

Java 程序说明文档

1. 程序功能说明

本次作业实现了一个文件系统监控程序，针对给定监控范围内的监控对象，以扫描方式探查监控对象相关属性的变化，从而触发规定的处理动作。

2. 程序运行所需环境和运行指令规范

java 环境：

jdk1.8.0_162 或 jdk-10

运行的 IDE：

Eclipse IDE for Java Developers

Version: Oxygen.2 Release (4.7.2)

Build id: 20171218-0600

在 eclipse 中导入 project 后直接运行 Main.java 文件即可，输入的字符串需要满足正确的格式否则会输出无效提示信息。

3. 合法输入说明

监控作业的输入格式为：

IF [path] renamed THEN record-summary

输入字符均为英文状态下，单词之间有一个空格，开头结尾无空格，方括号内为合法的文件或文件夹路径，path 必须为监控目标的绝对路径，路径格式以运行平台为准，path 与方括号之间无空格，renamed 可替换为 modified，path-changed，size-changed，所有输入除路径外大小写敏感，modified 必须小写；record-summary 可替换为 record-detail，recover。recover 不能搭配 modified、size-changed。

程序运行所需的输入监控任务由控制台输入。

本程序最多允许输入 500 行（包括有效请求，“END”和空行等其他不合法的请求），输入行数不足 500 行时程序会等待输入“END”来开始调度，输入行数等于 500 后程序将会关闭 System.in 不再读取输入，并直接开始监控线程，不再等待“END”。

一行只能输入 1 个监控任务。

本程序每行最多允许输入 10 个监控对象，达到 10 个监控对象时如果输入了新的监控对象，将不会把新的监控对象加入检测，并在控制台给出提示。

对于直接输入 END，程序不会输出任何信息。

对于同质任务请求，程序在输入后马上会进行判断，判断为同质请求进行输出。

所有的输入中都不可以含有多余的空格（包括 END）。

对于不合规的输入，程序在输入后会立即给出提示信息。

4. 程序输出格式说明

输出文件为"summary.txt"与"detail.txt"，位于程序的 project 所在的文件夹下。

summary.txt 中记录所有任务是"record-summary"的监控中 4 类触发器被触发的次数。每次触发时（即有次数变化时）都会追加写入，每次写入都会写入 4 种触发器的触发次数，每次程序重新运行时会将文件清空。

```
renamed: 0  
modified: 1  
path-changed: 0  
path-changed: 0  
-----  
renamed: 0  
modified: 1  
path-changed: 1  
path-changed: 0  
-----  
renamed: 0  
modified: 1  
path-changed: 2  
path-changed: 0  
-----
```

detail.txt 中记录所有任务是"record-detail"的监控，监控器被触发时的详细信息。每次触发时追加写入，每次写入都会写入文件名、修改时间、路径、文件大小四项信息。每次程序重新运行时将文件清空。

```
Trigger: modified
name: b.txt -> b.txt
path: D:\test -> D:\test
size: 28 -> 28
lastModify: 1524044798572 -> 1524044894731
-----
Trigger: path-changed
name: b.txt -> b.txt
path: D:\test -> D:\test\aaa
size: 28 -> 28
lastModify: 1524044894731 -> 1524044894731
-----
Trigger: path-changed
name: b.txt -> b.txt
path: D:\test\aaa -> D:\test
size: 28 -> 28
lastModify: 1524044894731 -> 1524044894731
-----
```

在控制台中，会输出格式无效提示信息，相同输入提示信息等，以及程序运行过程中的各种状态信息如触发器被触发的信息，以及监控对象消失提示信息等，作为对程序运行状况的辅助判定，因指导书并未对控制台输出作任何规定，具体请以实际输出为准，bug 判断依据请以实际文件输出为准。

5. SafeFile 类及测试线程补充说明

本程序已经包含了测试线程的模板，已把经测试线程在主线程中的启动顺序设为输入结束之后，请不要对测试线程的启动顺序作任何更改，测试样例的编写也应该严格依照模板，不能删除 sleep 语句。

编写测试样例时请使用 SafeFile 类中的 rename, move, createNewFile, mkdir, makedirs, delete, setLastModified, write 这几个静态方法来对监控对象作出更改，具体使用请参考程序源码及注释。例如：

```
SafeFile.move("D:\\test\\b.txt", "D:\\test\\aaa\\b.txt");
SafeFile.rename("D:\\test\\a.txt", "c.txt");//第二个参数不包括路径，仅包括文件名
SafeFile.write("D:\\test\\b.txt", "wewer");//第二个参数为要写入的内容字符串
SafeFile.createNewFile("D:\\test\\b.txt");
SafeFile.mkdir("D:\\test\\");
SafeFile.makedirs("D:\\test\\");
SafeFile.delete("D:\\test\\a.txt");
```

注意所输入的路径均为绝对路径，路径格式以运行平台为准，在 Windows 下路径分隔符请使用\\。

6. 补充说明

- (1) 测试样例执行结束结束，程序不会停止，需要手动关闭（可通过控制台）；
- (2) 监控对象消失后会在控制台提示并结束线程；
- (3) 强制要求不能对现有的目录作任何改动或删除，否则结果将不可预测；
- (4) **END** 必须作为单独一行输入，分散在多行，或与其他请求混合在同一行，则视为无效输入。
- (5) 对于文件夹的监控视为对文件夹内所有子文件和子目录下文件的监控，对四类触发器都是如此。
- (6) 监控对象不存在，或输入路径非绝对路径等不合法的输入不算在 10 个有效监控对象内的。
- (7) 输入路径在 Windows 下大小写不敏感，请不要将此作为测试点。
- (8) **recover** 操作对同一监控对象的其他触发器可能会产生影响，与多线程的运行机制有关，具有不确定性，具体以实际输出为准。
- (9) 进行重命名，新建文件等操作时输入的新文件名必须在 window 下有效，否则重命名可能不会成功。
- (10) 在某些边界条件下，多线程的运行机制具有一定的不确定性，运行的结果也不能百分之百地准确预测，所以对程序的功能进行测试时请尽量避免可能出现不同结果的边界情况。
- (11) 程序运行时 CPU 占用率较高，运行程序前请确保程序上一次运行的进程已经结束（可通过任务管理器查看），否则可能会对新的测试产生影响。
- (12) 仅允许输入英文状态下的字符。
- (13) 初始时监控范围内所有的文件数量不超过 50 个，且目录层次不能太深。
- (14) 对相同文件或文件副本的多次操作请保证在不同的扫描周期（1s 左右）内进行，否则结果将不可预测。
- (15) 对于程序运行结果出现不确定的情况，请多运行几遍观察，以多数次的结果为准。