

Chương 8

Files and streams

Contents

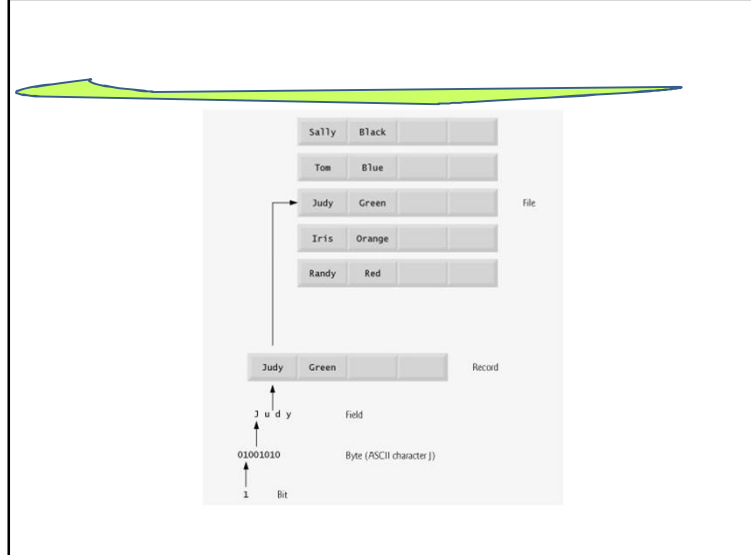
- Introduction
- Data Hierarchy
- Files and Streams
- Classes File and Directory
- Run file

Introduction

- Các biến và mảng chỉ lưu trữ dữ liệu tạm thời
- Các file (database) sẽ lưu trữ dữ liệu lâu dài
- Duy trì dữ liệu trong các tập tin thường được gọi là dữ liệu bền vững
- Máy vi tính lưu trữ các tập tin trên các thiết bị lưu trữ thứ cấp, như đĩa từ, đĩa quang và băng từ

Data Hierarchy

- Dữ liệu :
 - Được biểu diễn dưới dạng bit (0 và 1)
 - Ở dạng: decimal digits; letters; special symbols
 - Bảng mã ASCII
 - Character lưu bởi 1 byte, mỗi byte là 8 bit
- C# sử dụng :
 - Bảng mã Unicode® character set
 - Character lưu bởi 2 byte, mỗi byte là 8 bit



Files and Streams

- C # xem mỗi **tập tin** như là một **dòng (luồng)** tuần tự của **byte**
- Mỗi tập tin phải có một dấu kết thúc
- Các lớp **File** và **Directory** thuộc vùng **System.IO**
- Khi một chương trình thực hiện, môi trường thời gian chạy tạo ra ba đối tượng luồng :
 - **Console.Out**
 - **Console.In**
 - **Console.Error**

Classes File and Directory

Phương thức	Giải thích	Phương thức / property	Giải thích
AppendText()	Tạo một StreamWriter cho phép thêm văn bản vào tập tin	Attributes()	Thừa kế từ FileSystemInfo. Lấy/thiết đặt thuộc tính tập tin
Copy()	Sao chép một tập tin từ tập tin đã có	CreationTime	Thừa kế từ FileSystemInfo. Lấy/thiết đặt thời gian tạo tập tin
Create()	Tạo một tập tin mới	Directory	Lấy thư mục cha
CreateText()	Tạo một StreamWriter cho phép viết mới văn bản vào tập tin	Exists	Xác định tập tin có tồn tại chưa?
Delete()	Xóa một tập tin	Extension	Thừa kế từ FileSystemInfo. Phần mở rộng của tập tin
Exists()	Trả về đúng nếu tập tin tồn tại	FullName	Thừa kế từ FileSystemInfo. Đường dẫn đầy đủ của tập tin
GetAttributes()	Lấy/ thiết đặt các thuộc tính của một tập tin	LastAccessTime	Thừa kế từ FileSystemInfo. Thời điểm truy cập gần nhất
SetAttributes()		LastWriteTime	Thừa kế từ FileSystemInfo. Thời điểm ghi gần nhất
GetCreationTime()		Length	Kích thước tập tin
SetCreationTime()	Lấy/ thiết đặt thời gian tạo tập tin	Name	Tên tập tin
GetLastAccessTime()		AppendText()	Tạo đối tượng StreamWriter để ghi thêm vào tập tin
SetLastAccessTime()	Lấy/ thiết đặt thời gian truy cập tập tin lần cuối	CopyTo()	Sao chép sang một tập tin mới
GetLastWriteTime()		Create()	Tạo một tập tin mới
SetLastWriteTime()	Lấy/ thiết đặt thời gian chỉnh sửa tập tin lần cuối	Delete()	Xóa tập tin
Move()	Di chuyển tập tin đến vị trí mới, có thể dùng để đổi tên tập tin	MoveTo()	Di chuyển tập tin, cũng dùng để đổi tên tập tin
OpenRead()	Mở một tập tin để đọc (không ghi)	Open()	Mở một tập tin với các quyền hạn
OpenWrite()	Mở một tập tin cho phép ghi.	OpenRead()	Tạo đối tượng FileStream cho việc đọc tập tin
		OpenText()	Tạo đối tượng StreamReader cho việc đọc tập tin
		OpenWrite()	Tạo đối tượng FileStream cho việc ghi tập tin

Classes File and Directory

Phương thức	Giải thích
CreateDirectory()	Tạo tất cả các thư mục và thư mục con trong đường dẫn tham số.
Delete()	Xóa thư mục và các nội dung của nó.
Exists()	Trả về kết quả kiểu logic, đúng nếu đường dẫn đến thư mục tồn tại (có nghĩa là thư mục tồn tại).
GetCreationTime()	
SetCreationTime()	Lấy/thiết đặt ngày giờ tạo thư mục
GetCurrentDirectory()	
SetCurrentDirectory()	Lấy/thiết đặt thư mục hiện hành
GetDirectories()	Lấy về một mảng các thư mục con một thư mục
GetDirectoryRoot()	Trả về thư mục gốc của đường dẫn
GetFiles()	Trả về mảng chuỗi tên các tập tin chứa trong một thư mục
GetLastAccessTime()	
SetLastAccessTime()	Lấy/thiết đặt ngày giờ lần truy cập cuối cùng đến thư mục
GetLastWriteTime()	
SetLastWriteTime()	Lấy/thiết đặt ngày giờ lần chỉnh sửa cuối cùng lên thư mục
GetLogicalDrives()	Trả về tên của tất cả các ổ đĩa logic theo định dạng <ổ_đĩa>:\
GetParent()	Trả về thư mục cha của một đường dẫn.
Move()	Di chuyển thư mục (cả nội dung) đến một vị trí khác.

Phương thức/property	Ý nghĩa
Attributes	Thừa kế từ FileSystemInfo, lấy/thiết đặt thuộc tính của tập tin hiện hành.
CreationTime	Thừa kế từ FileSystemInfo, lấy/thiết đặt thời gian tạo tập tin
Exists	Trả về đúng nếu thư mục tồn tại
Extension	Thừa kế từ FileSystemInfo, phần mở rộng tập tin
FullName	Thừa kế từ FileSystemInfo, đường dẫn đầy đủ của tập tin hay thư mục
LastAccessTime	Thừa kế từ FileSystemInfo, ngày giờ truy cập cuối cùng
LastWriteTime	Thừa kế từ FileSystemInfo, ngày giờ chỉnh sửa cuối cùng
Name	Tên thư mục
Parent	Lấy thư mục cha
Root	Lấy thư mục gốc của đường dẫn.
Create()	Tạo một thư mục
CreateSubdirectory()	Tạo một hoặc nhiều thư mục con
Delete()	Xóa một thư mục và nội dung của nó
GetDirectories()	Trả về danh sách các thư mục con của thư hiện hiện có
GetFiles()	Lấy danh mục các tập tin của thư mục
GetFileSystemInfos()	Nhận về mảng các đối tượng FileSystemInfo
MoveTo()	Di chuyển DirectoryInfo và nội dung của nó sang đường dẫn khác
Refresh()	Làm tươi trạng thái đối tượng

Directory

- Tạo nút lệnh chọn folder

```
private void btbrows_Click(object sender, EventArgs e)
{
    string[] directoryList;
    directoryList = Directory.GetDirectories( fileName );
    outputtextBox.Text = MotaFolder( fileName );

    private string MotaFolder(string fileName)
    {
        string information;
        information = fileName + " exists\r\n\r\n";
        information += "Created: " + File.GetCreationTime(fileName) + "\r\n";
        information += "Last modified: " + File.LastWriteTime(fileName) + "\r\n";
        information += "Last accessed: " + File.LastAccessTime(fileName) + "\r\n";
        information += "Attributes: " + File.Attributes(fileName) + "\r\n" + "\r\n";
        return information;
    }
}
```

Directory

- Tạo folder
- Xóa folder
- Thuộc tính

```
private void btnxoafolder_Click(object sender, EventArgs e)
{
    if (txtnewfolder.Text == "")
    {
        MessageBox.Show("Đã xóa tập folder chưa?");
    }
}

//Khai báo biến toàn cục
FolderBrowserDialog fbd = new FolderBrowserDialog();
private void btnpath_Click(object sender, EventArgs e)
{
    fbd.ShowDialog(); txtpath.Text = fbd.SelectedPath;
}

private void btntaofolder_Click(object sender, EventArgs e)
{
    Directory.CreateDirectory(fbd.SelectedPath+"\" +txtfolder.Text);
    MessageBox.Show("Đã tạo xong thư mục con của thư mục vừa chọn!", "Chú ý", MessageBoxButtons.OK, MessageBoxIcon.Information);
}
}
```

Directory

```
Directory.Exists( fileName )
File.GetCreationTime( fileName )
File.LastWriteTime( fileName )
File.LastAccessTime( fileName )
File.Attributes( fileName )

DirectoryInfo dir = new DirectoryInfo(filename);
DirectoryInfo[] a = dir.GetDirectories();
foreach (object i in a)
{
    txtout.Text += "\r\n" + i.ToString();
}

<Đối tượng folder>.Attributes = FileAttributes.Hidden/ReadOnly/Normal
```

Dùng mảng để lưu trữ

File

- Kiểm tra file
- Duyệt file
 - Mô tả
 - Duyệt nội dung file
- Tạo file
- Ghi nội dung vào file
- Xoá file
- Thuộc tính file

Kiểm tra file

```
private void btnduyetfile_Click(object sender, EventArgs e)
{
    if (File.Exists(txttenfile.Text))
    {
        string filename = txttenfile.Text;
        txtout.Text = MotaFile(filename);
    }
}
```

Duyệt File

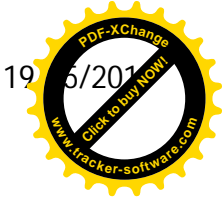
- Mô tả
- Nội dung

```
private string MotaFile(string fileName)
{
    string information;
    information = fileName + " exists\r\n\r\n";
    information += "Created: " + File.GetCreationTime(fileName) + "\r\n";
    information += "Last modified: " + File.GetLastWriteTime(fileName) + "\r\n";
    information += "Last accessed: " + File.GetLastAccessTime(fileName) + "\r\n";
    information += "Attributes: " + File.GetAttributes(fileName) + "\r\n" + "\r\n";
    return information;
}
```

```
StreamReader stream = new StreamReader(filename);
txtout.Text += stream.ReadToEnd();
```

Tạo file

```
private void btntaofile_Click(object sender, EventArgs e)
{
    File.Create(txttenfile.Text);
}
```



Ghi dữ liệu vào File

```
private void btnghifile_Click(object sender, EventArgs e)
{
    string file = txttenfile.Text;
    StreamWriter stream = new StreamWriter(file, true);
    stream.WriteLine("Đã nhập một dòng vào File " + file);
    stream.Close();
}
```

Xoá File

```
private void btnxoafile_Click(object sender, EventArgs e)
{
    File.Delete(@"D:\data.txt");
    MessageBox.Show("Đã xóa xong!");
}
```

```
private void btnthuocinhf_Click(object sender, EventArgs e)
{
    string file2 = txttenfile.Text;
    FileInfo fi = new FileInfo(file2);
    fi.Attributes = FileAttributes.Hidden/ReadOnly/Normal;
}
```

File

```
File.Exists( fileName )
File.GetCreationTime(fileName)
File.GetLastWriteTime(fileName)
File.GetLastAccessTime(fileName)
File.GetAttributes(fileName)

StreamReader stream = new StreamReader( fileName );
outputtextBox.Text += stream.ReadToEnd();

StreamWriter stream = new StreamWriter(file, true);
stream.WriteLine("Đã nhập một dòng vào File " + file);
stream.Close();

File.Delete(@"D:\data.txt");
<Đối tượng file>.Attributes = FileAttributes.Hidden/ReadOnly/Normal
```

```
DirectoryInfo dir = new DirectoryInfo(txtfolder.Text);
FileInfo[] a = dir.GetFiles();
foreach (FileInfo i in a)
{
    i.Attributes = FileAttributes.Hidden;
}

DirectoryInfo[] b = dir.GetDirectories();
foreach (DirectoryInfo j in b)
{
    j.Attributes = FileAttributes.Hidden;
}
```

Bắt lỗi

Khi làm việc với **File** và **Directory** rất dễ phát sinh lỗi, bạn cần tổ chức chương trình quản lý tốt vấn đề này

```
try
    { // đoạn lệnh để phát sinh lỗi }
catch (..Exception exp1)
    { // xử lý lỗi như thế nào }

finally
    { // làm việc gì đó mặc dù có lỗi }
```

```
try
{
    StreamReader stream = new StreamReader( fileName );
    outputtextBox.Text += stream.ReadToEnd();
} // end try

// handle exception if StreamReader is unavailable

catch ( IOException )
{
    MessageBox.Show( "Error reading from file", "File Error",
        MessageBoxButtons.OK, MessageBoxIcon.Error );
} // end catch
```

Run file

```
private void button1_Click(object sender, EventArgs e)
{
    Process k = new Process();
    k.StartInfo.FileName = "C:\\Windows\\system32\\mspaint.exe";
    k.Start();
}
```