

IO input-output :

文件或文件夹操作

输入输出流的操作

IO流的分类

根据数据的流向，分类：

- 1.输入流：从输入流中读数据，以Reader或InputStream结尾的类
- 2.输出流：向输出流中写数据，以Writer或OutputStream结尾的类

根据处理数据的单位分为：

1.字节流：可以处理任何类型的数据，以InputStream或OutputStream结尾的类

2.字符流：用于处理文本数据：以Reader或Writer结尾的类

根据流的功能分为：

- 1.节点流（低级流）：针对某种输入设备进行操作的流
- 2.处理流（高级流）：对已有流进行封装，使其具有新的功能

单个路径分隔符

windows: c:\a\b\c.txt

Unix : a/b/c.txt

多路径分隔符

windows: c:\a\b\c.txt;c:\a\v\d.txt

Unix : a/b/c.txt:a/b/d.txt

绝对路径：

从系统根目录开始，到某个文件的路径

相对路径：

从当前文件夹开始，到某个文件的路径

File类：

封装了文件或文件夹的操作

新建文件：

```
1 File file = new File("路径");  
2 file.createNewFile();
```

新建文件夹：

```
1 File dir = new File("路径");
2 dir.mkdir();或 dir.mkdirs();
```

判断文件或文件夹是否存在：

```
1 file.exists();
2 dir.exists();
```

判断是否是文件：

```
1 file.isFile();
2 dir.isFile();
```

获取文件名：

```
1 file.getName();
2 dir.getName();
```

获取绝对路径：

```
1 file.getAbsolutePath();
2 dir.getAbsoutlePath();
```

获取相对路径:

```
1 file.getPath();
2 dir.getPath();
```

获取文件大小（字节）：

```
1 file.length();
2 dir.length();
```

获取时间戳：

```
1 System.currentTimeMillis();
2 file.lastModified();
3 dir.lastModeified();
```

判断是否文件夹

```
1 file.isDirectory();
2 dir.isDirectory();
```

删除文件夹：

```
1 file.delete();
2 dir.delete();
```

获取文件夹内容:

```
1 File dir3 = new File("路径");
2 File[] files = dir3.listFiles();
3 for(File f : files){
```

```
4 System.out.println(f);  
5 }
```