Макаров Дмитрий Александрович
магистрант
ФГБОУ ВО «Армавирский государственный
педагогический университет»
г. Армавир, Краснодарский край
Коновалова Виктория Александровна
учитель обществознания
МАОУ – СОШ №7 им. Г.К. Жукова
г. Армавир, Краснодарский край

МОНИТОРИНГ ИЗМЕНЕНИЯ ФАЙЛОВ В ПАПКЕ СРЕДСТВАМИ JAVA

Аннотация: *данная статья посвящена средствам мониторинга изменения файлов. Авторы приходят к выводу о том, что API Watch Service реализует подход низкого уровня.*

Ключевые слова: *книга, публичная библиотека, образование, социальная структура населения.*

Мы постоянно работаем с приложениями, документами и файлами, внося в них свои изменения при необходимости редактируем, а также в случае ненадобности удаляем. Чтобы все это упорядочить и обеспечит «потокобезапасность», была введена служба Watch которая отслеживает объект на предмет изменений. Ниже реализована программа, фиксирующая события в определенном каталоге с подробными комментариями и пояснениями.

```
// Проверка заданного пути на принадлежность к каталогу
try {
Boolean isFolder = (Boolean) Files.getAttribute(path,
«basic:isDirectory», NOFOLLOW_LINKS);
if (!isFolder) {
throw new IllegalArgumentException(«Path: " + path + " is not a folder»);
} catch (IOException ioe) {
// Папка не существует
```

```
ioe.printStackTrace();
System.out.println(«Watching path: " + path);
// Получаем файловую систему пути
FileSystem fs = path.getFileSystem ();
// Мы создаем новый WatchService используя новый try() блок
try(WatchService service = fs.newWatchService()) {
// Регистрируем путь
// Следим за событием создания
path.register(service, ENTRY_CREATE);
// Запускаем бесконечный цикл опроса
WatchKey key = null;
while(true) {
key = service.take();
// извлечение события из очереди
Kind<?> kind = null;
for(WatchEvent<?> watchEvent : key.pollEvents()) {
// Получить тип события
kind = watchEvent.kind();
if (OVERFLOW == kind) {
continue;
} else if (ENTRY_CREATE == kind) {
// Был создан новый Path
Path newPath = ((WatchEvent<Path>) watchEvent).context();
// Вывод
System.out.println(«New path created: " + newPath);
if(!key.reset()) {
break;
} catch(IOException ioe) {
ioe.printStackTrace();
} catch(InterruptedException ie) {
```

```
ie.printStackTrace();
public static void main(String[] args) throws IOException,
InterruptedException {
// Папка за который мы будем следить
Path folder = Paths.get(System.getProperty(«user.home»));
watchDirectoryPath(folder);
```

Результат:

```
Watching path: C:\Users\Дмитрий
New path created: Новый текстовый документ.txt
New path created: Документ Microsoft Word.docx
```

Необходимо принять во внимание, что в нашем конкретном случае мы отслеживаем директорию Home, а не всю иерархию патока. Если вы пожелаете контролировать дерево каталогов, тогда вам необходимо будет зарегистрировать службу Watch для каждой папки в дереве.

Для того, чтобы получить производные от каталога, можно использовать интерфейс FileVisitor <T>. Он позволит обойти дерево каталога.

Кроме того, необходимо будет поддерживать коллекцию наблюдателей, в случае, если вы будете создавать или удалять уже существующие вложенные папки.

API Watch Service реализует подход низкого уровня, так что возможно будет лучше реализовать собственный механизм высокого уровня или использовать уже существующие решения.

Список литературы

1. Class FileSystem [Электронный ресурс]. – Режим доступа: https://docs.oracle.com/javase/7/docs/api/java/nio/file/FileSystem.html#newWatchService()

2. Interface WatchService [Электронный ресурс]. – Режим доступа: https://docs.oracle.com/javase/7/docs/api/java/nio/file/WatchService.html