复制文件及文件夹

1. 创建一个**FileUtils类**：方法copyFIle和copyDir。

复制文件夹显然需要利用**递归**的方法。

示例：

**public** **class** FileUtils {

**public** **static** **void** copyFile(File sourceFile, File destFile) {

InputStream is = **null**;

OutputStream os = **null**;

**if** (destFile.exists()) {// 文件已经存在

destFile = **new** File(destFile.getAbsoluteFile() + "(copy)");

**try** {

destFile.createNewFile();

} **catch** (IOException e) {

}

}

**try** {

is = **new** FileInputStream(sourceFile);

os = **new** FileOutputStream(destFile);

**byte**[] buff = **new** **byte**[1024];

**int** len = 0;

**while** (-1 != (len = is.read(buff))) {

os.write(buff, 0, len);

}

} **catch** (FileNotFoundException e) {

e.printStackTrace();

} **catch** (IOException e) {

e.printStackTrace();

} **finally** {

**try** {

os.flush();

is.close();

os.close();

} **catch** (IOException e) {

e.printStackTrace();

}

}

}

**public** **static** **void** copyFile(String sourcePath, String destPath) {

File sourceFile = **new** File(sourcePath);

File destFile = **new** File(destPath);

*copyFile*(sourceFile, destFile);

}

**public** **static** **void** copyDir(String sourcePath, String destPath) {

File sourceDir = **new** File(sourcePath);

File destDir = **new** File(destPath);

*copyDir*(sourceDir, destDir);

}

**public** **static** **void** copyDir(File sourceDir, File destDir) {

**if** (**null** == sourceDir) {

**return**;

}

**if** (sourceDir.isFile()) {

File newFile = **new** File(destDir, sourceDir.getName());

*copyFile*(sourceDir, newFile);

}

**else** **if** (sourceDir.isDirectory()) {

File newDir = **new** File(destDir, sourceDir.getName());

newDir.mkdir();

File[] dirs = sourceDir.listFiles();

**for** (File dir : dirs) {

*copyDir*(dir, newDir);

}

}}}

测试：

String sourcePath = "e:\\成绩";

System.***out***.println(sourceFile.exists());

File sourceDir = **new** File(sourcePath);

String destPath = "e:\\copy";

File destDir = **new** File(destPath);

FileUtils.*copyDir*(sourceDir, destDir);

FileUtils.*copyDir*(sourcePath, destPath);

1. 复制文件夹时，需要注意一个bug：

不允许拷贝到自己的子文件夹中去，否则会出现死循环，无限地复制，因为一直往自己的文件夹中复制东西，而复制的又是自己。

解决办法很简单：禁止复制到自己的子文件夹中。

在判断文件夹中，添加判断是否是自己的源文件夹的子文件夹即可。

**public** **static** **void** copyDir(File sourceDir, File destDir) {

**if** (**null** == sourceDir) {

**return**;

}

**if** (sourceDir.isFile()) {

File newFile = **new** File(destDir, sourceDir.getName());

*copyFile*(sourceDir, newFile);

}

**else** **if** (sourceDir.isDirectory()) {

File newDir = **new** File(destDir, sourceDir.getName());

***if(newDir.getAbsolutePath().contains(sourceDir.getAbsolutePath())){***

***throw new RuntimeException("不能拷贝到子文件夹中");***

***}***

newDir.mkdir();

File[] dirs = sourceDir.listFiles();

**for** (File dir : dirs) {

*copyDir*(dir, newDir);

}

}

}

1. 一个文件夹的体系（即内部文件夹层数太多），利用windows命令删除不了可以利用Java编程来解决。

**public** **static** **void** deleteFile(File desti) {

**if**(desti == **null**) {

**return**;

}

**if**(desti.isFile()) {

desti.delete();

**return**;

}**else** {

File[] files = desti.listFiles();

**for**(File file : files) {

*deleteFile*(file);

}

desti.delete();

}}

也可以利用软件工具：

robocopy.exe即可。

设置环境变量后，然后在命令行输入命令

Robocopy /MIR e:\\voiddir e:\\longlongdir

功能是把前一个文件夹复制到后一个文件夹中，复制一个空的就相当于删除。