OO 第六次作业指导书

一、 前言

- **作业目标**

本次作业的目标是训练同学们针对线程安全问题，如何平衡线程访问控制和共享对象之间的矛盾。

作业内容是实现一个监控程序，随时监控文件系统中文件状态的改变，并对这些改变做出相应的响应。

如：一个计算机，有多个用户在操作（增、删、改）各种文件和目录结构，其中一个用户在使用文件浏览器显示文件目录和列表时，浏览器窗口显示的目录和文件信息要随时刷新到最新状态。

- **作业背景知识**

IFTTT 是互联网的一种应用形态，它支持以 IF X THEN Y 的方式来定义任务，并能够在后台自动执行任务，比如：

IF {news.163.com has new message} THEN {drag the message to my blog}

其中 IF 和 THEN 为关键词，IF 后面的部分为触发器，THEN 后面部分为任务。

关于 IFTTT:（建议大家实验前先了解一下基本思想）

IFTTT_百科：http://baike.baidu.com/item/ifttt?fr=aladdin

二、 输入

通过控制台启动主程序后，再输入相关信息进行监控。输入可为一行或多行。但全部输入结束后，监控线程才开始启动。监控中，不可添加新的任务，即非并发输入。

触发器和任务的具体要求见指导书的第五节和第六节。可自行设计触发器和任务的输入格式，务必细致和准确。

触发器中指明监视的目录/文件，即监控范围。如果是目录，则意味目录内的任何子目录和文件都在监控之列；如果是普通文件（即不是目录），则意味监控范围为该文件。

基本的输入格式为：*IF 监视的目录/文件 触发器 THEN 任务*

其中: 监视的目录/文件为测试者自定义；触发器 为下文触发器部分(第五节)中给出的选项；任务 为下文任务部分(第六节)中给出的选项。

可能的任务格式举例：

IF /root modified THEN record-summary

//含义为：对/root 文件夹进行监控，一旦有文件被修改，则执行 record-summary。

IF [D:\test\demo.cpp] renamed THEN recover

//含义为：对 D:\test\demo.cpp 文件进行监控，一旦被重命名，则执行 recover。

触发器和任务的选择范围包括下文中给出的全部选项，每个输入为其中之一。

程序针对输入的每个不同的监视目录/文件，创建相应的监视线程。要求支持监控的目录/文件不少于 5 个，不多于 8 个。

对于同一个监视的目录/文件，可有设置多个触发器和任务组合。例如，对于输入的两个命令“IF x tr1 THEN ts1; IF x tr2 THEN ts2”，x 表示相同的目录/文件，而 tr1 可以和 tr2 不同，ts1 可以和 ts2 不同。如果 tr1 和 tr2 相同，且 ts1 和 ts2 相同，则视为相同命令，实际只处理其中一个，无需反馈提示。

如果监控的是目录，则监控目录下的所有文件，包括子目录（无论递归多少层）。

如果监控的目录不存在，则输出错误提示信息即可。

三、 输出

根据第六节任务部分的要求产生相应输出。结果 summary 和 detail 输出到特定文件，但建议控制台即时输出相关信息。

四、 硬性规定

针对桌面操作系统的文件系统，要求如下：

1. 使用线程安全设计，设计线程安全的文件访问类和其他有可能被共享的类，使用多线程进行检测和处理。提示：主要线程安全问题可能出现在 record-summary 和 record-detail 类型的任务中。
2. 要求文件访问类（线程安全类）提供文件修改（含创建文件、修改文件属性和删除文件）方法供测试使用

3. 使用 Java.io.File 类来实现文件的访问和相关操作(提示：Java.io.File is not thread safe!!!)。
4. 依据文件路径和文件全名来唯一识别一个文件，文件属性包括最后修改时间和文件大小。
5. 文件路径一律使用完整的绝对路径来表示。
6. 每个触发器对应一个独立的监视线程。
7. 如果一个触发器监视的是一个目录，则该顶层目录不允许有修改（包括删除、移动、重命名等）的操作，但可对监控目录下的子目录进行相应修改。
8. 为了能够正常访问包含中文字符的目录/文件，此次作业要求统一将编码格式设置为 UTF-8 编码。IDE 中的修改步骤见指导书末尾提示。

五、 触发器

要求支持的触发器包括:

1) "renamed" 文件重命名触发器

文件名称变化可简化定义为在同一层目录下，新增了一个文件，缺失了一个文件，新增的文件跟缺失的原来文件的最后修改时间相同，文件大小一样，但名称不一样的文件，即可定义为文件名称变化。（注：本程序并不是一个操作系统的文件管理系统，而是通过观察比较文件属性、位置和数量等变化来进行事件发现和响应的。）

例子：/root/a.txt => /root/b.txt。如果发现有文件缺失，但没有新增文件；或者有新增文件，但没有缺失文件，则都不能算为文件重命名。若新增了 x 个文件，缺失了 y 个文件且全部文件大小、最后修改时间均相同，仅名称不一样，则任意缺失到新增的映射均认为正确。如缺失/root/a.txt 和 /root/b.txt，新增 /root/c.txt 和 /root/d.txt，则认为 a.txt 改名到/root/c.txt 或 /root/d.txt，b.txt 改名到/root/d.txt 或 /root/c.txt 都合理。但不能认为两个不同文件改名成一个相同文件。

2) "Modified" 文件最后修改时间变化触发器

文件修改时间变化的含义是，监控范围内的一个文件在两次扫描中被发现最后修改时间不同（不处理时间的先后关系）。

3) "path-changed" 路径变化触发器

文件路径变化可定义为：在监控范围内，在不同路径下，新增一个和原来文件名字相同、规模相同、最后修改时间相同的文件且原来的文件消失。如果发现一个文件消失了，但是在监控范围内的其他路径下发现有文件名称相同、规模相同、最后修改时间相同的多个文件，则随意指定其中一个为原来的文件皆正确。

4) "size-changed"文件规模变化触发器

若监视范围为一个目录，则监控该目录下（递归定义）的文件/文件夹新增（0->x bytes，x>0）、删除(x->0 byte，x>0)（修改文件名或路径视为一删一增）以及修改内容产生文件大小变化(x->y bytes，x>0, y>0)；若监视一个具体文件，仅对该文件的删除和新增以及修改进行检测。

注意：

- 若监视一个目录，所有检测均需要支持子目录递归检测，即针对该目录下的所有文件及文件夹。
- 若监视一个具体文件，当文件重命名或路径移动，应该继续监视该重命名或路径移动后的文件。
- 限定路径移动和重命名仅针对普通文件而非文件夹。
- 一个目录文件的规模为其直接下层所有非目录文件规模的和。

六、 任务

支持的任务包括：

1) " record-summary " 记录 summary

构造一个 summary 记录对象，保存相应触发器的触发次数信息。每当触发器触发，触发器向 summary 对象登记信息，后者按触发器类别统计触发次数；summary 对象每隔一段时间（自行设定）保存信息至监控范围之外的某个特定文件（具体位置应在 readme 中交代说明）。

2) "record-detail" 记录 detail

构造一个 detail 记录对象，保存相应触发器触发时的相关详细信息：文件规模的前后变化、文件重命名的前后变化、文件路径的前后变化、文件修改时间的前后变化。每当触发器触发，触发器向 detail 对象登记信息，有后者按照触发器类别记录整理；detail 对象每隔一段时间（自行设定）保存信息至监控范围之外

的某个特定文件（具体位置应在 readme 中交代说明）。

3) "recover" 恢复文件路径（仅对文件）

仅可与重命名或路径改变触发器配合使用，其他情况应报错并忽略相应命令。效果为将重命名恢复原状以及将路径改变恢复原状。

七、 测试

测试者使用被测试者提供的线程安全类构造一个测试线程，模拟用户对文件的修改（与 IF 触发器匹配）--测试线程没有义务采用任何同步控制措施，由此导致的程序崩溃均为被测程序问题。

测试者在被测程序 main 方法的合适位置创建和启动测试线程。

测试者检查被测程序是否能够按照触发器正确检测到相应的变化，并按照任务要求进行处理。

被测试者提供的文件读写线程安全类（可以是多个）应该具备文件信息（名称、大小、修改时间）读取功能，同时可以重命名文件（对应重命名触发器）、移动文件以改变文件路径（对应路径改变触发器）、新增及删除文件和文件夹的功能和文件写入功能（用于造成文件大小和修改时间不同，对应规模变化触发器和修改时间触发器）。具体使用方法由被测试者在 readme 中说明。

无论测试者和被测试者，如果出现乱码问题，请自行研究解决。

八、 修改编码格式的方法

请同学们修改字符集编码，否则读取的文件名和路径名等可能显示为乱码。这里仅列出常用的 Eclipse 和 IDEA。使用其他 IDE 或编辑工具的建议在搜索引擎中寻找解决方案。

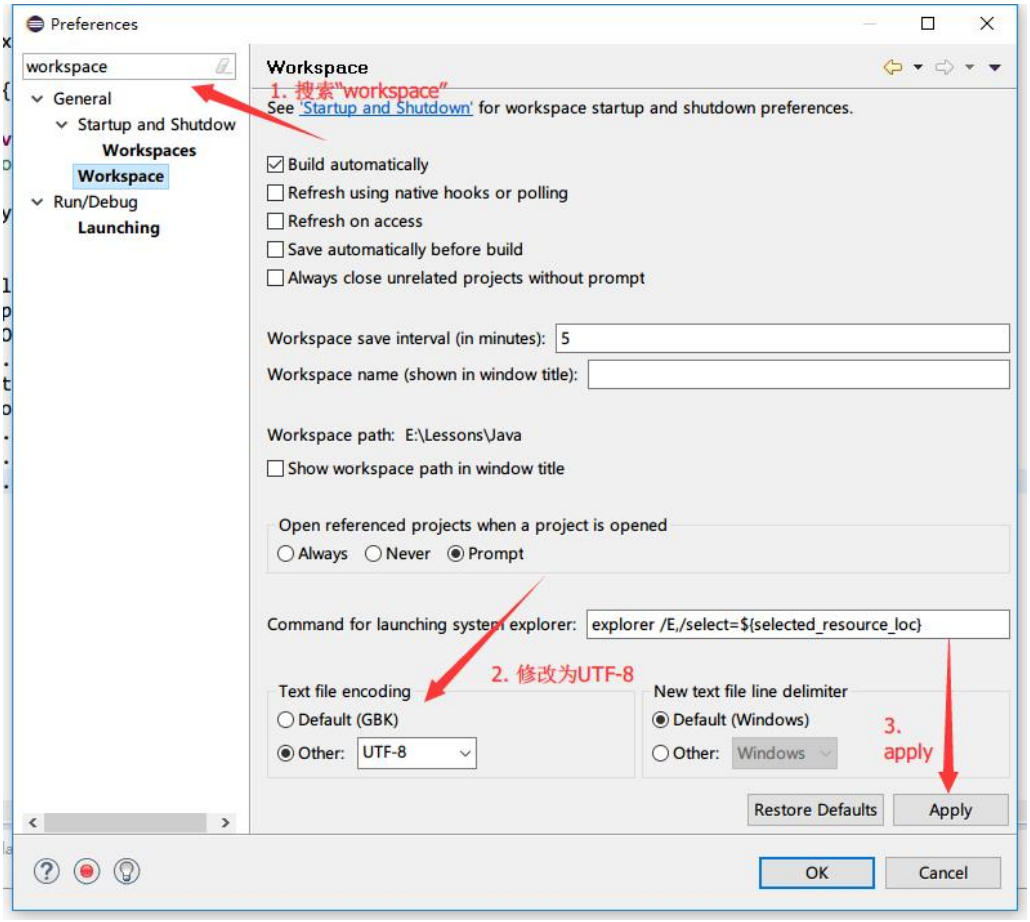

图 1：Eclipse IDE 中的字符集设定

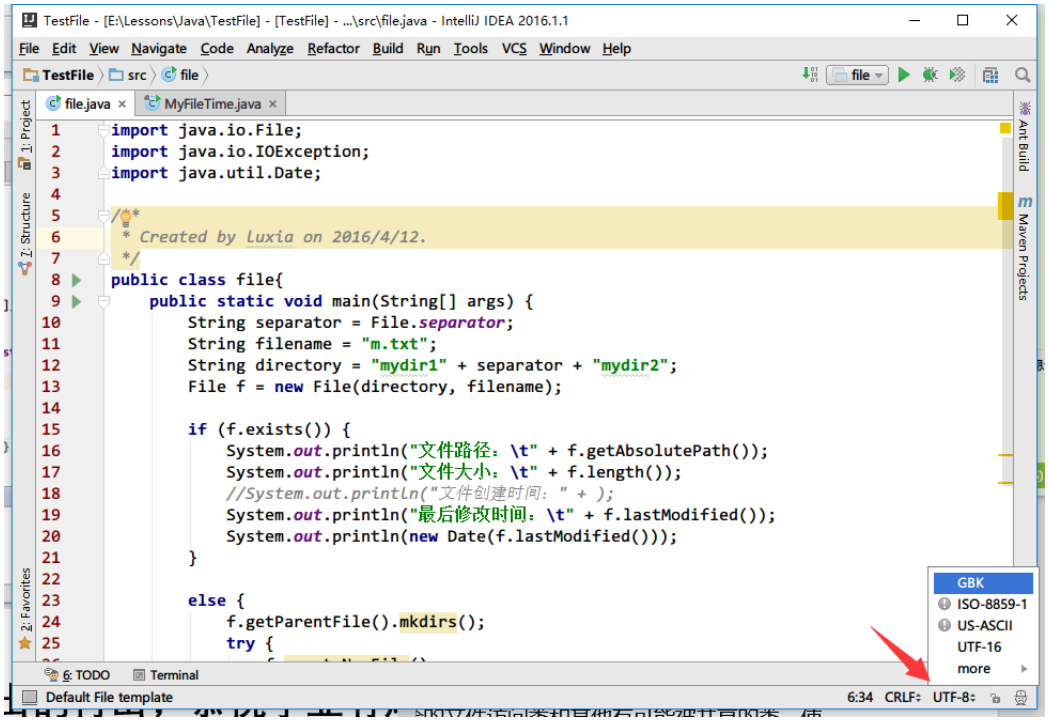

图 2： IDEA 中的字符集设定

九、 设计参考和补充说明

对于同一个修改，若满足多个触发器触发规则，则这些触发器要有响应。

对于一个 IF THEN 语句来说，有以下说明

监控对象：指 IF-THEN 语句里输入的文件或者目录。

工作区：若监控对象是目录，则目录自身和递归定义的下层目录和文件组成工作区；若监控对象是文件，则文件的父目录和其和递归定义的下层目录和文件组成工作区。

监控范围：若监控对象是目录，则目录自身和递归定义的下层目录和文件组成监控范围，也就是工作区；若监控对象是文件，则监控范围就是文件自身。

所有监控范围内的文件和目录都应该纳入触发检测，超出工作区的一切操作不予理会。

文件大小定义为文件字节数，目录大小定义为直接下层文件（不递归）的大小总和。

若监控范围内的一个目录或者文件大小发生变化，就触发一次 size-changed，包括新增删除以及修改大小。

举例：监控对象为/root

/root/test/a.txt 新增或删除，若 a.txt 是 0 字节，仅触发 a.txt 的 size-changed，若非 0 字节，则也触发/root/test/，但都与/root 无关。

/root/test/a.txt 修改大小，仅触发 a.txt 的 size-changed 和/root/test/的 size-changed，/root 仍然无关。

若监控范围内的一个目录或者文件最后修改时间发生变化，就触发一次 modified，前后文件都存在。

若监控范围内一个文件的名称发生变化但仍在同一个目录内，则触发一次 renamed，即两次扫描发现缺失一个文件，新增一个文件，且缺失的和新增的大小修改时间相同，名字不同。

若监控范围内一个文件的路径发生变化但名称没有变化，则触发一次 path-changed，即监控范围内缺失一个文件，工作区内新增一个文件，且新增的和缺失的路径不同名字相同大小相同修改时间相同。

若监控对象为文件，则触发 renamed 或者触发 path-changed 后需要跟踪新的

文件，其他的无需跟踪重命名和路径改变。

recover 造成的文件移动或重命名不会再次触发当前线程的 renamed 或者 path-changed。

两次扫描之间产生的变化若不是开始状态和最终状态则不理会，即扫描间隔内重命名完又重命名回去，不会触发，触发仅看扫描前后的状态。

文件内容不敏感，仅文件大小敏感。

新增 1：一个工作区一个线程，所有语句线程公用一个 summary 和 detail 对象。

新增 2：监控对象是文件，监控对象消失（非重命名或移动）之后，处理方式自定，监控对象在线程开始时必须保证存在