**Java中的目录**

创建目录：

File类中有两个方法可以用来创建文件夹：

mkdir( )方法创建一个文件夹，成功则返回true，失败则返回false。失败表明File对象指定的路径已经存在，或者由于整个路径还不存在，该文件夹不能被创建。

mkdirs()方法创建一个文件夹和它的所有父文件夹。

下面的例子创建 "/tmp/user/java/bin"文件夹：

CreateDir.java 文件代码：

import java.io.File;

public class CreateDir {

public static void main(String[] args) {

String dirname = "/tmp/user/java/bin";

File d = new File(dirname);

// 现在创建目录

d.mkdirs();

}

}

编译并执行上面代码来创建目录 "/tmp/user/java/bin"。

注意： Java 在 UNIX 和 Windows 自动按约定分辨文件路径分隔符。如果你在 Windows 版本的 Java 中使用分隔符 (/) ，路径依然能够被正确解析。

**读取目录**

一个目录其实就是一个 File 对象，它包含其他文件和文件夹。

如果创建一个 File 对象并且它是一个目录，那么调用 isDirectory() 方法会返回 true。

可以通过调用该对象上的 list() 方法，来提取它包含的文件和文件夹的列表。

下面展示的例子说明如何使用 list() 方法来检查一个文件夹中包含的内容：

DirList.java 文件代码：

import java.io.File;

public class DirList {

public static void main(String args[]) {

String dirname = "/tmp";

File f1 = new File(dirname);

if (f1.isDirectory()) {

System.out.println("目录 " + dirname);

String s[] = f1.list();

for (int i = 0; i < s.length; i++) {

File f = new File(dirname + "/" + s[i]);

if (f.isDirectory()) {

System.out.println(s[i] + " 是一个目录");

} else {

System.out.println(s[i] + " 是一个文件");

}

}

} else {

System.out.println(dirname + " 不是一个目录");

}

}

}

以上实例编译运行结果如下：

目录 /tmp

bin 是一个目录

lib 是一个目录

demo 是一个目录

test.txt 是一个文件

README 是一个文件

index.html 是一个文件

include 是一个目录