Lab11　IO

学习目标：

1. 掌握文件操作
2. 掌握目录操作

一、概述

C#中文件处理技术称为 I/O 技术，即输入与输出技术，或称为流处理技术或文件流处理技术。System.IO 命名空间包含允许读/写文件和数据流的类型以及提供基本文件和目录支持的类型，因此在使用这些类时需要引入 System.IO 命名空间。System.IO 命名空间中的常用类如表所示。

System.IO 命名空间的类

类名	功能和用途
Directory、DirectoryInfo	创建、删除并移动目录，通过属性获取特定目录的相关信息
File、FileInfo	创建、删除并移动文件，通过属性获取特定文件的相关信息
StreamReader、StreamWriter	读写文本数据信息
BinaryReader、BinaryWriter	读写二进制数据

System.IO 命名空间中的常用类大致分为操作目录的类、操作文件的类、文件读写类等。其中，Directory 类和 DirectoryInfo 类属于操作目录的类，File 类和 FileInfo 类属于操作文件的类，StreamReader 类和 StreamWriter 类属于文本文件读写的类，BinaryReader 类和 BinaryWriter 类属于二进制文件读写的类。

二、目录操作

在程序开发中，有时需要对文件目录进行操作，例如创建目录、删除目录等，为此 C#提供了 Directory 类和 DirectoryInfo 类。

1. Directory 类

Directory 类是静态类，提供了许多静态方法用于对目录进行操作，例如创建、删除和移动目录等。Directory 类的一些常用方法如表所示。

Directory 类的常用方法

方法	说明
CreateDirectory()	创建指定路径的目录
Exists()	判断目录是否存在
GetDirectoryRoot()	获取指定目录的根目录
GetDirectories()	获取当前目录之下的 Directory 对象数组
GetFiles()	获取当前目录下的 File 对象数组
Delete()	删除指定目录及其目录下的所有文件
Move()	将指定目录移动到新的位置

【注意】Directory 的 Delete()方法是永久删除，不把文件夹送到回收站；使用 Move()方法移动文件夹时，要注意不能跨磁盘移动，例如 C 盘的文件不能移到 D 盘下。

【实例】目录创建程序。

【操作步骤】

（1）启动 VS，新建一个 Windows 窗体应用程序 DirectoryApplication。

（2）双击 Form1.cs，切换到设计视图，从工具栏中拖拽 1 个 Label 控件、1 个 TextBox 控件和 1 个 Button 控件到窗体设计区，并调整控件大小进行布局。

（3）在窗体设计区中右击窗体 Form1 和每一个控件，设置窗体和控件的相关属性。下表列出了窗体及控件属性。

窗体及控件属性设置

窗体和控件	属性	属性值
Form1	Text	创建目录
label1	Name	请输入创建目录名称：
textBox1	Name	txtDirName
button1	Name	btnMake
	Text	创建

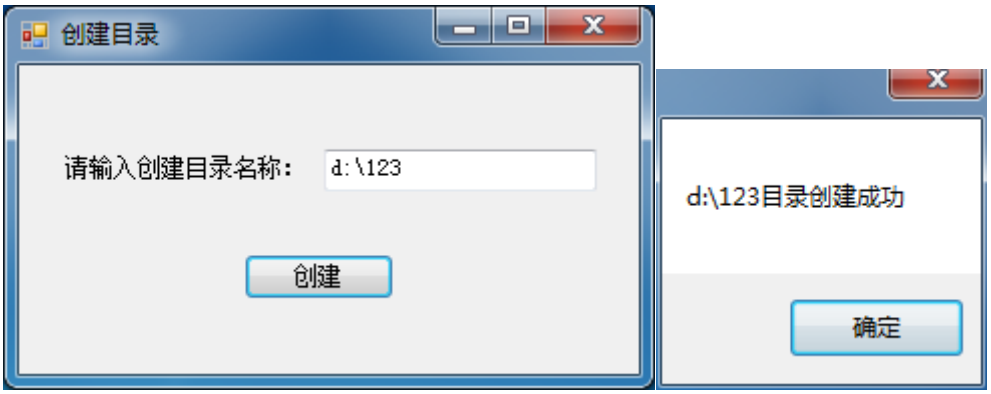
（4）双击“创建”按钮，为其添加单击事件处理程序，程序代码如下。

```
private void btnMake_Click(object sender, EventArgs e)
{
    string path = txtDirName.Text;
    if (Directory.Exists(path))
    {
        MessageBox .Show (path + "目录已经存在");
    }
    else
    {
        Directory.CreateDirectory(path);
        MessageBox.Show(path + "目录创建成功");
    }
}
```

以上代码功能是，首先定义一个 string 类型的变量 path 来保存用户输入的目录，然后调用 Directory 类的 Exists()方法来查找该目录是否存在，Exists()方法的参数是用户输入的目录，即变量 path，如果存在，提示“目录已经存在”；如果不存在，则调用 Directory 类的 CreateDirectory()方法创建该目录，并提示“目录创建成功”。

（5）在解决方案资源管理器中右击 DirectoryApplication 项目，将其设为启动项目。

（6）编译并运行，运行结果如图所示。



2. DirectoryInfo 类

DirectoryInfo 类的功能与 Directory 类相似，不同的是的 DirectoryInfo 类是一个实例类，所有方法都是实例方法。也就是说，要想使用 DirectoryInfo 类所提供的方法必须实例化一个属于 DirectoryInfo 类的对象。因此，如果需要对同一个目录进行多次重复操作时，应该考虑使用 DirectoryInfo 类的实例方法。DirectoryInfo 类不仅拥有与 Directory 类功能相似的方法，而且还具有一些特有的属性，如下表所示。

DirectoryInfo 类的常用属性

属性	说明
Name	获取当前 DirectoryInfo 对象的名称
Root	获取路径的根目录
Parent	获取指定子目录的父目录
FullName	获取目录或文件的完整目录
Exists	判断指定目录是否存在

【实例】目录浏览程序。

【操作步骤】

（1）启动 VS，新建一个 Windows 窗体应用程序 DirectoryInfoApplication。

（2）双击 Form1.cs，切换到设计视图，从工具栏中拖拽 2 个 GroupBox 控件、4 个 Label 控件、4 个 TextBox 控件、1 个 ListBox 控件和 1 个 Button 控件到窗体设计区，调整控件大小进行布局。

（3）在窗体设计区中右击窗体 Form1 和每一个控件，设置窗体和控件的相关属性。下表列出了窗体及控件属性。

窗体及控件属性设置

窗体和控件	属性	属性值
Form1	Text	浏览目录
label1	Name	请输入目录名称：
textBox1	Name	txtDirName
label2	Text	目录名称：
textBox2	Name	txtName
label3	Text	根目录名称：
textBox3	Name	txtRoot
label4	Text	父目录名称：
textBox4	Name	txtParent
groupBox1	Text	详细信息
groupBox2	Text	目录列表
button1	Name	btnBrowse
	Text	浏览
listBox1	Name	listGetDirectories

（4）定义一个 dirList()方法，用于在列表框中循环输出指定目录中的所有子目录。

```
public void dirList(DirectoryInfo dir)
    {
        listGetDirectories.Items.Clear();
        DirectoryInfo[] dirs = dir.GetDirectories();
        for (int i = 0; i < dirs.Length; i++)
        {
            listGetDirectories.Items.Add(dirs[i]);
        }
    }
```

以上代码的功能是，封装一个dirList()方法，该方法没有返回值，有一个DirectoryInfo类型的参数，方法体中首先清空列表框，并通过DirectoryInfo对象的GetDirectories()方法获得目录中的所有子目录，所有子目录构成一个DirectoryInfo类型的数组，然后在列表框中循环输

出该数组中的每一个元素的值。

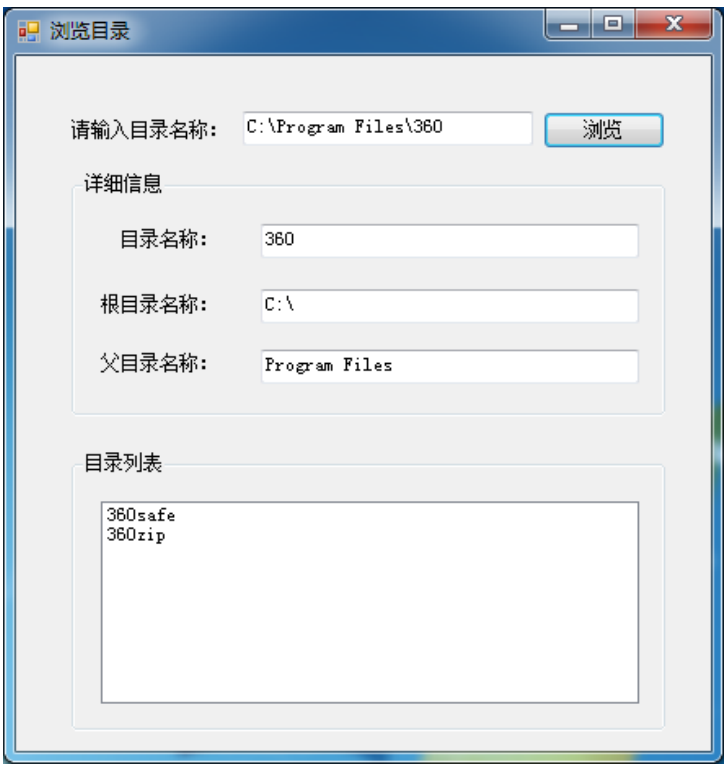
（5）双击“浏览”按钮，为其添加单击事件处理程序，程序代码如下。

```
private void btnBrowse_Click(object sender, EventArgs e)
 {
     string path = txtDirName.Text;
     if (!string.IsNullOrEmpty(path))
     {
         DirectoryInfo dir = new DirectoryInfo(path);
         txtName.Text = dir.Name;
         txtRoot.Text = dir.Root.ToString();
         txtParent.Text = dir.Parent.ToString();
         //输出子目录
         dirList(dir);
     }
     else
     {
         MessageBox.Show("请输入目录名称！");
     }
 }
```

以上代码的功能是，首先定义一个string类型的变量path，用来存储用户输入的目录，然后通过调用string的IsNullOrEmpty()方法来判断path值是否为空，即判断用户是否已输入目录名称，如果不为空，则创建该目录的DirectoryInfo对象，并在相应文本框中输出用户指定目录的名称、根目录名称、父目录名称，通过调用步骤（4）中封装的dirList()方法输出所有子目录。

（6）在解决方案资源管理器中右击 DirectoryInfoApplication 项目，将其设为启动项目。

（7）编译并运行，运行结果如下图所示。



三、文件操作

File 类和 FileInfo 类主要提供有关文件的各种操作，包括创建、复制、移动和删除文件等。

1.File 类

File 类是一个静态类，它提供了许多静态方法，用于处理文件。File 类的一些常用方法如下表所示。

File 类的常用方法

方法	说明
Create()	创建文件
Open()	打开指定路径上的文件，返回 FileStream 对象
Copy()	将文件复制到指定位置
Move()	将指定文件移动到新位置
Delete()	删除文件
Exists()	判断指定文件是否存在

【注意】Directory 和 File 提供的方法都是共享方法，如果执行一次操作，使用共享方法的效率比较高；但如果针对一个目录或文件多次操作，可以考虑使用 DirectoryInfo 和 FileInfo 提供的实例方法。

【实例】文件删除程序。

【操作步骤】

（1）启动 VS，新建一个 Windows 窗体应用程序 FileApplication。

（2）双击 Form1.cs，切换到设计视图，从工具栏中拖拽 1 个 Label 控件、1 个 TextBox 控件和 1 个 Button 控件到窗体设计区，并调整控件大小进行布局。

（3）在窗体设计区中右击窗体 Form1 和每一个控件，设置窗体和控件的相关属性。下表列出了窗体及控件属性。

窗体及控件属性设置

窗体和控件	属性	属性值
Form1	Text	删除文件
label1	Name	请输入删除文件名称：
textBox1	Name	txtDirName
button1	Name	btnDelete
	Text	删除

（4）双击“删除”按钮，为其添加单击事件处理程序，程序代码如下。

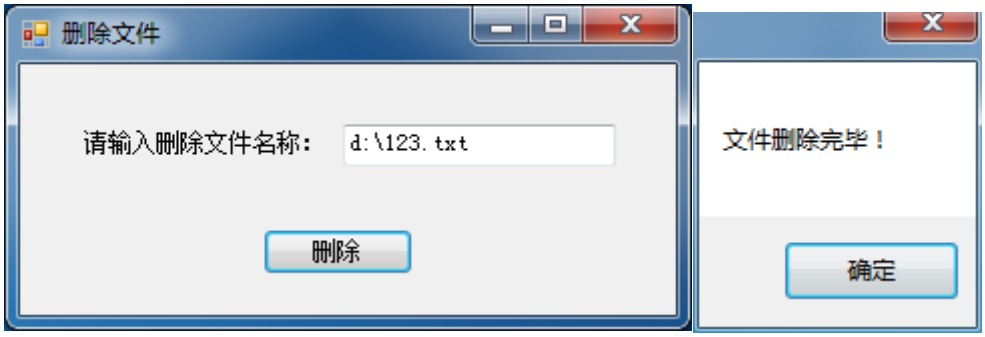
```
private void btnDelete_Click(object sender, EventArgs e)
{
    string path = txtDirName.Text;
    if (File.Exists(path))
    {
        File.Delete(path);
        MessageBox.Show("文件删除完毕！");
    }
    else
    {
        Console.WriteLine("文件不存在！");
    }
```

```
}
```

以上代码功能是，通过File类的Exists()方法判断用户输入的文件是否存在，如果存在，通过调用File类的Delete()方法删除该文件，并提示用户“文件删除完毕!”；如果不存在，提示“文件不存在!”。

（5）在解决方案资源管理器中右击 FileApplication 项目，将其设为启动项目。

（6）编译并运行，运行结果如下图所示。



2. FileInfo 类

FileInfo 类与 File 类有些类似，它们都可以对磁盘上的文件进行操作。不同的是 FileInfo 类是实例类，所有的方法必须实例化对象后才能调用。FileInfo 类除了拥有与 File 类相似的方法外，同时也有它特有的属性如下表所示。

FileInfo 类的常用属性

属性	说明
Directory	获取父目录的实例
DirectoryName	获取表示目录的完整路径的字符串
FullName	获取目录或文件的完整目录
Length	获取当前文件的大小

【实例】文件浏览程序。

【操作步骤】

（1）启动 VS，新建一个 Windows 窗体应用程序 FileInfoApplication。

（2）双击 Form1.cs，切换到设计视图，从工具栏中拖拽 1 个 GroupBox 控件、3 个 Label 控件、3 个 TextBox 控件和 1 个 Button 控件到窗体设计区，调整控件大小进行布局。

（3）在窗体设计区中右击窗体 Form1 和每一个控件，设置窗体和控件的相关属性。下表列出了窗体及控件属性。

窗体及控件属性设置

窗体和控件	属性	属性值
Form1	Text	浏览文件
label1	Name	请输入文件名称：
textBox1	Name	txtFileName
label2	Text	文件当前目录：
textBox2	Name	txtDir
label3	Text	文件大小：
textBox4	Name	txtSize
groupBox1	Text	详细信息
button1	Name	btnBrowse
	Text	浏览

（4）双击“浏览”按钮，为其添加单击事件处理程序，程序代码如下。

```
private void btnBrowse_Click(object sender, EventArgs e)
{
    FileInfo file = new FileInfo(txtFileName.Text);
    if (file.Exists)
    {
        txtDir.Text = file.Directory.ToString();
        txtSize.Text = file.Length.ToString();
    }
    else
    {
        file.Create();
        MessageBox .Show ("文件已经创建成功！");
    }
}
```

以上代码的功能是，首先创建FileInfo类型的对象file，FileInfo类的构造方法中有一个参数，即用户输入的文件名，通过调用对象file的Exists属性来判断该文件是否存在，如果用户指定的文件存在，则输出文件的当前目录和文件大小，如果文件不存在，则通过调用对象的Create()方法创建该文件，并提示“文件已经创建成功!”。

（5）在解决方案资源管理器中右击 FileInfoApplication 项目，将其设为启动项目。

（6）编译并运行，运行结果下图所示。