**ЛАБОРАТОРНА РОБОТА № 3**

***Створення утиліти «DiskInfo»***

**Хід роботи:**

***Завдання 1.*** Використовуючи виклики системних функцій, отримати інформацію про дискову підсистему:

1. Список усіх логічних дисків в системі.
2. Отримати тип кожного диску присутнього в системі, та дати пояснення для кожного типу диску.
3. Отримати інформацію про диски в системі та про файлові системи Які Використовують на них.
4. Отримати інформацію про зайнятості та вільне місце на кожному з дисків.
5. Отримати інформацію про системну пам'ять.
6. Отримати інформацію про Назву комп'ютера
7. Отримати Назву поточного користувача
8. Отримати інформацію про поточний системний каталог, Тимчасовий каталог, поточний робочий каталог.
9. Для обраних каталогу на диску, Включити спостереження за змінами, продемонструвати відслідковування більше однієї зміни. Зміни записувати в лог файл.

Посилання на репозиторій:

https://github.com/andreylion06/SystemNetworkProgramming-Lab3

***Звіт про програму:***

Програма "ConsoleAppLab3" призначена для отримання інформації про дискову систему комп’ютера та деяку інформацію про систему. Вона дозволяє користувачу отримувати такі дані, як список логічних дисків в системі, типи дисків, інформацію про файлові системи на них, вільне та зайняте місце на кожному диску, інформацію про системну пам'ять, назву комп'ютера, назву поточного користувача, а також інформацію про поточні каталоги та можливість спостерігати за змінами у вибраному каталозі та записувати ці зміни в лог-файл.

Рішення реалізоване за допомогою консольного додатку використовуючи мову програмування C#. Консольний додаток має меню з опціями, що описане класом Menu. Кожен пункт меню в класі – це, по суті, елемент словника, де ключ – це назва опції типу string та значення це делегат Action.

Структура програми:

• Program.cs: Включає в себе основний код програми, створює екземпляри класів DiskInfo, SystemInfo та Menu, та обробляє основний цикл вибору опцій меню, в одному з пунктів меню створюється екземпляр класу DirectoryWatcher.

• Menu.cs: Містить клас Menu, який відповідає за відображення меню та вибір опцій користувачем.

• DiskInfo.cs: Містить клас DiskInfo, який надає інформацію про диски в системі.

• SystemInfo.cs: Містить клас SystemInfo, який надає інформацію про сис-тему.

• DirectoryWatcher.cs: Містить клас DirectoryWatcher, який відслідковує зміни в каталозі.

***Лістинг програми:***

using ConsoleAppLab3.Logic;

// Ім'я файлу: Program.cs

// Ремарка: Program.cs використовує класи Menu, DiskInfo, SystemInfo,

// DirectoryWatcher та демонструє правильність їх роботи

// Автор: Андрій Сахно

var diskInfo = new DiskInfo();

var systemInfo = new SystemInfo();

var items = new Dictionary<string, Action>

{

{ "Display logic drivers", () => diskInfo.DisplayLogicalDrives() },

{ "Display disk types", () => diskInfo.DisplayDiskTypes() },

{ "Display file system info", () => { diskInfo.DisplayFileSystemInfo(); } },

{ "Display disk space", () => { diskInfo.DisplayDiskSpace(); } },

{ "Display memory info", () => { systemInfo.DisplayMemoryInfo(); } },

{ "Display machine name", () => { systemInfo.DisplayMachineName(); } },

{ "Display user name", () => { systemInfo.DisplayUserName(); } },

{ "Display directory info", () => { systemInfo.DisplayDirectoryInfo(); } },

{ "Start monitoring the folder with the path", () => {

string pathToDirectory = @"C:\Users\User\Desktop\TestFolder";

DirectoryWatcher watcher = new DirectoryWatcher(pathToDirectory);

watcher.StartWatching();

Console.WriteLine("Press 'q' to quit the sample.");

while (Console.Read() != 'q');

watcher.StopWatching();

} },

{ "Exit 3", () => { Console.WriteLine("Exiting the program"); Environment.Exit(0); } }

};

var menu = new Menu(items);

while (true)

{

menu.DisplayOptions();

string? option = menu.ChooseOption();

menu.HandleInput(option);

menu.WaitNextIteration();

}

// Кінець файлу

// Ім'я файлу: Menu.cs

// Ремарка: Menu відображає консольне меню опцій, обробляє вибір користувача та виконує відповідні дії.

// Автор: Андрій Сахно

namespace ConsoleAppLab3.Logic

{

class Menu

{

private Dictionary<string, Action> menuItems;

public Menu(Dictionary<string, Action> items)

{

menuItems = items;

}

public void DisplayOptions()

{

Console.Clear();

Console.WriteLine("\n\n================ Menu ================\n");

int i = 1;

foreach (var item in menuItems)

{

Console.WriteLine($" Option {i}: {item.Key}");

i++;

}

Console.WriteLine("\n=======================================");

}

public string? ChooseOption()

{

Console.Write("\nYour choice ==> ");

string? userInput = Console.ReadLine();

Console.WriteLine();

return userInput;

}

public void HandleInput(string? userInput)

{

try

{

int operationNumber = int.Parse(userInput!);

menuItems.ElementAt(operationNumber - 1).Value.Invoke();

}

catch (Exception ex)

{

Console.WriteLine("Unknown operation. Please, try again.");

}

}

public void WaitNextIteration()

{

Console.Write("\nPress any key to continue.");

Console.ReadLine();

}

}

}

// Кінець файлу

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

// Ім'я файлу: DiskInfo.cs

// Ремарка: DiskInfo надає інформацію про диски в системі.

// Автор: Андрій Сахно

namespace ConsoleAppLab3.Logic

{

public class DiskInfo

{

// Список усіх логічних дисків в системі.

public void DisplayLogicalDrives()

{

string[] drives = Directory.GetLogicalDrives();

Console.WriteLine("Logical Drives:");

foreach (string drive in drives)

{

Console.WriteLine(drive);

}

}

// Отримати тип кожного диску присутнього в системі, та дати пояснення для кожного типу диску.

public void DisplayDiskTypes()

{

string[] drives = Directory.GetLogicalDrives();

Console.WriteLine("Disk Types:");

foreach (string drive in drives)

{

DriveInfo info = new DriveInfo(drive);

string typeDescription = GetTypeDescription(info.DriveType);

Console.WriteLine($"{drive} Type: {info.DriveType} ({typeDescription})");

}

}

private string GetTypeDescription(DriveType type)

{

switch (type)

{

case DriveType.CDRom:

return "An optical disc drive that contains audio or video data";

case DriveType.Fixed:

return "A disk hard drive that is physically attached (fixed) to the device";

case DriveType.Network:

return "A drive that is mapped to a network resource";

case DriveType.NoRootDirectory:

return "A drive that does not have a root directory";

case DriveType.Ram:

return "A RAM disk";

case DriveType.Removable:

return "A disk drive that has removable media, such as a floppy drive, thumb drive, or flash card reader";

case DriveType.Unknown:

return "The type of drive is unknown";

default:

return "Not recognized drive type";

}

}

// Отримати інформацію про диски в системі та про файлові системи які Використовують на них.

public void DisplayFileSystemInfo()

{

string[] drives = Directory.GetLogicalDrives();

Console.WriteLine("File System Info:");

foreach (string drive in drives)

{

DriveInfo info = new DriveInfo(drive);

if (info.IsReady)

{

Console.WriteLine($"{drive} File system: {info.DriveFormat}");

}

}

}

// Отримати інформацію про зайнятості та вільне місце на кожному з дисків.

public void DisplayDiskSpace()

{

string[] drives = Directory.GetLogicalDrives();

Console.WriteLine("Disk Space Info:");

foreach (string drive in drives)

{

DriveInfo info = new DriveInfo(drive);

if (info.IsReady)

{

Console.WriteLine($"{drive} Total size: {info.TotalSize} bytes");

Console.WriteLine($"{drive} Available space: {info.AvailableFreeSpace} bytes\n");

}

}

}

}

}

// Кінець файлу

using System;

using System.Collections.Generic;

using System.Diagnostics;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

// Ім'я файлу: SystemInfo.cs

// Ремарка: SystemInfo надає інформацію про систему, включаючи системну пам'ять,

// назву комп'ютера, ім'я користувача та інформацію про каталоги.

// Автор: Андрій Сахно

namespace ConsoleAppLab3.Logic

{

public class SystemInfo

{

// Отримати інформацію про системну пам'ять.

public void DisplayMemoryInfo()

{

Console.WriteLine("Counting in progress...");

var info = new PerformanceCounter("Memory", "Available MBytes");

Console.WriteLine($"Available system memory: {info.NextValue()}MB");

}

// Отримати інформацію про Назву комп'ютера

public void DisplayMachineName()

{

Console.WriteLine($"Machine name: {Environment.MachineName}");

}

// Отримати Назву поточного користувача

public void DisplayUserName()

{

Console.WriteLine($"Current user: {Environment.UserName}");

}

// Отримати інформацію про поточний системний каталог, Тимчасовий каталог, поточний робочий каталог.

public void DisplayDirectoryInfo()

{

Console.WriteLine($"Current system directory: {Environment.SystemDirectory}");

Console.WriteLine($"Temporary directory: {Path.GetTempPath()}");

Console.WriteLine($"Current working directory: {Environment.CurrentDirectory}");

}

}

}

// Кінець файлу

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

// Ім'я файлу: DirectoryWatcher.cs

// Ремарка: DirectoryWatcher відслідковує зміни в обраному каталозі та записує ці зміни в лог-файл.

// Автор: Андрій Сахно

namespace ConsoleAppLab3.Logic

{

public class DirectoryWatcher

{

// Для обраних каталогу на диску, Включити спостереження за змінами,

// продемонструвати відслідковування більше однієї зміни.Зміни

// записувати в лог файл.

private FileSystemWatcher watcher;

public DirectoryWatcher(string path)

{

watcher = new FileSystemWatcher(path);

watcher.NotifyFilter = NotifyFilters.LastAccess | NotifyFilters.LastWrite | NotifyFilters.FileName | NotifyFilters.DirectoryName;

watcher.Changed += OnChildrenWereChanged;

watcher.Created += OnChildrenWereChanged;

watcher.Deleted += OnChildrenWereChanged;

watcher.Renamed += OnChildrenWereRenamed;

}

public void StartWatching()

{

watcher.EnableRaisingEvents = true;

Console.WriteLine("Started watching directory.");

}

public void StopWatching()

{

watcher.EnableRaisingEvents = false;

Console.WriteLine("Stopped watching directory.");

}

private void OnChildrenWereChanged(object source, FileSystemEventArgs e)

{

string log = $"File: {e.FullPath} {e.ChangeType}";

Console.WriteLine(log);

File.AppendAllText("log.txt", log + Environment.NewLine);

}

private void OnChildrenWereRenamed(object source, RenamedEventArgs e)

{

string log = $"File: {e.OldFullPath} renamed to {e.FullPath}";

Console.WriteLine(log);

File.AppendAllText("log.txt", log + Environment.NewLine);

}

}

}

// Кінець файлу

***Висновок:*** У результаті виконання лабораторної роботи було розроблено програму "ConsoleAppLab3", яка надає користувачу зручний інтерфейс для отримання різноманітної інформації про дискову підсистему комп'ютера. Завдяки ретельно структурованому коду та використанню системних функцій, програма забезпечує швидкий доступ до даних про логічні диски, типи дисків, інформацію про файлові системи, пам'ять та інші параметри системи. Всі ці можливості роблять програму корисним інструментом для аналізу та моніторингу дискового простору та інших системних ресурсів.