

# LẬP TRÌNH HƯỚNG SỰ KIỆN

Giảng viên: ThS. Phan Thanh Toàn

# **BÀI 5**

## **XỬ LÝ NHẬP XUẤT TRONG C#**

Giảng viên: ThS. Phan Thanh Toàn

## MỤC TIÊU BÀI HỌC

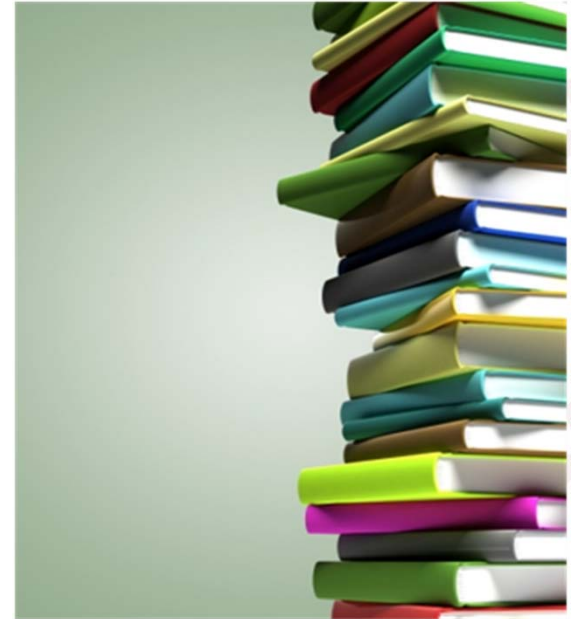
- Phân biệt các loại tệp.
- Liệt kê được các lớp cơ bản xử lý tệp tin và thư mục.
- Liệt kê được các thao tác cơ bản xử lý tệp tin và thư mục.
- Vận dụng được ngôn ngữ C# vào triển khai các ứng trên hệ thống tệp.



## CÁC KIẾN THỨC CẦN CÓ

Để học được môn học này, sinh viên phải học xong các môn học:

- Cơ sở lập trình;
- Lập trình hướng đối tượng;
- Cơ sở dữ liệu;
- Hệ quản trị cơ sở dữ liệu SQL Server.





## HƯỚNG DẪN HỌC

- Đọc tài liệu tham khảo.
- Thảo luận với giáo viên và các sinh viên khác về những vấn đề chưa hiểu rõ.
- Trả lời các câu hỏi của bài học.



## CẤU TRÚC NỘI DUNG

**5.1**

Khái niệm về các loại tệp tin

**5.2**

Xử lí hệ thống tệp tin và thư mục qua các lớp

**5.3**

Các thao tác đọc, ghi tệp

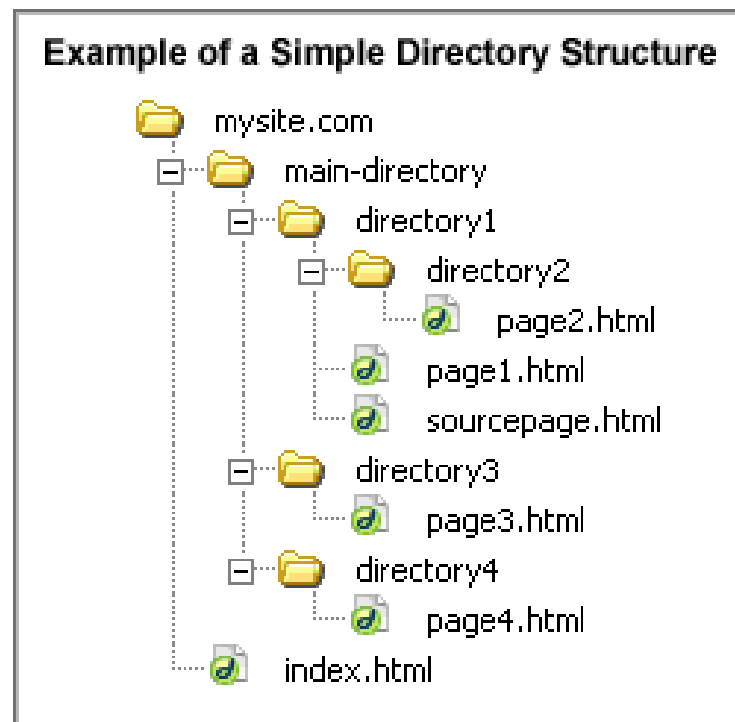
## 5.1. KHÁI NIỆM VỀ CÁC LOẠI TỆP TIN

5.1.1. Khái niệm tệp tin và thư mục

5.1.2. Phân loại tệp tin

### 5.1.1. KHÁI NIỆM TỆP TIN THƯ MỤC

- Tập tin là tập dữ liệu được lưu trữ trong bộ nhớ ngoài.
- Các tập tin được quản lí qua tên tệp.
- Dữ liệu trong tệp là dữ liệu bền vững.
- Dữ liệu trong tệp tin có kích thước không giới hạn (chỉ phụ thuộc vào hệ điều hành).
- Thư mục dùng để lưu trữ các tệp tin, thư mục thường được lưu trữ theo cấu trúc cây thư mục.





### 5.1.2. PHÂN LOẠI TỆP TIN

- Có 3 loại tệp tin:
  - Tệp văn bản (Text File);
  - Tệp định kiểu (Typed File);
  - Tệp không định kiểu (Untyped File).
- Tệp văn bản: là tệp sử dụng lưu trữ dữ liệu dạng kí tự, các dữ liệu được lưu trữ dưới dạng mã ASCII.
- Các dữ liệu được lưu trữ theo cấu trúc từng dòng, mỗi dòng được kết thúc bởi kí tự xuống dòng.
- Mỗi tệp được kết thúc bởi kí tự EOF (End Of File), kí tự EOF mã ASCII là 26.
- Tệp văn bản truy xuất theo kiểu tuần tự.



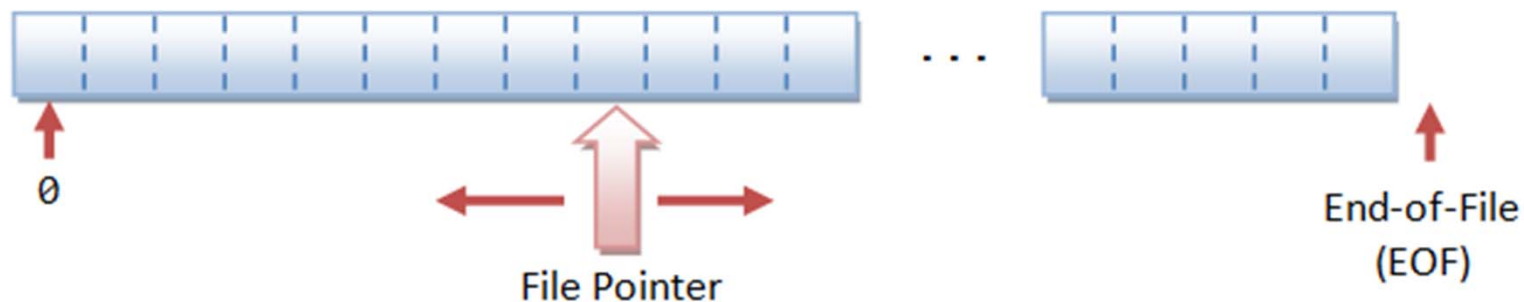
### 5.1.2. PHÂN LOẠI TỆP TIN (tiếp theo)

- Tập định kiểu: là tệp tin bao gồm nhiều phần tử, các phần tử đều có cùng kiểu dữ liệu như int, float, double hoặc các kiểu dữ liệu do người dùng tự định nghĩa.
- Dữ liệu trong tệp định kiểu được lưu trữ dưới dạng các byte liên tiếp.
- Tệp không định kiểu: Lưu trữ dữ liệu và người dùng không quan tâm đến cấu trúc của dữ liệu mà chỉ quan tâm đến độ lớn dữ liệu được lưu trữ.



### 5.1.2. PHÂN LOẠI TẬP TIN (tiếp theo)

- Con trỏ tệp tin: Khi một tệp tin được mở để làm việc sẽ có một con trỏ tệp tin hoạt động và dịch chuyển qua các phần tử của tệp, phần tử được xử lý chính là phần tử được trỏ bởi con trỏ tệp.
- Tại một thời điểm con trỏ tệp trỏ đến một phần tử duy nhất trong tệp.
- Khi mở tệp con trỏ tệp luôn trỏ đến phần tử đầu tiên trong tệp.
- Khi đọc/ghi xong một phần tử con trỏ tệp sẽ tự dịch chuyển sang phần tử kế tiếp về phía cuối tệp.
- Phần tử nhận biết dấu hiệu kết thúc tệp là EOF.
- Truy cập dữ liệu trong tệp thông qua biến tệp.



## 5.2. XỬ LÝ HỆ THỐNG TỆP TIN VÀ THƯ MỤC QUA CÁC LỚP

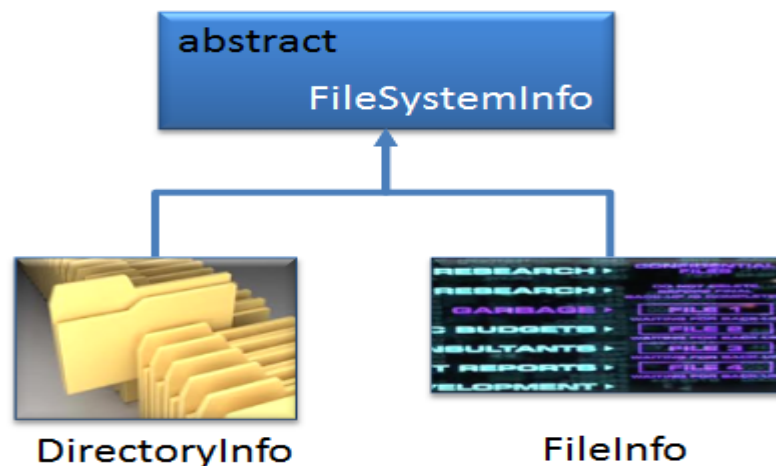
5.2.1. Các lớp đối tượng xử lý tệp tin và thư mục

5.2.2. Một số ví dụ

### 5.2.1. CÁC LỚP ĐỐI TƯỢNG XỬ LÝ TỆP TIN VÀ THƯ MỤC

- Namespace System.IO chứa một tập hợp các lớp hỗ trợ xử lý tệp tin và thư mục.
- Các class được chia thành 2 loại: information (thông tin) và utility (tiện ích).
- Các class loại thông tin (information) đều dẫn xuất từ class cơ sở FileSystemInfo, các class này cung cấp các phương thức và thuộc tính liên quan đến thông tin tệp tin và thư mục.
- Có 3 class quan trọng: FileInfo, DirectoryInfo và DriveInfo.
- Các class thuộc loại utility cung cấp các phương thức thực thi trên các đối tượng tệp tin và thư mục.

System.IO namespace



### 5.2.1. CÁC LỚP ĐỐI TƯỢNG XỬ LÝ TỆP TIN VÀ THƯ MỤC (tiếp theo)

- Lớp FileSystemInfo:
  - Cung cấp các lớp cơ bản FileInfo và DirectoryInfo, đây là 2 lớp cơ bản cung cấp các thông tin về tệp tin và thư mục.
  - Các thuộc tính:

Thuộc tính	Ý nghĩa
Attributes	Gán hoặc lấy các thuộc tính của tệp tin hoặc thư mục.
CreationTime	Gán hoặc lấy giá trị thời gian tạo lập tệp tin hoặc thư mục.
Exists	Kiểm tra tệp tin hay thư mục có tồn tại không.
Extension	Trả về thông tin về phần mở rộng của tệp tin.
LastAccessTime	Gán hoặc lấy giá trị về thời gian truy cập cuối cùng trên tệp tin hoặc thư mục.
LastWriteTime	Gán hoặc lấy giá trị về thời gian ghi cuối cùng trên tệp tin hoặc thư mục.
Name	Trả về tên tệp tin hoặc thư mục.
FullName	Trả về đường dẫn đầy đủ của tệp tin hoặc thư mục.



### 5.2.1. CÁC LỚP ĐỐI TƯỢNG XỬ LÝ TỆP TIN VÀ THƯ MỤC (tiếp theo)

- Các phương thức của lớp FileSystemInfo:

Phương thức	Ý nghĩa
Delete	Xóa tệp tin hoặc thư mục.
Refresh	Cập nhật lại dữ liệu với thông tin mới nhất từ hệ thống tệp.

### 5.2.1. CÁC LỚP ĐỐI TƯỢNG XỬ LÝ TỆP TIN VÀ THƯ MỤC (tiếp theo)

- Lớp FileInfo:
  - Lớp FileInfo được sử dụng để truy cập đến một tệp tin trong hệ thống tệp.
  - Các thuộc tính của lớp FileInfo.

Thuộc tính	Ý nghĩa
Attributes	Gán hoặc lấy các thuộc tính của tệp tin hoặc thư mục.
CreationTime	Gán hoặc lấy giá trị thời gian tạo lập tệp tin hoặc thư mục.
Exists	Kiểm tra tệp tin hay thư mục có tồn tại không.
Extension	Trả về thông tin về phần mở rộng của tệp tin.
LastAccessTime	Gán hoặc lấy giá trị về thời gian truy cập cuối cùng trên tệp tin hoặc thư mục.
LastWriteTime	Gán hoặc lấy giá trị về thời gian ghi cuối cùng trên tệp tin hoặc thư mục.
Name	Trả về tên tệp tin hoặc thư mục.
FullName	Trả về đường dẫn đầy đủ của tệp tin hoặc thư mục.
IsReadOnly	Gán hoặc lấy thông tin về tệp tin chỉ đọc.
Length	Kích thước tệp tin.

### 5.2.1. CÁC LỚP ĐỐI TƯỢNG XỬ LÝ TỆP TIN VÀ THƯ MỤC (tiếp theo)

- Các phương thức của lớp FileInfo

Phương thức	Ý nghĩa
AppendText	Tạo đối tượng StreamWriter cho phép ghi thêm dữ liệu vào tệp tin.
CopyTo	Sao chép tệp tin tới thư mục khác.
Create	Tạo tệp tin mới.
CreateText	Tạo đối tượng StreamWriter cho phép ghi dữ liệu vào tệp tin.
Decrypt	Giải mã tệp tin đã được mã hóa.
Encrypt	Mã hóa tệp tin.
MoveTo	Di chuyển tệp tin tới thư mục khác.
Open	Mở tệp tin để đọc hoặc ghi.
OpenRead	Mở tệp tin ở chế độ chỉ đọc.
OpenWrite	Mở tệp tin ở chế độ ghi.

### 5.2.1. CÁC LỚP ĐỐI TƯỢNG XỬ LÝ TỆP TIN VÀ THƯ MỤC (tiếp theo)

- Ví dụ: Kiểm tra sự tồn tại của tệp tin và hiển thị thông tin về tên tệp, đường dẫn và kích thước tệp qua đối tượng FileInfo

```
FileInfo file = new FileInfo(@"C:\DATA\ProgWinFormC#.pdf");  
if (file.Exists)  
{  
    Console.WriteLine("Name of the file: " + file.Name);  
    Console.WriteLine("Full Name of the file: " + file.FullName);  
    Console.WriteLine("Size of the file: " +  
        file.Length.ToString());  
}
```

### 5.2.1. CÁC LỚP ĐỐI TƯỢNG XỬ LÝ TỆP TIN VÀ THƯ MỤC (tiếp theo)

- Lớp DirectoryInfo:
  - Lớp DirectoryInfo cung cấp các thuộc tính và phương thức về đối tượng thư mục trong hệ thống tệp.
  - Các thuộc tính lớp DirectoryInfo:

Thuộc tính	Ý nghĩa
Attributes	Gán hoặc lấy các thuộc tính của tệp tin hoặc thư mục.
CreationTime	Gán hoặc lấy giá trị thời gian tạo lập tệp tin hoặc thư mục.
Exists	Kiểm tra tệp tin hay thư mục có tồn tại không.
Extension	Trả về thông tin về phần mở rộng của tệp tin.
LastAccessTime	Gán hoặc lấy giá trị về thời gian truy cập cuối cùng trên tệp tin hoặc thư mục.
LastWriteTime	Gán hoặc lấy giá trị về thời gian ghi cuối cùng trên tệp tin hoặc thư mục.
Name	Trả về tên tệp tin hoặc thư mục.
FullName	Trả về đường dẫn đầy đủ của tệp tin hoặc thư mục.
Parent	Trả về thư mục cha của thư mục hiện tại.
Root	Trả về thư mục gốc.

### 5.2.1. CÁC LỚP ĐỐI TƯỢNG XỬ LÝ TỆP TIN VÀ THƯ MỤC (tiếp theo)

- Các phương thức lớp DirectoryInfo:

Phương thức	Ý nghĩa
Create()	Tạo thư mục.
CreateSubdirectory()	Tạo thư mục con.
GetDirectories()	Trả về mảng đối tượng kiểu DirectoryInfo gồm các thư mục con trong thư mục hiện thời.
GetFiles()	Trả về mảng đối tượng kiểu FileInfo gồm các tệp tin trong thư mục hiện hành.
GetFileSystemInfos()	Trả về mảng đối tượng kiểu FileSystemInfo gồm các tệp tin và thư mục trong thư mục hiện hành.
MoveTo()	Di chuyển thư mục hiện hành sang vị trí khác.



### 5.2.1. CÁC LỚP ĐỐI TƯỢNG XỬ LÝ TỆP TIN VÀ THƯ MỤC (tiếp theo)

Ví dụ: Duyệt thư mục và lấy thông tin về các tệp tin trong thư mục

```
DirectoryInfo dir = new DirectoryInfo(@"C:\DATA");  
FileInfo[] file = dir.GetFiles();  
foreach(FileInfo f in file)  
{  
    Console.WriteLine("File Name: " + f.Name);  
    Console.WriteLine("Created Time: " +  
        f.CreationTime.ToShortTimeString());  
}
```

### 5.2.1. CÁC LỚP ĐỐI TƯỢNG XỬ LÝ TỆP TIN VÀ THƯ MỤC (tiếp theo)

- Lớp DriveInfo:
  - Cung cấp các thuộc tính và phương thức cơ bản để truy xuất vào đối tượng ổ đĩa trong hệ thống.
  - Các thuộc tính của lớp DriveInfo:

Thuộc tính	Ý nghĩa
AvailableFreeSpace	Trả về kích thước còn trống của ổ đĩa.
DriveFormat	Trả về thông tin kiểu bảng FAT (FAT32, NTFS).
DriveType	Trả về kiểu ổ đĩa (ổ đĩa cứng, CDROM...).
IsReady	Trả về trạng thái ổ đĩa (có sẵn sàng cho truy xuất không).
Name	Trả về tên ổ đĩa.
TotalFreeSpace	Trả về tổng kích thước còn trống của ổ đĩa.
TotalSize	Trả về tổng kích thước của ổ đĩa.
VolumeLabel	Trả về nhãn ổ đĩa.

### 5.2.1. CÁC LỚP ĐỐI TƯỢNG XỬ LÝ TỆP TIN VÀ THƯ MỤC (tiếp theo)

- Các phương thức của lớp DriveInfo:

Phương thức	Ý nghĩa
GetDrives	Trả về tất cả các ổ đĩa hiện có trên hệ thống.

- Các thành phần kiểu liệt kê của Drive Type:

Thành phần	Ý nghĩa
CDROM	Ổ đĩa CDROM.
Fixed	Ổ đĩa cố định.
NetWork	Ổ đĩa mạng.
NoRootDirectory	Ổ đĩa không có thư mục gốc.
Ram	Ổ đĩa RAM (được tạo từ RAM).
Removeable	Ổ đĩa di động (USB).
UnKnown	Ổ đĩa không xác định.

### 5.2.2. MỘT SỐ VÍ DỤ

- Ví dụ 1: Chương trình hiển thị các thông tin cơ bản của tất cả các ổ đĩa trên hệ thống (tên ổ đĩa, tổng kích thước theo GB, loại bảng FAT)

```
DriveInfo[] drives = DriveInfo.GetDrives();
foreach (DriveInfo drive in drives)
{
    if (drive.IsReady == true)
    {
        Console.WriteLine("Name: " + drive.Name);
        Console.WriteLine("Size: " + drive.TotalSize/(1024*1024*1024)
            + "GB");
        Console.WriteLine("Type: " + drive.DriveType.ToString());
        Console.WriteLine("Fat type: " +
            drive.DriveFormat.ToString());
    }
}
```

### 5.2.2. MỘT SỐ VÍ DỤ (tiếp theo)

- Ví dụ 2: Chương trình sao chép tất cả các tệp tin trong thư mục DATA sang thư mục DATA1. Chương trình sẽ kiểm tra nếu chưa tồn tại thư mục DATA1 sẽ tạo ra thư mục DATA1

```
DirectoryInfo source = new DirectoryInfo(@"C:\DATA");  
DirectoryInfo destination = new DirectoryInfo(@"C:\DATA1");  
if (!destination.Exists)  
    destination.Create();  
FileInfo[] files = source.GetFiles();  
foreach (FileInfo f in files)  
{  
    f.CopyTo(destination.FullName.ToString() + "\\\" + f.Name);  
}
```

## 5.3. CÁC THAO TÁC ĐỌC GHI TỆP

5.3.1. Các lớp đối tượng  
đọc ghi tệp tin trong C#

5.3.2. Một số ví dụ



### 5.3.1. CÁC LỚP ĐỐI TƯỢNG XỬ LÝ ĐỌC GHI TẬP TIN

- Lớp File:
  - Cung cấp các thuộc tính, phương thức cho phép đọc, ghi tập tin;
  - Các phương thức của lớp File:

Phương thức	Ý nghĩa
AppendAllText	Thêm một xâu kí tự vào tập tin, nếu tập tin chưa tồn tại sẽ tạo tập tin mới.
AppendText	Mở tập tin và trả về đối tượng StreamWriter, sử dụng để ghi dữ liệu vào tập tin.
Copy	Sao chép tập tin thành một tập tin mới, tập mới phải chưa tồn tại trên đĩa.
Create	Tạo một tập tin và trả về đối tượng FileStream.
CreateText	Tạo hoặc mở một tập tin và trả về đối tượng StreamWriter, dùng để ghi dữ liệu vào tập tin.
Move	Di chuyển tập tin đến vị trí khác.
OpenRead	Mở một tập tin đã có và trả về đối tượng FileStream ở chế độ chỉ đọc.

### 5.3.1. CÁC LỚP ĐỐI TƯỢNG XỬ LÝ ĐỌC GHI TỆP TIN

- Các phương thức của lớp File (tiếp theo):

Phương thức	Ý nghĩa
OpenText	Mở một tệp tin đã có và trả về đối tượng StreamReader, được sử dụng để đọc dữ liệu từ tệp.
OpenWrite	Mở một tệp tin đã có và trả về đối tượng StreamWriter, được sử dụng để ghi dữ liệu vào tệp tin.
ReadAllBytes	Mở một tệp tin và đọc dữ liệu vào một mảng kiểu byte.
ReadAllLines	Mở một tệp tin và đọc dữ liệu vào một mảng kiểu chuỗi ký tự.
ReadAllText	Mở một tệp tin và đọc toàn bộ nội dung tệp vào một biến chuỗi.
WriteAllBytes	Mở một tệp tin và ghi một mảng kiểu byte vào tệp (ghi đè nội dung đã có).

### 5.3.1. CÁC LỚP ĐỐI TƯỢNG XỬ LÝ ĐỌC GHI TỆP TIN

- Các phương thức của lớp File (tiếp theo):

Phương thức	Ý nghĩa
WriteAllLines	Mở một tệp tin và ghi toàn bộ mảng kiểu xâu vào tệp (ghi đè nội dung đã có).
WriteAllText	Mở tệp và ghi toàn bộ nội dung một xâu kí tự vào tệp (ghi đè nội dung đã có).
Copy	Sao chép tệp tin thành tệp tin mới.
Create	Tạo tệp tin mới và trả về đối tượng FileStream.
CreateText	Tạo mới hoặc mở một tệp tin đã có và trả về đối tượng StreamWriter.

### 5.3.1. CÁC LỚP ĐỐI TƯỢNG XỬ LÝ ĐỌC GHI TỆP TIN (tiếp theo)

- Các phương thức của lớp File (tiếp theo):

Phương thức	Ý nghĩa
Move	Di chuyển tệp tin đến vị trí khác
Open	Mở một tệp tin đã tồn tại và trả về giá trị FileStream
OpenRead	Mở một tệp tin đã tồn tại và trả về giá trị FileStream ở chế độ chỉ đọc
OpenText	Mở một tệp tin đã có và trả về giá trị StreamReader
OpenWrite	Mở một tệp tin đã có và trả về giá trị StreamWriter

### 5.3.1. CÁC LỚP ĐỐI TƯỢNG XỬ LÝ ĐỌC GHI TỆP TIN (tiếp theo)

- Các phương thức của lớp File (tiếp theo):

Phương thức	Ý nghĩa
ReadAllBytes	Mở tệp tin và đọc nội dung vào mảng kiểu byte.
ReadAllLines	Mở tệp tin và đọc nội dung vào một mảng kiểu chuỗi.
ReadAllText	Mở một tệp tin và đọc toàn bộ nội dung tệp vào một biến chuỗi.
WriteAllBytes	Mở một tệp tin và ghi nội dung một mảng kiểu byte vào tệp.
WriteAllLines	Mở một tệp tin và ghi nội dung một mảng kiểu chuỗi ký tự vào tệp.
WriteAllText	Mở một tệp tin và ghi nội một chuỗi ký tự vào tệp.

### 5.3.1. CÁC LỚP ĐỐI TƯỢNG XỬ LÝ ĐỌC GHI TỆP TIN (tiếp theo)

- Lớp FileStream:
  - Cung cấp các thuộc tính và phương thức cơ bản đọc, ghi nội dung tệp tin.
  - Các thuộc tính lớp FileStream:

Thuộc tính	Ý nghĩa
CanRead	Kiểm tra xem có đọc được dữ liệu từ tệp tin hay không?
CanWrite	Kiểm tra xem có ghi được dữ liệu vào tệp tin không?
Length	Độ dài của tệp
Position	Vị trí của con trỏ trong stream



### 5.3.1. CÁC LỚP ĐỐI TƯỢNG XỬ LÝ ĐỌC GHI TỆP TIN (tiếp theo)

➤ Các phương thức của FileStream (tiếp theo):

Phương thức	Ý nghĩa
Read	Đọc một khối byte từ tệp
ReadByte	Đọc một byte từ tệp và dịch chuyển con trỏ sang vị trí tiếp theo
Write	Ghi một khối byte lên tệp
WriteByte	Ghi một byte lên tệp
Flush	Ghi toàn bộ dữ liệu từ bộ đệm lên tệp
Seek	Di chuyển con trỏ đến vị trí yêu cầu
Close	Đóng tệp

### 5.3.1. CÁC LỚP ĐỐI TƯỢNG XỬ LÝ ĐỌC GHI TẬP TIN (tiếp theo)

- Lớp StreamReader:
  - Cung cấp các thuộc tính, phương thức cơ bản để đọc dữ liệu nhận được từ lớp Stream.
  - Các thuộc tính:

Thuộc tính	Ý nghĩa
BaseStream	Trả về đối tượng Stream đang đọc dữ liệu
CurrentEncoding	Trả về mã hóa hiện tại đang sử dụng trong lớp Stream
EndOfStream	Kiểm tra xem đã đọc hết Stream hay chưa

### 5.3.1. CÁC LỚP ĐỐI TƯỢNG XỬ LÝ ĐỌC GHI TẬP TIN (tiếp theo)

➤ Các phương thức:

Phương thức	Ý nghĩa
Close	Đóng Stream đang đọc dữ liệu
Peek	Trả về kí tự kế tiếp trong stream và không di chuyển con trỏ tệp
Read	Đọc một nhóm kí tự kế tiếp trong stream
ReadBlock	Đọc một khối kí tự kế tiếp trong stream
ReadLine	Đọc một dòng trong stream
ReadToEnd	Đọc tất cả dữ liệu trong stream

### 5.3.1. CÁC LỚP ĐỐI TƯỢNG XỬ LÝ ĐỌC GHI TẬP TIN (tiếp theo)

Ví dụ: Chương trình sử dụng lớp StreamReader để đọc và hiển thị nội dung tệp dữ liệu theo từng dòng

```
StreamReader reader = File.OpenText(@"C:\DATA\text.txt");  
while (!reader.EndOfStream)  
{  
    string line = reader.ReadLine();  
    Console.WriteLine(line);  
}
```

### 5.3.1. CÁC LỚP ĐỐI TƯỢNG XỬ LÝ ĐỌC GHI TẬP TIN (tiếp theo)

- Lớp StreamWriter:

- Cung cấp các thuộc tính và phương thức cơ bản để ghi dữ liệu vào tệp.
- Các thuộc tính:

Thuộc tính	Ý nghĩa
BaseStream	Trả về đối tượng Stream đang đọc dữ liệu
Encoding	Trả về mã hóa hiện tại đang sử dụng trong lớp Stream
NewLine	Lấy và thiết lập kí tự xuống dòng

- Các phương thức:

Phương thức	Ý nghĩa
Close	Đóng tệp
Write	Ghi dữ liệu vào stream
WriteLine	Ghi dữ liệu vào tệp và xuống dòng mới

### 5.3.1. CÁC LỚP ĐỐI TƯỢNG XỬ LÝ ĐỌC GHI TỆP TIN (tiếp theo)

**Ví dụ:** Chương trình sẽ tạo tệp songuyen.txt và thực hiện ghi vào tệp 10 số nguyên từ 1 đến 10, mỗi số trên một dòng

```
StreamWriter writer = File.CreateText(@"C:\DATA\SONGUYEN.txt");  
for (int i = 1; i <= 10; i++)  
writer.WriteLine(i.ToString());  
writer.Close();
```

### 5.3.2. VÍ DỤ

Chương trình tìm tất cả các số nguyên tố trong đoạn 1 – n và lưu kết quả vào tệp Prime.txt

Thuật toán:

Input:

Số nguyên n

Output: các số nguyên tố trong đoạn [1, n]

Algorithm:

Xây dựng hàm isPrime(x): kiểm tra số nguyên x có phải số nguyên tố hay không.

Duyệt các số từ 1 đến n, áp dụng hàm isPrime cho tất cả các số.



### 5.3.2. VÍ DỤ (tiếp theo)

- Hàm isPrime(x) trả về true nếu x là số nguyên tố, false nếu x là hợp số.

```
static bool isPrime(int x)
{
    int i;
    if (x <= 3)
        return true;
    else
    {
        i = 2;
        while (x % i != 0) i++;
        if (i < x) return false;
        else return true;
    }
}
```

### 5.3.2. VÍ DỤ (tiếp theo)

- Chương trình sử dụng hàm isPrime để tìm các số nguyên tố và lưu vào tệp Prime.txt

```
int n;  
Console.WriteLine("NHAP SO N:");  
n = Convert.ToInt32(Console.ReadLine());  
string path = @"C:\DATA\Prime.txt";  
StreamWriter writer = File.CreateText(path);  
for (int i = 1; i <= n; i++)  
if (isPrime(i))  
    writer.Write(i.ToString() + ",");  
writer.Close();
```

## TÓM LƯỢC CUỐI BÀI

Trong bài này, chúng ta đã nghiên cứu các nội dung chính sau:

- Khái niệm và phân loại tệp tin;
- Các lớp thao tác với tệp tin và thư mục;
- Các lớp đọc/ghi dữ liệu vào tệp tin.