

文件操作首先需要引用命名空间 `System.IO`;

File类(文件)

```
1 File.Exists(); //判断文件是否存在
2 File.ReadAllText(); //读取所有文本
3 File.WriteAllText(); //写入所有文本
4 File.AppendAllText(); //追加所有文本
5 File.ReadAllLines(); //读取所有行返回字符串数组
6 File.WriteAllLines(); //写入所有行
7 File.ReadAllBytes(); //读取所有字节
8 File.WriteAllBytes(); //写入所有字节
9 File.Delete(); //删除
```

Encoding编码和解码

```
1 获取编码格式:
2 Encoding encoding_utf8= Encoding.GetEncoding("Utf-8");//Utf-8是编码的名字
3 编码:文本>>字节数组
4 Byte[] bytes=encoding_utf8.GetBytes("文本");
5 解码:字节数组>>文本
6 string str=encoding_utf8.GetString(bytes);
```

Directory(目录)

```
1 Directory.CreateDirectory(); //创建目录
2 Directory.Delete(); //删除
3 Directory.Exists(); //目录是否存在
4 Directory.GetFiles(); //获取所有文件
5 Directory.GetDirectories(); //获取所有子目录
6 Directory.GetCurrentDirectory(); //获取当前程序的工作目录
7 Directory.Move(); //移动目录和目录中的内容
8 Directory.GetLogicalDrives(); //获取电脑上的盘符信息
```

Path类(路径)本质是一个字符串操作类

```
1 Path.Combine(); //连接
2 Path.GetExtension(); //获取扩展名
3 Path.GetFileName(); //获取文件名
4 Path.GetFileNameWithoutExtension(); //获取没有扩展名的文件名
5 Path.GetFullPath(); //获取完整路径
```

FileStream文件流操作

为什么要使用文件流？

1. 超大文件不能一次性载入内存, 内存装不下
2. 一次性读取超大数据需要耗费较长时间

常见的流：

- | | |
|------|-------------|
| 1文件流 | 硬盘到内存 |
| 2内存流 | 内存到内存 |
| 3网络流 | 一台机器到另外一台机器 |

```
1 //创建文件流的两种方式
2 FileStream fs = new FileStream(路径, 文件模式, 文件流权限, 共享);
3 File.Open(); 或 File.Create();
4 /*参数 描述
5 FileMode(打开文件后指针放哪里, 如果文件已经存在怎么处理)
6 FileAccess(想对文件进行什么样的操作, 是读还是写)
7 FileShare(别的应用程序还能怎样操作这个文件)
8 */
```

文件流常用方法

```
1 Read(); //读
2 Write(); //写
3 WriteByte(); //写一个字节
4 Flush(); //清除缓冲区, 使缓冲区内容全部写到文件中
5 Seek(); //跟相对位置定位指针
6 Position; //指针位置
```

```
7 Close(); //关闭文件流
8 Dispose(); //释放
```

流的读写器(只能操作文本)

StreamReader读取器

```
1 BaseStream();
2 CurrentEncoding();
3 Read();
4 ReadLine();
5 ReadToEnd();
6 EndOfStream();
```

StreamWriter写入器

```
1 StreamWriter写入器
2 Write();
3 WriteLine();
4 Flush();
5 Close();
6 Dispose();
```

IO操作相关的类:

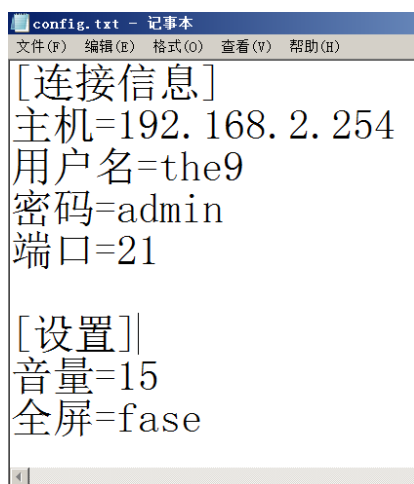
1	BinaryReader	从二进制流读取原始数据。
2	BinaryWriter	以二进制格式写入原始数据。
3	BufferedStream	字节流的临时存储。
4	Directory	有助于操作目录结构。
5	DirectoryInfo	用于对目录执行操作。
6	DriveInfo	提供驱动器的信息。
7	File	有助于处理文件。
8	FileInfo	用于对文件执行操作。
9	FileStream	用于文件中任何位置的读写。
10	MemoryStream	用于随机访问存储在内存中的数据流。
11	Path	对路径信息执行操作。

12	StreamReader	用于从字节流中读取字符。
13	StreamWriter	用于向一个流中写入字符。
14	StringReader	用于读取字符串缓冲区。
15	StringWriter	用于写入字符串缓冲区。

作业:

- 1.定义一个使用文件流复制文件的方法
- 2.自学递归(方法自己调自己),能在控制台输出D盘所有文件
- 3.Ini文件读取类IniFile

比如有这样一个Ini文件



```
config.txt - 记事本
文件(F)  编辑(E)  格式(O)  查看(V)  帮助(H)

[连接信息]
主机=192.168.2.254
用户名=the9
密码=admin
端口=21

[设置]
音量=15
全屏=fase
```

可以使用如下的方式访问拿到ip地址:

```
1 IniFile ini=new IniFile();
2 ini.ReadFile(@"D:\config.txt");
3 string ip= ini.GetString("连接信息","主机")
4 print(ip); //输出192.168.2.254
```