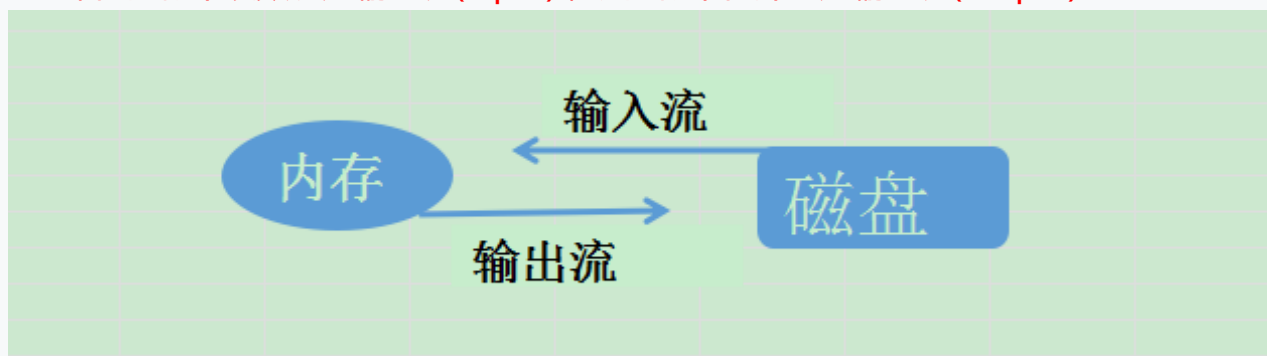


09 IO流_描述

JAVAAEE高级

一：IO流描述

- 1、Java中使用IO(输入输出)来读取和写入，读写设备上的数据、硬盘文件、内存、键盘.....
- 2、根据数据的走向可分为输入流和输出流，这个走向是以内存为基准的
- 3、即往内存中读数据是输入流(input)，从内存中往外写是输出流(output)



- 4、长用标准的流
System.in的类型是InputStream
System.out的类型是PrintStream，其是OutputStream的子类FilterOutputStream的子类

二：File 类

- 1、File 能新建、删除、重命名文件和目录
- 2、但 File 不能访问文件内容本身
- 3、如果需要访问文件内容本身，则需要使用输入/输出流
- 4、File对象可以作为参数传递给流的构造函数

```

1.  @Test
2.      public void getVoid() throws IOException {
3.          File dir1 = new File("D:\\IO\\dir1");
4.          if (!dir1.exists()) { // 如果D:/IO/dir1不存在, 就创建为目录
5.              dir1.mkdirs();
6.              System.out.println("。 。 。 。 。");
7.          }
8.          // 创建以dir1为父目录, 名为"dir2"的File对象
9.          File dir2 = new File(dir1, "dir2");
10.         if (!dir2.exists()) { // 如果还不存在, 就创建为目录
11.             dir2.mkdirs();
12.         }
13.         // 创建以dir2为父目录, 名为"test.txt"的File对象
14.         File file = new File(dir2, "test.txt");
15.         if (!file.exists()) { // 如果还不存在, 就创建为文件
16.             file.createNewFile();
17.         }
18.
19.     }

```

5、删除文件与文件夹

```

1.  @Test
2.      public void getVoid() throws IOException {
3.          //删除文件
4.          File file1 = new File("D:\\IO\\dir1\\dir2\\test.txt");
5.          System.out.println(file1.delete());
6.
7.          //如果删除一个文件夹, 那么文件夹必须是空的
8.          File file1 = new File("D:\\IO\\dir1\\dir2");
9.          System.out.println(file1.delete());
10.     }

```

6、修改文件的名称

注意:

如果路径名相同, 就是改名

如果路径名不同, 就是改名并剪切

```

1.  @Test

```

```

2.     public void getVoid() throws IOException {
3.         File file1 = new File("D:\\IO\\dir1\\修改前.txt");
4.         File file2 = new File("D:\\IO\\dir1\\修改后.txt");
5.         System.out.println(file1.renameTo(file2));
6.     }

```

7、判断

```

1.     @Test
2.     public void getVoid() throws IOException {
3.         File file1 = new File("D:\\IO\\dir2\\13");
4.         System.out.println(file1.isDirectory()); // 判断是否是文件夹
5.         file1 = new File("D:\\IO\\修改后.txt");
6.         System.out.println(file1.isFile()); // 判断是否是文件
7.         System.out.println(file1.isHidden()); // 判断是否是隐藏文件
8.     }

```

8、获取的方法

```

1.     @Test
2.     public void getVoid() throws IOException {
3.         File file1 = new File("D:\\IO");
4.         String[] arr = file1.list(); // 仅为了获取文件名
5.         for (String string : arr) {
6.             System.out.println(string);
7.         }
8.         File[] subFiles = file1.listFiles();
9.         for (File file : subFiles) { // 获取文件对象
10.            System.out.println(file);
11.        }
12.        System.out.println("绝对路径:"+file1.getAbsolutePath());
13.        System.out.println("构造方法中传入路径:"+file1.getPath());
14.        System.out.println("文件或者文件的名称:"+file1.getName());
15.        System.out.println("获取长度。字节数:"+file1.length());
16.
17.        SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");
18.        System.out.println("文件的最后修改时间:"+sdf.format(file1.lastModified()));
19.    }

```

三：案例<打印某个地址的.jar文件名>

1、从键盘输入接收一个文件夹路径,打印出该文件夹下所有的.jar文件名

```
1.  @Test
2.      public void getVoid() throws IOException {
3.          File dir = getDir();
4.          printJavaFile(dir);
5.      }
6.      /*
7.       * 获取键盘录入的文件夹路径
8.       * 1、返回值类型File
9.       */
10.     public static File getDir() {
11.         Scanner sc = new Scanner(System.in); // 创建键盘录入对象
12.         System.out.println("请输入一个文件夹路径");
13.         while (true) {
14.             String line = sc.nextLine(); // 将键盘录入的文件夹路径存储
15.             File dir = new File(line); // 封装成File对象
16.             if (!dir.exists()) {
17.                 System.out.println("您录入的文件夹路径不存在,请重新录入");
18.             } else if (dir.isFile()) {
19.                 System.out.println("您录入的是文件路径,请重新录入文件夹路
20. 径");
21.             } else {
22.                 return dir;
23.             }
24.         }
25.
26.         /*
27.          * 获取文件夹路径下的所有.jar文件
28.          */
29.         public static void printJavaFile(File dir) {
30.             // 1,获取到该文件夹路径下的所有的文件和文件夹,存储在File数组中
31.             File[] subFiles = dir.listFiles();
32.             // 2,遍历数组,对每一个文件或文件夹做判断
33.             for (File subFile : subFiles) {
34.                 // 3,如果是文件,并且后缀是.java的,就打印
35.                 if (subFile.isFile() && subFile.getName().endsWith(".jar"))
36.                 {
37.                     System.out.println(subFile);
38.                 }
39.             }
40.         }
41.     }
42. }
```

```
37.         // 4如果是文件夹,就递归调用
38.         } else if (subFile.isDirectory()) {
39.             printJavaFile(subFile);
40.         }
41.     }
42. }
```

四：流的分类

1、Java用于操作流的类都在IO包中

2、流按流向分为两种：输入流，输出流

3、流按操作类型分为两种

字节流：

字节流可以操作任何数据,因为在计算机中任何数据都是以字节的形式存储的

字节流的抽象父类：

InputStream

OutputStream

字符流：

字符流只能操作纯字符数据，比较方便

字符流的抽象父类：

Reader

Writer