**面向对象第六次课程作业指导说明**

**一、程序功能**

如指导书中所说，该程序可以实现对中小规模文件进行简单监控操作的功能。

**二、程序运行所需环境和使用规范**

运行环境：与课程提供的JDK和Eclipse相同即可。

运行步骤：将全部.java导入工程文件，从属fileScan包，该工程文件中需要有JRE System Library，之后运行run Begin文件，即可在控制台输入IF THEN信息。

**三、正确输入格式和使用操作说明**

**1. 控制台输入：**

IF THEN语句的使用，要求需满足如下格式，”IF XXX trigger\_type THEN task\_type”。

其中trigger\_type有如下四种，renamed，modified，pathchanged，sizechanged，需要完全匹配；task\_type有如下三种，summary，detail，recover,同样需要完全匹配。

在上述格式中，每两个部分之间有且仅有一个空格，同时，限制文件或目录中不能出现空格字符，如果有会显示Not Match字样提示。

对于系统内要求的不能作为文件名的字符，输入中未作要求，但是由于使用testCase或者提前布置现场的方法均无法创建，所以实际上属于不会起作用的无效输入。同时由于不同操作系统对文件名长度限制有一定差异，限制文件名不超过20个字符。

**2.TestCase制造与使用**

应指导书要求，除了提供testThread类可以供测试者进行修改，和main中testThread.start()的位置可在一定范围内移动以外，其他程序部分按理说不能修改，因为修改后无法保证程序的正确性，所以不建议改动。

提供的testThread类有如下几种方法：

Boolean addFile(String file\_name);

Boolean makeDirectory(String file\_name);

Boolean rename(String from, String to);

Boolean deleteFile(String file\_name);

Boolean move(String from, String to);

Boolean changeSize(String file\_name);

Boolean changeTime(String file\_name);

其中，addFile的参数file\_name为文件的绝对路径，如addFile(“H:\\1\\1.txt”)，表示增加所在目录为H:\1的1.txt文件，由于JAVA转义字符的影响，字符串内应当以\\来表示文件的层级关系。

MakeDirectory（“H:\\1\\2”），表示新建一个所在目录为H:\1的文件夹2.

Rename（”H:\\1.txt”, “H:\\2.txt”），表示将目录为H:\的1.txt文件改名为2.txt，使用时需注意保证前后两者的所在目录相同，以满足rename的含义。

DeleteFile（”H:\\1.txt”），表示将目录为H:\的1.txt文件删除，文件删除的结果不会对监控产生作用。

Move（”H:\\1.txt”, “H:\\1\\1.txt”），表示将目录为H:\的1.txt文件移动至目录为H:\1的地方，使用时需注意保证前后两者的名称相同，以满足move的含义。

ChangeSize（“H:\\1.txt”），表示修改目录为H:\的1.txt文件大小和上一次修改时间。

ChangeTime（”H:\\1.txt”），表示修改目录为H:\的1.txt文件上一次修改时间。

上述文件均可以自由组合，同时建议使用Thread.Sleep()保证不同操作之间有一定的间隔，以防止扫描结束周期内多次操作导致的结果合并缺失，示例如下。

1. **public** **boolean** testcase(){
2. **int** i;
3. addFile("H:\\1.txt");
4. **for**(i= 0; i < 10; i++){
5. **try** {
6. Thread.sleep(300);
7. } **catch** (InterruptedException e) {
8. // TODO Auto-generated catch block
9. e.printStackTrace();
10. }
11. **if**(!changeSize("H:\\3.txt"))
12. **return** **false**;
13. }
14. **if**(!rename("H:\\1.txt", "H:\\2.txt"))
15. **return** **false**;
16. **try** {
17. Thread.sleep(600);
18. } **catch** (InterruptedException e) {
19. // TODO Auto-generated catch block
20. e.printStackTrace();
21. }
22. **if**(!deleteFile("H:\\2.txt"))
23. **return** **false**;
25. **try** {
26. Thread.sleep(1000);
27. } **catch** (InterruptedException e) {
28. // TODO Auto-generated catch block
29. e.printStackTrace();
30. }

33. **return** **true**;
34. }

一切测试请在testcase（）中完成，具体使用方式与课上讲义相同。

注意，run（）方法请不要修改。

1. **public** **void** run(){
2. **if**(!testcase()){
3. System.out.println("File Operation Error.");
4. }
5. **else**{
6. System.out.println("File Operation Over.");
7. }
8. file\_operation\_over = **true**;
9. }

当testcase（）中有某一步操作失败后，程序将终止测试样例，可能的失败比如add一个已有的File，delete一个没有的文件等不合理的操作。

**3.输出结果的说明**

运行后会在程序所在的目录下生成两个文件，detail.txt和summary.txt，其中detail存放所有应该被记录的文件修改细节，包括大小、名称、路径和修改时间，summary中存放某一类触发器被触发，且要求执行summary记录的次数，包括触发器类型和被要求记录的次数。

比如，

**IF H:\1.txt modified THEN detail**

**IF H:\1.txt modified THEN summary**

**IF H:\ modified THEN detail**

**END**

上述对1.txt进行changeTime操作，会被记录两次detail和一次summary；

**IF H:\1.txt modified THEN detail**

**IF H:\1.txt modified THEN summary**

**IF H:\ modified THEN summary**

**END**

而上述对1.txt进行changeTime操作，会被记录一次detail和两次summary。

**四、特殊情况和边界可能**

1.多个recover只会触发一次

比如：

IF H:\ renamed THEN recover

IF H:\1.txt renamed THEN recover

当对1.txt文件进行rename操作时，屏幕提示会出现两次，但是实际只会执行一次，保证Rename操作被取消。

2.对于多次rename操作

比如：

IF H:\1.txt renamed THEN summary

rename("H:\\1.txt", "H:\\2.txt")) -- A

rename("H:\\2.txt", "H:\\3.txt")) -- B

若AB操作之间的空隙不够，在两次扫描中完成了AB两个操作，那么输出将只有1.txt -> 3.txt；若A、B操作之间的空隙足够长，跨过一次扫描，那么将有两个输出1.txt -> 2.txt 与 2.txt -> 3.txt。该条说明对Detail和Summary均有效。

3.由于指导书说明极度不明确，截止到作业提交结束前，都未搞清楚监控对象不超过10个是什么意思，所以限制有效且不重复的IF THEN语句120个，超出范围会提示Monitors Number Over 120!，剩下的怎么都好TAT。

4.关于程序的扫描周期，小数据量的时候扫描周期为1S，基本可以保证准确，但是当监控对象层次结果过深时，会出现扫描时间极长的可能，所以应当避免结构过深可能导致的栈溢出与等待时间延长。

5.若由于监控的文件过多，在扫描时间内未能完成对监控对象的遍历，而此时程序会认为该段时间内扫描范围中没有发生任何变化。若此时发生文件变动，很有可能不能被记录，比如监控自己堆满杂物的系统盘等。可以通过增加testThread的线程时间来解决，或者其实小一点的数据量就能看出来程序有没有问题了。

6.对于名称不同、其他均相同的文件，不应在同一个扫描周期内同时操作，否则对于rename与changePath的可能出现AB均变为C，而D文件被忽略的情况。

7.对于testThread.start()测试线程开始的位置，在Begin类中main函数里有一小块地方使用。

