

# File

---

## 一、创建文件对象

1.根据文件路径创建文件对象

```
public File(String pathname)
```

2.根据父路径和子路径名字创建文件对象

```
public File(String parent,String child)
```

3.根据父路径对应的文件对象和子路径名字创建文件对象

```
public File(File parent,String child)
```

## 二、判断文件类型

1.判断当前文件对象对应的路径是否存在

```
public boolean exists(File file)
```

2.判断当前文件对象指代的是否为文件

```
public boolean isFile(File file)
```

3.判断当前文件对象指代的是否为文件夹

```
public boolean isDirectory(File file)
```

## 三、获取文件信息

1.获取文件名称（含后缀名）

```
public String getName(File file)
```

2.获取文件的大小（字节数）

```
public long length(File file)
```

3.获取文件最后的修改时间

```
public long lastModified(File file)
```

4.获取创建文件对象时所使用的路径

```
public String getPath(File file)
public String getAbsolutePath(File file)//绝对路径
```

## 四、使用文件对象进行创建和删除文件（夹）

### 1. 创建一个新的空文件

```
public boolean createNewFile()
```

示例如下：

```
File f=new File("D:\\ZSH-Computer Science\\Java\\zsh1.txt");//此时还不存在zsh1.txt  
这个文件  
f.createNewFile();//调用完此方法后，会自动帮我们在指定路径处生成一个zsh1.txt的空文件  
//注意：若再次调用上述方法，会返回false，因为zsh1.txt这个文件已存在
```

### 2. 创建文件夹

```
public boolean mkdir();//只能创建一级文件夹  
public boolean mkdirs();//可以创建多级文件夹
```

### 3. 删除文件或者空文件夹，删除后的文件不会进入回收站，而是直接永久删除。

注意：无法删除非空文件夹。

```
public boolean delete()
```

## 五、遍历文件夹

### 1. 获取当前目录下所有“一级文件名称”，存入一个字符串数组中并返回

```
public String[] list()
```

示例如下：

```
File f=new File("D:\\ZSH-Computer Science");  
String[] fileNames=f.list();  
for(String fileName : fileNames){  
    System.out.println(fileName);  
}
```

### 2. (重点) 获取当前目录下所有“一级文件对象”，存入一个文件对象数组中并返回

```
public File[] listFiles()
```

示例如下：

```
File f=new File("D:\\ZSH-Computer Science");  
File[] files=f.listFiles();  
for(File file : files){  
    file.delete();  
}
```

使用 `listFiles` 方法时的注意事项如下：

- 当主调是文件，或者路径不存在时，返回null
- 当主调是空文件夹时，返回一个长度为0的数组
- 当主调是一个有内容的文件夹时，将里面所有一级文件和文件夹的路径放在File数组中返回
- 当主调是一个文件夹，且里面有隐藏文件时，将里面所有文件和文件夹的路径放在File数组中返回，包含隐藏文件
- 当主调是一个文件夹，但是没有权限访问该文件夹时，返回null

3.经典问题：在某个大文件夹（比如D盘）下搜索某个小文件

算法思想：方法递归

思路分析：

- (1) 使用 `listFiles` 方法找出大文件夹下的所有一级文件对象
- (2) 对所有一级文件对象进行遍历，判断是否为文件
- (3) 若是文件，则判断是否为目标文件
- (4) 若是文件夹，则需要进入该文件夹，并重复上述流程（递归）

代码实现如下：

```
package File;
import java.io.File;

public class FileSearch {
    public static void main(String[] args) {
        //需求：从D盘中找出myFirstFile.txt这个文件并输出其绝对路径
        searchFile(new File("D:/"), "myFirstFile.txt");
    }

    /**
     * 在某个大文件夹（比如D盘）下搜索某个小文件
     * @param dir      大文件夹
     * @param fileName 目标文件的名称
     */
    public static void searchFile(File dir, String fileName) {
        /*1.拦截非法情况*/
        if (dir == null || !dir.exists() || dir.isFile()) {
            return; //代表无法搜索
        }

        /*2.若dir是非空文件夹，执行以下核心逻辑*/

        //2.1 获取当前目录下的所有一级文件对象，并存入到文件对象数组files中
        File[] files = dir.listFiles();

        //2.2 判断当前目录下是否存在一级文件对象
        if (files != null && files.length > 0) {
            //2.3 遍历所有一级文件对象
            for (File f : files) {
```

```
//2.4.1 若f是文件，则判断其名称是否与目标文件的名称相一致
if (f.isFile()) {
    if (f.getName().contains(fileName)) {
        System.out.println("The file's pathname is " +
f.getAbsolutePath());
    }
}

//2.4.2 若f是文件夹，则继续重复上述流程（递归）
else{
    searchFile(f,fileName);
}
}
}
}
```