Escuela Técnica Superior de Ingeniería

Prácticas
de
Programación Concurrente y Distribuida

3º Curso de Grado en Ingeniería Informática

Curso 2017-18

EXAMEN

Enero de 2018

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 un taller de cambio de neumáticos trabajan cuatro operarios. Al taller llegan turismos y camiones para cambiar los neumáticos. Para cambiar los neumáticos de un turismo basta con un operario, pero los camiones necesitan que dos operarios se encarguen del cambio. Los camiones tendrán prioridad de paso con respecto a los coches.

PROYECTO 1.

Tiempo estimado: 40 minutos.
Puntos: 4

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

- **`Taller`**. La clase `Taller` mantendrá el estado de ocupación del Taller e implementará los siguientes métodos:
  - **`EntraCamion`**. Que deberá ser invocado por los Camiones cuando quieren acceder al Taller .
  - **`SaleCamion`**. Que deberá ser invocado por los Camiones al salir del Taller.
  - **`EntraCoche`**. Que deberá ser invocado por los coches cuando quieren entrar en el Taller.
  - **`SaleCoche`**. Que deberá ser invocado por los coches al salir del Taller.
- **`Camion`**. Representará cada uno de los Camiones mediante un hilo. El hilo se creará heredando de la clase `Thread`. El hilo pondrá un mensaje de inicio indicando su identificador, intentará acceder al Taller usando la clase `Taller`, permanecerá en el Taller una cantidad de tiempo aleatoria de entre 3 y 5 segundos y saldrá del Taller.
- **`Coche`**. Representará cada uno de los coches mediante un hilo. El hilo se creará implementando el *interface* `Runnable`. El hilo pondrá un mensaje de inicio indicando su identificador, intentará acceder al Taller usando la clase `Taller`, permanecerá en el Taller una cantidad de tiempo aleatoria de entre 2 y 4 segundos y saldrá del Taller.

- **`Generador`**. Contendrá el método `main` y será quién comience la ejecución. Debe lanzar, de forma aleatoria, coches y camiones a intervalos de tiempo de entre 1 a 3 segundos. La frecuencia de llegada de coches será del 70% y la de Camiones del 30%. Deberá esperar a que finalicen todos los hilos para finalizar.

El control de la concurrencia y la sincronización se realizará en la clase `Taller`, 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 `Taller` 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 3 hilos para lanzar los coches. Al finalizar, cada coche deberá devolver el tiempo que ha tardado en realizar el cambio, y generador pondrá un mensaje final indicando el tiempo total de ocupación del Taller por los coches.

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 del Taller y las colas de espera del *Proyecto 1*.