

## Лабораторна робота № 7

### ОСНОВИ РОБОТИ З ФАЙЛАМИ В C#

**Мета роботи:** ознайомитися з основними правилами роботи з файлами та каталогами в C#.

#### Варіант 15

#### Завдання для лабораторної роботи

**Завдання 1.** Кожне з цих завдань вирішите з використанням класів StreamReader и StreamWriter. Реалізуйте в консольному додатку.

Варіант 15. У вхідному файлі input.txt міститься текст. У файл output.txt вивести відредагований текст, в якому видалені зайві пропуски між словами.

```
using System;
using System.IO;
using System.Text.RegularExpressions;

class Program
{
    static void Main()
    {
        try
        {
            Console.WriteLine("Введіть текст:");
            string text = Console.ReadLine();

            using (StreamWriter sw = new StreamWriter("input.txt"))
            {
                sw.Write(text);
            }
        }
    }
}
```

```
Console.WriteLine("Текст був успішно записаний у вхідний файл.");

using (StreamReader sr = new StreamReader("input.txt"))
{
    string content = sr.ReadToEnd();

    string editedText = Regex.Replace(content, @"\s+", " ");

    using (StreamWriter sw = new StreamWriter("output.txt"))
    {
        sw.Write(editedText);
    }

    Console.WriteLine("Вміст файлу був успішно відредагований.");

    using (StreamReader srOutput = new StreamReader("output.txt"))
    {
        string outputText = srOutput.ReadToEnd();
        Console.WriteLine("Вміст файлу output.txt:\n" + outputText);
    }
}

catch (Exception e)
{
    Console.WriteLine("Сталася помилка: " + e.Message);
}
}
```

}

```
Введіть текст:  
мене          Звати          Олександр          Новак  
Текст був успішно записаний у вхідний файл.  
Вміст файлу був успішно вдредагований.  
Вміст файлу output.txt:  
мене Звати Олександр Новак
```

Рис. 1.1 – Вивід консолі завдання 1

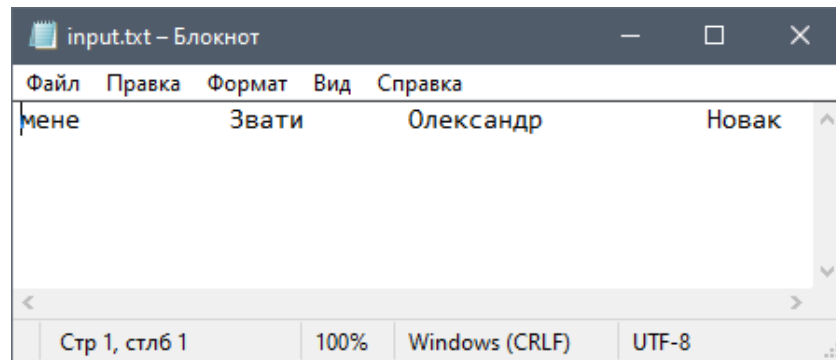


Рис. 1.2 – Вміст файлу input.txt після відпрацювання

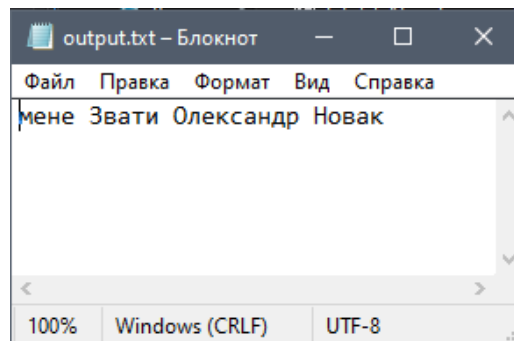


Рис. 1.2 – Вміст файлу output.txt після відпрацювання

**Завдання 2.** Кожне з цих завдань вирішіть трьома способами - з використанням символьного, байтового і двоїчного потоків. Реалізуйте в консольному додатку.

Варіанти 5,10,15,20. Створити програмним чином файл input.txt і заповнити його 50 цілими числами, отриманими за допомогою генератора випадкових чисел з діапазону [-10, 10]. Написати програму, що виводить на

екран і в файл output.txt статистику повторень чисел з файлу input.txt, у вигляді <число, частота>.

***Спосіб 1 (символьний потік):***

```
using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;

class Program
{
    static void Main()
    {
        string inputFilePath = "input.txt";
        GenerateRandomNumbers(inputFilePath);

        string outputFilePath = "output_symbol.txt";
        ProcessFile(inputFilePath, outputFilePath);

        Console.WriteLine("Статистика була збережена у файлі: " + outputFilePath);
    }

    static void GenerateRandomNumbers(string filePath)
    {
        Random random = new Random();
        using (StreamWriter writer = new StreamWriter(filePath))
        {
            for (int i = 0; i < 50; i++)
```

```

    {
        writer.WriteLine(random.Next(-10, 11));
    }
}

```

```

static void ProcessFile(string inputFilePath, string outputFilePath)
{
    Dictionary<int, int> frequencyMap = new Dictionary<int, int>();

    using (StreamReader reader = new StreamReader(inputFilePath))
    {
        string line;
        while ((line = reader.ReadLine()) != null)
        {
            int number = int.Parse(line);
            if (frequencyMap.ContainsKey(number))
            {
                frequencyMap[number]++;
            }
            else
            {
                frequencyMap[number] = 1;
            }
        }
    }
}

```

```

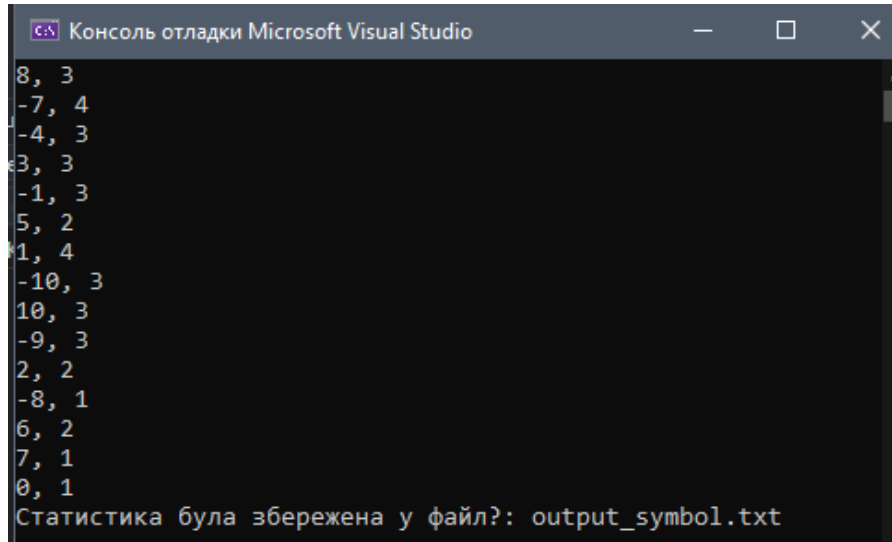
using (StreamWriter writer = new StreamWriter(outputFilePath))

```

```

{
    foreach (var entry in frequencyMap)
    {
        writer.WriteLine($"{entry.Key}, {entry.Value}");
        Console.WriteLine($"{entry.Key}, {entry.Value}");
    }
}
}
}

```

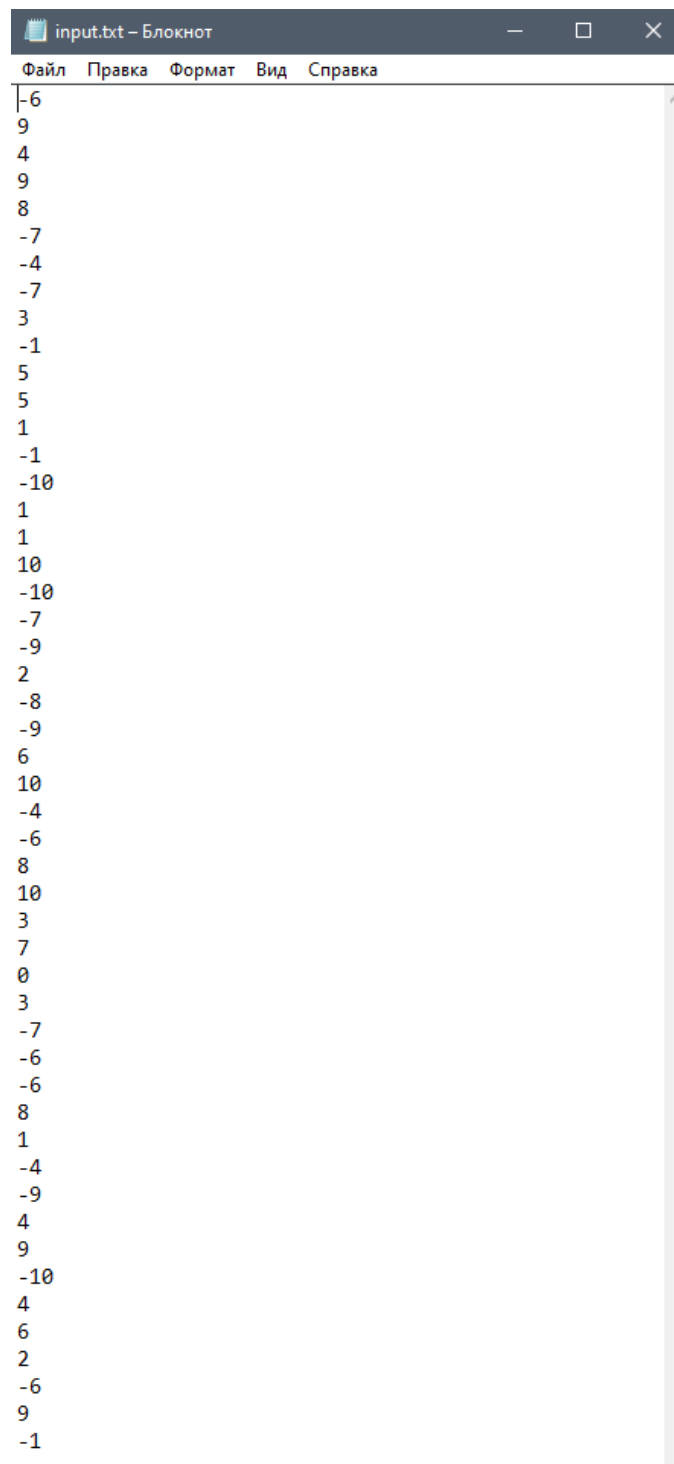


```

Консоль отладки Microsoft Visual Studio
8, 3
-7, 4
-4, 3
3, 3
-1, 3
5, 2
1, 4
-10, 3
10, 3
-9, 3
2, 2
-8, 1
6, 2
7, 1
0, 1
Статистика була збережена у файл?: output_symbol.txt

```

Рис. 2.1.1 – Вивід консолі завдання 2, спосіб 1



The image shows a Notepad window with the title bar 'input.txt - Блокнот'. The menu bar includes 'Файл', 'Правка', 'Формат', 'Вид', and 'Справка'. The text area contains a vertical list of integers, each on a new line. The numbers are: -6, 9, 4, 9, 8, -7, -4, -7, 3, -1, 5, 5, 1, -1, -10, 1, 1, 10, -10, -7, -9, 2, -8, -9, 6, 10, -4, -6, 8, 10, 3, 7, 0, 3, -7, -6, -6, 8, 1, -4, -9, 4, 9, -10, 4, 6, 2, -6, 9, and -1.

```
-6
9
4
9
8
-7
-4
-7
3
-1
5
5
1
-1
-10
1
1
10
-10
-7
-9
2
-8
-9
6
10
-4
-6
8
10
3
7
0
3
-7
-6
-6
8
1
-4
-9
4
9
-10
4
6
2
-6
9
-1
```

Рис. 2.1.2 – Вміст файлу input.txt після відпрацювання

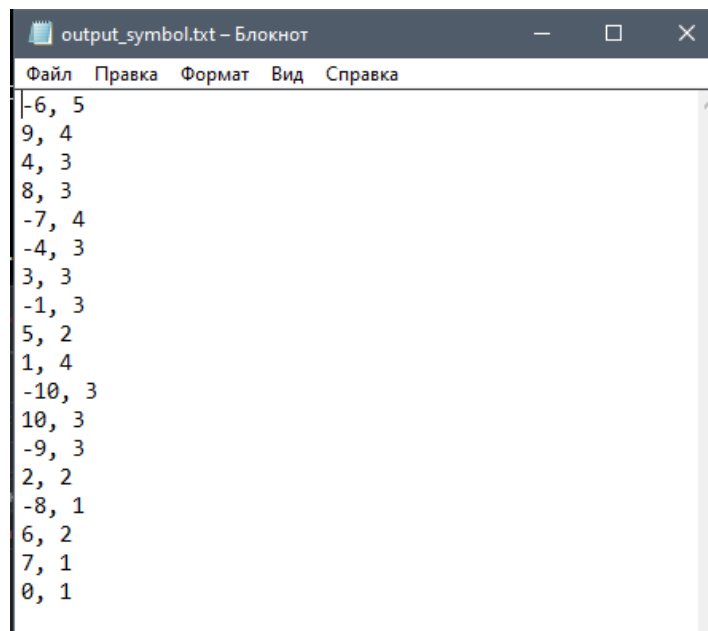


Рис. 2.1.3 – Вміст файлу output.txt після відпрацювання

***Список 2 (байтовий список):***

```
using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;

class Program
{
    static void Main()
    {
        string inputFilePath = "input.txt";
        GenerateRandomNumbers(inputFilePath);

        string outputFilePath = "output_byte.txt";
        ProcessFile(inputFilePath, outputFilePath);
    }
}
```



```
    Console.WriteLine("Статистика була збережена у файлі: " + outputPath);  
}
```

```
static void GenerateRandomNumbers(string filePath)  
{  
    Random random = new Random();  
    using (FileStream fs = new FileStream(filePath, FileMode.Create))  
    using (BinaryWriter writer = new BinaryWriter(fs))  
    {  
        for (int i = 0; i < 50; i++)  
        {  
            writer.Write(random.Next(-10, 11));  
        }  
    }  
}
```

```
static void ProcessFile(string inputFilePath, string outputPath)  
{  
    Dictionary<int, int> frequencyMap = new Dictionary<int, int>();  
  
    using (FileStream fs = new FileStream(inputFilePath, FileMode.Open))  
    using (BinaryReader reader = new BinaryReader(fs))  
    {  
        while (fs.Position < fs.Length)  
        {  
            int number = reader.ReadInt32();  
            if (frequencyMap.ContainsKey(number))  
            {
```

```
        frequencyMap[number]++;  
    }  
    else  
    {  
        frequencyMap[number] = 1;  
    }  
}  
}
```

```
using (StreamWriter writer = new StreamWriter(outputFilePath))  
{  
    foreach (var entry in frequencyMap)  
    {  
        writer.WriteLine($"{entry.Key}, {entry.Value}");  
        Console.WriteLine($"{entry.Key}, {entry.Value}");  
    }  
}  
}
```



## ***Список 2 (двоичный список):***

```
using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;

class Program
{
    static void Main()
    {
        string inputFilePath = "input.txt";
        GenerateRandomNumbers(inputFilePath);

        string outputFilePath = "output_binary.txt";
        ProcessFile(inputFilePath, outputFilePath);

        Console.WriteLine("Статистика була збережена у файлі: " + outputFilePath);
    }

    static void GenerateRandomNumbers(string filePath)
    {
        Random random = new Random();
        using (BinaryWriter writer = new BinaryWriter(File.Open(filePath,
FileMode.Create)))
        {
            for (int i = 0; i < 50; i++)
            {
                writer.Write(random.Next(-10, 11));
            }
        }
    }
}
```

```
    }  
  }  
}
```

```
static void ProcessFile(string inputFilePath, string outputFilePath)  
{  
    Dictionary<int, int> frequencyMap = new Dictionary<int, int>();  
  
    using (BinaryReader reader = new BinaryReader(File.Open(inputFilePath,  
FileMode.Open)))  
    {  
        while (reader.BaseStream.Position < reader.BaseStream.Length)  
        {  
            int number = reader.ReadInt32();  
            if (frequencyMap.ContainsKey(number))  
            {  
                frequencyMap[number]++;  
            }  
            else  
            {  
                frequencyMap[number] = 1;  
            }  
        }  
    }  
  
    using (StreamWriter writer = new StreamWriter(outputFilePath))  
    {  
        foreach (var entry in frequencyMap)
```

```

{
    writer.WriteLine($"{entry.Key}, {entry.Value}");
    Console.WriteLine($"{entry.Key}, {entry.Value}");
}
}
}
}
}

```

```

-4, 2
-8, 3
-2, 3
3, 3
5, 1
-5, 4
-10, 2
8, 2
-3, 5
0, 3
-1, 3
7, 3
-7, 4
2, 1
9, 3
-6, 3
6, 3
1, 2
Статистика була збережена у файл?: output_binary.txt

```

Рис. 2.3.1 – Вивід консолі завдання 2, спосіб 3

input.txt – Блокнот

Файл Правка Формат Вид Справка

ыяяшыяяюяяя ыяяцяяя эяя яяяя юяяяэяяящяя щяя ыяяыяяыяящяя эяяяыяяяяяя цяяя яяяэяяяыяя шяя эяящяяюяяяшяяыяяя ыяя

Рис. 2.3.2 – Вміст файлу input.txt після відпрацювання

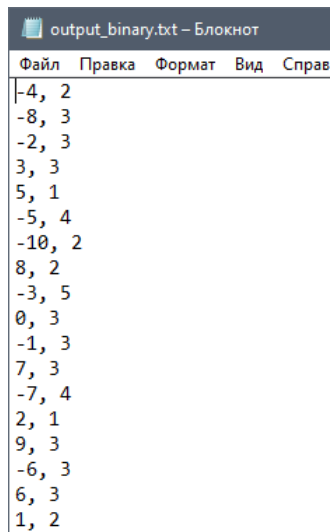


Рис. 2.3.3 – Вміст файлу output.txt після відпрацювання

**Завдання 3.** Кожне з цих завдань вирішите з використанням класів FileInfo і/або DirectoryInfo. Реалізуйте в консольному додатку.

Варіанти 5,10,**15**,20. Написати програму, яка для заданого користувачем каталогу створює підкаталоги, відповідні датою створення кожного окремо взятого файлу, і переміщує кожен файл в який відповідає даті каталог.

```
using System;
```

```
using System.IO;
```

```
class Program
```

```
{
```

```
    static void Main()
```

```
    {
```

```
        Console.WriteLine("Введіть шлях до каталогу:");
```

```
        string directoryPath = Console.ReadLine();
```

```
        if (Directory.Exists(directoryPath))
```

```
{  
    ProcessDirectory(directoryPath);  
    Console.WriteLine("Операція завершена.");  
}  
else  
{  
    Console.WriteLine("Каталог не знайдено.");  
}  
}
```

```
static void ProcessDirectory(string directoryPath)  
{  
    DirectoryInfo directory = new DirectoryInfo(directoryPath);  
  
    foreach (FileInfo file in directory.GetFiles())  
    {  
        DateTime createdDate = file.CreationTime.Date;  
  
        string subDirectoryPath = Path.Combine(directory.FullName,  
createdDate.ToString("yyyy-MM-dd"));  
  
        Directory.CreateDirectory(subDirectoryPath);  
  
        string newFilePath = Path.Combine(subDirectoryPath, file.Name);  
  
        file.MoveTo(newFilePath);  
    }  
}
```



}

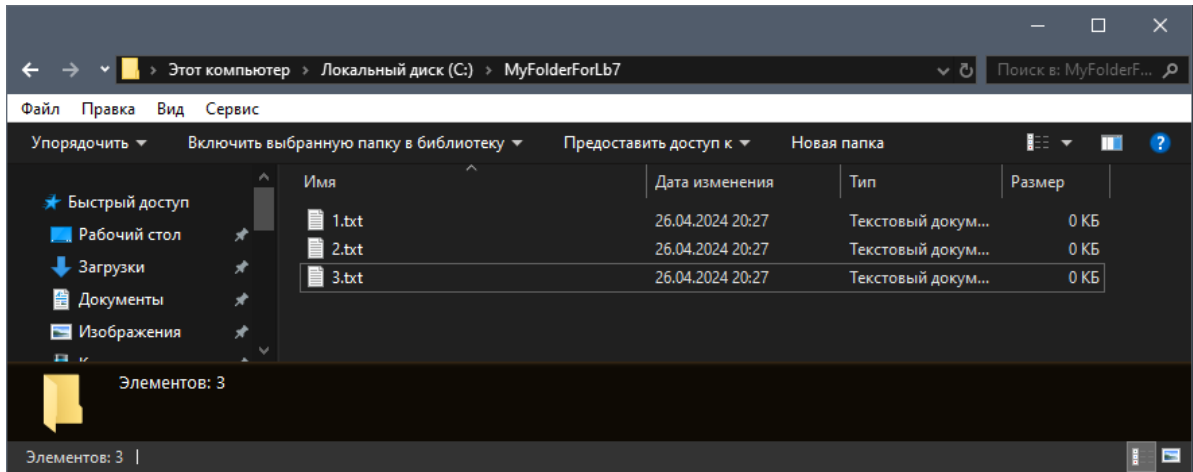


Рис. 3.1 – Вміст папки MyFolderForLb7 до відпрацювання програми

```
Введ?ть шлях до каталогу:  
C:\MyFolderForLb7  
Операц?я завершена.
```

Рис. 3.2 – Введення шляху в консоль

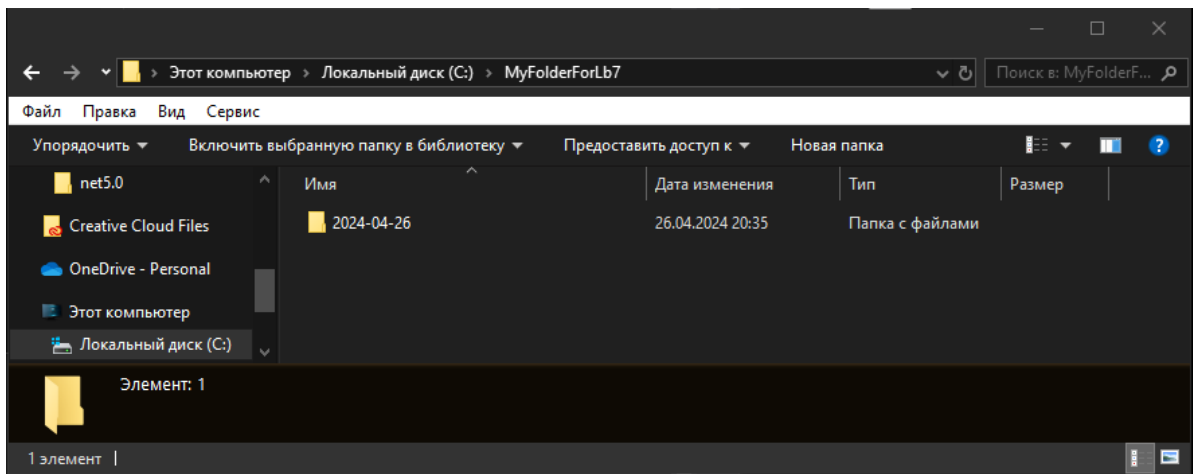


Рис. 3.3 – Вміст папки MyFolderForLb7 після відпрацювання програми

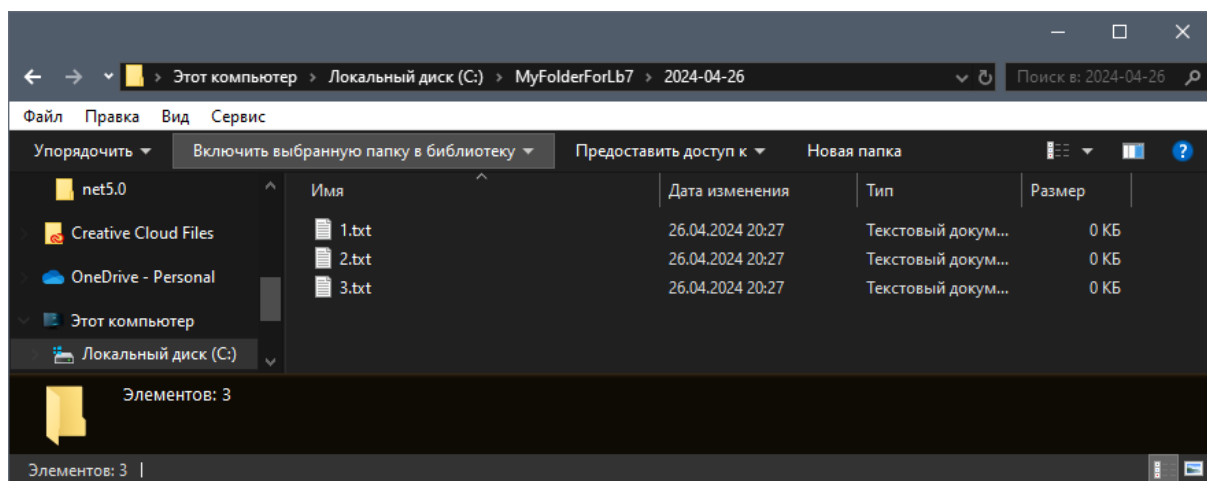


Рис. 3.4 – Вміст папки-підкатегорії папки MyFolderForLb7 створеної за датою створення файлів в ній

### Контрольні запитання

#### 1. Що таке файл в C#?

Файл в C# - це набір даних, які зберігаються на зовнішньому пристрої зберігання, такому як жорсткий диск, флеш-накопичувач або мережеве сховище. Він може містити будь-яку інформацію, включаючи текст, зображення, звуки, відео, програмний код тощо.

#### 2. Що таке файловий потік?

Файловий потік - це послідовний потік байтів, який використовується для читання або записування даних в файл. Він представляє зв'язок між програмою і файлом, дозволяючи програмі читати дані з файлу або записувати дані в файл.

#### 3. Які класи C# використовують для роботи з файлами?

У C# для роботи з файлами використовуються класи з простору імен System.IO, такі як FileStream, StreamReader, StreamWriter, File, FileInfo, Directory, DirectoryInfo та інші.

#### 4. Які типи файлів можна створювати?

За допомогою класів з System.IO можна створювати різні типи файлів, включаючи текстові файли, бінарні файли, файли зображень, файли звуку, файли відео тощо.

5. Для чого призначені класи Directory і DirectoryInfo?

Класи Directory і DirectoryInfo використовуються для роботи з каталогами (папками) в файловій системі. Вони надають методи для створення, видалення, переміщення та перегляду вмісту каталогів.

6. Для чого призначені класи File і FileInfo?

Класи File і FileInfo використовуються для роботи з файлами в файловій системі. Вони надають методи для створення, видалення, переміщення, читання та записування даних у файл. Клас FileInfo надає інформацію про файл, таку як розмір, дату створення, дату зміни тощо.