# 文件系统

1. 文件系统类的介绍
2. 说明

文件操作类大都在 *System.IO* 命名空间里。 *FileSystemInfo* 类是任何文件系统类的基类； *FileInfo* 与 *File* 表示文件系统中的文件； *DirectoryInfo* 与 *Directory* 表示文件系统中的文件夹； *Path* 表示文件系统中的路径； *DriveInfo* 提供对有关驱动器的信息的访问。注意， *XXXInfo* 与 *XXX* 类的区别是： *XXX* 是静态类， *XXXInfo* 类可以实例化。

还有个较为特殊的类 *System.MarshalByRefObject* 允许在支持远程处理的应用程序中跨应用程序域边界访问对象。

1. 类关系图



二、FileInfo 与 File 类

class Program{

static void Main( string [] args){

FileInfo file =new FileInfo ( @"E:\ 学习笔记\C#平台\test.txt");// 创建文件

 Console.WriteLine("创建时间：" + file.CreationTime);

 Console.WriteLine(" 路径： " + file.DirectoryName);

 StreamWriter sw = file.AppendText(); // 打开追加流

 sw.Write("李志伟" ); // 追加数据

 sw.Dispose();// 释放资源,关闭文件

 File.Move(file.FullName,@"E:\学习笔记\test.txt");// 移动

 Console.WriteLine("完成！" );

 Console.Read();

}

}

Pat屏幕上有字

描述已自动生成h:

三、DirectoryInfo 与 Directory 类

下面的示例演示如何复制目录及其内容。

string sourceDirFullName = @"E:\STUDY";

string targetDirFullName = @"E:\学习笔记2";

DirectoryInfo sourceDir = new DirectoryInfo(sourceDirFullName);

DirectoryInfo targetDir = new DirectoryInfo(targetDirFullName);

CopyAll(sourceDir, targetDir);

*Console*.*WriteLine*("完成！！");

*Console*.*ReadKey*();

}

public static void CopyAll(DirectoryInfo source,DirectoryInfo target) {

if (source.*FullName*.*ToLower*() == target.*FullName*.*ToLower*()) return;

if (Directory.Exists(target.*FullName*) == false) {

Directory.CreateDirectory(target.*FullName*);

}

//复制文件到另一个路径

foreach(*FileInfo* fi in source.GetFiles()) {

*Console*.*WriteLine*(@"Copying {0}\{1}", target.*FullName*, fi.*Name*);

fi.*CopyTo*(Path.Combine(target.ToString(), fi.*Name*), true);

}

foreach(DirectoryInfo sourceDir in source.GetDirectories()) {

DirectoryInfo targetDir = target.CreateSubdirectory(sourceDir.Name);

CopyAll(sourceDir, targetDir);//递归复制文件

}

}电脑屏幕的截图

描述已自动生成

下面的示例演示目录下文件的大小。

DirectoryInfo d = new DirectoryInfo(@"E:\学习笔记2");

*Console*.*WriteLine*("当前项目文件的大小为{0}字节", DirSize(d));

*Console*.*WriteLine*("完成！！");

*Console*.*ReadKey*();

}

public static long DirSize(DirectoryInfo d) {

long Size = 0;

*FileInfo*[] files = d.GetFiles();

foreach (*FileInfo* fi in files) Size += fi.*Length*;

DirectoryInfo[] dirs = d.GetDirectories();

foreach (DirectoryInfo dir in dirs) Size += DirSize(dir);

return Size;

}

手机屏幕的截图

描述已自动生成

四、Path 类

class Program{

static void Main( string [] args){

 Console.WriteLine(Path.Combine(@"E:\学习笔记\C#平台",@"Test\Test.TXT"));//连接

 Console.WriteLine("平台特定的字符："+Path.DirectorySeparatorChar);

 Console.WriteLine("平台特定的替换字符："+Path.AltDirectorySeparatorChar);

  Console .Read();

 }

}

五、DriveInfo 类

class Program{

static void Main( string [] args){

  DriveInfo [] drives = DriveInfo.GetDrives();

  foreach(DriveInfo d in drives){

   if (d.IsReady) {

   Console.WriteLine("总容量："+ d.TotalFreeSpace);

   Console.WriteLine("可用容量："+ d.AvailableFreeSpace);

   Console.WriteLine("驱动器类型"+ d.DriveFormat);

   Console.WriteLine("驱动器的名称："+ d.Name +"\n");

    }

  }

  Console .WriteLine( "OK!" );

  Console .Read();

}

}