

Gestion des fichiers

TP 11

R2.01

Malo Mouron

B1

Table des matières

1. Fournir le code nécessaire pour réaliser ce travail.	2
2. Donnez le code nécessaire pour extraire les informations demandées d'un nom de fichier complet.....	3
3. Donnez le code nécessaire pour copier un fichier donné en fichier d'un autre nom dans le même répertoire (§ capture ci-dessous).....	4
4. Donnez le code nécessaire pour copier un fichier donné dans un répertoire donné, mais en conservant le même nom (§ capture ci-dessous)	5
5. Donnez le code nécessaire pour ouvrir un fichier (choisir un petit fichier) et afficher sur la console son contenu, octet par octet, en code hexadécimal, comme sur la capture ci-dessous :.....	6
6. Donnez le code nécessaire pour afficher sur la console le contenu d'un fichier texte dont le chemin est donné, en lisant le fichier en une fois (choisir un PETIT fichier) :.....	7
7. Donnez le code nécessaire pour réaliser ce programme	8
8. Donnez le code nécessaire pour afficher, sur la console, le contenu d'un fichier .zip dont le chemin est donné (nom, taille originale, taille compressée de chaque fichier), comme sur la capture ci-dessous :	9

1. Fournir le code nécessaire pour réaliser ce travail.

```
/// <summary>
/// Traite un répertoire et retourne une représentation en chaîne de
/// chaque fichier dans le répertoire.
/// </summary>
/// <param name="cheminRepertoire">Le chemin du répertoire à
/// traiter.</param>
/// <returns>Une représentation en chaîne de chaque fichier dans le
/// répertoire.</returns>
public static string TraiterRepertoire(string cheminRepertoire)
{
    DirectoryInfo infoRepertoire = new DirectoryInfo(cheminRepertoire);
    FileInfo[] fichiers = infoRepertoire.GetFiles();
    string resultat = "";

    foreach (FileInfo fichier in fichiers)
    {
        FileInfoDetails detailsFichier = new FileInfoDetails(fichier);
        resultat += detailsFichier.GetFileInfo() + "\n";
    }

    return resultat;
}
```

```
Console.WriteLine(FileInfoDetails.TraiterRepertoire("C:/Users/malom/Desktop/mm076547/S2/R2_07 - Graphes/TP/"));
```

1 - Premiers pas.pdf	69599	06/03/2024 13:36:34
2 - Gestion de la carte.pdf	151350	08/03/2024 08:26:28
3 - Voisins.pdf	68974	21/03/2024 10:21:19
4 - Iron Quest.pdf	112355	22/03/2024 15:38:32

```

/// <summary>
/// Initialise une nouvelle instance de la classe FileInfoDetails.
/// </summary>
/// <param name="file">L'objet FileInfo à partir duquel extraire les détails.</param>
public FileInfoDetails(FileInfo file)
{
    _nomFichier = file.Name;
    _nomFichierSansExtension = file.Name.Substring(0, file.Name.Length - file.Extension.Length);
    _extension = file.Extension;
    _nomRepertoire = file.DirectoryName;
    _repertoireRacine = file.Directory.Root.ToString();
    _repertoireParent = file.Directory.Parent.ToString();
    _dateCreation = file.CreationTime;
    _derniereDateEcriture = file.LastWriteTime;
    _derniereDateAcces = file.LastAccessTime;
    _tailleFichier = file.Length;
}

```

Ci-dessus la class que j'utilise pour initialiser toute les données du fichier charger en paramètre qui seront servit dans d'autres fonctions ci-dessous

2. Donnez le code nécessaire pour extraire les informations demandées d'un nom de fichier complet.

```

/// <summary>
/// Obtient des informations de base sur le fichier.
/// </summary>
/// <returns>
/// Une chaîne de caractères contenant le répertoire , le nom du fichier et l'extention du fichier.
/// </returns>
public string GetFileInfoRac()
{
    return "Base Folder : " + NomRepertoire + "\n" +
        "File Name : " + NomFichier + "\n" +
        "Extension : " + Extension;
}

```

```

FileInfo file = new FileInfo("C:/Users/malom/Desktop/mm076547/S2/R2_07 - Graphes/TP/4 - Iron Quest.pdf");
FileInfoDetails fileInfoDetails = new FileInfoDetails(file);
Console.WriteLine(fileInfoDetails.GetFileInfoRac());

```

```

-----
Base Folder :C:\Users\malom\Desktop\mm076547\S2\R2_07 - Graphes\TP
File Name :4 - Iron Quest.pdf
Extension :.pdf

```

3. Donnez le code nécessaire pour copier un fichier donné en fichier d'un autre nom dans le même répertoire (§ capture ci-dessous)



```
Give me a file path :c:\temp\Computer.cs
Copy c:\temp\Computer.cs to c:\temp\Computer (copy).cs done.
```

```
/// <summary>
/// Copies the current file to a new file with the same name, but with "(copy)" appended before the
extension.
/// The new file is created in the same directory as the current file.
/// </summary>
public void CopyMyFile()
{
    // Combine the directory path and the new file name to create the path for the new file
    string newFilePath = Path.Combine(_nomRepertoire, _nomFichierSansExtension + "(copy)" +
_extension);

    // Combine the directory path and the current file name to create the path for the current file
    string oldFilePath = Path.Combine(_nomRepertoire, _nomFichier);

    // Copy the current file to the new file
    File.Copy(oldFilePath, newFilePath);
}
```

```
FileInfo file = new FileInfo("C:/Users/malom/Desktop/mm076547/S2/R2_07 - Graphes/TP/4 - Iron
Quest.pdf");
FileInfoDetails fileInfoDetails = new FileInfoDetails(file);
fileInfoDetails.CopyMyFile();
```

 4 - Iron Quest (copy).pdf	22/03/2024 15:38	Chrome HTML Do...	110 Ko
 4 - Iron Quest.pdf	22/03/2024 15:38	Chrome HTML Do...	110 Ko

4. Donnez le code nécessaire pour copier un fichier donné dans un répertoire donné, mais en conservant le même nom (§ capture ci-dessous)

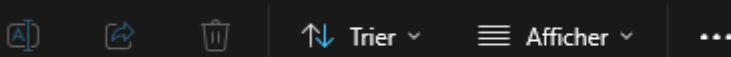

```
Give me a file path :c:\temp\CPU.cs
Give me destination folder path :c:\temp\test
Copy c:\temp\CPU.cs to c:\temp\test\CPU.cs done.
```

```
/// <summary>
/// Copies the current file to a new file with the same name in a specified directory.
/// </summary>
/// <param name="newPath">The directory where the new file will be created.</param>
public void CopyMyFileIntoPath(string newPath)
{
    // Combine the new directory path and the current file name to create the path for the new file
    string newFilePath = Path.Combine(newPath, _nomFichier);

    // Combine the current directory path and the current file name to create the path for the current file
    string oldFilePath = Path.Combine(_nomRepertoire, _nomFichier);

    // Copy the current file to the new file
    File.Copy(oldFilePath, newFilePath);
}
```

```
FileInfo file = new FileInfo("C:/Users/malom/Desktop/mm076547/S2/R2_07 - Graphes/TP/4 - Iron
Quest.pdf");
FileInfoDetails fileInfoDetails = new FileInfoDetails(file);
fileInfoDetails.CopyMyFileIntoPath("C:/Users/malom/Desktop/mm076547/S2/R2_07 - Graphes");
```

Bureau > mm076547 > S2 > R2_07 - Graphes >		
		
Nom	Modifié le	Type
 4 - Iron Quest.pdf	22/03/2024 15:38	Chron

5. Donnez le code nécessaire pour ouvrir un fichier (choisir un petit fichier) et afficher sur la console son contenu, octet par octet, en code hexadécimal, comme sur la capture ci-dessous :

```
Give me a file path :c:\temp\CPU.cs
63 6c 61 73 73 20 43 50
55 7b 0d 0a 0d 0a 7d
```

```
/// <summary>
/// Affiche le contenu du fichier en hexadécimal.
/// </summary>
public static void DisplayFileContentInHexadecimal()
{
    Console.WriteLine("Give me a file path : ");
    var filePath = Console.ReadLine();
    using (var fileStream = File.OpenRead(filePath))
    {
        var buffer = new byte[1];

        while (fileStream.Read(buffer, 0, buffer.Length) > 0)
            Console.WriteLine(buffer[0].ToString("X2") + " ");
    }
    Console.WriteLine();
}
```

```
FileInfoDetails.DisplayFileContentInHexadecimal();
```

```
Give me a file path : C:\Users\malom\Desktop\mm076547\S2\R2_07 - Graphes\TP\4
- Iron Quest.pdf
25 50 44 46 2D 31 2E 35 0A 25 E4 F0 ED F8 0A 31 36 20 30 20 6F 62 6A 0A 3C 3C
2F 54 79 70 65 2F 58 4F 62 6A 65 63 74 2F 53 75 62 74 79 70 65 2F 49 6D 61 67
65 2F 57 69 64 74 68 20 35 37 30 2F 48 65 69 67 68 74 20 35 34 30 2F 43 6F 6C
```

6. Donnez le code nécessaire pour afficher sur la console le contenu d'un fichier texte dont le chemin est donné, en lisant le fichier en une fois (choisir un PETIT fichier) :

```
/// <summary>
///  afficher sur la console le contenu d'un fichier texte dont le chemin est donné.
/// </summary>
public static void DisplayFileContent()
{
    // Prompt the user for a file path
    Console.WriteLine("Give me a file path : ");
    var filePath = Console.ReadLine();

    // Assert that the file path is not null
    Debug.Assert(filePath != null, nameof(filePath) + " != null");

    // Open the file at the given path for reading
    using var reader = new StreamReader(filePath);

    // Read the entire content of the file
    var contenu = reader.ReadToEnd();

    // Print the content of the file to the console
    Console.WriteLine(contenu);
}
```

```
FileInfoDetails.DisplayFileContent();
```

```
-----
Veuillez entrer le chemin du fichier : C:/test.txt
Ceci est un fichier test et sont contenue
sdbcsbc
scsbcjisncsc
```


7. Donnez le code nécessaire pour réaliser ce programme

```
/// <summary>
/// Writes user input to a file.
/// </summary>
public static void WriteUserInputToFile()
{
    // Prompt the user for a file path
    Console.Write("Give me a file path: ");
    var filePath = Console.ReadLine();

    // Check if the file path is writable
    if (!File.Exists(filePath) || !new FileInfo(filePath).IsReadOnly)
    {
        // Initialize a boolean variable to control the loop
        bool continueWriting = true;

        // Enter a loop that continues until the user enters "fin"
        while (continueWriting)
        {
            // Prompt the user for a text
            Console.Write("Please enter a text (enter 'fin' to stop): ");
            var text = Console.ReadLine();

            // If the text is "fin", set continueWriting to false
            if (text == "fin")
            {
                continueWriting = false;
            }
            else
            {
                // Otherwise, append the text as a new line in the file
                using (var streamWriter = File.AppendText(filePath))
                {
                    streamWriter.WriteLine(text);
                }
            }
        }
    }
    else
    {
        // If the file path is not writable, inform the user
        Console.WriteLine("The file path is not writable.");
    }
}
```

```
FileInfoDetails.WriteUserInputToFile();
```

```
-----
Give me a file path: C:/test.txt
Please enter a text (enter 'fin' to stop): Ligne1
Please enter a text (enter 'fin' to stop): Ligne2
Please enter a text (enter 'fin' to stop): Test
Please enter a text (enter 'fin' to stop): fin
```

```
Ceci est un fichier test et sont contenue
sdbcsbc
scsbcjisncsc
Ligne1
Ligne2
Test
```

8. Donnez le code nécessaire pour afficher, sur la console, le contenu d'un fichier .zip dont le chemin est donné (nom, taille originale, taille compressée de chaque fichier), comme sur la capture ci-dessous :

```
Give me a zip-file path :c:\temp\temp.zip
Computer.cs - 295 / 148
CPU.cs - 15 / 15
instrument.cs - 28 / 28
Musicien.cs - 303 / 164
Orchestra.cs - 600 / 231
Printer.cs - 0 / 0
```

```
/// <summary>
/// Displays the details of a zip file.
/// </summary>
public static void DisplayZipFile()
{
    // Prompt the user for a zip file path
    Console.Write("Give me a zip-file path : ");
    string zipFilePath = Console.ReadLine();

    // Open the zip file for reading
    using (ZipArchive archive = ZipFile.OpenRead(zipFilePath))
    {
        // Iterate over each entry in the zip file
        foreach (ZipArchiveEntry entry in archive.Entries)
        {
            // Print the name, original size, and compressed size of each entry
            Console.Write($"Nom : {entry.FullName} ; ");
            Console.Write($"Taille originale : {entry.Length} bytes ; ");
            Console.WriteLine($"Taille compressée : {entry.CompressedLength} bytes");
        }
        // Print a separator line
        Console.WriteLine("-----");
    }
}
```

```
FileInfoDetails.DisplayZipFile();
```

```
-----
Give me a zip-file path : C:\Users\malom\Desktop\test.zip
Nom : test - Copie.txt ; Taille originale : 86 bytes ; Taille compressée : 74 bytes
Nom : test.txt ; Taille originale : 86 bytes ; Taille compressée : 74 bytes
-----
```

