

## | 学习目标

- \* 能够掌握IO流的概述

- \* InputStream , OutputStream ( I/O 流 )

- \* 字节流 , 字符流

- \* 能够掌握File对象的创建方式

- \* 三种 : path , ( parent ( String , File ) , path )

- \* 能够掌握File对象常用的方法

- \* getName , length , lastModified , getPath , getAbsolutePath, getCanonicalPath

- \*

exists, isFile, isDirectory, createNewFile, mkdir, mkdirs, delete, list, listFiles(FileFilter), listRoots

---

- \* 能够掌握IO流的概述

- \* 应用程序与设备之间的数据传输

- \* 键盘 : 输入数据

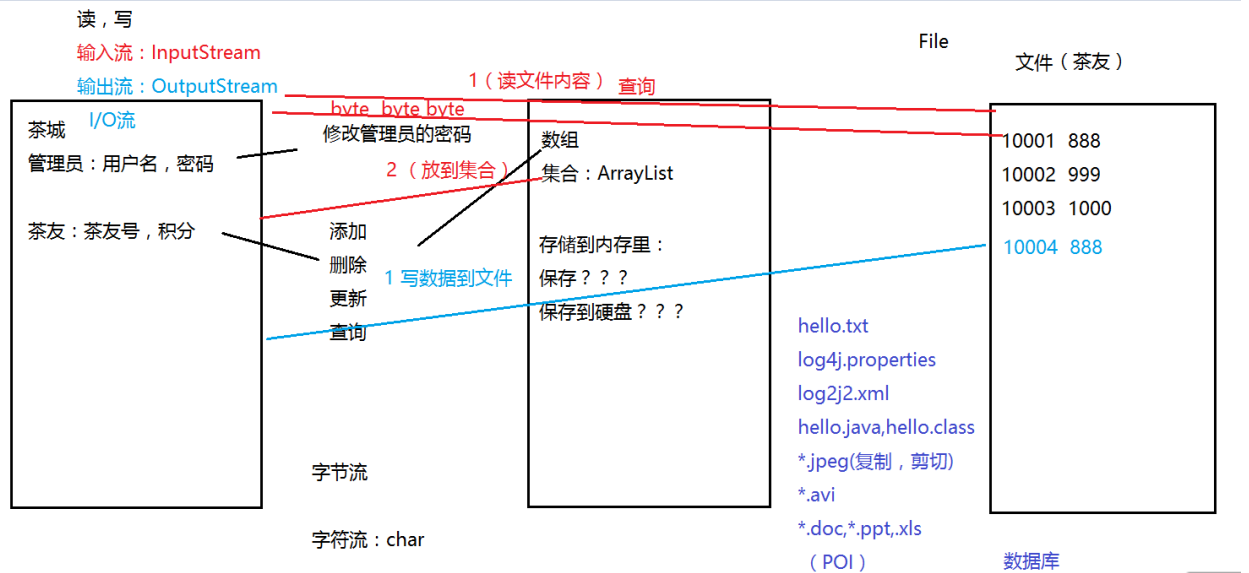
- \* 显示器 : 显示数据

- \* 在Java中 , 通过不同输入输出设备之间的数据传输抽象的表述为“流” , 程序允许通过流的方式与输入输出设备进行数据传输。

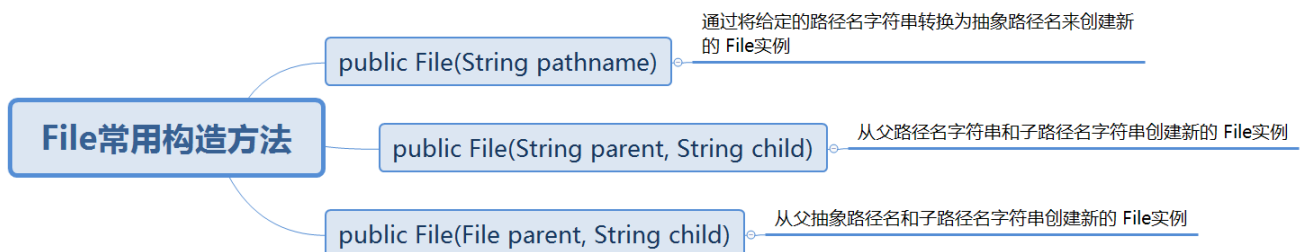
- \* 设备 : 键盘 , 内存 , 显示器 , 网络等

- \* 输入流和输出流是相对于内存设备而言的 , 将外设中的数据读取到内存中即输入 , 将内存的数据写入到外设中即输出。

- \* Java中的“流”主要位于java.io包中 , 称之为IO ( 输入输出 ) 流。输入也叫做读取数据 , 输出也叫做写出数据。



## \* 能够掌握File对象的创建方式



```

1 public static void main(String[] args) throws IOException {
2     // File:构建方式1
3     //E:\
4     // 转义字符
5     // 构建了File对象
6     File f1=new File("E:\\cc.txt");
7     //createNewFile();
8     f1.createNewFile();
9     // String str="\"";
10    // System.out.println(str);
11    // File:构建方式2
12    File f2=new File("E:\\bb","cc.txt");
13    // 确保parent路径是存在 否则创建会报出: java.io.IOException: 系统找不到指定
14    f2.createNewFile();
15
16    // File:构建方式3
17    File parent=new File("E:\\bb");

```

```
18     File f3=new File(parent,"dd.txt");
19     f3.createNewFile();
20     System.out.println("程序结束...");
21 }
22 * 温馨提示:
23     * 一个File对象代表硬盘中实际存在的一个文件或者目录。
```

\* 能够掌握File对象常用的方法



当且仅当具有该名称的文件尚不存在时，创建一个新的空文件

public boolean createNewFile()

删除由此File表示的文件或目录

public boolean delete()

创建由此File表示的目录

public boolean mkdir()

创建由此File表示的目录，包括任何必需但不存在的父目录

public boolean mkdirs()

返回一个String数组，表示该File目录中的所有子文件或目录

public String[] list()

返回一个File数组，表示该File目录中的所有的子文件或目录

public File[] listFiles()

返回一个File数组，表示根目录数组

public static File[] listRoots()

返回此抽象路径名表示的文件最后一次被修改的时间

public long lastModified()

## File常用的方法

1 \* 方法学习一：

```
2 public static void main(String[] args) throws IOException {
3     // 构建File文件对象
4     File f1=new File("E:\\bb.txt");
5     // 创建文件
6     f1.createNewFile();
7     // 获取文件的名字
8     String fileName=f1.getName();
9     System.out.println(fileName);
10    System.out.println("-----");
11    // 获取文件的长度
12    long length=f1.length();
13    System.out.println(length);
14
15    // 绝对路径和相对路径（开发中常用相对路径）
16    // 相对于当前项目创建
17    File f2=new File("cc.txt");
18    f2.createNewFile();
19    // . 代表当前项目下
20    File f3=new File("./dd.txt");
```

```

21     f3.createNewFile();
22     // .. 代表当前目录下上一层目录
23     File f4=new File("../ee.txt");
24     f4.createNewFile();
25
26     // getPath,getAbsolutePath,getCanonicalPath
27     printPath(f1);
28     System.out.println("-----");
29     printPath(f2);
30     System.out.println("-----");
31     printPath(f3);
32     System.out.println("-----");
33     printPath(f4);
34
35 }
36
37 public static void printPath(File f) throws IOException {
38     System.out.println("Path:"+f.getPath());
39     System.out.println("AbsolutePath:"+f.getAbsolutePath());
40     System.out.println("CanonicalPath:"+f.getCanonicalPath());
41 }

```

42 结果（文件创建结果：通过查看具体位置有没有）：

```

43 bb.txt
44 -----
45 0
46 Path:E:\bb.txt
47 AbsolutePath:E:\bb.txt
48 CanonicalPath:E:\bb.txt
49 -----
50 Path:cc.txt
51 AbsolutePath:E:\ws0722\26\cc.txt
52 CanonicalPath:E:\ws0722\26\cc.txt
53 -----
54 Path:..\dd.txt
55 AbsolutePath:E:\ws0722\26\..\dd.txt
56 CanonicalPath:E:\ws0722\26\dd.txt
57 -----
58 Path:..\..\ee.txt
59 AbsolutePath:E:\ws0722\26\..\..\ee.txt
60 CanonicalPath:E:\ws0722\ee.txt

```

```
61
62 * 学习方法二:
63     public static void main(String[] args) throws IOException {
64         File f1=new File("ff.txt");
65         //假如文件不存在我才创建
66         if(!f1.exists()) {
67             System.out.println("文件不存在, 创建文件");
68             f1.createNewFile();
69         }
70         System.out.println("f1 是否文件:"+f1.isFile());
71         System.out.println("f1 是否目录:"+f1.isDirectory());
72         System.out.println("-----");
73
74         // 在当前项目下创建目录
75         File d1=new File("test1");
76         d1.mkdir();
77         System.out.println("d1 是否文件:"+d1.isFile());
78         System.out.println("d1 是否目录:"+d1.isDirectory());
79
80         // 在当前项目创建多个层次的目录
81         File d2=new File("test/test2");
82         d2.mkdirs();
83
84         // d1.delete();
85         // d2.delete();
86
87         // delete方法, 如果此File表示目录, 则目录必须为空才能删除
88         File d3=new File("test");
89         d3.delete();
90
91         System.out.println("-----");
92         // 获得所有的盘符
93         File[] filePaths = File.listRoots();
94         for(File file:filePaths) {
95             System.out.println(file.getCanonicalPath());
96         }
97
98         //获得文件最近访问修改时间
99         long lastModified = f1.lastModified();
100        Date date=new Date(lastModified);
```

```

101         SimpleDateFormat sdf=new SimpleDateFormat("yyyy-MM-dd hh:mm:ss");
102         System.out.println(sdf.format(date));
103     }
104     结果:
105     f1 是否文件:true
106     f1 是否目录:false
107     -----
108     d1 是否文件:false
109     d1 是否目录:true
110     -----
111     C:\
112     D:\
113     E:\
114     F:\
115     G:\
116     2019-09-02 08:30:29
117
118     * 遍历单层目录
119     public static void main(String[] args) throws IOException {
120         File dir=new File("E:\\test");
121         if(dir.isDirectory()) {
122             String[] paths = dir.list();
123             for(String path:paths) {
124                 System.out.println(path);
125             }
126         }
127         System.out.println("-----");
128         if(dir.isDirectory()) {
129             File[] listFiles = dir.listFiles();
130             for(File file:listFiles) {
131                 System.out.println(file.getCanonicalPath());
132             }
133         }
134
135     }
136     结果:
137
138     * 遍历目录下所有目录和文件
139     public static void main(String[] args) throws IOException {
140         // 构建File 对象

```

```

141     File dir=new File("E:\\test");
142     // 调用打印的方法
143     printDir(dir);
144 }
145
146 public static void printDir(File dir) throws IOException {
147     if(dir==null) {
148         throw new IllegalArgumentException("dir is null...");
149     }
150     // 获取文件列表
151     File[] listFiles = dir.listFiles();
152     for(File file:listFiles) {
153         if(file.isFile()) {
154             // 假如是文件直接打印
155             System.out.println(file.getCanonicalPath());
156         }else {
157             // 假如是目录打印该目录，继续调用遍历的目录（递归）
158             System.out.println(file.getCanonicalPath());
159             printDir(file);
160         }
161     }
162 }
163 }
164

```

165 结果:

```

166 E:\test\test1
167 E:\test\test1\bbb.java
168 E:\test\test1\hello
169 E:\test\test1\hello\aaa.txt
170 E:\test\test2
171 E:\test\test2\ccc.txt
172 E:\test\test3
173 E:\test\test3\a
174 E:\test\test3\b
175 E:\test\test3\b\fff.txt
176 E:\test\test3\ddd.txt
177 * 可以调试查看理解调用栈的过程
178
179 * 搜索test目录找出txt文件
180 * 第一种写法

```



```
181 public static void main(String[] args) throws IOException {
182     // 构建File 对象
183     File dir = new File("E:\\test");
184     // 调用打印的方法
185     printDir(dir);
186 }
187
188 public static void printDir(File dir) throws IOException {
189     if (dir == null) {
190         throw new IllegalArgumentException("dir is null...");
191     }
192     // 获取文件列表
193     File[] listFiles = dir.listFiles();
194     for (File file : listFiles) {
195         if (file.isFile()) {
196             if (file.getName().endsWith(".txt")) {
197                 // 假如是文件直接打印
198                 System.out.println(file.getCanonicalPath());
199             }
200         } else {
201             printDir(file);
202         }
203     }
204 }
205
206 * 第二种写法
207 public static void main(String[] args) throws IOException {
208     // 构建File 对象
209     File dir = new File("E:\\test");
210     // 调用打印的方法
211     printDir(dir);
212 }
213
214 public static void printDir(File dir) throws IOException {
215     if (dir == null) {
216         throw new IllegalArgumentException("dir is null...");
217     }
218     // 获取文件列表
219     File[] listFiles = dir.listFiles(new FileFilter() {
220         //
```

```

221         @Override
222         public boolean accept(File pathname) {
223             return pathname.getName().endsWith(".txt") || pathname.isDirect
224         }
225     });
226
227     for (File file : listFiles) {
228         if(file.isFile()) {
229             System.out.println(file.getCanonicalPath());
230         }else {
231             printDir(file);
232         }
233     }
234 }
235 }
236 * 第三种写法: lamda表达式写法
237     public static void main(String[] args) throws IOException {
238         // 构建File 对象
239         File dir = new File("E:\\test");
240         // 调用打印的方法
241         printDir(dir);
242     }
243
244     public static void printDir(File dir) throws IOException {
245         if (dir == null) {
246             throw new IllegalArgumentException("dir is null...");
247         }
248         // 获取文件列表
249         File[] listFiles = dir.listFiles((pathname) -> {
250             return pathname.getName().endsWith(".txt") || pathname.isDirectory(
251         });
252
253         for (File file : listFiles) {
254             if (file.isFile()) {
255                 System.out.println(file.getCanonicalPath());
256             } else {
257                 printDir(file);
258             }
259         }
260     }

```

```
261
262 * 结果:
263 E:\test\test1\hello\aaa.txt
264 E:\test\test2\ccc.txt
265 E:\test\test3\b\fff.txt
266 E:\test\test3\ddd.txt
267
```