Кам’янець-Подільський національний університет

 імені Івана Огієнка

КАФЕДРА КОМП’ЮТЕРНИХ НАУК

Навчальна дисципліна «Об’єктно-орієнтоване програмування»

ЛАБОРАТОРНА РОБОТА #0108

Тема: Робота з файлами

Варіант №3

Виконав:

студент 1-го курсу

групи KNms1-B21

Волошин В.Ю.

Прийняла:

старший викладач,

Мястковська М.О.

Кам’янець-Подільський – 2023

1. **Хід роботи.**
2. **Завдання до виконання.**

### Завдання 1.

Реалізувати наступні операції з файловою системою:

1) На диску D: створіть каталог OOP\_lab08.

2) Cтворіть у ньому наступні каталоги (назви англійською мовою):

* Номер\_вашої\_групи (підставте свою групу);
* Ваше\_прізвище (підставте своє прізвище);
* Sources;
* Reports;
* Texts.

3) Скопіюйте каталоги Texts, Sources та Reports до каталогу Ваше\_прізвище.

4) Перемістіть каталог Ваше\_прізвище до каталогу Номер\_вашої\_групи.

5) У каталозі Texts створіть текстовий файл dirinfo.txt у який помістіть усю інформацію про даний каталог (див. приклад вище).

**Завдання 2.**

Дано файл f, компоненти якого є цілими числами. Знайти:

1) кількість парних чисел серед компонент;

2) кількість подвоєних парних чисел серед компонент;

3) кількість квадратів непарних чисел серед компонент.

1. **Лістинг програми.**

class FileSystemManager

{

    // Метод для налаштування файлової системи

    public static void SetupFileSystem()

    {

        try

        {

            // Задання кореневого каталогу

            string rootDirectory = @"D:\OOP\_lab08";

            Directory.CreateDirectory(rootDirectory);

            // Задання номеру групи та вашого прізвища

            string groupNumber = "KNms1b23";

            string yourLastName = "Voloshin";

            // Формування шляхів до підкаталогів

            string groupDirectory = Path.Combine(rootDirectory, groupNumber);

            string yourLastNameDirectory = Path.Combine(rootDirectory, yourLastName);

            string sourcesDirectory = Path.Combine(rootDirectory, "Sources");

            string reportsDirectory = Path.Combine(rootDirectory, "Reports");

            string textsDirectory = Path.Combine(rootDirectory, "Texts");

            Console.WriteLine("Крок 1: Створення кореневого каталогу");

            CreateDirectory(rootDirectory);

            Console.WriteLine("Крок 2: Створення підкаталогів");

            CreateDirectory(groupDirectory);

            CreateDirectory(yourLastNameDirectory);

            CreateDirectory(sourcesDirectory);

            CreateDirectory(reportsDirectory);

            CreateDirectory(textsDirectory);

            Console.WriteLine("Крок 3: Копіювання каталогів");

            CopyDirectory(textsDirectory, Path.Combine(yourLastNameDirectory, "Texts"));

            CopyDirectory(sourcesDirectory, Path.Combine(yourLastNameDirectory, "Sources"));

            CopyDirectory(reportsDirectory, Path.Combine(yourLastNameDirectory, "Reports"));

            Console.WriteLine("Крок 4: Переміщення каталогу");

            MoveDirectory(yourLastNameDirectory, Path.Combine(groupDirectory, yourLastName));

            Console.WriteLine("Крок 5: Створення файлу dirinfo.txt");

            string dirinfoPath = Path.Combine(textsDirectory, "dirinfo.txt");

            CreateDirInfoFile(dirinfoPath, textsDirectory);

            Console.WriteLine("Операції виконано успішно.");

        }

        catch (Exception ex)

        {

            Console.WriteLine($"Помилка: {ex.Message}");

        }

    }

    // Метод для створення каталогу

    private static void CreateDirectory(string directoryPath)

    {

        Console.WriteLine($"Створення каталогу: {directoryPath}");

        Directory.CreateDirectory(directoryPath);

    }

    /// Метод для копіювання каталогу

    private static void CopyDirectory(string sourceDirName, string destDirName)

    {

        Console.WriteLine($"Копіювання каталогу з {sourceDirName} у {destDirName}");

        DirectoryInfo sourceDir = new DirectoryInfo(sourceDirName);

        DirectoryInfo destDir = new DirectoryInfo(destDirName);

        // Create the destination directory if it doesn't exist

        if (!destDir.Exists)

        {

            CreateDirectory(destDir.FullName);

        }

        foreach (FileInfo file in sourceDir.GetFiles())

        {

            file.CopyTo(Path.Combine(destDir.FullName, file.Name), false);

        }

        foreach (DirectoryInfo subDir in sourceDir.GetDirectories())

        {

            string subDirPath = Path.Combine(destDir.FullName, subDir.Name);

            CopyDirectory(subDir.FullName, subDirPath);

        }

    }

    // Метод для переміщення каталогу

    private static void MoveDirectory(string sourceDirName, string destDirName)

    {

        Console.WriteLine($"Переміщення каталогу з {sourceDirName} у {destDirName}");

        Directory.Move(sourceDirName, destDirName);

    }

    // Метод для створення файлу dirinfo.txt

    private static void CreateDirInfoFile(string filePath, string directoryPath)

    {

        Console.WriteLine($"Створення файлу dirinfo.txt в {directoryPath}");

        using (StreamWriter writer = new StreamWriter(filePath))

        {

            writer.WriteLine($"Інформація про каталог: {directoryPath}");

            writer.WriteLine($"Створено: {DateTime.Now}");

            writer.WriteLine("Зміст:");

            string[] files = Directory.GetFiles(directoryPath);

            foreach (string file in files)

            {

                writer.WriteLine($"- {Path.GetFileName(file)}");

            }

            string[] subdirectories = Directory.GetDirectories(directoryPath);

            foreach (string subdirectory in subdirectories)

            {

                writer.WriteLine($"- {Path.GetFileName(subdirectory)}/");

            }

        }

    }

}

using System;

using System.IO;

class NumberAnalyzer

{

    private int[] numbers;

    // Конструктор класу NumberAnalyzer, який зчитує числа з файлу за вказаним шляхом

    public NumberAnalyzer(string filePath)

    {

        try

        {

            string[] lines = File.ReadAllLines(filePath);

            numbers = Array.ConvertAll(lines, int.Parse);

        }

        catch (Exception ex)

        {

            Console.WriteLine($"Помилка: {ex.Message}");

        }

    }

    // Метод для аналізу чисел та виведення різних характеристик

    public void AnalyzeNumbers()

    {

        // Перевірка, чи числа були успішно зчитані

        if (numbers == null || numbers.Length == 0)

        {

            Console.WriteLine("Немає чисел для аналізу.");

            return;

        }

        // Виведення початкового масиву чисел

        Console.WriteLine("Початкові числа:");

        PrintNumbers();

        // Підрахунок кількості парних чисел та їх виведення

        int countEvenNumbers = CountEvenNumbers();

        Console.WriteLine($"1) Кількість парних чисел: {countEvenNumbers}");

        // Підрахунок кількості подвоєних парних чисел та їх виведення

        int countDoubledEvenNumbers = CountDoubledEvenNumbers();

        Console.WriteLine($"2) Кількість подвоєних парних чисел: {countDoubledEvenNumbers}");

        // Підрахунок кількості квадратів непарних чисел та їх виведення

        int countSquaresOfOddNumbers = CountSquaresOfOddNumbers();

        Console.WriteLine($"3) Кількість квадратів непарних чисел: {countSquaresOfOddNumbers}");

    }

    // Виведення чисел

    private void PrintNumbers()

    {

        foreach (var number in numbers)

        {

            Console.Write(number + " ");

        }

        Console.WriteLine();

    }

    // Підрахунок кількості парних чисел та їх виведення

    private int CountEvenNumbers()

    {

        int count = 0;

        Console.WriteLine("Парні числа:");

        foreach (var number in numbers)

        {

            if (number % 2 == 0)

            {

                Console.Write(number + " ");

                count++;

            }

        }

        Console.WriteLine();

        return count;

    }

    // Підрахунок кількості подвоєних парних чисел та їх виведення

    private int CountDoubledEvenNumbers()

    {

        int count = 0;

        Console.WriteLine("Подвоєні парні числа:");

        foreach (var number in numbers)

        {

            if (number % 2 == 0 && Array.IndexOf(numbers, number \* 2) != -1)

            {

                Console.Write(number + " ");

                count++;

            }

        }

        Console.WriteLine();

        return count;

    }

    // Підрахунок кількості квадратів непарних чисел та їх виведення

    private int CountSquaresOfOddNumbers()

    {

        int count = 0;

        Console.WriteLine("Квадрати непарних чисел:");

        foreach (var number in numbers)

        {

            if (number % 2 != 0 && Array.IndexOf(numbers, number \* number) != -1)

            {

                Console.Write(number + " ");

                count++;

            }

        }

        Console.WriteLine();

        return count;

    }

}

class Program

{

    static void Main()

    {

        // task 1

        FileSystemManager.SetupFileSystem();

        Console.WriteLine();

        // task 2

         string filePath = "numbers.txt";  // Ім'я файлу для збереження чисел

        // Створення файлу та запис в нього випадкових цілих чисел

        GenerateRandomNumbersFile(filePath, 10);  // 10 - кількість випадкових чисел

        // Створення екземпляра класу NumberAnalyzer та аналіз чисел

        NumberAnalyzer numberAnalyzer = new NumberAnalyzer(filePath);

        numberAnalyzer.AnalyzeNumbers();

    }

    // Метод для генерації файлу з випадковими цілими числами

    static void GenerateRandomNumbersFile(string filePath, int count)

    {

        Random random = new Random();

        using (StreamWriter writer = new StreamWriter(filePath))

        {

            for (int i = 0; i < count; i++)

            {

                writer.WriteLine(random.Next(1, 100));  // Генерація випадкового числа в діапазоні [1, 100)

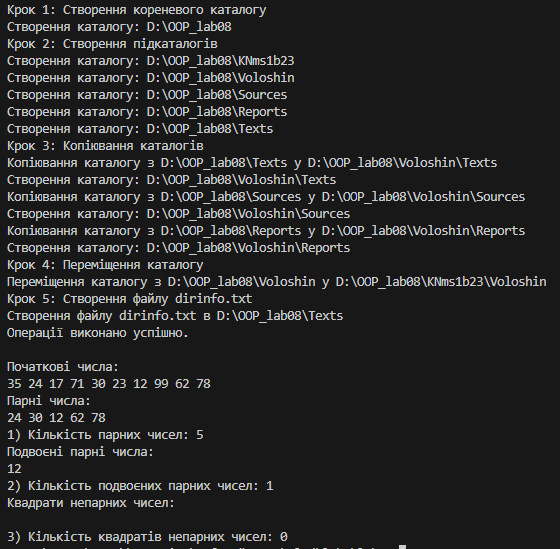
            }

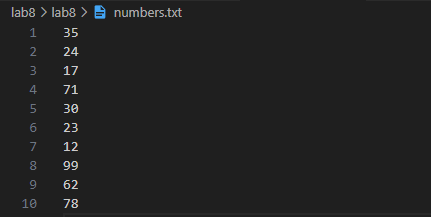
        }

    }

}

1. **Результат роботи.**





1. **Посилання на проект.**