

```
1 IO:input-output
```

一. 文件或文件夹操作

输入输出流的操作

File封装了文件和文件夹操作

1. 单个路径的分隔符

```
1 Windows:c:\\a\\b\\c.txt
```

```
1 Unix:/a/b/c.txt
```

```
1 System.out.println(File.separator);  
2 System.out.println(File.separatorChar);
```

2. 多个路径的分割符

```
1 Windows:c:\\a\\b\\c.txt;\\a\\b\\d.txt  
2 Unix:/a/b/c.txt:/a/b/c.txt  
3 System.out.println(File.pathSeparator);  
4 System.out.println(File.pathSeparatorChar);
```

二. 相对路径与绝对路径

绝对路径:从系统的根目录开始, 到某个文件的路径

```
1 比如:C:\\Users\\lanou\\Desktop\\上课代码  
2 JavaPro\\JavaPro07\\src\\com\\company\\Main.java
```

相对路径:从当前文件夹开始, 到某个文件夹的路径

```
1 比如:src/com/company/Main.java
```

```
1 比如:./src/com/company/Main.java
```

```
1 比如:../JavaPro07/src/com/company/Main.java  
2 .代表当前目录, ..代表上级目录
```

新建文件

```
1 File file = new File("src/main/java/com/company/a.txt");  
2 System.out.println(file.createNewFile());
```

新建文件夹

```
1 File dir = new File("src/main/java/com/company/b");
2 System.out.println(dir.mkdir());
```

新建文件夹

```
1 File dir1 = new File("src/main/java/com/company/c1/c2");
2 System.out.println(dir1.mkdirs());
```

注:mkdir支持单机目录, mkdirs支持多级目录

1. 判断文件或文件夹是否存在file.exists()
 2. 判断是否是文件夹file.isDirectory()
 3. 获取文件名file.getName()
 4. 获取文件绝对路径file.getAbsolutePath()
 5. 获取文件的相对路径file.getPath()
 6. 获取文件的大小(字节)file.length()
 7. 获取文件的最后修改时间(时间戳)
- ?时间戳:1970年1月1日到某个时间的(毫)秒数

获取当前的时间戳

```
System.out.println(System.currentTimeMillis());
```

```
file.lastModified()
```

8. 删除文件file.delete()

9. 获取文件夹的内容(遍历)

```
1 File dir3 = new File("src/com/company/abc");
2 File[] files = dir3.listFiles();
3 for (File f : files) {
4     System.out.println(f);
5 }
```

遍历文件夹中的所有文件

```
1 static void allFiles(File dir) {
2     //除文件夹
3     if (!dir.isDirectory()) {
4         return;
5     }
```

```

6  for (File f : dir.listFiles()) {
7  System.out.println(f);
8  if (f.isDirectory()) {
9  allFiles(f);
10 }
11 }
12 }
13 static void deleteDir(File dir){
14 for (File f : dir.listFiles()) {
15 if (f.isDirectory()){
16 deleteDir(f);
17 }else {
18 f.delete();
19 }
20 }
21 dir.delete();
22 }

```

listFiles:只能获取一层目录下的内容

三. 工程目录

```

1  src:源代码
2  -main:工程
3  -java:工程的源代码
4  -resources:配置文件
5  -test:测试工程
6  -java:测试工程的源代码
7  -resources:配置文件
8
9  pom.xml:依赖管理()
10 target:编译后的工程

```

四. IO流的分类

根据数据的流向, 分为:

输入流:从输入流中读数据, 以Reader或InputStream结尾的类

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)针对某个输入输出设备进行操作的流

2. 高级流:(高级流)对已有流进行封装,使其有新的功能readz字符

