

C# FOR BASE

GV: Bùi Quang Đăng

Contents

1

File và Directory

2

Thao tác với file, directory

3

Đọc và ghi vào file

4

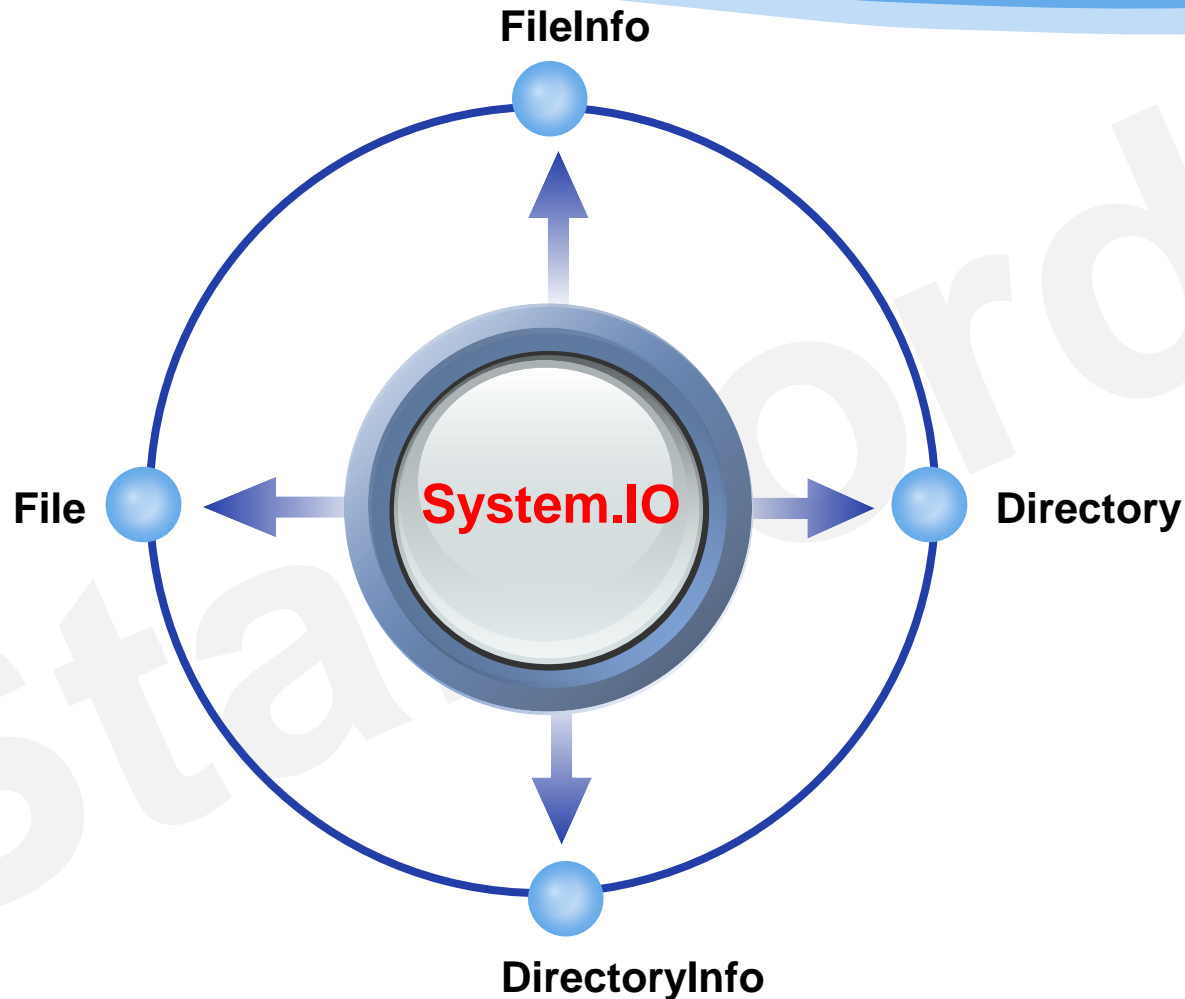
Exercises

FILE & DIRECTORY

File & Directory

- ❖ Sử dụng để thao tác với các tập tin, thư mục trên hệ thống máy tính.
- ❖ Để thao tác với file, directory sử dụng không gian tên **System.IO**.
- ❖ Đọc thông tin từ file, truy xuất thông tin thư mục.
- ❖ Ghi dữ liệu ra file, tạo thư mục mới.

File & Directory



File & Directory

```

using System;
using System.Runtime.InteropServices;
using System.Security.AccessControl;

namespace System.IO
{
    public static class Directory
    {
        public static DirectoryInfo CreateDirectory(string path);
        public static DirectoryInfo CreateDirectory(string path, DirectorySecurity directorySecurity);
        public static void Delete(string path);
        public static void Delete(string path, bool recursive);
        public static bool Exists(string path);
        public static DirectorySecurity GetAccessControl(string path);
        public static DirectorySecurity GetAccessControl(string path, AccessControlSections includeSections);
        public static DateTime GetCreationTime(string path);
        public static DateTime GetCreationTimeUtc(string path);
        public static string GetCurrentDirectory();
        public static string[] GetDirectories(string path);
        public static string[] GetDirectories(string path, string searchPattern);
        public static string[] GetDirectories(string path, string searchPattern, SearchOption searchOption);
        public static string GetDirectoryRoot(string path);
        public static string[] GetFiles(string path);
        public static string[] GetFiles(string path, string searchPattern);
        public static string[] GetFiles(string path, string searchPattern, SearchOption searchOption);
        public static string[] GetFileSystemEntries(string path);
    }
}

```

File & Directory

```

using System;
using System.Runtime.InteropServices;
using System.Security.AccessControl;

namespace System.IO
{
    ...public sealed class DirectoryInfo : FileSystemInfo
    {
        ...public DirectoryInfo(string path);

        ...public override bool Exists { get; }
        ...public override string Name { get; }
        ...public DirectoryInfo Parent { get; }
        ...public DirectoryInfo Root { get; }

        ...public void Create();
        ...public void Create(DirectorySecurity directorySecurity);
        ...public DirectoryInfo CreateSubdirectory(string path);
        ...public DirectoryInfo CreateSubdirectory(string path, DirectorySecurity directorySecurity);
        ...public override void Delete();
        ...public void Delete(bool recursive);
        ...public DirectorySecurity GetAccessControl();
        ...public DirectorySecurity GetAccessControl(AccessControlSections includeSections);
        ...public DirectoryInfo[] GetDirectories();
        ...public DirectoryInfo[] GetDirectories(string searchPattern);
        ...public DirectoryInfo[] GetDirectories(string searchPattern, SearchOption searchOption);
    }
}
    
```

File & Directory

```

using System;
using System.Runtime.InteropServices;
using System.Security.AccessControl;
using System.Text;

namespace System.IO
{
    ...public static class File
    {
        ...public static void AppendAllText(string path, string contents);
        ...public static void AppendAllText(string path, string contents, Encoding encoding);
        ...public static StreamWriter AppendText(string path);
        ...public static void Copy(string sourceFileName, string destFileName);
        ...public static void Copy(string sourceFileName, string destFileName, bool overwrite);
        ...public static FileStream Create(string path);
        ...public static FileStream Create(string path, int bufferSize);
        ...public static FileStream Create(string path, int bufferSize, FileOptions options);
        ...public static FileStream Create(string path, int bufferSize, FileOptions options, FileSecurity fileSecurity);
        ...public static StreamWriter CreateText(string path);
        ...public static void Decrypt(string path);
        ...public static void Delete(string path);
        ...public static void Encrypt(string path);
        ...public static bool Exists(string path);
        ...public static FileSecurity GetAccessControl(string path);
        ...public static FileSecurity GetAccessControl(string path, AccessControlSections includeSections);
        ...public static FileAttributes GetAttributes(string path);
    }
}
    
```


File & Directory

```
using System;
using System.Runtime.InteropServices;
using System.Security.AccessControl;

namespace System.IO
{
    ...public sealed class FileInfo : FileSystemInfo
    {
        ...public FileInfo(string fileName);

        ...public DirectoryInfo Directory { get; }
        ...public string DirectoryName { get; }
        ...public override bool Exists { get; }
        ...public bool IsReadOnly { get; set; }
        ...public long Length { get; }
        ...public override string Name { get; }

        ...public StreamWriter AppendText();
        ...public FileInfo CopyTo(string destFileName);
        ...public FileInfo CopyTo(string destFileName, bool overwrite);
        ...public FileStream Create();
        ...public StreamWriter CreateText();
        ...public void Decrypt();
        ...public override void Delete();
        ...public void Encrypt();
        ...public FileSecurity GetAccessControl();
    }
}
```

File & Directory

- ❖ **FileSystemInfo** – Lớp đối tượng cơ sở biểu diễn các file đối tượng hệ thống.
- ❖ **FileInfo and File** – Các lớp này thể hiện một file trên file hệ thống.
- ❖ **DirectoryInfo and Directory** – Các lớp này thể hiện một folder trên tập tin hệ thống.
- ❖ **Path** – Lớp này chứa các bộ phận tĩnh dùng thao tác với các pathnames.

File & Directory

❖ Các thuộc tính của lớp cơ bản FileSystemInfo

Name	Description
CreationTime	Thời gian file, folder được tạo
DirectoryName (FileInfo), Parent (DirectoryInfo)	Tên đường dẫn của folder chức đựng
Exists	Xác định file ,folder hiện hữu có tồn tại hay không
Extension	Tên mở rộng của file, trả về khoảng trắng nếu là folder
FullName	Tên đường dẫn của file ,folder
LastAccessTime	Thời gian file, folder truy xuất lần cuối
LastWriteTime	Time file or folder was last modified
Name	Name of the file or folder

File & Directory

❖ Các thuộc tính của lớp cơ bản FileSystemInfo

Name	Description
Root	Đường dẫn gốc
Length	Kích thước file tính bằng bytes (chỉ FileInfo)

❖ Các phương thức

Name	Description
Create()	Tạo một folder hoặc một file rỗng
Delete()	Xóa file, folder
MoveTo()	Di chuyển hoặc sửa tên file, folder.
CopyTo()	(FileInfo only) Sao chép file, không sao chép phương thức cho folders.

File & Directory

❖ Các phương thức

Name	Description
GetDirectories()	(DirectoryInfo only) Trả về một mảng các đối tượng của DirectoryInfo đại diện tất cả folders được chứa trong folder này.
GetFiles()	(DirectoryInfo only) Trả về một mảng các đối tượng của FileInfo đại diện tất cả folders được chứa trong folder này
GetFileSystemObjects()	(DirectoryInfo only) Trả về đối tượng FileInfo và DirectoryInfo như mảng của tham khảoFileSystemInfo.

File & Directory

❖ Tạo một đối tượng DirectoryInfo

- Bắt đầu làm việc với lớp DirectoryInfo bằng cách khai báo một đường dẫn cụ thể ví dụ : "C:\", "D:\WINNT", ...
nếu muốn truy cập thư mục của ứng dụng đang thi hành dùng ký hiệu "."

Ví dụ:

```
//Tạo một thư mục mới từ thư mục hiện hành trở đi  
DirectoryInfo dir1 = new DirectoryInfo(".");  
//Tạo một thư mục mới từ thư mục C:\Foo\Bar trở đi  
DirectoryInfo dir2 = new DirectoryInfo(@"C:\Foo\Bar");
```

File & Directory

❖ Lớp Path

- Lớp Path gồm các phương thức tĩnh để thực hiện các phép toán trên tên đường dẫn dễ dàng hơn.

Ví dụ: muốn hiển thị tên đường dẫn cho một file ReadMe.txt trong folder C:\My Documents. Bạn có thể tìm đường dẫn đến file như sau:

```
Console.WriteLine(Path.Combine(@"C:\My Documents", "ReadMe.txt"));
```

- Sử dụng lớp Path dễ dàng hơn nhiều so với khi bạn thực hiện các ký hiệu bằng tay nhất là bởi vì lớp Path nhận biết được các định dạng khác nhau của đường dẫn trên các hệ điều hành khác nhau.

Thao tác với File & Directory

Thao tác với File & Directory

❖ Tạo file, thư mục (Create)

- Tạo thư mục sử dụng phương thức **Directory.CreateDirectory**("Tên thư mục"); hoặc
DirectoryInfo dr = new DirectoryInfo("Tên thư mục");
if (!dr.Exists)
{
 dr.Create();
}

Thao tác với File & Directory

❖ Tạo file, thư mục (Create)

Ví dụ:

```
Directory.CreateDirectory("C:\\Stanford");
```

```
DirectoryInfo dr = new DirectoryInfo("C:\\Stanford");
```

```
if (!dr.Exists)
{
    dr.Create();
}
```

Thao tác với File & Directory

❖ Tạo file, thư mục (Create)

- Tạo file sử dụng phương thức **File.Create**("Tên file");
hoặc

```
FileInfo fi = new FileInfo("Tên file");
```

```
if (fi.Exists)
```

```
{
```

```
    fi.Create();
```

```
}
```

Thao tác với File & Directory

❖ Tạo file, thư mục (Create)

Ví dụ:

```
File.Create("C:\\Stanford_CSharp.txt");
```

```
FileInfo fi = new FileInfo("C:\\Stanford_CSharp.txt");  
if (fi.Exists)  
{  
    fi.Create();  
}
```

Thao tác với File & Directory

❖ Sao chép (Copy)

- Sử dụng phương thức **File.Copy**("Đường dẫn nguồn", "Đường dẫn đích", overwrite); hoặc
FileInfo fi = new FileInfo("Đường dẫn file");
fi.CopyTo("Đường dẫn file đích", overwrite);

❖ Di chuyển (Move)

- Sử dụng phương thức **File.Move**("Đường dẫn file nguồn", "Đường dẫn file đích"); hoặc
FileInfo fi = new FileInfo("Đường dẫn file");
fi.MoveTo("Đường dẫn file đích");

Thao tác với File & Directory

❖ Xóa (Delete)

- Sử dụng phương thức **Directory.Delete**("Tên thư mục"); hoặc

`DirectoryInfo dr = new DirectoryInfo("Tên thư mục");`

```
if (dr.Exists)
{
    dr.Delete();
}
```

Thao tác với File & Directory

❖ Xóa (Delete)

- Sử dụng phương thức **File.Delete**("Tên file"); hoặc
`FileInfo fi = new FileInfo("Tên file");`

```
if (fi.Exists)
{
    fi.Delete();
}
```

ĐỌC VÀ GHI VÀO FILE

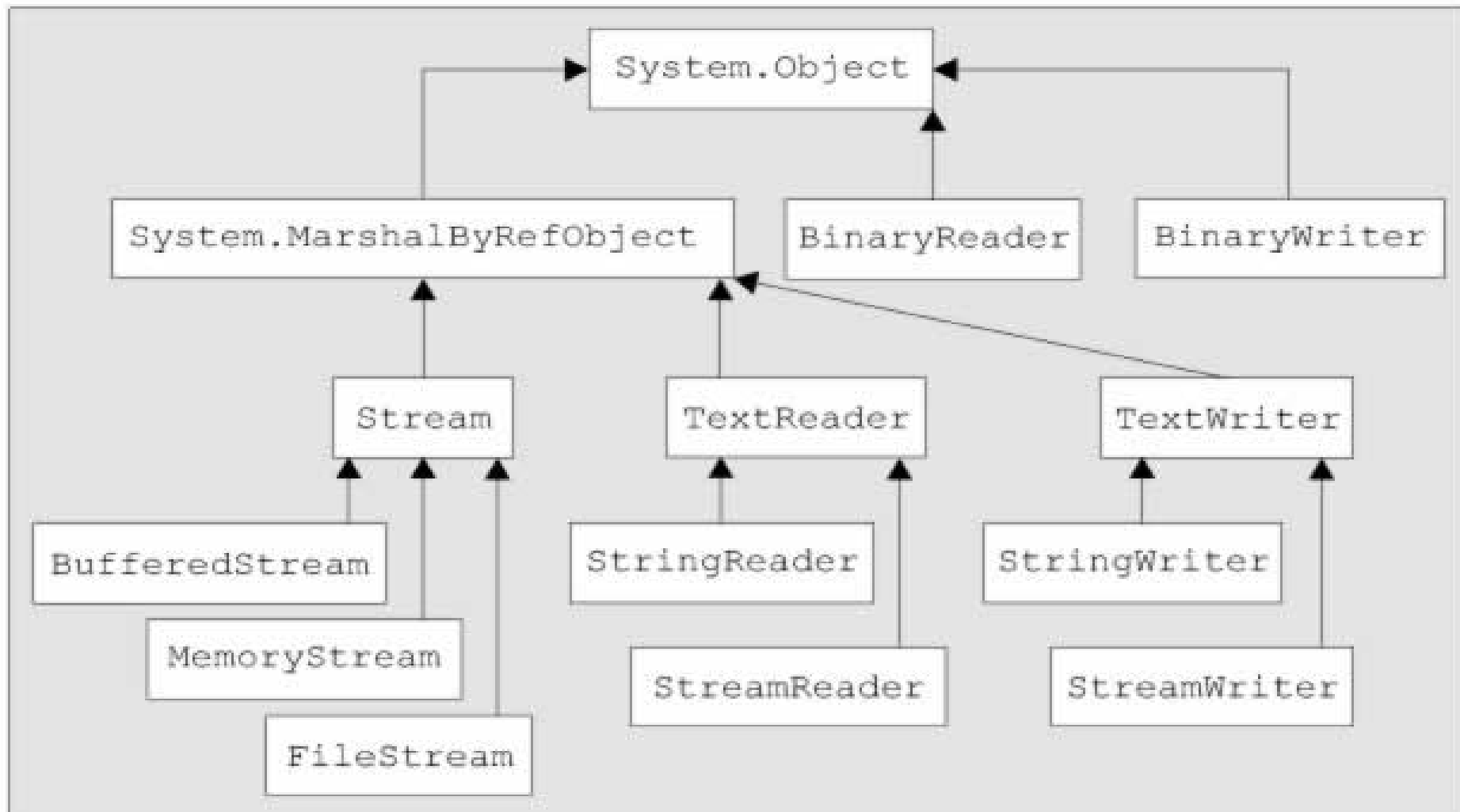
Streams

- ❖ Đọc và ghi dữ liệu sẽ được thực hiện qua lớp Stream.
- ❖ Đây là một thực thể có khả năng nhận được hoặc tạo ra một “nhóm” dữ liệu.
- ❖ Sử dụng namespace: **System.IO.Stream**.

Streams

- ❖ Một stream là một đối tượng dùng để chuyển dữ liệu. Dữ liệu có thể được truyền theo hai hướng:
 - Nếu dữ liệu được truyền từ nguồn bên ngoài vào trong chương trình của bạn, ta gọi là đọc dữ liệu.
 - Nếu dữ liệu được truyền từ chương trình của bạn ra nguồn bên ngoài, ta gọi là viết dữ liệu.

Streams



Streams

❖ FileStream

- Lớp FileStream đem lại việc thực thi cho những thành viên của lớp abstract Stream theo một thể thức thích hợp đối với các file-base streaming giống như các lớp **DirectoryInfo** và **FileInfo**.
- Lớp FileStream cho phép mở những tập tin hiện hữu cũng như tạo mới file. Khi tạo tập tin, lớp FileStream thường dùng những enum FileMode, FileAccess và FileShare

Streams

❖ FileStream

- FileStream được sử dụng đọc và viết dữ liệu vào hoặc từ một file. Để khởi tạo một FileStream, bạn cần 4 phần sau:
 - **file** bạn muốn truy xuất .
 - **mode**, cho biết bạn muốn mở file như thế nào.
 - **access**, cho biết bạn muốn truy xuất file như thế nào – bạn định đọc hoặc viết file hoặc cả hai.
 - **share access** – khả năng truy xuất file.

Streams

❖ FileStream

Ví dụ:

```
// Tạo một tập tin mới trên thư mục làm việc  
FileStream myFStream = new  
FileStream("Stanford_CSharp.txt", FileMode.Op  
enOrCreate, FileAccess.ReadWrite);
```

Streams

❖ FileStream

Enumeration	Values
FileMode	Append, Create, CreateNew, Open, OpenOrCreate, or Truncate
FileAccess	Read, ReadWrite, or Write
FileShare	Inheritable, None, Read, ReadWrite, or Write

Streams

❖ FileStream

Ví dụ: Đọc file dữ liệu có tên cs081201cb.txt

```
FileStream strm = new FileStream("C:\\cs081201cb.txt", FileMode.Open, FileAccess.Read);
StreamReader rdr = new StreamReader(strm);

int i = -1;
while (rdr.Peek() >= 0)
{
    i++;
    string str = rdr.ReadLine();
    if (str == null || str == "")
    {
        return;
    }
}
rdr.Close();
strm.Close();
```

I

Streams

❖ FileStream

Ví dụ: Ghi file dữ liệu có tên cs081201cb.txt

```
string filename = "C:\\cs081201cb.txt";
FileStream fs;
FileInfo fi = new FileInfo(filename);
if (fi.Exists)
{
    fs = new FileStream(filename, FileMode.Truncate);
}
else
{
    fs = new FileStream(filename, FileMode.OpenOrCreate);
}
StreamWriter w = new StreamWriter(fs);

w.WriteLine("Xin chào các bạn học viên khóa cs081201cb");
w.WriteLine("Chúc các bạn học tập tốt !");
w.WriteLine("Stanford - đào tạo học việc lập trình");
w.Flush();
w.Close();
fs.Close();
```

File Text

❖ Làm việc với những tập tin văn bản

- Đọc ghi dữ liệu với file có thể sử dụng StreamReader và StreamWriter.
- Sử dụng các phương thức được hỗ trợ trong lớp File và FileInfo.

Ví dụ:

```
string filename = "C:\\Stanford_CSharp.txt";  
File.WriteAllText(filename, "Xin chào lớp C# for base");
```



Thank You !