

Path和File类

一、Path类

用来存放文件路径相关

```
string str = @"E:\Code\C# for VS_Code\Path类\老牛.wav";
//获取文件名的老方法:
int index = str.LastIndexOf(@"\");
str = str.Substring(index + 1);
System.Console.WriteLine(str);

//使用Path类的新方法:
//GetFileName() : 直接获取文件名
string fileName = Path.GetFileName(str);

//GetFileNameWithoutExtension() : 直接获取不含扩展名的文件名
fileName = Path.GetFileNameWithoutExtension(str);

//GetExtension() : 获取文件扩展名
string fileExtension = Path.GetExtension(str);

//GetDirectoryName() : 获取文件所在的文件夹路径
string fileDirectory = Path.GetDirectoryName(str);
System.Console.WriteLine(fileDirectory);    //结果为: E:\Code\C# for
VS_Code\Path类

//GetFullPath() : 获取文件的绝对路径
string fileFullPath = Path.GetFullPath(str);

//Combine() : 连接两个字符串作为路径
string combinePath = Path.Combine(@"C:\system\",@"TXT\a.txt");
System.Console.WriteLine(combinePath);    //结果为: C:\system\TXT\a.txt
```

二、File类

**注: File类的缺点: 读取时会一次性读取全部数据, 非常消耗内存, 因此只能读取小文件
读取大文件要用文件流: FileStream**

1.创建文件

```
//Create() : 在指定路径创建一个文件
File.Create(@"C:\Users\Admin\Desktop\new.txt");
System.Console.WriteLine("创建成功");
```

2.删除文件

```
//Delete() : 删除指定文件
File.Delete(@"C:\Users\Admin\Desktop\new.txt");
```

3.复制文件

```
//Copy() : 复制一个文件

File.Copy(@"C:\Users\Admin\Desktop\new.txt",@"C:\Users\Admin\Desktop\newNew.txt");
```

4.读取内容

```
//ReadAllBytes() : 以字节形式读取文件内容,返回字节数组
byte[] byteArray = File.ReadAllBytes(@"C:\Users\Admin\Desktop\new.txt");

//ReadAllLines() : 以字符串形式整行地读取文件,返回字符串数组
string[] contents = File.ReadAllLines(@"C:\Users\Admin\Desktop\new.txt");

//ReadAllText() : 以字符串形式整篇地读取文件,返回字符串
string str = File.ReadAllText(@"C:\Users\Admin\Desktop\new.txt");

//注: 在ReadAllLines()和ReadAllText()中均可添加encoding参数,用以指定编码形式
//当已经引入System.Text.Encoding库时可简化
File.ReadAllLines(@"C:\Users\new.txt",encoding:System.Text.Encoding.Unicode);
File.ReadAllText(@"C:\Users\new.txt",encoding:System.Text.Encoding.UTF8);

//注: C:\Users\Admin\Desktop\new.txt这种事绝对路径
//直接写new.txt则是相对路径,但需要把文件放在本程序(.exe)同一目录下
```

5.写入数据

1.覆盖写入:(会删除原数据后再写入)

```
//WriteAllBytes(string path,byte[] bytes) : 将指定字节数组写入文件
byte[] byteArray = {97,98,99,10,100,101,102};
File.WriteAllBytes(@"C:\Users\Admin\Desktop\newNew.txt",byteArray);

System.Console.WriteLine(File.ReadAllText(@"C:\Users\Admin\Desktop\newNew.txt"));
//读取结果为: abc \n def

//File.WriteAllLines(string path,string[] contents)    以字符串形式整行地写入文件
string[] strArray = {"abc","def","ghi","jkl","mno","pqr","stu","vwx","yz"};
File.WriteAllLines(@"C:\Users\Admin\Desktop\new.txt",strArray);

//WriteAllText(string path,string contents)    以字符串形式整篇写入文件
```

```
string str = "语文、数学、英语\n生物、物理、化学\n政治、历史、地理";  
File.WriteAllText(@"C:\Users\Admin\Desktop\new.txt",str);
```

2.追加写入：(不会覆盖原数据)

```
//追加写入，不覆盖原数据  
//AppendAllLines(string path,string[] contents)  
string[] strArray = {"abc","def","ghi","jkl","mno","pqr","stu","vwx"};  
File.AppendAllLines(@"C:\Users\Admin\Desktop\new.txt",strArray);  
  
//AppendAllText(string path,string contents)  
File.AppendAllText(@"C:\Users\Admin\Desktop\new.txt","你看会不会覆盖");
```