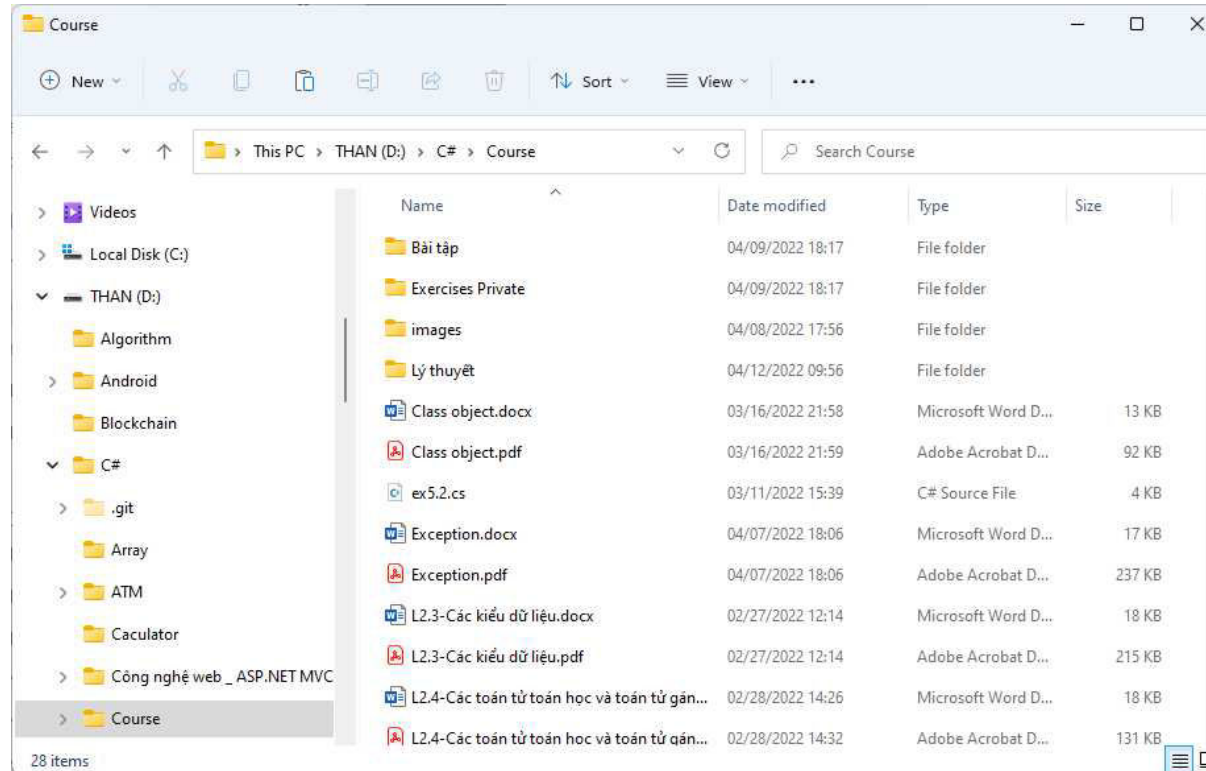


Bài 9.1: Thao tác với file và thư mục

- ✓ Mục đích sử dụng file, thư mục
- ✓ Tìm hiểu namespace System.IO
- ✓ Tìm hiểu lớp DirectoryInfo
- ✓ Tìm hiểu lớp FileInfo
- ✓ Ví dụ minh họa

Mục đích sử dụng của file, thư mục

- ✓ Sử dụng file, thư mục để quản lý, lưu trữ dữ liệu dài hạn trong máy tính.
- ✓ Thư mục được sử dụng để quản lý các thư mục con và các file bên trong nó.
- ✓ File hay tệp dùng để chứa dữ liệu với các định dạng khác nhau: hình ảnh, âm thanh, text.



Namespace System.IO

- ✓ Namespace System.IO chứa các lớp cha để quản lý các thao tác đọc ghi file, quản lý thư mục.
- ✓ Ngoài ra trong đó cũng chứa các interface, enum, struct, delegate phục vụ các thao tác với file, thư mục.
- ✓ Các kiểu trong namespace này tập trung vào việc dùng chương trình để thao tác với file, thư mục vật lý.
- ✓ Một số kiểu còn lại hỗ trợ thao tác đọc ghi với bộ đệm cũng như địa chỉ gốc của bộ nhớ.

Namespace System.IO

Các kiểu	Mục đích sử dụng
<code>BinaryReader</code>	Các lớp này cho phép lưu trữ và truy cập các giá trị của kiểu nguyên thủy(integer, string, bool,..) giống như một giá trị nhị phân.
<code>BinaryWriter</code>	
<code>BufferedStream</code>	Lớp này cung cấp vùng nhớ tạm cho luồng các byte sẽ được lưu vào bộ nhớ về sau.
<code>Directory</code>	Sử dụng các lớp này để quản lý cấu trúc thư mục trong máy tính. Lớp <code>Directory</code> sử dụng các phương thức static. Lớp <code>DirectoryInfo</code> hỗ trợ các tính năng tương tự nhưng sử dụng đối tượng để tham chiếu.
<code>DirectoryInfo</code>	
<code>DriveInfo</code>	Lớp này cung cấp thông tin chi tiết về ổ đĩa mà máy đang sử dụng.
<code>File</code>	Các lớp này hỗ trợ quản lý các file trong máy tính. Lớp <code>File</code> quản lý thao tác qua các phương thức static. Lớp <code>FileInfo</code> cung cấp các chức năng tương tự nhưng qua tham chiếu đối tượng.
<code>FileInfo</code>	
<code>FileStream</code>	Hỗ trợ truy cập file ngẫu nhiên trong đó dữ liệu được biểu diễn ở dạng luồng các byte.
<code>FileSystemWatcher</code>	Lớp này hỗ trợ giám sát sự thay đổi của các file ngoại vi trong thư mục được chỉ định.
<code>MemoryStream</code>	Lớp này hỗ trợ truy cập ngẫu nhiên vào dữ liệu lưu trong bộ nhớ thay vì lưu trong các file vật lý.
<code>Path</code>	Lớp này cung cấp các thông tin về tên file, thư mục và đường dẫn của chúng.
<code>StreamWriter</code>	Sử dụng các lớp này để ghi, đọc dữ liệu dạng text. Không hỗ trợ truy cập ngẫu nhiên.
<code>StreamReader</code>	
<code>StringWriter</code>	Giống hai lớp bên trên, các lớp này cũng hỗ trợ thao tác với dữ liệu dạng text. Mặt khác, nó lưu trữ dữ liệu trong bộ đệm string thay vì lưu trong file vật lý.
<code>StringReader</code>	

Lớp FileSystemInfo

- ✓ Lớp File và Directory kế thừa trực tiếp từ lớp Object.
- ✓ Trong khi đó FileInfo và DirectoryInfo cung cấp chức năng tương tự nhưng kế thừa từ lớp trừu tượng FileSystemInfo.
- ✓ Sử dụng FileInfo và DirectoryInfo là lựa chọn tốt hơn để quản lý chi tiết một file hoặc thư mục vì các phương thức của nó thường trả về một đối tượng thay vì string.
- ✓ Trong hầu hết các trường hợp ta sử dụng các thành phần của lớp cha FileSystemInfo để truy xuất thông tin đặc điểm của một file hoặc thư mục nào đó.

Thuộc tính	Mô tả
Attributes	Truy cập hoặc thiết lập các giá trị đặc trưng liên kết với file hiện tại.
CreationTime	Đọc ra hoặc thiết lập thời gian tạo file, thư mục.
Exists	Xác định xem liệu một file, thư mục cho trước có tồn tại không.
Extension	Truy cập phần mở rộng(đuôi file).
FullName	Lấy đường dẫn đầy đủ của một file hoặc thư mục.
LastAccessTime	Lấy hoặc thiết lập thời gian truy cập cuối vào file hoặc thư mục hiện thời.
LastWriteTime	Lấy hoặc thiết lập thời gian cuối cùng file/thư mục được ghi dữ liệu vào.
Name	Quản lý tên của thư mục/file hiện tại.

Lớp DirectoryInfo

- ✓ Lớp này chứa tập các phương thức cho phép tạo mới, di chuyển, xóa, liệt kê các thư mục và thư mục con.
- ✓ Các thành phần chính của lớp:

Thành phần	Mô tả
Create()	Tạo mới một thư mục khi cung biết tên đường dẫn.
CreateSubdirectory()	Tạo mới một hoặc 1 tập thư mục con khi cung cấp tên đường dẫn.
Delete()	Xóa một thư mục và tất cả nội dung bên trong.
GetDirectories()	Trả về một mảng các đối tượng DirectoryInfo đại diện cho tất cả các thư mục con hiện có trong thư mục được chỉ định.
GetFiles()	Trả về một mảng các đối tượng FileInfo đại diện cho các file có trong thư mục được chỉ định.
MoveTo()	Di chuyển một thư mục và nội dung bên trong nó đến một nơi khác.
Root	Lấy tên ổ cứng/ thư mục gốc
Parent	Lấy tên thư mục cha của thư mục hiện tại.

Ví dụ

Ví dụ sau truy cập vào thư mục Wallpaper của hệ điều hành window và liệt kê các thông tin về thư mục, file ảnh .jpg trong đó:

```
DirectoryInfo dir = new DirectoryInfo(@"C:\Windows\Web\Wallpaper");
Console.WriteLine("=> Directory infomation:");
Console.WriteLine("FullName: " + dir.FullName);
Console.WriteLine("Name: " + dir.Name);
Console.WriteLine("Parent: " + dir.Parent);
Console.WriteLine("Creation: " + dir.CreationTime);
Console.WriteLine("Attributes: " + dir.Attributes);
Console.WriteLine("Root: " + dir.Root);
var files = dir.GetFiles("*.jpg", SearchOption.AllDirectories);
foreach (var file in files)
{
    Console.WriteLine("File name: " + file.FullName);
    Console.WriteLine("File size: " + file.Length);
    Console.WriteLine("File creation: " + file.CreationTime);
    Console.WriteLine("File attributes: " + file.Attributes);
    Console.WriteLine("=====");
}
```


Lưu ý

Nếu chạy chương trình C# trên nền tảng không phải window ta sử dụng:

- ✓ Path.VolumeSeparatorChar để lấy kí tự phân tách giữa ổ đĩa và kí tự phân tách thư mục.
- ✓ Path.DirectorySeparatorChar để lấy kí tự phân tách các thư mục.

```
var volumSeparator = Path.VolumeSeparatorChar;  
var dirSeparator = Path.DirectorySeparatorChar;  
var path = $"C{volumSeparator}{dirSeparator}Windows{dirSeparator}Web{dirSeparator}Wallpaper";
```

```
DirectoryInfo dir = new DirectoryInfo(path);  
Console.WriteLine("==> Directory infomation:");  
Console.WriteLine("FullName: " + dir.FullName);  
Console.WriteLine("Name: " + dir.Name);  
Console.WriteLine("Parent: " + dir.Parent);  
Console.WriteLine("Creation: " + dir.CreationTime);  
Console.WriteLine("Attributes: " + dir.Attributes);  
Console.WriteLine("Root: " + dir.Root);  
var files = dir.GetFiles("*.jpg", SearchOption.AllDirectories);
```

- ✓ Để truy cập đến thư mục hiện tại ta sử dụng 1 dấu chấm.

Kết quả minh họa

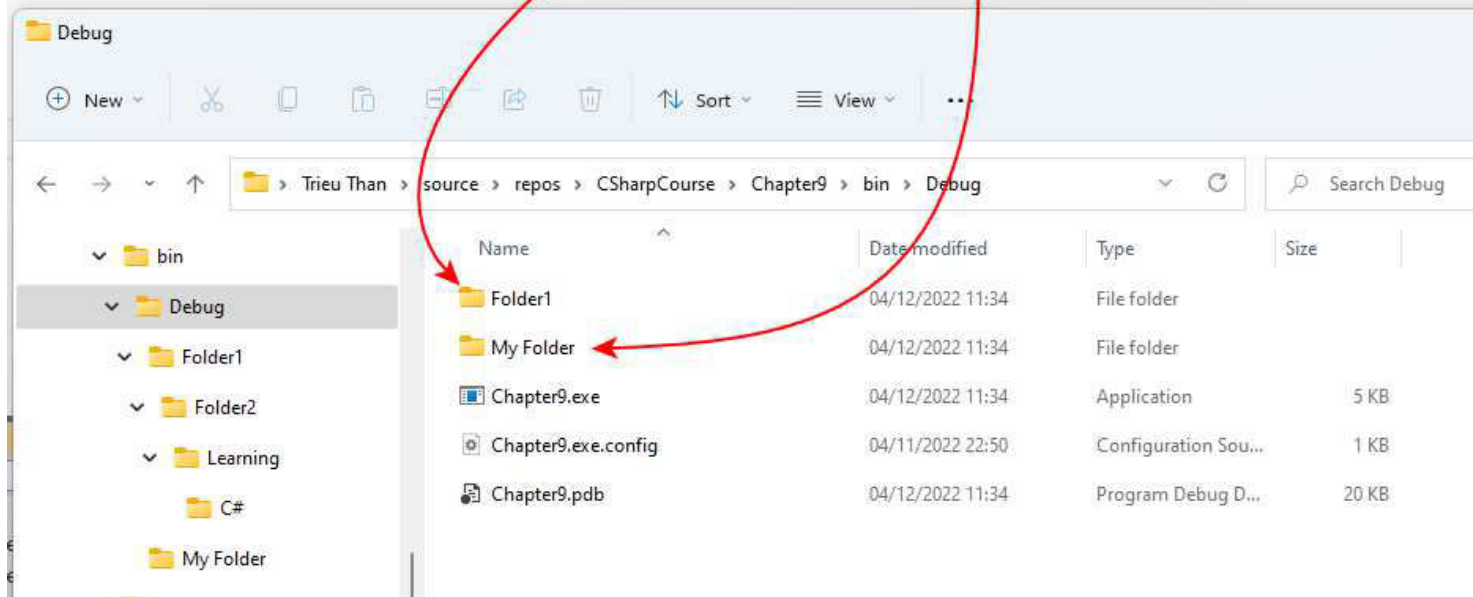
```
==> Directory information:
FullName: C:\Windows\Web\Wallpaper
Name: Wallpaper
Parent: Web
Creation: 06/05/2021 19:10:49
Attributes: Directory
Root: C:\
File name: C:\Windows\Web\Wallpaper\ThemeA\img20.jpg
File size: 826734
File creation: 06/05/2021 19:06:16
File attributes: Archive
=====
File name: C:\Windows\Web\Wallpaper\ThemeA\img21.jpg
File size: 833851
File creation: 06/05/2021 19:06:16
File attributes: Archive
=====
File name: C:\Windows\Web\Wallpaper\ThemeA\img22.jpg
File size: 760748
File creation: 06/05/2021 19:06:16
File attributes: Archive
=====
File name: C:\Windows\Web\Wallpaper\ThemeA\img23.jpg
File size: 907112
File creation: 06/05/2021 19:06:16
File attributes: Archive
```

Ví dụ tạo thư mục

Ví dụ sau tạo các thư mục mới và thư mục con trong thư mục hiện tại.

```
// tạo một thư mục mới trong thư mục hiện tại
DirectoryInfo myDir = new DirectoryInfo(@"..\My Folder");
myDir.Create();

// tạo chuỗi các thư mục con
DirectoryInfo dir = new DirectoryInfo(".");
dir.CreateSubdirectory(@"Folder1\Folder2\Learning\C#");
```



Lớp FileInfo

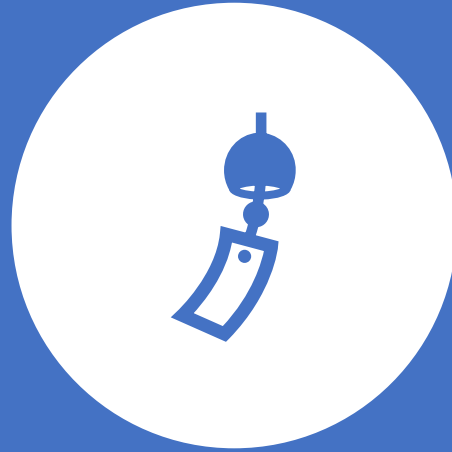
Cho phép quản lý thông tin chi tiết các file tồn tại trong ổ cứng và cho phép thực hiện các hành động tạo mới, sao chép, di chuyển, xóa file.

Thành phần	Mô tả
<code>AppendText()</code>	Tạo một StreamWriter để thêm dữ liệu dạng text vào file.
<code>Create()</code>	Tạo mới một file khi cho biết tên đường dẫn.
<code>CopyTo()</code>	Sao chép một file đã tồn tại và nội dung của nó vào một file mới.
<code>CreateText()</code>	Tạo một StreamWriter ghi nội dung mới vào file.
<code>Delete()</code>	Xóa file mà đối tượng FileInfo đang tham chiếu tới.
<code>Directory</code>	Lấy đối tượng của thư mục cha.
<code>DirectoryName</code>	Lấy đường dẫn đầy đủ đến thư mục cha.
<code>Length</code>	Lấy kích thước của file hiện tại tính theo byte.
<code>Name</code>	Lấy tên file.
<code>MoveTo()</code>	Di chuyển một file đến một nơi khác.
<code>Open()</code>	Mở một file với nhiều quyền đọc ghi, chia sẻ.
<code>OpenRead()</code>	Mở một đối tượng FileStream ở chế độ chỉ đọc.
<code>OpenText()</code>	Tạo một đối tượng StreamReader cho phép đọc từ một file text cho trước.
<code>OpenWrite()</code>	Mở một FileStream ở chế độ chỉ ghi.

Ví dụ

```
// ví dụ sau tạo file input1.txt
var fileName = "input1.txt";
try
{
    FileInfo fileInfo = new FileInfo(fileName);
    var fs = fileInfo.Create();
    // thực hiện đọc ghi... với đối tượng file stream vừa tạo
    fs.Close();
    // fileInfo.Delete(); // xóa bỏ file vừa tạo
}
catch (IOException e)
{
    Console.WriteLine(e);
}

// ví dụ sau mở file input.txt để đọc ghi
var fileToOpen = "input.txt";
FileInfo file = new FileInfo(fileToOpen);
var reader = file.Open(FileMode.OpenOrCreate, FileAccess.ReadWrite, FileShare.None);
try
{
    int data;
    while ((data = reader.ReadByte()) != -1)
    {
        // do something here
    }
}
finally
{
    reader.Close();
}
```



Nội dung tiếp theo

Thao tác đọc ghi file text