TRƯỜNG ĐẠI HỌC MỞ TP. HCM
Cơ hội học tập cho mọi người

# LẬP TRÌNH GIAO DIỆN

Nguyễn Thị Mai Trang

1

1

TRƯỜNG ĐẠI HỌC MỞ TP. HCM
Cơ hội học tập cho mọi người

## Chương 7

# Làm việc với File và Thư mục

2

2

TRƯỜNG ĐẠI HỌC MỞ TP. HCM
Cơ hội học tập cho mọi người

## MỤC TIÊU

- Tạo, đọc, ghi và cập nhật được file
- Sử dụng được lớp File và Directory để truy xuất được thông tin về file và thư mục lưu trữ trên máy tính
- Thao tác thành thạo khi truy cập file tuần tự
- Sử dụng được các lớp FileStream, StreamReader, StreamWriter để đọc và ghi file text
- Sử dụng được lớp FileStream và BinaryFormater để đọc và ghi các đối tượng vào file

3

3

TRƯỜNG ĐẠI HỌC MỞ TP. HCM
Cơ hội học tập cho mọi người

## NỘI DUNG

1. Lớp File, Directory và Stream
2. Làm việc với file và thư mục
3. Serialization

4

4

TRƯỜNG ĐẠI HỌC MỞ TP. HCM
Cơ hội học tập cho mọi người

## 7.1 Lớp File, Directory và Stream

- Giới thiệu
- Lớp File
- Lớp Directory

5

5

TRƯỜNG ĐẠI HỌC MỞ TP. HCM
Cơ hội học tập cho mọi người

## Giới thiệu

- Khi muốn đọc hay ghi dữ liệu vào/ra tập tin hay muốn truyền dữ liệu từ máy này sang máy khác, ta phải tổ chức dữ liệu theo cấu trúc tuần tự các byte hay các gói tin ….
- Thư viện .NET Framework cung cấp lớp **Stream** (và các lớp kế thừa) để chương trình có thể sử dụng trong các thao tác nhập xuất dữ liệu như đọc/ghi tập tin, truyền dữ liệu qua mạng,…
- Trong C#, để có thể sử dụng các lớp, đối tượng thao tác với file và thư mục, cần khai báo không gian tên ***System.IO***

6

6

TRƯỜNG ĐẠI HỌC MỞ TP. HCM
Cơ hội học tập cho mọi người

## Giới thiệu (tt)

- **Một số lớp Stream thông dụng:**
  - **Stream**: Lớp trừu tượng, cung cấp chức năng đọc/ghi dữ liệu theo byte
  - **BinaryReader**: Đọc dữ liệu nhị phân
  - **BinaryWriter**: Ghi dữ liệu nhị phân
  - **File, FileInfo, Directory, DirectoryInfo**: cung cấp các phương thức cho phép thao tác với tập tin và thư mục như tạo, xóa, đổi tên, liệt kê file và thư mục,...
  - **FileStream**: Đọc/ ghi tập tin theo cơ chế đồng bộ / bất đồng bộ, mặc định là đồng bộ.
  - **TextReader, TextWriter**: lớp trừu tượng cho phép đọc, ghi ký tự
  - **StringReader, StringWriter**: kế thừa từ TextReader, TextWriter, cài đặt thêm các phương thức đọc, ghi chuỗi.

7

7

TRƯỜNG ĐẠI HỌC MỞ TP. HCM
Cơ hội học tập cho mọi người

## Lớp File

- **Một số phương thức static của class File:**
  - AppentText: ghi nội dung văn bản vào cuối file.
  - Copy: sao chép file.
  - CreateText: tạo file văn bản.
  - Delete: Xóa file.
  - Exists: kiểm tra sự tồn tại của file.
  - GetCreationTime: trả về đối tượng DateTime là thời điểm file được tạo.
  - GetLastAccessTime: trả về đối tượng DateTime là lần cuối cùng truy cập file

8

8

TRƯỜNG ĐẠI HỌC MỞ TP. HCM
Cơ hội học tập cho mọi người

## Lớp File (tt)

- **Một số phương thức static của class File (tt):**
  - GetLastWriteTime: trả về đối tượng DateTime là lần cuối cùng cập nhật nội dung file.
  - Move: di chuyển file.
  - Open: mở file.
  - OpenRead: mở file chỉ để đọc.
  - OpenText: mở file văn bản.
  - OpenWrite: mở file để ghi.

9

9

TRƯỜNG ĐẠI HỌC MỞ TP. HCM
Cơ hội học tập cho mọi người

## Lớp Directory

- **Một số phương thức của class Directory:**
  - CreateDirectory: tạo thư mục.
  - Delete: Xóa thư mục.
  - Exists: kiểm tra sự tồn tại của thư mục.
  - GetCreationTime: trả về đối tượng DateTime là thời điểm thư mục được tạo.
  - GetLastAccessTime: trả về đối tượng DateTime là lần cuối cùng truy cập thư mục.
  - GetLastWriteTime: trả về đối tượng DateTime là lần cuối cùng cập nhật nội dung thư mục.
  - Move: di chuyển thư mục.

10

10

TRƯỜNG ĐẠI HỌC MỞ TP. HCM
Cơ hội học tập cho mọi người

# 7.2 Làm việc với file và thư mục

- **Đọc, ghi tập tin nhị phân**
  - Sử dụng lớp cơ sở Stream. Lớp Stream có rất nhiều phương thức nhưng quan trọng nhất là các phương thức Read(), Write(), BeginRead(), BeginWrite() và Flush().
  - Cách thực hiện như sau:
    - Tạo đối tượng Stream để đọc hoặc ghi.
    - Sử dụng phương thức File.OpenRead để đọc và File.OpenWrite để ghi vào file

11

11

TRƯỜNG ĐẠI HỌC MỞ TP. HCM
Cơ hội học tập cho mọi người

# Làm việc với file và thư mục (tt)

- Ví dụ: đọc nội dung file E:\ball.jpg ghi vào file E:\ball1.jpg

```
static void BinaryReadWrite()
{
    //tạo đối tượng Stream để đọc tập tin ball.jpg trên đĩa E
    Stream inputStream = File.OpenRead(@"E:\ball.jpg");
    //tạo đối tượng Stream để ghi vào tập tin ball1.jpg
    Stream outputStream = File.OpenWrite(@"E:\ball1.jpg");
    //khai báo bộ đệm chứa các byte lúc đọc
    byte[] buffer = new Byte[1024];
    int bytesRead;//biến lưu số byte thực sự đọc được
    //đọc tuần tự cho đến cuối tập tin
    while ((bytesRead = inputStream.Read(buffer, 0, SizeBuff)) > 0)
    {   //ghi vào tập tin
        outputStream.Write(buffer, 0, bytesRead);
    }
    //đóng file
    inputStream.Close();
    outputStream.Close();
}
```

12

12

## Làm việc với file và thư mục (tt)

- **Đọc, ghi tập tin văn bản:**
  - Sử dụng hai lớp **StreamReader, StreamWriter**
    - **StreamReader**: đọc tập tin với các hàm **Read**, **ReadLine**
    - **StreamWriter**: ghi vào tập tìn với các hàm **Write**, **WriteLine**
  - Ví dụ

```
//tạo đối tượng StreamReader để đọc tập tin test.txt
StreamReader reader = new StreamReader(@"D:\test.txt");
//tạo đối tượng StreamWriter để ghi vào tập tin test1.txt
StreamWriter writer = new StreamWriter(@"D:\test1.txt");
string text;//biến lưu từng dòng mỗi khi đọc
do
{
    text = reader.ReadLine();//đọc từng dòng từ test.txt
    writer.WriteLine(text);//ghi vào test1.txt
} while (text != null);
//đóng tập tin
reader.Close();
writer.Close();
```

13

13

## Làm việc với file và thư mục (tt)

- **Tạo, xóa, di chuyển file**: sử dụng các phương thức static của lớp File:
  - File.Create ( "file_name" );
  - File.Delete ( "file_name" );
  - File.Move ( "source_file" , "dest_file" )
- **Tạo, xóa, di chuyển thư mục**: sử dụng các phương thức static của lớp Directory:
  - Directory.CreateDirectory ( "dir_name" );
  - Directory.Delete ( "dir_name" );
  - Directory.Move ( "source_dir" , "dest_dir")

14

14

TRƯỜNG ĐẠI HỌC MỞ TP. HCM
Cơ hội học tập cho mọi người

## Làm việc với file và thư mục (tt)

- **Truy xuất thông tin của file**: sử dụng các phương thức static của lớp File:
  - File.GetCreationTime ("file_name");
  - File.GetLastAccessTime ("file_name");
  - File.GetAttributes ("file_name");
  - File. GetLastWriteTime ( "file_name");
- **Truy xuất thông tin của thư mục**: sử dụng các phương thức static của lớp Directory
  - Directory.GetCreationTime("dir_name");
  - Directory.GetLastAccessTime("dir_name");
  - Directory.GetAttributes("dir_name");
  - Directory.GetLastWriteTime( "dir_name" );

15

15

TRƯỜNG ĐẠI HỌC MỞ TP. HCM
Cơ hội học tập cho mọi người

## Làm việc với file và thư mục (tt)

- **Liệt kê file và thư mục con trong thư mục**: sử dụng các phương thức static của lớp Directory
  - Directory.GetFiles ("dir_name");
  - Directory.GetDirectories ("dir_name");

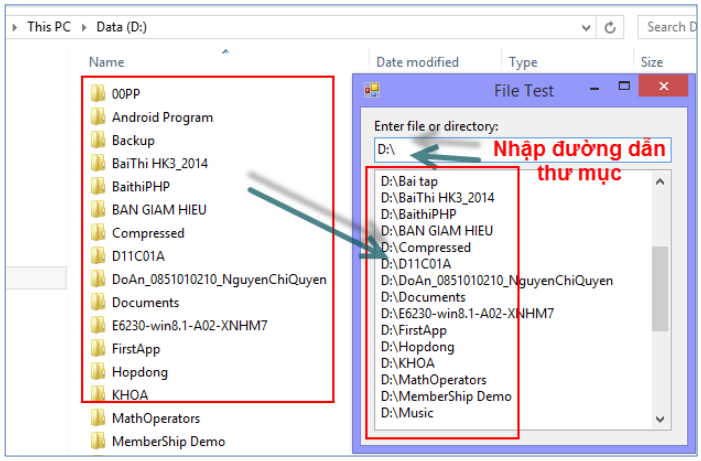
16

16

TRƯỜNG ĐẠI HỌC MỞ TP. HCM
Cơ hội học tập cho mọi người

# Làm việc với file và thư mục (tt)

– Ví dụ: nhập đường dẫn ổ đĩa/thư mục, enter, danh sách tập tin và thư mục trên ổ đĩa sẽ hiển thị trên textbox bên dưới



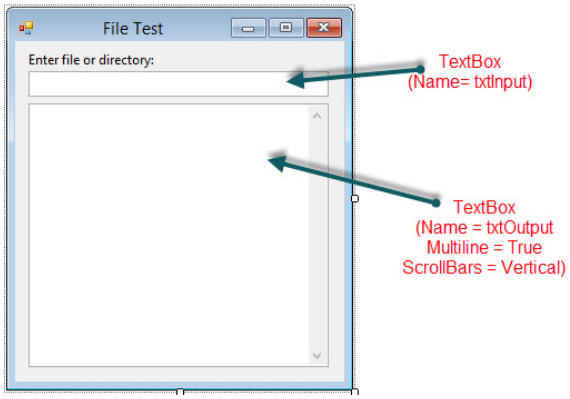

17

17

TRƯỜNG ĐẠI HỌC MỞ TP. HCM
Cơ hội học tập cho mọi người

# Làm việc với file và thư mục (tt)

• **Thiết kế giao diện**




18

18

TRƯỜNG ĐẠI HỌC MỞ TP. HCM
Cơ hội học tập cho mọi người

# Làm việc với file và thư mục (tt)

– Phương thức GetInformation hiển thị thông tin của thư mục

```
private void GetInformation(string fileName)
{
    txtOutput.Clear();
    txtOutput.AppendText(fileName + " exists\n");
    txtOutput.AppendText("Created: " +
       File.GetCreationTime(fileName) + "\n");
    txtOutput.AppendText("Last modified: " +
       File.GetLastWriteTime(fileName) + "\n");
    txtOutput.AppendText("Last accessed: " +
       File.GetLastAccessTime(fileName) + "\n");
}
```

19

19

TRƯỜNG ĐẠI HỌC MỞ TP. HCM
Cơ hội học tập cho mọi người

# Làm việc với file và thư mục (tt)

– Khai báo hàm xử lý sự kiện KeyDown trên textbox txtInput

```
private void txtInput_KeyDown( object sender, KeyEventArgs e )
{
   if ( e.KeyCode == Keys.Enter )//nếu nhấn phím Enter
   { //Lấy tên file từ textbox txtInput
      string fileName = txtInput.Text;
      if ( File.Exists(fileName))//nếu là file
      {
         GetInformation(fileName);
         //đọc nội dung file lên textbox txtOutput
         StreamReader stream = new StreamReader(fileName);
         try  {
               txtOutput.AppendText( stream.ReadToEnd() );
         }
         catch (IOException) {
            MessageBox.Show("Error reading from file","File Error",
                 MessageBoxButtons.OK, MessageBoxIcon.Error );
         }
         finally{
            stream.Close();
         }
      }
```

20

20

TRƯỜNG ĐẠI HỌC MỞ TP. HCM
Cơ hội học tập cho mọi người

# Làm việc với file và thư mục (tt)

- Xử lý sự kiện KeyDown (tt)

```
        else if (Directory.Exists(fileName))//nếu là thư mục
        {
           GetInformation( fileName );
           //lấy danh sách các thư mục con
           string[] arrDir = Directory.GetDirectories(fileName );
           txtOutput.AppendText( "Directory contents:\n" );
            //hiển thị các thư mục con trong textbox txtOuput
           foreach (string dir in arrDir )
              txtOutput.AppendText (dir + "\n" );
        }
        else // tên tập tin hoặc thư mục không hợp lệ
        {
           MessageBox.Show( txtInput.Text + " does not exist",
               "File Error", MessageBoxButtons.OK,
               MessageBoxIcon.Error );
        }
     }
}
```

21

21

TRƯỜNG ĐẠI HỌC MỞ TP. HCM
Cơ hội học tập cho mọi người

# 7.3 Serialization

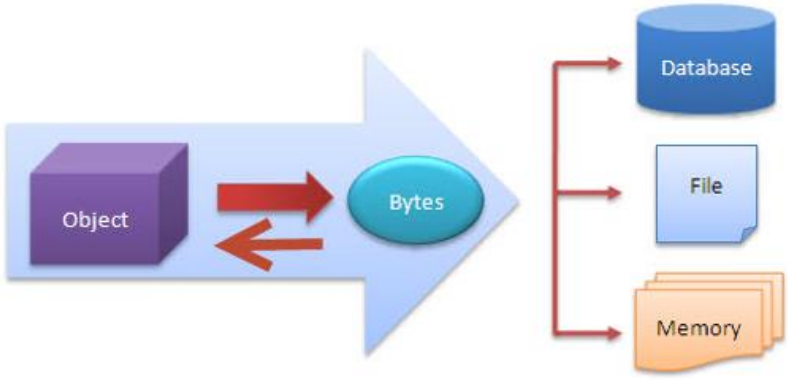
- **Serialization**
  - Là quá trình chuyển đổi một cấu trúc dữ liệu hoặc đối tượng thành một định dạng có thể lưu trữ được vào file, bộ nhớ, hoặc vận chuyển thông qua mạng.
  - Ví dụ: tạo một đối tượng hình tròn, vẽ trên giao diện chương trình, sau đó lưu thành file.
- **Deserialization**:
  - Quá trình phục hồi dữ liệu lưu trong file trở lại trạng thái ban đầu.
  - Ví dụ: đọc file đã lưu đối tượng hình tròn trước đó, vẽ lại trên giao diện.

22

22

TRƯỜNG ĐẠI HỌC MỞ TP. HCM
Cơ hội học tập cho mọi người

## Serialization (tt)




23

23

TRƯỜNG ĐẠI HỌC MỞ TP. HCM
Cơ hội học tập cho mọi người

## Serialization (tt)

- .NET Framework cung cấp 2 kỹ thuật serialize:
  - Binary serialize (serialize nhị phân):
    - Giữ nguyên kiểu dữ liệu → giữ nguyên cấu trúc đối tượng.
    - Được dùng để chia sẻ đối tượng giữa các ứng dụng bằng cách serialize vào vùng nhớ clipboard; serialize vào các luồng, đĩa từ, bộ nhớ, trên mạng …; truyền cho máy tính khác trên mạng.
  - XML và SOAP Serialize:
    - Chỉ serialize các trường dữ liệu public → không giữ nguyên kiểu dữ liệu.
    - XML và SOAP thường được sử dụng trong việc truyền dữ liệu từ các dịch vụ như web service.

24

24

TRƯỜNG ĐẠI HỌC MỞ TP. HCM
Cơ hội học tập cho mọi người

## Serialization (tt)

- Các đối tượng cơ sở đều có khả năng serialize.
- Đối với các đối tượng do người sử dụng định nghĩa → đặt trên khai báo lớp chỉ thị **[Serializable]**.
- Muốn loại trừ một thành phần (phương thức, biến thành viên, thuộc tính,…) không muốn được serialize → ta đặt trước các khai báo **[NonSerialized]**
- Để có thể serialize đối tượng của lớp → sử dụng các đối tượng Formatter trong .NET → thực thi interface IFormatter (namespace System.Runtime.Serialization)

25

25

TRƯỜNG ĐẠI HỌC MỞ TP. HCM
Cơ hội học tập cho mọi người

## Serialization (tt)

- Các đối tượng được sử dụng để serialize trong .Net:
  - BinaryFormatter: serialize đối tượng thành một tập tin nhị phân, thuộc namespace System.Runtime.Serialization.Formatters.Binary.
  - SoapFormatter: serialize đối tượng thành định dạng XML để truyền tải thông tin giữa các ứng dụng qua mạng thông qua giao thức HTTP, thuộc namespace System.Runtime.Serialization.Formatters.Soap.
- Interface IFormatter chỉ có hai phương thức là **Serialize()** và **Deserialize().**

26

26

TRƯỜNG ĐẠI HỌC MỞ TP. HCM
Cơ hội học tập cho mọi người

## Serialization (tt)

- **Các bước Serialization:**
  - Thiết kế lớp khả tuần tự (khai báo**[Serialize]** trên khai báo lớp).
  - Khai báo các namespace cần thiết
    - using System.IO;
    - using System.Runtime.Serialization;
    - using System.Runtime.Serialization.Formatters.Binary;
  - Tạo đối tượng stream
    - FileStream stream = new FileStream(“filename", FileMode.Create, FileAccess.Write);
  - Tạo đối tượng BinaryFormatter
    - BinaryFormatter bf = new BinaryFormatter();
  - Gọi phương thức Serialize
    - bf.Serialize(stream, objclass);

27

27

TRƯỜNG ĐẠI HỌC MỞ TP. HCM
Cơ hội học tập cho mọi người

## Serialization (tt)

- **Các bước Deserialization:**
  - Khai báo các namespace cần thiết
    - using System.IO;
    - using System.Runtime.Serialization;
    - using System.Runtime.Serialization.Formatters.Binary;
  - Tạo đối tượng stream
    - FileStream stream = new FileStream(“filename", FileMode.Open,FileAccess.Read);
  - Tạo đối tượng BinaryFormatter
    - BinaryFormatter bf = new BinaryFormatter();
  - Gọi phương thức Deserialize
    - objclass = (objclass) bf. Deserialize(stream);

28

28