

# Bài 9: Làm việc với File & Thư mục

Lương Trần Hy Hiến

FIT, HCMUP

*Lập trình Windows Form với C#*

# Nội dung

- **Quản lý ổ đĩa, thư mục, file**
- **Move, Copy, Delete file**
- **Đọc/Ghi file**

# Quản lý ổ đĩa, Thư mục và File

- **Namespace: `System.IO`**
- **Thao tác quản lý File và Thư Mục**
  - Xem danh sách ổ đĩa
  - Danh sách thư mục, file
  - Xem các thuộc tính của thư mục, file

# Quản lý ổ đĩa

- **Class : DriveInfo**
- **Cho phép thao tác, đọc thông tin trên ổ đĩa**
- **Danh sách các thuộc tính & hàm**

Tên	Ý nghĩa
TotalFreeSpace	Xem dung lượng đĩa trống
TotalSize	Xem tổng dung lượng đĩa
DriveType	Cho biết loại ổ đĩa. Kiểu dữ liệu trả về là DriveType: CDRom, Fixed, Network, Ram, Removable, Unknown
IsReady	Cho biết ổ đĩa đã sẵn sàng Read/Write.
Name	Tên ổ đĩa
GetDrives	Lấy danh sách ổ đĩa hiện có

## Quản lý ổ đĩa

```
DriveInfo[] arrDrives = DriveInfo.GetDrives();  
foreach (DriveInfo drive in arrDrives)  
{  
    if (drive.IsReady)  
    {  
        Console.WriteLine("Drive name:" + drive.Name);  
        Console.WriteLine("Free space: " + drive.TotalFreeSpace);  
        Console.WriteLine("Total size: " + drive.TotalSize);  
        Console.WriteLine("Root directory:" + drive.RootDirectory.FullName);  
    }  
}
```

Drive name:C:\

Free space: 6853718016

Total size: 20971528192

Root directory:C:\

# Quản lý thư mục

- **Class: DirectoryInfo**
- **Thao tác trên thư mục: tạo, xóa, sửa tên thư mục**
- **Danh sách các hàm và thuộc tính**

Tên	Ý nghĩa
CreationTime	Xem hoặc thiết lập thời gian tạo thư mục
Exists	Kiểm tra thư mục tồn tại trên ổ đĩa
FullName	Lấy đường dẫn của tới thư mục
LastAccessTime	Cho biết thời gian cuối cùng thư mục (file) được truy cập
Name	Cho biết tên của thư mục
<b>DirectoryInfo</b> Parent	Trả về thư mục cha.
<b>FileAttributes</b> Attributes	Cho biết thuộc tính của thư mục (file) FileAttributes là 1 enum gồm các giá trị như: Directory, Readonly, Hidden,...

# Quản lý thư mục

```
DirectoryInfo di = new DirectoryInfo(@"D:\Rac .NET\MyApp");  
if (di.Exists)  
{  
    Console.WriteLine("Created Date: " + di.CreationTime.ToShortDateString());  
    Console.WriteLine("Full name: " + di.FullName);  
    Console.WriteLine("Name: " + di.Name);  
    Console.WriteLine("Parent Directory: " + di.Parent.Name);  
    Console.WriteLine("Root: " + di.Root.Name);  
}
```

Created Date: 3/15/2008

Full name: D:\Rac .NET\MyApp

Name: MyApp

Parent Directory: Rac .NET

Root: D:\

# Quản lý thư mục

## Danh sách các hàm và thuộc tính

Tên	Ý nghĩa
Create	Tạo thư mục
Delete	Xóa thư mục
GetDirectories	Trả danh sách thư mục con hiện thời.
GetFiles	Trả về danh sách file của thư mục hiện thời.
MoveTo	Di chuyển thư mục đến nơi khác; hoặc đổi tên thư mục



# Quản lý thư mục

## ➤ Tạo thư mục

```
DirectoryInfo di = new DirectoryInfo(@"D:\Testing");  
if (!di.Exists)  
    di.Create();  
return;
```

## ➤ Xóa thư mục

```
DirectoryInfo di = new DirectoryInfo(@"D:\Testing");  
if (di.Exists)  
    di.Delete();  
return;
```

# Quản lý thư mục

## ➤ Đổi tên thư mục

```
DirectoryInfo di = new DirectoryInfo(@"D:\Testing");  
if (di.Exists)  
    di.MoveTo(di.Root.Name + @"\\" + "Hello");  
return;
```

## ➤ Di chuyển thư mục

```
DirectoryInfo di = new DirectoryInfo(@"D:\Testing");  
if (di.Exists)  
    di.MoveTo(@"C:\Hello");  
return;
```

# Quản lý thư mục

## ➤ Truy xuất thư mục con.

```
DirectoryInfo di = new DirectoryInfo(@"D:\Hello");  
if (di.Exists)  
{  
    DirectoryInfo[] arrSubFolders = di.GetDirectories();  
    foreach (DirectoryInfo subFolder in arrSubFolders)  
        Console.WriteLine("Name: " + subFolder.Name);  
}
```

## ➤ Truy xuất danh sách files

```
DirectoryInfo di = new DirectoryInfo(@"D:\Hello");  
if (di.Exists)  
{  
    FileInfo[] arrFiles = di.GetFiles();  
    foreach (FileInfo file in arrFiles)  
        Console.WriteLine("Name: " + file.Name);  
}
```

## Quản lý file

Tên	Ý nghĩa
CreationTime	Thời gian tạo file hoặc folder
DirectoryName (FileInfo), Parent (DirectoryInfo)	Đường dẫn đầy đủ của folder chứa file hoặc folder hiện thời
Exists	File hay folder có tồn tại hay không?
Extension	Phần mở rộng
FullName	Tên đầy đủ, cả đường dẫn
LastAccessTime	Thời gian lần truy cập cuối
LastWriteTime	Thời gian lần sửa chữa cuối
Name	Tên file hay folder
Root	Folder gốc (chỉ với DirectoryInfo)
Length	Dung lượng (bytes), chỉ với FileInfo

## Quản lý file (tt)

//khởi tại biến myFile trỏ đến tập tin

```
FileInfo myFile = new FileInfo(@"d:\\HyHienProfile.txt");
```

//sao chép sang ổ đĩa E

```
myFile.CopyTo(@"E:\\test.txt");
```

//kiểm tra sự tồn tại

```
MessageBox.Show(myFile.Exists.ToString());
```

//thông tin thời điểm tạo file

```
MessageBox.Show(myFile.CreationTime.ToString());
```

//cập nhật thời điểm tạo file

```
myFile.CreationTime = new DateTime(2009, 7, 27, 12, 0, 0);
```

# Quản lý file (tt)

```
DirectoryInfo theFolder = new DirectoryInfo(folderFullName);  
if (!theFolder.Exists)  
    throw new DirectoryNotFoundException("Folder not found: " + folderFullName);  
string currentPath = theFolder.FullName;  
//Lấy tên các thư mục con của thư mục hiện thời
```

```
ArrayList folders = new ArrayList();  
foreach (DirectoryInfo folder in theFolder.GetDirectories())  
    folders.Add(folder.Name);
```

```
//Lấy tên các file trong thư mục hiện thời  
ArrayList files = new ArrayList();  
foreach (FileInfo file in theFolder.GetFiles())  
    files.Add(file.Name);
```

## Moving, Copying, and Deleting Files

- **Có thể sao chép, di chuyển hoặc xóa file.**
  - Phương thức `Path.Combine(string, string)`: trả về tên đầy đủ của file tạo từ đường dẫn và tên file.
  - Phương thức `File.Delete(string)`: xóa tập tin.
  - Phương thức `File.Move(string, string)`: di chuyển file từ vị trí cũ đến vị trí mới.
  - Phương thức `File.Copy(string, string)`: sao chép file sang thư mục mới

# Đọc ghi file

- Dựa trên khái niệm **stream** (luồng dữ liệu):
  - **stream** là đối tượng dùng để chuyển dữ liệu. Do đó stream có thể là luồng dựa trên bộ nhớ, trên tập tin, trên mạng,...
  - **FileStream**: đối tượng dùng để đọc ghi file nhị phân.
  - **StreamReader** và **StreamWriter**: đối tượng dùng để đọc ghi file text
  - Chú ý: Các tác vụ đọc ghi hầu hết đều sử dụng buffer. Do đó, với tác vụ ghi, phải đưa dữ liệu từ buffer lên đĩa trước khi đóng file.



# Đọc ghi file nhị phân

- **Hàm tạo:** chỉ cần chỉ ra filename, FileMode, FileAccess, FileShare.
  - **FileMode:** kiểu mở file, có thể là Append, Create, CreateNew, Open, OpenOrCreate hoặc Truncate.
  - **FileAccess:** kiểu truy cập, có thể là Read, ReadWrite hoặc Write.
  - **FileShare:** kiểu chia sẻ giữa các thread, có thể là Inheritable, None, Read, ReadWrite hoặc Write.
- **Để đọc và ghi byte, dùng các hàm:**
  - **ReadByte():** đọc 1 byte từ stream
  - **WriteByte(byte):** ghi 1 byte vào stream
  - **Read/Write(byte[], int off, int count):** đọc/ghi một mảng byte bắt đầu từ off, độ dài count
- **Close():** Đóng file

# Đọc ghi file văn bản

- Dùng **StreamReader** và **StreamWriter**
  - Có thể khởi tạo StreamReader dựa trên:
    - Tên file cần đọc
    - Một FileStream khác
    - Một FileInfo với phương thức **OpenText()**
  - Có thể khởi tạo StreamWriter dựa trên:
    - Tên file cần đọc, mã encode
    - Một FileStream khác
    - Một FileInfo với phương thức **CreateText()**
  - Để đọc và ghi, dùng các hàm:
    - **Read()** hoặc **Write()**: đọc hoặc ghi 1 ký tự
    - **ReadLine()** hoặc **WriteLine()**: đọc hoặc ghi 1 dòng
    - **ReadToEnd()**: đọc đến hết file

## VD: Hàm ghi file văn bản

```
void GhiFileText(string sfilename, string chuoi)
{
    FileStream fstream = new FileStream(sfilename,
        FileMode.Append, FileAccess.Write, FileShare.None);
    StreamWriter mywriter = new StreamWriter(fstream);
    mywriter.WriteLine(DateTime.Now.ToString() + " - " +
chuoi);
    mywriter.Close();
    fstream.Close();
}
```

## VD: Hàm đọc file văn bản

```
string DocFileText(string sfilename)
{
    StreamReader myreader = null;
    string result = string.Empty;
    try {
        myreader = new StreamReader(sfilename);
        result = myreader.ReadToEnd();
    } catch {
        throw;
    } finally {
        if(myreader != null) myreader.Close();
    }
    return result;
}
```

