# Windows form

## Bài 18.3

Làm việc với File và Thư Mục

C Sharp

Dễ Hiểu - Dễ học


# Part 3 : FileStream Đọc - ghi file C#

# 1. Stream

## ❑ 1. Stream :

*Stream trong C# là một cơ chế hỗ trợ đọc ghi dữ liệu đặc biệt*

✓*Tại sao phải dùng stream :*

*Khi đọc dữ liệu từ một file lớn, nếu đọc toàn bộ dữ liệu cùng lúc,chương trình có thể treo vì không thể xử lý khối lượng dữ liệu quá lớn.*

** Stream giúp đọc dữ liệu theo từng khối nhỏ hoặc từng byte riêng rẽ.*

** Chương trình sau đó đọc dữ liệu từ stream.*

✓*Hiểu đơn giản stream giống như một đường ống nối chương trình với nguồn dữ liệu và cho phép một chuỗi byte (giống như dòng nước) chạy qua.*

## 2.FileStream

### ❑ 2. Khái quát FileStream :

❖ *FileStream là một loại stream đặc biệt chuyên dùng để đọc ghi dữ liệu với file*

❖ *Các phương thức chuyên dụng để làm việc với dữ liệu cấp cao theo từng loại định dạng cụ thể:*

- ✓*lớp StreamWriter/StreamReader để làm việc với văn bản;*
- ✓*lớp BinaryWriter/BinaryReader để làm việc với các kiểu dữ liệu cơ sở (int, bool,char...);*
- ✓*lớp XmlWriter/XmlReader làm việc với xml.*

## 2.FileStream

❑ *3 . StreamWriter/StreamReader để làm việc với văn bản*

### 3.1 StreamWriter : Ghi file văn bản

```
FileStream fs = new FileStream("data1.txt", FileMode.Create, FileAccess.ReadWrite);
StreamWriter sWriter = new StreamWriter(fs);
sWriter.Write("Hello world");
sWriter.Flush();
fs.Close();
```

*Chú ý :*
*sWriter.Flush(); sẽ chuyển mọi thứ trong Luồng ra ngoài tệp.( xả luồng )*
*✓Điều này có thể được thực hiện ở giữa quá trình sử dụng Luồng và bạn có thể tiếp tục viết.*
*✓ fs.Close();*
*đóng Luồng . Điều này bao gồm Xả luồng một lần cuối cùng.*

*✓FileMode.CreateNew tạo file mới*
*✓FileMode.Create tạo mới, nếu file đang có bị ghi đè*
*✓FileMode.Open mở file đang tồn tại*
*✓FileMode.OpenOrCreate mở file đang tồn tại, tạo mới nếu không có*
*✓FileMode.Truncate mở file đang tồn tại và làm rỗng file*
*✓FileMode.Append mở file đang tồn tại và tới cuối file, hoặc tạo mới*

*File ccess kiểu liệt kê FileAccess, cho biết muốn truy cập file như thế nào*
*✓FileAccess.Read chỉ đọc*
*✓FileAccess.Write chỉ ghi*
*✓FileAccess.ReadWrite đọc và ghi*

## 2.FileStream

### ❑ 3 . *StreamWriter/StreamReader để làm việc với văn bản*

### 3.2 StreamReader : Đọc file văn bản

```
fs = new FileStream("data1.txt", FileMode.OpenOrCreate, FileAccess.Read);
StreamReader sReader = new StreamReader(fs);
var str = sReader.ReadToEnd();
Console.WriteLine($"str = {str}");
fs.Close();
```

### ❑ 4 . BinaryWriter / BinaryReader

*Để làm việc với các kiểu dữ liệu cơ sở (int, bool,char...);*

```
FileStream fs2 = new FileStream("data2.txt", FileMode.Create, FileAccess.ReadWrite);
BinaryWriter bWriter = new BinaryWriter(fs2);
bWriter.Write(1234);
fs2.Close();  //đóng Luồng .
fs2 = new FileStream("data2.txt", FileMode.OpenOrCreate, FileAccess.Read);
BinaryReader bReader = new BinaryReader(fs2);
var i = bReader.ReadInt32();
Console.WriteLine($"i = {i}");
```

*Chú ý :*

*Luôn phải nhớ sử dụng .Close để đóng file sau khi sử dụng xong*

## 2.FileStream

### ❑ 5 . *Using*

***Trong các ví dụ trên, sau khi kết thúc làm việc với file, chúng ta phải tự mình gọi lệnh đóng luồng file. Đây là một thao tác rất hay bị bỏ quên.***

***✓Để giải phóng người lập trình khỏi việc phải tự mình hủy bỏ các object như vậy, C# cung cấp một cấu trúc mới: using block.***

***✓ Khi kết thúc khối code, biến fs sẽ tự bị hủy bỏ***

```
using (FileStream fs = new FileStream("data3.txt", FileMode.Create, FileAccess.ReadWrite))
{
    BinaryWriter bWriter = new BinaryWriter(fs);
    bWriter.Write(1234);
    StreamWriter sWriter = new StreamWriter(fs);
    sWriter.Write("Hello world");
    sWriter.Flush();
}
using (var fs = new FileStream("data3.txt", FileMode.OpenOrCreate, FileAccess.Read))
{
    BinaryReader bReader = new BinaryReader(fs);
    var i = bReader.ReadInt32();
    StreamReader sReader = new StreamReader(fs);
    var str = sReader.ReadToEnd();
    Console.WriteLine($"i = {i}");
    Console.WriteLine($"str = {str}");
}
```

## 3. Phương thức tắt

### ❑ 6 . Một số phương thức "tắt"

*Để đơn giản hóa việc ghi/ đọc dữ liệu với file ta có thể dùng các p thức "tắt" :*

- ✓*File.WriteAllText*
- ✓*File.ReadAllText*
- ✓*File.WriteAllBytes*
- ✓*File.ReadAllBytes*
- ✓*File.WriteAllLines*
- ✓*File.ReadAllLines*
- ✓*File.OpenRead*
- ✓*File.OpenWrite*
- ✓*File.Create*

*Chú ý : các lệnh đọc tắt này đọc toàn bộ dữ liệu vào bộ nhớ,vốn rất không hiệu quả nếu file lớn.*

*Khi sử dụng các phương thức "tắt" file được mở và đóng tự động.Chúng ta không cần thực hiện các thao tác làm việc với file thông thường*

## 3. Phương thức tắt

### ❑ 7 . File.WriteAllText :

*Ghi toàn bộ string (1 chuỗi )vào file*

*// Nếu file chưa có thì sẽ tạo file mới*

*// File đã tồn tại sẽ xóa hết nội dung cũ, ghi đè ND mới*

```
//Set thư mục làm việc hiện tại
string Dir = @"D:\Tuhoc.cc";
Directory.SetCurrentDirectory(Dir);
DirectoryInfo currDir = new DirectoryInfo(".");
Console.WriteLine(currDir.FullName);

string filemoi = "filetaomoi.txt";
string noidung = @"chào các bạn đến với http://tuhoc.cc
    kênh tự học của những ng chăm học
    Hội chăm  học nhất VN";
File.WriteAllText(filemoi,noidung);
```

### ❑ 8 . File.WriteAllLines

*Ghi  string[] (mảng kiểu string),*

*mỗi phần tử trong mảng sẽ được viết vào 1 dòng trong file*

```
string[] noidung2 = { "dòng 1", "Dòng 2", "dòng 3" };
File.WriteAllLines(filemoi,noidung2);
```

## 3. Phương thức tắt

### ❑ 9 . File.AppendAllLines

*Ghi nối đuôi string[] (mảng kiểu string),*

```
string[] noidung4 = { "dòng 4", "Dòng 5", "dòng 6" };
File.AppendAllLines(filemoi,noidung4);
```

### ❑ 10 . File.ReadAllText

*Sẽ đọc tất cả các dòng trong file và trả về 1 string.*

```
string noiDungDoc = File.ReadAllText(filemoi);
Console.WriteLine(noiDungDoc);
Console.WriteLine(noiDungDoc.GetType().ToString());
```

### ❑ 11 . File.ReadAllLines

*Trả về giá trị kiểu string[] (mảng kiểu string)*

```
string[] arr_Noidung = File.ReadAllLines(filemoi);
Console.WriteLine("Nội dung theo dòng");
int dem = 0;
foreach (string line in arr_Noidung)
{
    Console.WriteLine(line);
}
Console.WriteLine(arr_Noidung[0]); // xuất dòng index 0
```

## 3. Phương thức tắt

### ❑ 12 . File.Move

*Di chuyển và đổi tên nếu tên file đích khác tên gốc*

```
string path1 = "filetaomoi.txt";
string path2 = "filedichuyen.txt";
File.Move(path1, path2);
// di chuyển file filetaomoi.txt đổi thành filedichuyen.txt
```

### ❑ 13 . File.Copy()

*Copy file .*

```
File.Copy("file1.txt", "file1-copy.txt");
Console.WriteLine("đã copy");
```

### ❑ 14 . File.Delete()

```
File.Delete("file1-copy.txt");
Console.WriteLine("Đã xóa");
```