

ACTIVIDAD 1

Enunciado

En su arranque el programa creará tres ficheros (Fichero1.txt, Fichero2.txt y Fichero3.txt) con nombre y prioridad y saquen un mensaje cada uno cuando acabe su tarea. Además de crear los ficheros cada hilo debe de escribir 100 líneas la misma frase indicando el nombre del hilo y su prioridad. A su vez un hilo Demonio que cuenta líneas. Este mostrará cada segundo un mensaje con el número de líneas que tiene cada fichero.

Desarrollo

El programa presenta 2 clases `MyThread` y `MyDaemonThread` que extienden la interfaz `Runnable` y la clase principal que contiene el `main()`.

- `main()`: encargado de crear los hilos, establecer nombres, prioridad 0-10(máx) y lanzarlos. Aun así en este caso en concreto establecer prioridades no asegura que el primer hilo sea el primero en terminar debido a que la cantidad de trabajo es pequeña y hay recursos disponibles todo el tiempo.

```
Thread thread1 = new Thread(new MyThread("Fichero1.txt"), "Thread-1" );
//thread2 y thread3

Thread daemonThread = new Thread(new MyDaemonThread("Fichero1.txt",
"Fichero2.txt", "Fichero3.txt"), "DaemonThread" );
thread1.setPriority(10);
//thread2 y thread3
daemonThread.setDaemon(true);

thread1.start();
//thread2 y thread3
daemonThread.start();
```

- `MyThread.java`: con el método estático `Thread.currentThread()` obtenemos el hilo específico que invoca dicho método. El punto clave del ejercicio se encuentra dentro del bucle `for`, en el cual a cada iteración se instancia un nuevo `FileWriter` y se cierra, para que el archivo resulte modificado y el `daemonThread` tenga filas ya escritas en los archivos mientras están siendo modificados.

```

@Override
    public void run() {

        this.threadName = Thread.currentThread().getName();
        this.threadPriority = "" + Thread.currentThread().getPriority();
        int i;
        FileWriter fw = null;

        for(i = 0; i<100; i++){
            try {
                Thread.sleep(100);
                fw = new FileWriter(file, true);
                writeLine(fw);
                fw.close();
            } catch (Exception ex) {
                Logger.getLogger(MyThread.class.getName()).log(Level.SEVERE, null, ex);
            }
        }
        System.out.println(threadName+": "+fileName+" completo");
    }

```

- MyDaemonThread.java: Este hilo no presenta nada especial, de esta forma obtenemos todo el fichero como un `Stream<String>` en el que cada `String` es una linea, solo queda contar el número de elementos con `.count()`.

```

System.out.println(
    "Fichero: "+fileName1+" num lineas: " +
    Files.lines(Paths.get(fileName1), StandardCharsets.UTF_8).count()
);

```