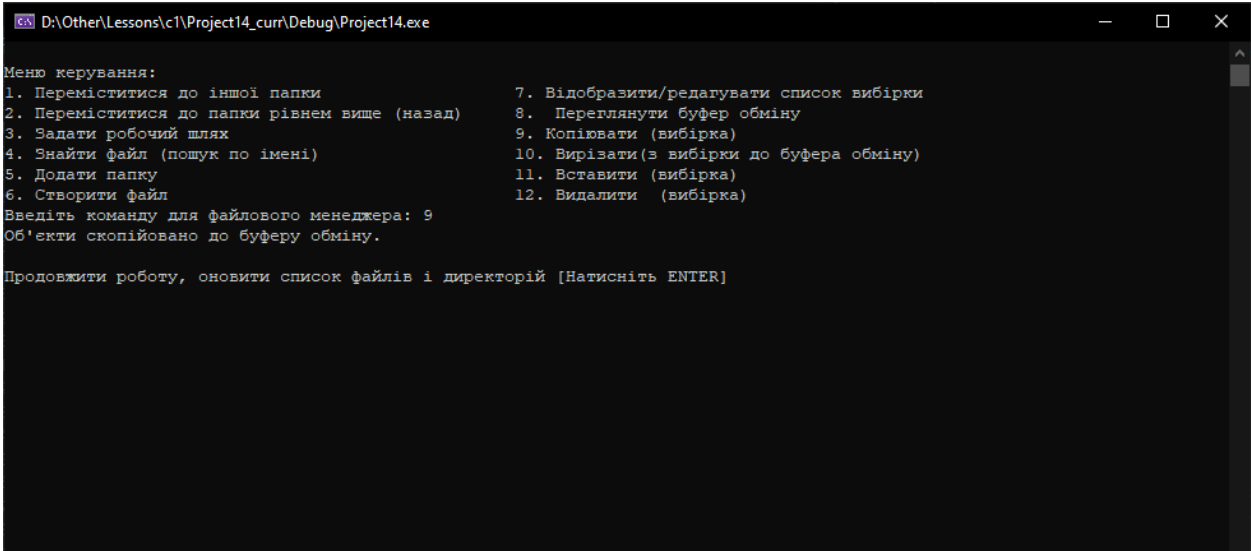
Меню вибірки:
1. Додати об'єкт
2. Прибрати об'єкт
3. Очистити вибірку
4. Відобразити вміст списку
5. Повернутися
Оберіть пункт меню: 1
Введіть назву файлу/папки в поточній директорії: Test_file.txt
Об'єкту додано.
Оберіть пункт меню: 1
Введіть назву файлу/папки в поточній директорії: TestFolder
Об'єкту додано.
Оберіть пункт меню:

```

Рисунок 3.8 – додавання об'єктів до списку вибірки

Пункти меню 9, 10 і 11 відповідають за копіювання, вирізання і вставку і об'єктів.

Для того щоб копіювати об'єкт або групу об'єктів, необхідно спочатку додати їх до списку вибірки, потім, необхідно обрати пункт 9 («Копіювати»), щоб скопіювати вміст вибірки до буферу обміну (Рисунок 3.9). Після цього, список елементів які знаходяться у буфері обміну можна буде переглянути обравши пункт 8 (Рисунок 3.10).



```

D:\Other\Lessons\c1\Project14_curr\Debug\Project14.exe
Меню керування:
1. Переміститися до іншої папки
2. Переміститися до папки рівнем вище (назад)
3. Задати робочий шлях
4. Знайти файл (пошук по імені)
5. Додати папку
6. Створити файл
7. Відобразити/редагувати список вибірки
8. Переглянути буфер обміну
9. Копіювати (вибірка)
10. Вирізати (з вибірки до буфера обміну)
11. Вставити (вибірка)
12. Видалити (вибірка)
Введіть команду для файлового менеджера: 9
Об'єкти скопійовано до буферу обміну.
Продовжити роботу, оновити список файлів і директорій [Натисніть ENTER]

```

Рисунок 3.9 – копіювання об'єктів до буферу обміну

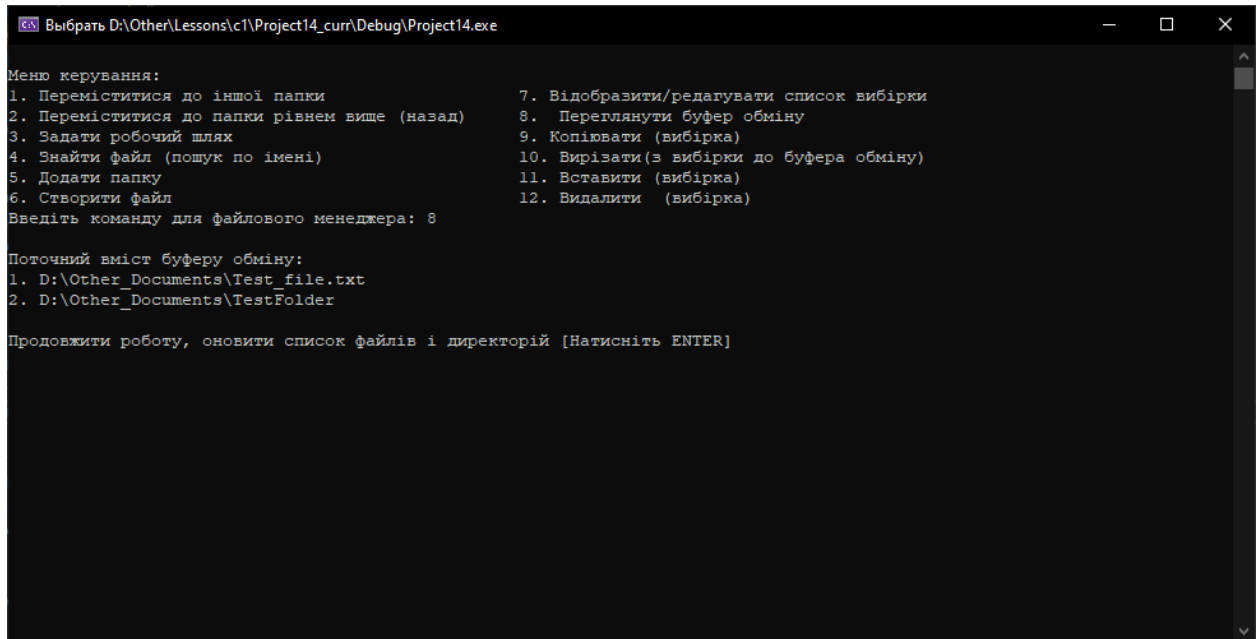


Рисунок 3.10 – перегляд поточного вмісту буферу обміну

Для того щоб вставити скопійовані об'єкти, необхідно перейти до директорії призначення і обрати пункт 11 («Вставити»), після цього програма повідомить користувача про виконає операції і запропонує користувачу натиснути клавішу «ENTER» для оновлення списку файлів та директорій (Рисунок 3.11).

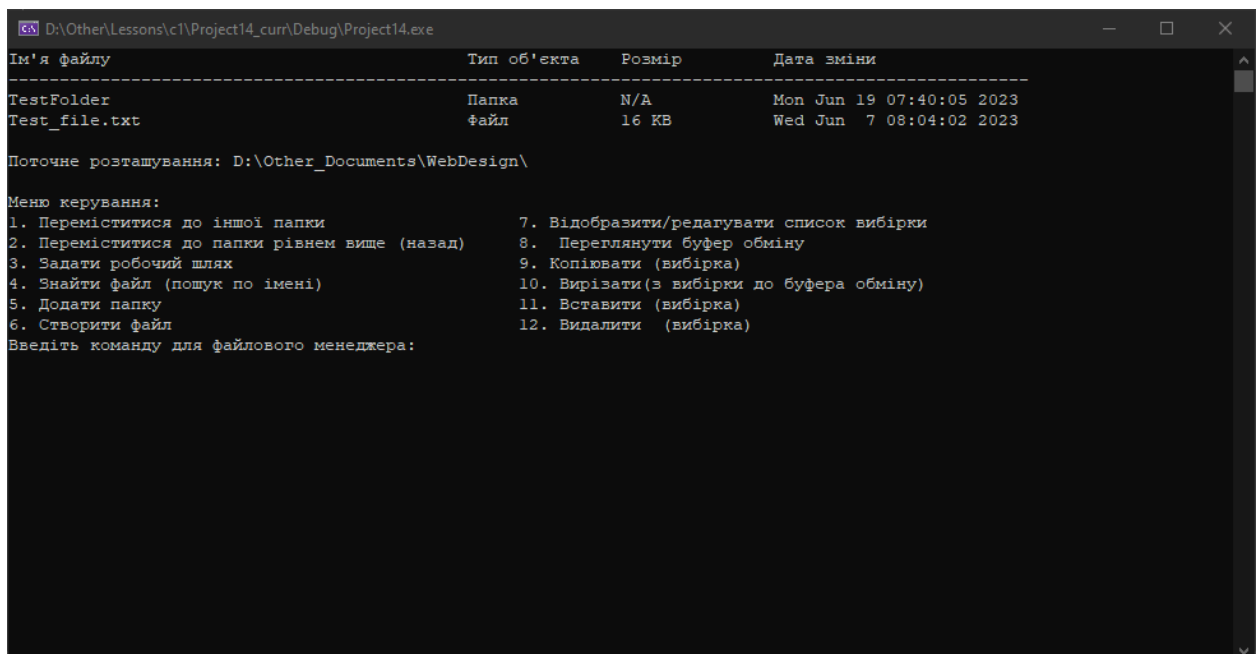


Рисунок 3.11 – вставлені файл «Test_file.txt» і папка «TestFolder» до директорії «WebDesign»

Обравши пункт 10 («Вирізати») програма аналогічним чином скопіює об'єкти зі списку вибірки до буферу обміну, але при вставці, файли і папки будуть видалені з попереднього розташування.

Видалення об'єктів виконується за допомогою пункту 12 («Видалити»), для цього необхідно попередньо додати потрібні файли і папки до списку вибірки. Після цього, програма запитає чи точно користувач впевнений, що хоче видалити усі об'єкти які є у списку вибірки (Рисунок 3.12), після підтвердження, програма виконає видалення і запропонує оновити список файлів і директорій (Рисунок 3.13).

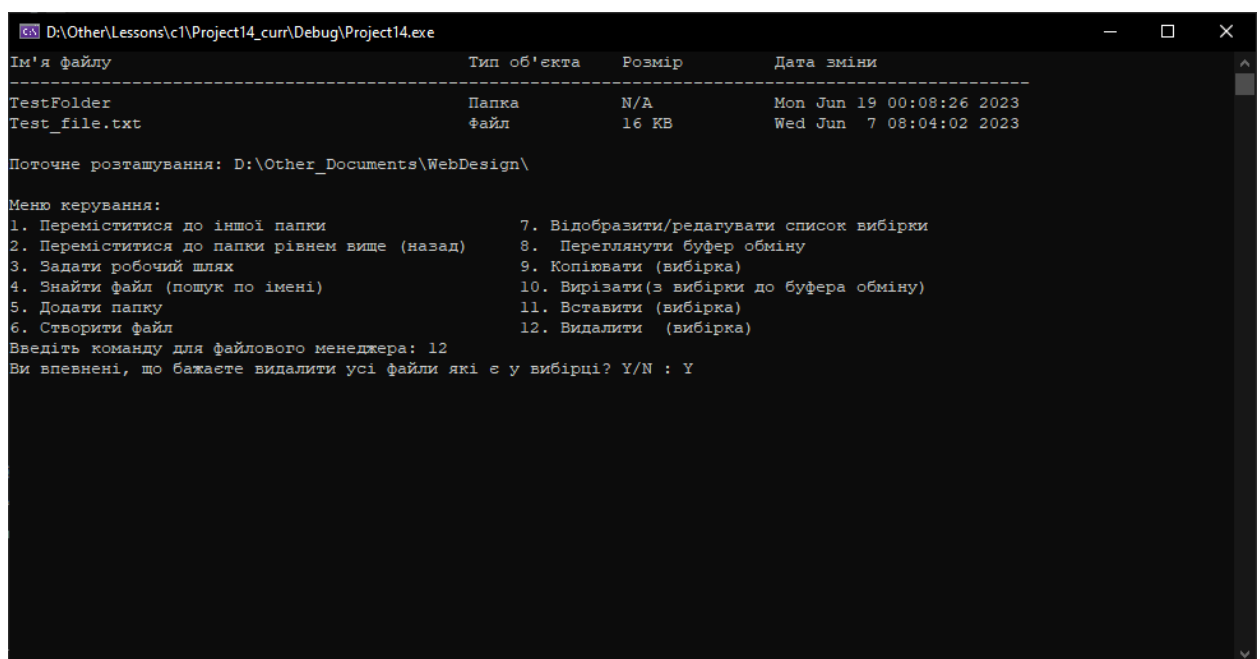


Рисунок 3.12 – підтвердження видалення об'єктів

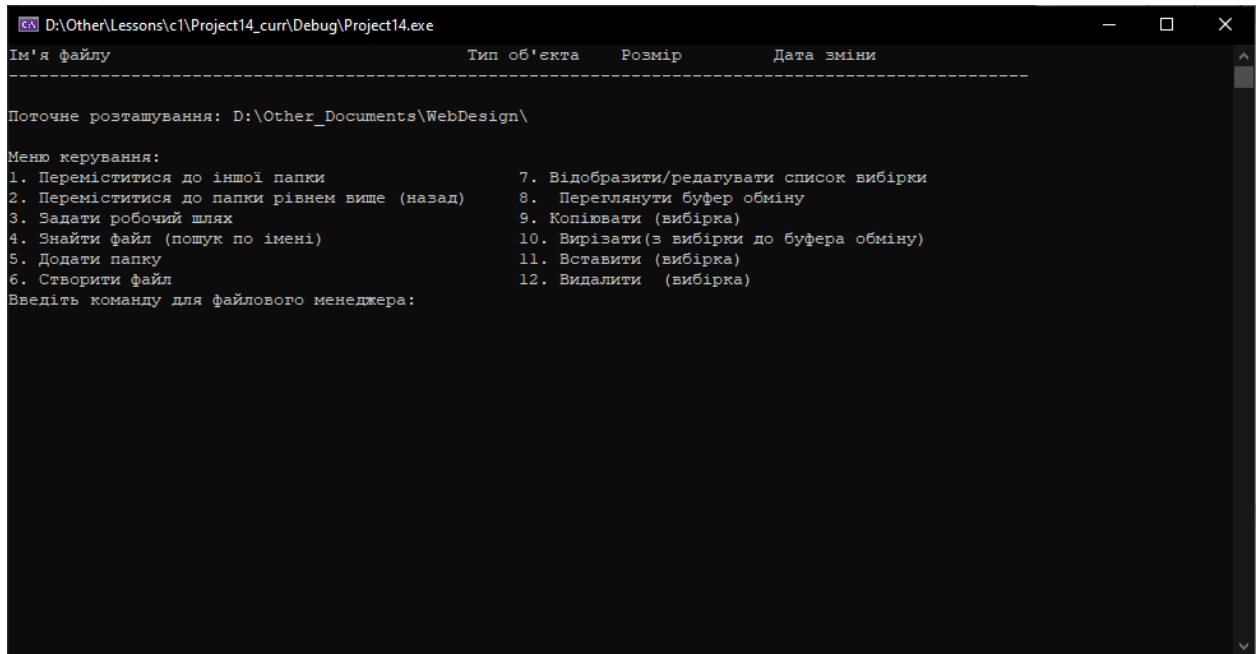


Рисунок 3.13 – папка «TestFolder» і файл «Test_file.txt» були видалені з директорії «WebDesign»

Під час відображення списку файлів і папок користувач може ввести у рядку команд файлового менеджера назву файлу чи папки які є поточній директорії, після чого відкриється меню дій які можна виконати над даним об'єктом. Наприклад, повернемося назад до директорії «Other_Documents», ввівши для цього комбінацію символів «..», щоб повернутися до папки рівнем вище і введемо назву файлу «Test_file.txt» який є у цій папці. Програма відкриє невелике меню операцій які можна виконати над даним об'єктом (Рисунок 3.14).

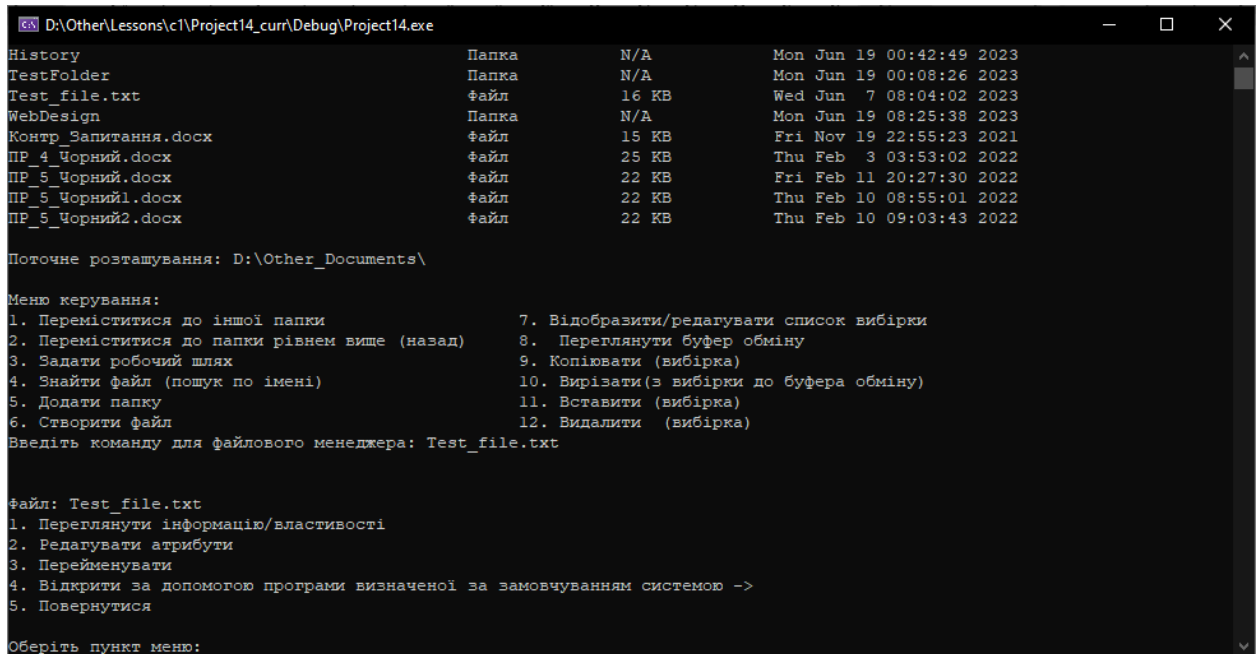


Рисунок 3.14 – меню дій які можна виконати над файлом «Test_file.txt»

Обравши пункт 1, користувач може переглянути поточну інформацію про об'єкт та його властивості (Рисунок 3.15).

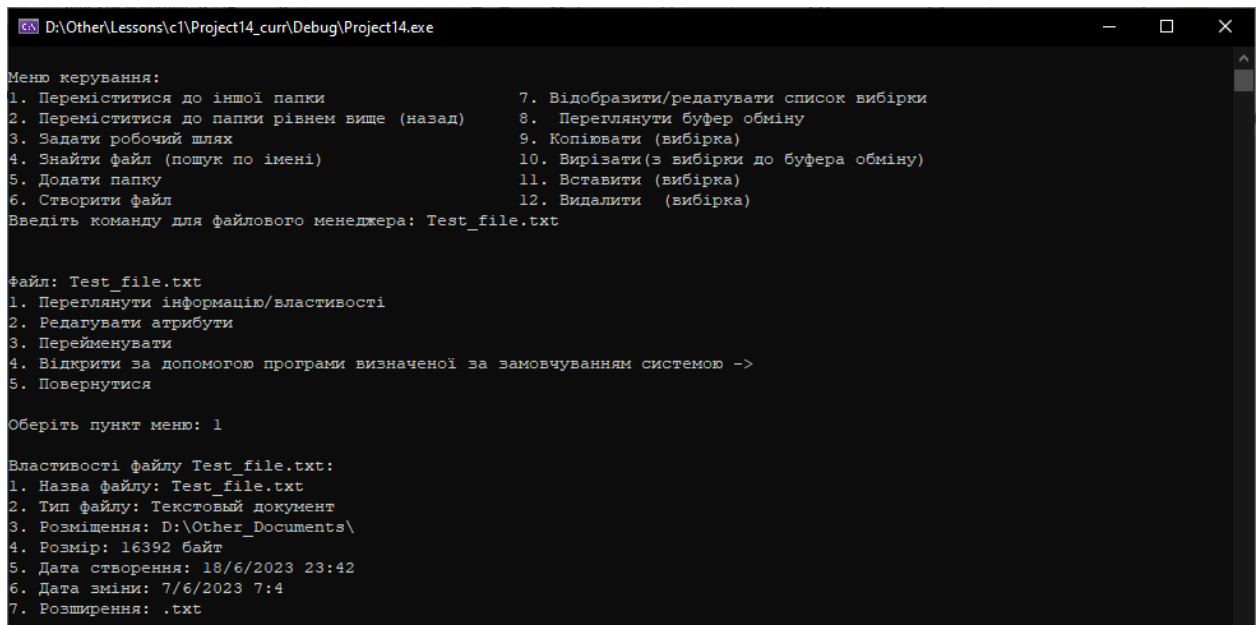


Рисунок 3.15 – інформація про файл «Test_file.txt» та його властивості

Для того щоб відредагувати атрибути файлу, необхідно обрати пункт 2, після цього програма відобразить невелике меню «Редагування атрибутів» (Рисунок 3.16). У цьому меню, обравши пункт 1, можна додати один атрибут до файлу, для цього програма запропонує користувачу ввести літеру атрибуту, якщо літера атрибуту введена коректно, програма сповістить користувача про

успішну зміну атрибутів об'єкту (Рисунок 3.17). Після цього, обравши пункт 1, можна буде переглянути поточний список атрибутів цього файлу чи папки (Рисунок 3.18).

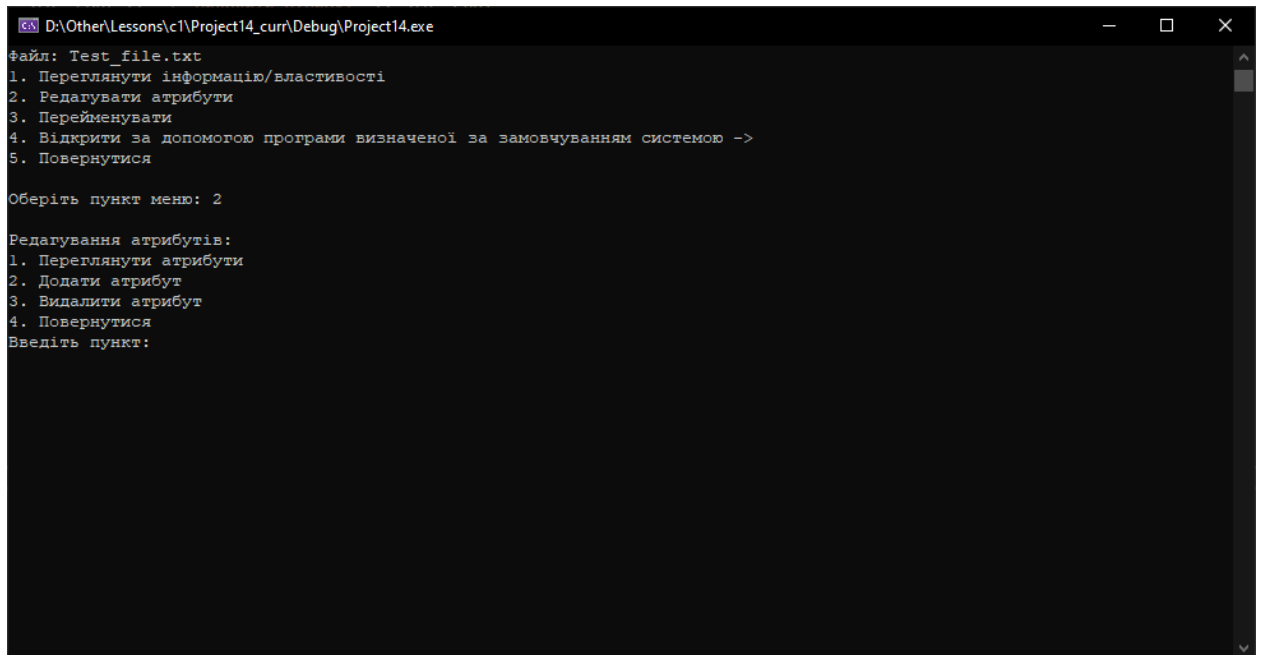


Рисунок 3.16 – меню «Редагування атрибутів»

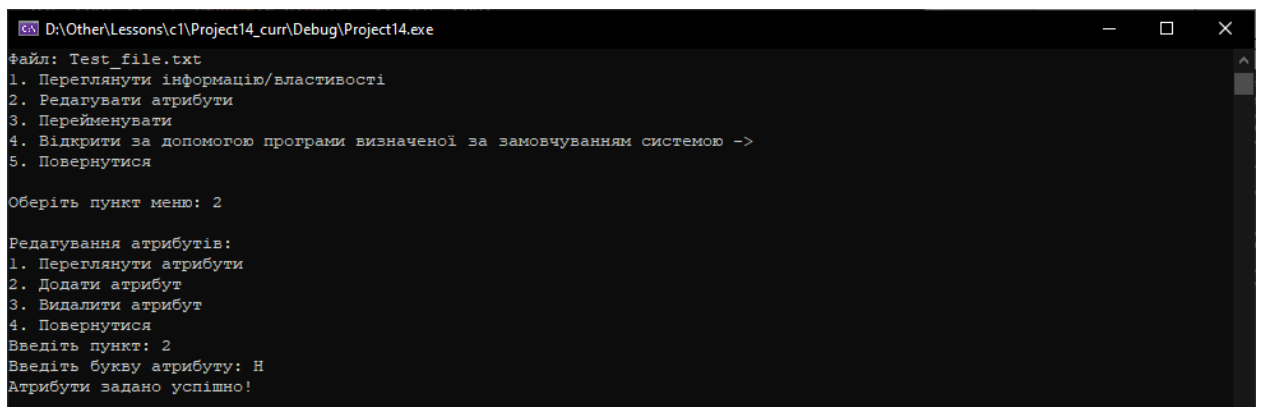


Рисунок 3.17 – додавання до файлу атрибуту «H» (прихований)



Рисунок 3.18 – поточний список атрибутів файлу «Test_file.txt»

Аналогічним чином виконується видалення атрибуту, для цього потрібно обрати пункт 3 і ввести літеру атрибуту який необхідно видалити.

Наступний пункт меню операцій над даним файлом чи папкою під номером 3 («Перейменування»), дозволяє перейменувати даний об'єкт, для цього програма запропонує користувачу ввести нове ім'я для об'єкту, перевірить його на коректність і повідомить про успішне перейменування (Рисунок 3.19).

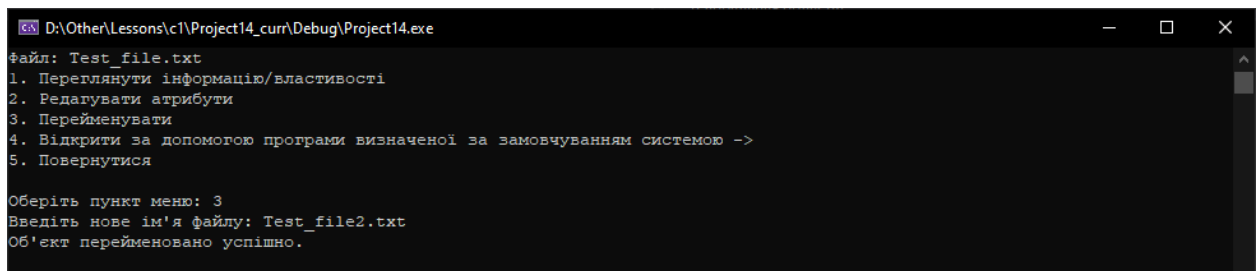


Рисунок 3.19 – перейменування файлу «Test_file.txt» на «Test_file2.txt»

Пункт меню 4 («Відкрити за допомогою програми визначеної за замовчуванням системою ->») дозволяє користувачу відкрити даний файл за допомогою внутрішньої програмної асоціації системи для файлів цього типу. Обравши цей пункт, файл одразу ж буде відкритий програмою встановленою для файлів цього типу за замовчуванням (Рисунок 3.20).

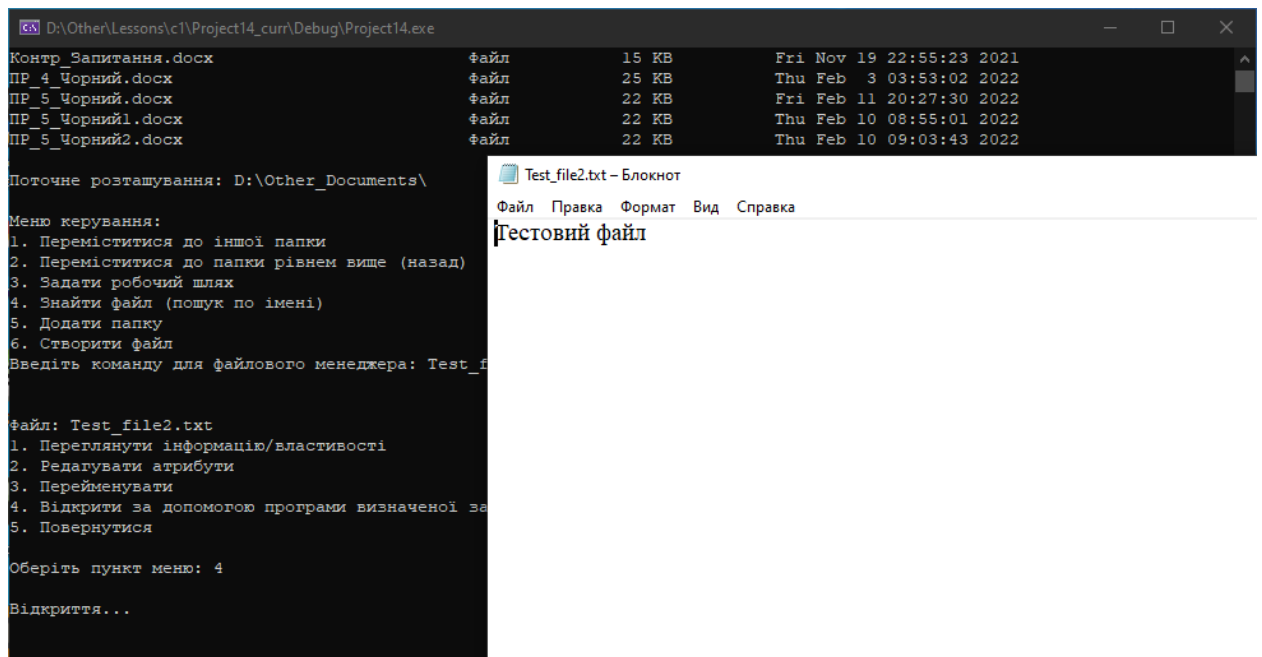


Рисунок 3.20 – відкриття файлу за допомогою програми за замовчуванням

ВИСНОВКИ

В курсовій роботі було розроблено програму «Файловий менеджер» в якій було реалізовано основні функції для ефективної навігації та маніпулювання об'єктами файлової системи. Реалізовані функції включають можливість переміщення між папками, задання робочого шляху, пошук файлів і папок за назвою, створення нових папок, додавання файлів, перейменування файлів, виконання стандартних операцій копіювання, вирізання, вставлення та видалення використовуючи список вибірки.

Цей універсальний програмний продукт може бути використаний будь-яким користувачем, якому потрібен зручний спосіб керування та організації файлів і папок у своїй системі. За допомогою інтуїтивно зрозумілого меню програми користувачі можуть легко переміщатися між структурою каталогів файлів, знаходити та отримувати доступ до об'єктів файлової системи, переглядати, змінювати властивості, атрибути файлів і папок і відкривати файли використовуючи внутрішні асоціації системи.

Додаткові функції, такі як відображення та редагування списку вибірки, а також можливість перегляду вмісту буфера обміну, підвищують зручність роботи користувача, надаючи досить повний набір інструментів для керування файлами. Крім того, програму можна налаштувати відповідно до індивідуальних уподобань кожного користувача, дозволяючи індивідуально налаштовувати її зовнішній вигляд і функціональність.

Програму «Файловий менеджер» було розроблено для необмеженого використання користувачами, щоб ефективно задовольняти їхні потреби в організації даних у своїй системі. Усі поставлені вимоги були успішно реалізовані в програмі.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. МЕТОДИЧНІ ВКАЗІВКИ ДО ВИКОНАННЯ КУРСОВИХ РОБІТ ДЛЯ ЗДОБУВАЧІВ ОСВІТИ СПЕЦІАЛЬНОСТІ 121 «ІНЖЕНЕРІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ»: МЕТОДИЧНІ ВКАЗІВКИ / Відокремлений структурний підрозділ «Полтавський політехнічний фаховий коледж Національного технічного університету «Харківський політехнічний інститут»; [уклад.: В.В.Олійник]. – Полтава: ВСП ППФК НТУ “ХПІ”, 2022. – 22 с.
2. Олексій В. Програмування на C++ в прикладах і задачах Васильєв Олексій / Васильєв Олексій., 2021. – 368 с.
3. Трофименко О. Г. C++. Основи програмування. Теорія та практика [Електронний ресурс] / О. Г. Трофименко // Фенікс. – 2010. – Режим доступу до ресурсу: <http://programming.in.ua/files/programming/book/c-plus-plus-trofumenko.rar>.
4. Stroustrup B. Programming: Principles and Practice Using C++ [Електронний ресурс] / Bjarne Stroustrup // RR Donnelley. – 2014. – Режим доступу до ресурсу: <https://dl.icdst.org/pdfs/files3/fe0590f02fa06bb42cba558fbc9e51c.pdf>.
5. B. Lippman S. C++ Primer [Електронний ресурс] / Stanley B. Lippman // Addison-Wesley Professional. – 2013. – Режим доступу до ресурсу: https://zhjwpku.com/assets/pdf/books/C++.Primer.5th.Edition_2013.pdf.
6. GetLogicalDrives function (fileapi.h). Microsoft Learn. URL: <https://learn.microsoft.com/en-us/windows/win32/api/fileapi/nf-fileapi-getlogicaldrives> (дата звернення: 11.05.2023).
7. GetFileAttributesA function (fileapi.h). Microsoft Learn. URL: <https://learn.microsoft.com/en-us/windows/win32/api/fileapi/nf-fileapi-getfileattributesa> (дата звернення: 11.05.2023).

8. GetVolumeInformationA function (fileapi.h). Microsoft Learn. URL:
<https://learn.microsoft.com/en-us/windows/win32/api/fileapi/nf-fileapi-getlogicaldrives> (дата звернення: 06.06.2023).

ДОДАТОК А. ЛІСТИНГ ПРОГРАМИ

Репозиторій програми: <https://github.com/Roma-Cherny/Coursework>



Додаток А1 – репозиторій на програмний продукт