

## 文件监控器说明文档

又是一个个的不眠夜,亲爱的你是不是写 OO 写到吐血了呢? 虽然大家都很累但是要害进行互测, 辛苦你了, 请认真阅读一下有关输入格式以及一些乱七八糟现象的解释。( 谁让助教说可以 readme 呢? ) 每个人的程序都千差万别, 希望你能耐住性子测试我的程序, 谢谢~

### 程序运行要求:

拿到我的程序后你应该能看到所有的类, 其中 `TestThread` 类为你需要编写的类, 能够用到的方法都在 `run` 方法的上方, 在后面的文档会进行详细解释。编写完 `TestThread` 后请运行 `Main` 文件, 其中 `TestThread` 线程我已经在合适位置为你创建好了( 你总不能把测试线程先于监控线程开启吧? 主要保证一切准备妥当后开始文件操作 ), 在控制台输入 `IFTTT` 格式的字符串若干, 具体格式在输入部分要求, 然后以 “END” 结束输入后就会开启所有监控线程并自动开始测试线程, 待测试线程执行完所有文件操作后, 你就可以在指定目录看到 `Summary.txt` 文件和 `Detail.txt` 文件来获取监控结果了。

### IFTTT 输入部分:

由于这次要求不允许测格式问题, 所以对你的输入要求比较高。首先 `IFTTT` 方面需要输入符合以下要求的字符串: “IF Path Triger THEN Task” 以空格分割共五个部分, 要求除中间这四个分割空格之外不能有任何空格, 包括文件夹或文件的名字中也不允许有空格。Path 为监控的绝对路径, 要求真实存在, 否则会报错。Triger 为一个字符串, 内容为 `renamed/modified/path-changed/size-changed` 中的任意一个。Task 为一个字符串, 内容为 `record-summary/record-detail/recover` 中的任意一个, 其余

的 IF 和 THEN 则是强制规定，所有的字母大小写敏感。搭配上倘若 Task 为 recover 时，Triger 只能为 renamed 或 path-changed，否则会报错。可能的合法输入如下：

```
IF D:\Monitor\test.txt path-changed THEN recover
```

一行只能输入一条 IFTTT，以回车分割，实质相同的 IFTTT 会输出 #Same 于控制台，监控路径不存在的 IFTTT 会输出 Invalid Path 并且不开启监控线程，所以请首先在操作系统把现场布置好，保证文件或文件夹的存在，但是程序一旦开始，便不可以通过操作系统改变文件，只能运行编写好的测试线程。

本程序要求构建不超过 10 个对象的监控线程，超过部分的监控并不会被成功构造。当所有的 IFTTT 都输入完毕后，输入“END”便可以结束 IFTTT 输入线程，开启所有监控线程和后续的测试线程。

测试线程部分：

测试线程设计了部分文件操作方法，包括 mkdir 新建目录，addFile 新建文件，rename 重命名，delete 删除文件，move 移动文件，changSize 改变文件大小（向指定文件追加五字节），changTime 改变文件最后修改时间，每个方法的参数均在上方有定义。根据指导书，对已有的目录进行重命名、删除等操作是禁止的。

在文件操作过程中倘若出现错误，例如对不存在的文件重命名，会使测试线程输出提示信息并退出，所以请一定确定编写测试样例的合法合理。Testcase 方法中我已经为你注释掉了一个可能的测试样例格式，可以参考。由于可以规定相邻两条操作之间的间隔，这里要求至少等待五秒，即使用注释掉的

```
try {Thread.sleep(5000);} catch (InterruptedException e) {return false;}
```

在测试线程结束后，线程等待 10 秒待相应文件写入操作完全完成后，程序自动退出。

## 输出部分

倘若你的电脑有 D 盘，那么所有的 summary 和 detail 均会输出到 D:\Detail.txt 和 D:\Summary.txt 两个文件中，如果你没有 D 盘，请把 OutputDetail 和 OutputSummary 中的文件路径修改为不在监控范围内的一个合理的位置。

有关 record-summary，其会记录相应触发器的实际触发次数总和，例如监控 a 文件的 renamed 和 b 文件的 renamed 进行 record-summary，且都触发了一次，那么文件中 renamed 触发器触发次数为 2。倘若监控一个文件夹，那么其中任意文件的触发均会使相应触发器触发次数+1。可能的文件内容如下：

renamed:1

modified:2

path-changed:3

size-changed:4

有关 record-detail，每当其对应的触发器触发一次，便会记录下该文件的信息变化，并且以追加的形式输出于 Detail.txt，以空行分隔每一次记录，可能的输出如下：

D:\Monitor\test.txt path-changed

D:\Monitor\test.txt->D:\Monitor\222\test.txt

test.txt->test.txt

8->8

1524011978951->1524011978951

分别输出文件绝对路径 对应触发器

旧绝对路径->新绝对路径

旧文件名->新文件名

旧大小->新大小

旧修改时间->新修改时间

由于要求 record-detail 输出全部信息变化，所以有可能一些信息是没有变化的。

监控线程部分：

倘若监控对象是一个目录，则该线程永远不会结束( 除非测试线程结束导致的程序退出 )。

倘若监控对象是一个文件，那么当该文件丢失时，线程直接结束且不会再度开启。

各触发器对于丢失的定义为在新的快照里找不到监控对象，也找不到满足触发的文件。

文件的增删操作不会触发触发器，仅会使某些监控线程监控对象丢失从而导致线程结束。

线程调度的特殊现象解释：

这个部分希望你仔细阅读，因为很多多线程的情况是不可预知的，可能会产生不同的现象，但是都是合理说得通的，向助教咨询后得到答复为只要是合理说得通的不算 bug。

以下为你举几个例子。并且希望在你的测试过程中，尽量减少容易导致不可预料结果的操作，或者在得到不同现象时思考一下是否正常，以避免错误申报 bug。

首先为你解释一下我的监控策略：输入监控结束后，为所有合理( 路径存在，搭配合理 ) 的监控任务建立线程，即每个 IFTTT 一个线程。线程内部存在两个 snapshot 快照，每隔一段时间进行 snapshot 快照更新，即把新的快照当作旧的快照，并且更新新的 snapshot 快照。这两个快照对应本次扫描前后的监控范围情况。扫描后针对监控对象( 单个文件或文件夹中所有文件 ) 利用两个快照进行 trigger 的判断，如果触发则根据 task 进行相应操作。一切结束后线程暂停一段时间，进行新一轮的快照更新和判断。

以下为一些可能的例子：

IF D:\Monitor\test.txt path-changed THEN recover

IF D:\Monitor path-changed THEN record-detail

对 test.txt 操作进行路径改变，当第一条监控监控到触发时进行 recover，那么对于第二条监控，获取快照有两个情况：

- 1、 该文件已经 recover，则相当于没有扫描到变化，不进行任何 record-detail
- 2、 该文件还没有 recover，触发，进行 record-detail。在下一次扫描时发现文件 recover，再次触发，进行 record-detail

这两种情况都是合理的。

IF D:\Monitor renamed THEN recover

IF D:\Monitor\test.txt modified THEN record-detail

对 test.txt 文件进行改名操作，当第一条监控触发时进行 recover，那么对于第二条监控，获取快照有两个情况：

- 1、 该文件已经 recover，则相当于没有扫描到变化，监控对象没有变化，不进行任何 record-detail
- 2、 该文件还没有 recover，第二条监控丢失监控对象，线程结束

这两种情况都是合理的

IF D:\Monitor\test.txt path-changed THEN record-detail

IF D:\Monitor path-changed THEN recover

对 test.txt 操作，向内部文件夹移动进行路径改变，例如移动到 D:\Monitor\test\test.txt，当第二条监控监控到触发时进行 recover，那么对于第二条监控，获取快照有两个情况：

- 1、该文件没有 recover，触发，进行 record-detail，且根据指导书要求继续追踪移动后的文件，监控范围缩小，recover 之后获取下一次快照，此时该监控对象返回了上层目录，则该监控线程丢失了监控对象，线程结束。
- 2、该文件已经 recover，则相当于没有扫描到变化，监控对象没有变化，不进行任何 record-detail

这两种情况都是合理的

由于 recover 的存在，与以上情况类似的情况还有很多，包括 record-summary 的计数为 0 或 2，监控线程中监控对象的丢失等都与多线程的随机性有关，请测试者比对结果的时候一定确认真实的扫描情况，结合控制台输出进行比对。

其余未提到的各种要求均符合指导书以及 issue 里的各种明确正确的解释。

望手下留情