

# Лабораторна робота № 2

Виконав:

студент ПМІ-44

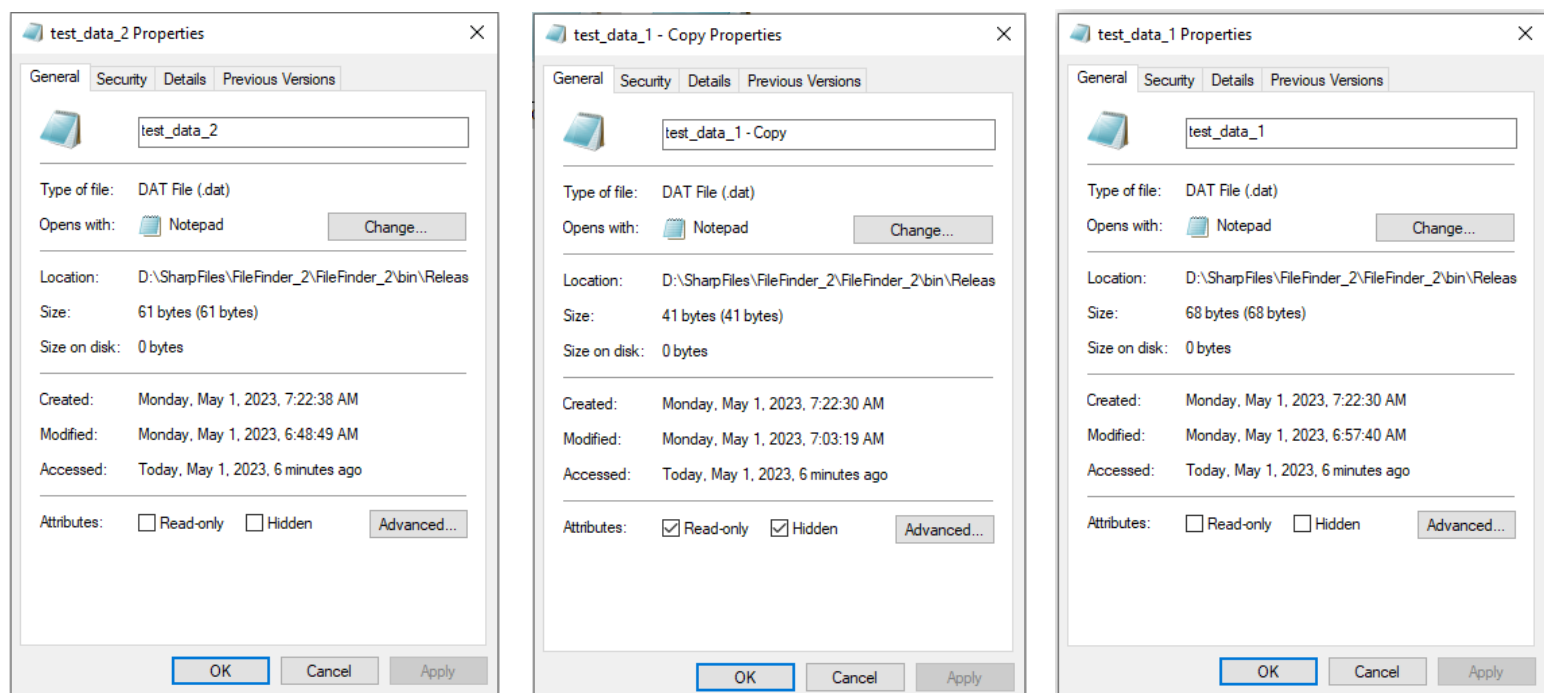
Копина Ілля

**Тема:** Атрибути та властивості файлів

## Хід Роботи:

Була створена програма, яка вміє шукати файли типу «.dat», обробляти їх властивості / атрибути та вираховувати суму з чисел, які зберігаються в цих файлах. Також були створені та заповнені 3 тестових файли цього розширення для тестування самої програми.

Після створення програми, усі тестові дані були перевірені та кожному з них були присвоєні свої певні властивості та атрибути. Окрім цього, один з файлів знаходиться безпосередньо у директорії програми, а інші два файли були переміщені в папку problems, щоб ускладнити задачу пошуку потрібних файлів.



## Основні Компоненти Програми:

```
0 references
public static void Main(string[] args)
{
    string folderPath = Directory.GetCurrentDirectory();
    var fileNames = GetAllDatFiles(folderPath);
}

2 references
private static List<string> GetAllDatFiles(string folderPath)
{
    string[] fileNames = Directory.GetFileSystemEntries(folderPath);

    var resultList = new List<string>();
    foreach (string fileName in fileNames)
    {
        var fileInfo = new FileInfo(fileName);

        if (fileInfo.Attributes.HasFlag(FileAttributes.Directory))
        {
            resultList.AddRange(GetAllDatFiles(fileName));
        }
        else if (fileInfo.Extension == ".dat")
        {
            resultList.Add(fileName);
        }
    }

    return resultList;
}
```

Це основний функціонал пошуку потрібних нам файлів. Він представляє собою рекурсивну функцію, яка переглядає усі наявні файли в початковій папці та шукає серед них файли з розширенням «.dat». Якщо в циклі ми знаходимо іншу папку то викликаємо цю ж функцію, але уже для щойно знайденої папки. Таким чином у результаті буде отримано список з потрібних нам файлів.

Після чого ми передаємо цей список у наступний цикл, який за допомогою класу **FileInfo** допоможе нам знайти усі потрібні властивості та атрибути файлів. Для того, щоб вивести результат та оформити його у вигляді таблиці використовувався допоміжний клас **ConsoleTable**. Також варто зазначити, що атрибути отримувались безпосередньо з властивості **Attributes** використовуючи спеціальний Enum. Також варто зазначити, що розташування останнього файлу було збережене для зчитування чисел та виконання операції суми.

```

var lastFileName = "";
foreach (string fileName in fileNames)
{
    var fileInfo = new FileInfo(fileName);

    var table = new ConsoleTable("File Parameters", "Value");
    table.Options.EnableCount = false;

    table.AddRow("File name", fileInfo.Name)
        .AddRow("File path", fileInfo.FullName)
        .AddRow("File size", fileInfo.Length)
        .AddRow("File creation time", fileInfo.CreationTime)
        .AddRow("File last access time", fileInfo.LastAccessTime)
        .AddRow("File last write time", fileInfo.LastWriteTime)
        .AddRow("Read only", fileInfo.Attributes.HasFlag(FileAttributes.ReadOnly))
        .AddRow("Hidden", fileInfo.Attributes.HasFlag(FileAttributes.Hidden))
        .AddRow("System", fileInfo.Attributes.HasFlag(FileAttributes.System))
        .AddRow("Archive", fileInfo.Attributes.HasFlag(FileAttributes.Archive))
        .AddRow("Compressed", fileInfo.Attributes.HasFlag(FileAttributes.Compressed))
        .AddRow("Encrypted", fileInfo.Attributes.HasFlag(FileAttributes.Encrypted))
        .AddRow("Normal", fileInfo.Attributes.HasFlag(FileAttributes.Normal));

    table.Write();

    lastFileName = fileName;
}

if(string.IsNullOrEmpty(lastFileName))
{
    return;
}

```

Остання частина програми представляє собою цикл, який перевіряє та обраховує числа з файлу. Окрім звичайного перетворення чисел, тут також є можливість ввести своє значення замість поганого числа з файлу. Якщо ж відповідь буде не число, а будь яка літера чи слово то програма буде використовувати значення 0 і для всіх інших подібних випадків.

```

var data = File.ReadAllText(lastFileName).Trim('\n', '\r');

Console.WriteLine("\nData from last file: \n" + data + "\n");

string[] numbers = data.Split();

int count = 0;
double sum = 0;
bool isDefaultSet = false;
foreach (string number in numbers)
{
    if (double.TryParse(number, out var actualNumber))
    {
        sum += actualNumber;
    }
    else if (!isDefaultSet)
    {
        Console.WriteLine("\nNumber #" + count + " '" + number + "' is damaged.\nPlease input new value (Or any non numeric value to set zero by default)");
        Console.Write(" --> ");
        var newValue = Console.ReadLine();

        if (double.TryParse(newValue, out actualNumber))
        {
            sum += actualNumber;
        }
        else
        {
            isDefaultSet = true;
        }
    }

    count++;
}

Console.WriteLine("\nSum: " + sum);

```

Результат Работи:

