

Un hilo solo puede ser iniciado Una vez

Un hilo No se puede iniciar 2 veces.

Si se lanza una segunda vez, java lanza una:

`IllegalThreadStateException` - if the thread was already started.

From the Java API Specification for the `Thread.start` method:

It is never legal to start a thread more than once.

In particular, a thread may not be restarted once it has completed execution.

So yes, a Thread can only be started once.

Y si queremos lanzar el hilo una segunda vez?

If a Thread needs to be run more than once, then one should

make an new instance of the Thread and call start on it.

Proteger Recursos Compartidos

Para proteger un recurso compartido EN JAVA (por ejemplo una lista de proyectiles-->Ver ejemplo dummies)

que pueda ser accedido por 2 o más hilos (y una tarea programada es un hilo) tengo 2 opciones:

- a) SEMÁFOROS
- b) MONITORES

a) Meterlo en una clase que use la clase semaphore o una llave "programada":

es decir programarmelo todo yo con wait, notify y una variable booleana.

b) Meterlo en una clase que tenga todas las operaciones que puedo hacer con el recurso(lista x ej)

dentro de métodos synchronized.

Java se encarga de poner "la llave".

Nota: Si un lenguaje no tuviera "Synchronized" solo tendría la opción a.

RESUMEN GLOBAL

Concurrencia:

Cuando 2 o + hilos intentan acceder a un recurso compartido al q solo queremos q acceda 1 a la vez (o N a la vez)

Para ello debemos proteger el recurso compartido con mecanismo de sincronización:

- semáforo (Permite el paso de N hilos a la SC)
- monitor (Permite el paso de 1 hilo a la SC)

Sincronización:

Cuando quiero q 2 o + hilos accedan a un recurso en el orden q yo quiero.

(Si no se usara se harían las tareas "a lo loco"(sin orden))

Se hace usando wait y notify