ДНІПРОВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

ІМЕНІ ОЛЕСЯ ГОНЧАРА

Факультет прикладної математики

Кафедра математичного забезпечення ЕОМ

ЗВІТ

з лабораторної роботи № 4

за курсом «Архітектура, модулі та компоненти програмних систем»

Тема: «Робота з файловою системою у мовах програмування високого рівня»

Спеціальність 121

Група ПЗ-22у-1

Виконав Андрущенко В.А.

Перевірив Антонюк В.А.

Дніпро, 2023р.

1. Постановка задачі

Використовуючи мову програмування високого рівня (C, C++, C#, іншу мову за згодою з викладачем, але не скриптові мови) створити програму, яка виконує наступне: Варіант 1. Підрахувати сумарний обсяг файлів в каталозі.

1. Опис розв’язку

Аналізуючи попередню роботу було використано подібний алгоритм, але скрипт мовою С#, для програми, яка виконує: підрахунок сумарного обсягу файлів в каталозі, представлено на рисунках 1.1 – 1.5.

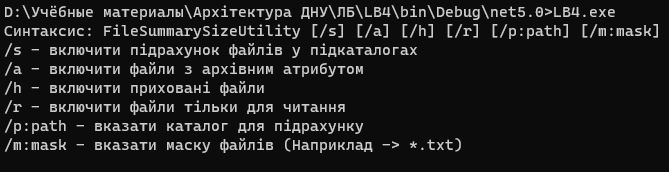


Рисунок 1.1 – Запуск без параметрів



Рисунок 1.2 – Робота з вказаним каталогом



Рисунок 1.3 – Робота з вказаними файлами за шаблоном \*.cs



Рисунок 1.4 – Запуск з параметрами та без врахування підкаталогів

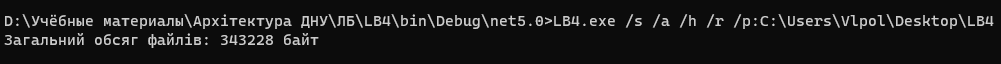


Рисунок 1.5 – Врахування атрибутів файлів (запуск з архівами, прихованими та тільки для читання)

1. Вихідний текст програми

Текст програми з відповідними коментарями представлено в лістингу 1.1.

Лістинг 1.1 – Main

using System;

using System.IO;

namespace LB4{

class Program{

static void Main(string[] args){

bool paramSubdir = false;

bool paramHidden = false;

bool paramRead = false;

bool paramArchive = false;

string paramPath = Directory.GetCurrentDirectory();

string paramMask = "\*.\*";

long totalSize = 0;

// Descriptions

if (args.Length == 0 || args[0].ToLower() == "/?")

{

Console.WriteLine("Синтаксис: FileSummarySizeUtility [/s] [/a] [/h] [/r] [/p:path] [/m:mask]");

Console.WriteLine("/s - включити пiдрахунок файлiв у пiдкаталогах");

Console.WriteLine("/a - включити файли з архiвним атрибутом");

Console.WriteLine("/h - включити прихованi файли");

Console.WriteLine("/r - включити файли тiльки для читання");

Console.WriteLine("/p:path - вказати каталог для пiдрахунку");

Console.WriteLine("/m:mask - вказати маску файлiв (Наприклад -> \*.txt)");

Environment.Exit(0);

}

// Params

for (int i = 0; i < args.Length; i++)

{

if (args[i].ToLower() == "/s")

{

paramSubdir = true;

}

else if (args[i].ToLower() == "/a")

{

paramArchive = true;

}

else if (args[i].ToLower() == "/h")

{

paramHidden = true;

}

else if (args[i].ToLower() == "/r")

{

paramRead = true;

}

else if (args[i].ToLower().StartsWith("/p:"))

{

paramPath = args[i].Substring(3);

}

else if (args[i].ToLower().StartsWith("/m:"))

{

paramMask = args[i].Substring(3);

}

else

{

Console.WriteLine("Помилка! Невiдомий параметр: " + args[i]);

Environment.Exit(1);

}

}

// Summary size file

try{

var files = Directory.GetFiles(paramPath, paramMask, paramSubdir ? SearchOption.AllDirectories : SearchOption.TopDirectoryOnly);

foreach (var file in files)

{

var attributes = File.GetAttributes(file);

if (paramArchive || !attributes.HasFlag(FileAttributes.Archive))

{

if (paramHidden || !attributes.HasFlag(FileAttributes.Hidden))

{

if (paramRead || !attributes.HasFlag(FileAttributes.ReadOnly))

{

totalSize += new FileInfo(file).Length;

}

}

}

}

Console.WriteLine("Загальний обсяг файлiв: " + totalSize + " байт");

Environment.Exit(0);

}

catch (Exception ex){

Console.WriteLine("Помилка! Спробуйте ще раз! Повiдомлення: " + ex.Message);

Environment.Exit(1);

}

}

}

}

Висновок: в результаті цієї лабораторної роботи було ознайомлено з, роботою з файловою системою у мовах програмування високого рівня.