<http://blog.csdn.net/magic_jss/article/details/51472205>

Java提供File类，让我们对文件进行操作，简单整理了一下File类的用法。



1.基本概念

File：文件和目录路径名的抽象表示形式，代表文件或者文件夹。

2.构造方法

// 根据parent抽象路径名和child路径名字符串创建一个新File实例

File(File parent, String child)

// 通过将给定路径名字符串转换为抽象路径名来创建一个新File实例

File(String pathname)

// 根据parent路径名字符串和child路径名字符串创建一个新File实例

File(String parent, String child)

// 通过将给定的file:URI转换为一个抽象路径名来创建一个新的File实例

File(URI uri)

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8

3.常用方法

(1).创建功能

// 创建此抽象路径名指定的目录   
boolean mkdir()   
// 创建此抽象路径名指定的目录，包括所有必需但不存在的父目录   
boolean mkdirs()   
// 当且仅当不存在具有此抽象路径名指定名称的文件时，不可分地创建一个新的空文件   
boolean createNewFile()   
// 在默认临时文件目录中创建一个空文件，使用给定前缀和后缀生成其名称   
static File createTempFile(String prefix, String suffix)   
// 在指定目录中创建一个新的空文件，使用给定的前缀和后缀字符串生成其名称   
static File createTempFile(String prefix, String suffix, File directory)

// 构造方法一

File file1 = new File("f://file1");

// 构造方法二

File file2 = new File("f://file1", "file2");

// 构造方法三

File file3 = new File(file2, "file3.txt");

// 创建目录并返回是否创建成功,如果目录存在则返回false

boolean b1 = file1.mkdir();

System.out.println(b1);// true

// 创建目录

boolean b2 = file2.mkdir();

System.out.println(b2);// true

// 创建文件

// 在F盘下创建/file1/file2/file3.txt文件

boolean b3 = file3.createNewFile();

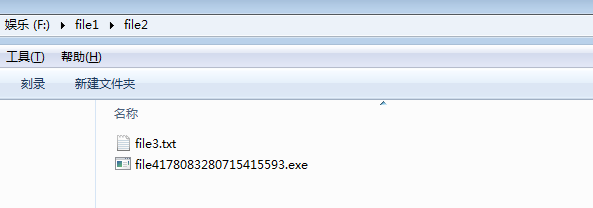
System.out.println(b3);// true

// 创建空文件并指定前缀和后缀

// 在F盘下创建/file1/file2/file4.....exe文件

File.createTempFile("file4", ".exe", file2);

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9
* 10
* 11
* 12
* 13
* 14
* 15
* 16
* 17
* 18
* 19
* 20
* 21



注：三种构造方法效果是等同的，没有本质区别；创建目录mkdir()、mkdirs()方法，mkdirs()方法创建目录时，如果待创建目录的上几级目录不存在则一并创建，mkdir()则只能创建单级目录。

(2).删除功能

// 删除此抽象路径名表示的文件或目录   
boolean delete()

//删除目录

System.out.println(file1.delete()); // false

//删除文件

System.out.println(file3.delete());// true

* 1
* 2
* 3
* 4

注：删除操作时，删除的是目录，则必须保证是空目录。

(3).判断功能

// 测试此抽象路径名表示的文件或目录是否存在   
boolean exists()   
// 测试此抽象路径名表示的文件是否是一个目录   
boolean isDirectory()   
// 测试此抽象路径名表示的文件是否是一个标准文件   
boolean isFile()   
// 测试此抽象路径名指定的文件是否是一个隐藏文件   
boolean isHidden()   
// 测试应用程序是否可以读取此抽象路径名表示的文件   
boolean canRead()   
// 测试应用程序是否可以修改此抽象路径名表示的文件   
boolean canWrite()

File file = new File("F://hello");

File file2 = new File(file, "blog.txt");

// 判断是否存在

if (!file.exists()) {

// 创建目录

file.mkdir();

}

if (file2.exists()) {

// 创建文件

file2.createNewFile();

}

// 判断是否是目录

System.out.println(file.isDirectory());// true

// 判断是否是文件

System.out.println(file.isFile());// false

System.out.println(file2.isDirectory());// false

System.out.println(file2.isFile());// true

System.out.println(file2.isHidden());// false

// 判断是否是隐藏的

System.out.println(file2.isHidden());// false

// 判断是否可读

System.out.println(file2.canRead());// true

// 判断是否可写

System.out.println(file2.canWrite());// true

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9
* 10
* 11
* 12
* 13
* 14
* 15
* 16
* 17
* 18
* 19
* 20
* 21
* 22
* 23
* 24
* 25
* 26

注：可以自主修改文件可读性，查看不同输出。

(4).获取功能

(1).基本获取功能

// 返回由此抽象路径名表示的文件或目录的名称   
String getName()   
// 返回此抽象路径名的绝对路径名形式   
File getAbsoluteFile()   
// 返回此抽象路径名的绝对路径名字符串   
String getAbsolutePath()   
// 将此抽象路径名转换为一个路径名字符串   
String getPath()   
// 返回此抽象路径名表示的文件最后一次被修改的时间   
long lastModified()

File file = new File("F://hello");

File file2 = new File(file, "blog.txt");

// 判断文件或目录是否存在

if (!file.exists()) {

// 不存在则创建

file.mkdir();

}

if (file2.exists()) {

// 创建文件

file2.createNewFile();

}

//获取文件名或者目录名

System.out.println(file2.getName());// blog.txt

//获取文件或目录的绝对路径

System.out.println(file2.getAbsolutePath());// F:\hello\blog.txt

//获取文件或目录的路径名（绝对路径或者相对路径）

System.out.println(file2.getPath());// F:\hello\blog.txt

//获取文件或目录修改的最后时间返回毫秒值

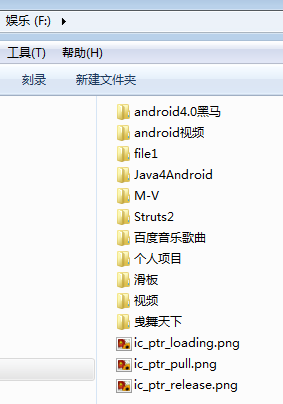
System.out.println(file2.lastModified());// 1463734158963

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9
* 10
* 11
* 12
* 13
* 14
* 15
* 16
* 17
* 18
* 19

(2).迭代获取功能，过滤器功能

// 返回一个字符串数组，这些字符串指定此抽象路径名表示的目录中的文件和目录   
String[] list()   
// 返回一个字符串数组，这些字符串指定此抽象路径名表示的目录中满足指定过滤器的文件和目录   
String[] list(FilenameFilter filter)   
// 返回一个抽象路径名数组，这些路径名表示此抽象路径名表示的目录中的文件   
File[] listFiles()   
// 返回抽象路径名数组，这些路径名表示此抽象路径名表示的目录中满足指定过滤器的文件和目录   
File[] listFiles(FileFilter filter)   
// 返回抽象路径名数组，这些路径名表示此抽象路径名表示的目录中满足指定过滤器的文件和目录   
File[] listFiles(FilenameFilter filter)

图示：先看下F盘里面的东东



File file = new File("F://");

// 获取抽象路径名下的文件和目录

String[] s = file.list();

// 过滤文件或者目录名

String[] ss = file.list(new FilenameFilter() {

public boolean accept(File dir, String name) {

// 返回以.png结尾的文件或者目录名

// 控制返回值判断是否添加到数组中

return name.endsWith(".png");

}

});

// 增强for输出

for (String string : s) {

System.out.print(string + " ");// $RECYCLE.BIN android4.0黑马 android视频......等等

}

// 增强for

for (String string : ss) {

System.out.print(string + " ");// ic\_ptr\_loading.png ic\_ptr\_pull.png ic\_ptr\_release.png

}

// 获取抽象路径名下的文件和目录对象

File[] files = file.listFiles();

// 获取抽象路径名下的文件和目录对象，添加文件过滤

File[] files2 = file.listFiles(new FileFilter() {

public boolean accept(File pathname) {

// 判断是否是隐藏目录

return (pathname.isDirectory()&&pathname.isHidden());

}

});

// 获取抽象路径名下的文件和目录对象,添加文件名过滤

File[] files3 = file.listFiles(new FilenameFilter() {

public boolean accept(File dir, String name) {

// 判断是否是以png结尾的文件

return (new File(dir, name).isFile())&&name.endsWith(".png");

}

});

for (File f : files) {

System.out.print(f.getName()+" ");//$RECYCLE.BIN android4.0黑马 android视频......等等

System.out.println();

for (File f : files2) {

System.out.print(f.getName()+" ");//$RECYCLE.BIN ghos

}

System.out.println();

for (File f : files3) {

System.out.print(f.getName());//ic\_ptr\_loading.pngic\_ptr\_pull.pngic\_ptr\_release.png

}

(5).重命名功能

// 重新命名此抽象路径名表示的文件   
boolean renameTo(File dest)

// 重新命名此抽象路径名表示的文件

// boolean renameTo(File dest)

File file = new File("f://");

File[] files = file.listFiles();

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

File f = files[i];

//判断是否是以.png结尾的文件

if (f.isFile() && f.getName().endsWith(".png")) {

// 更改文件名，renameTo()接收File对象 ，这里该对象并未指定盘符

boolean b = f.renameTo(new File("pic" + i + ".png"));

System.out.println(b);

// true

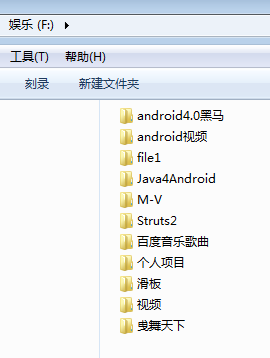
// true

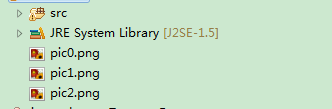
// true

}

}

图示：





注：更改文件或目录名时，renameTo()方法参数对象如果不指定盘的话，默认会将文件剪切到项目目录下（由上面截图可以看到）；指定盘符的话会根据指定位置剪切到该位置。renameTo()方法相当于剪切加重命名。

注：更多方法查看API

版权声明：本文为博主原创文章，未经博主允许不得转载