Escuela Técnica Superior de Ingeniería

Prácticas
de
Programación Concurrente y Distribuida

3º Curso de Grado en Ingeniería Informática

Curso 2016-17

EXAMEN

Enero de 2017

CONSIDERACIONES PREVIAS:

- **No se permite el uso de ningún tipo de documentación.**
- **El acceso a Internet está desactivado conscientemente.**
- **Apague el teléfono móvil.**

ANTES DE COMENZAR EL EXAMEN:

- Cree una carpeta con su nombre y primer apellido en el **Escritorio** separados por un guión bajo (ejemplo: **`Pedro_Abad`**).
- En dicha carpeta aparecerá un proyecto por cada una de las preguntas del examen. Dichos proyectos se denominarán **`Proyecto1`**, **`Proyecto2`**,…, **`Proyecto4`**.

ENUNCIADO:

En una fábrica de tornillos se fabrican tornillos de acero inoxidable y de hierro. Una vez fabricados, los tornillos pasan por una vibradora para pulirlos. En la vibradora caben dos tornillos de hierro o tres de inoxidable, pero no pueden mezclarse en su interior los tornillos de hierro y los inoxidables.

PROYECTO 1.

Tiempo estimado: 40 minutos.
Puntos: 4

Será el proyecto base para solucionar el enunciado. Contendrá las siguientes clases:

- **`Vibradora`**. La clase `Vibradora` mantendrá el estado de ocupación de la vibradora e implementará los siguientes métodos:
  - **`EntraInox`**. Que deberá ser invocado por los tornillos inoxidables cuando quieren acceder a la vibradora .
  - **`SaleInox`**. Que deberá ser invocado por los tornillos inoxidables al salir de la vibradora.
  - **`EntraHierro`**. Que deberá ser invocado por los tornillos de hierro cuando quieren entrar en la vibradora.
  - **`SaleHierro`**. Que deberá ser invocado por los tornillos de hierro al salir de la vibradora.
- **`Inoxidable`**. Representará cada uno de los tornillos inoxidables mediante un hilo. El hilo se creará heredando de la clase `Thread`. El hilo pondrá un mensaje de inicio indicando su identificador, intentará acceder a la vibradora usando la clase `Vibradora`, permanecerá en el vibradora una cantidad de tiempo aleatoria de entre 3 y 5 segundos y saldrá.
- **`Hierro`**. Representará cada uno de los tornillos de hierro mediante un hilo. El hilo se creará implementando el *interface* `Runnable`. El hilo pondrá un mensaje de inicio indicando su identificador, intentará acceder a la vibradora usando la clase `Vibradora`, permanecerá en la vibradora una cantidad de tiempo aleatoria de entre 2 y 5 segundos y saldrá.

- **`Generador`**. Contendrá el método `main` y será quién comience la ejecución. Debe lanzar, de forma aleatoria, tornillos inoxidables o tornillos de hierro a intervalos de tiempo de entre 1 a 2 segundos. La frecuencia de llegada de tornillos inoxidables será del 60% y la de tornillos de hierro del 40%. Deberá esperar a que finalicen todos los hilos para finalizar.

El control de la concurrencia y la sincronización se realizará en la clase `Vibradora`, mediante las primitivas de Java `wait(), notify()` y/o `notifyAll().`

PROYECTO 2.

Tiempo estimado: 20 minutos.
Puntos: 3

Se modificará el *Proyecto1* para que la clase `Vibradora` controle la concurrencia mediante `ReentrantLocks` y `Conditions.`

No podrá usarse el método `signalAll()` de las `Conditions.`

PROYECTO 3.

Tiempo estimado: 15 minutos.
Puntos: 2

Tomará como base el *Proyecto1* y se modificará la clase **`generador`** para que use un *ThreadPool* con un tamaño fijo de 4 hilos para lanzar los tornillos de hierro. Al finalizar, cada tornillo deberá devolver el tiempo que ha empleado en pulirse, y generador pondrá un mensaje final indicando el tiempo total de ocupación de la vibradora por los tornillos de hierro.

PROYECTO 4.

Tiempo estimado: Depende de la implementación que se pretenda
Puntos: 1

Se creará un *Applet* que visualice de forma gráfica, mediante un *Canvas*, la situación de la vibradora y las colas de espera del *Proyecto 1*.