Escuela Técnica Superior de Ingeniería

Prácticas
de
Programación Concurrente y Distribuida

3º Curso de Grado en Ingeniería Informática

Curso 2018-19

EXAMEN

Enero de 2019

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:

Una piscina, con capacidad para 5 personas, dispone para sus clientes de 6 aletas y 5 palas. Los usuarios de la piscina, después de un tiempo de calentamiento, necesitan usar, a la vez, dos aletas y dos palas durante su entrenamiento, finalizado el cual, las dejarán libres y saldrán de la piscina. Las aletas siempre se cogerán antes que las palas, es decir, no se cogen las palas hasta que no se dispone de las aletas.

Se debe diseñar una solución ausente de interbloqueos.

PROYECTO 1.

Tiempo estimado: 40 minutos.
Puntos: 4

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

- **`Piscina`**. La clase `Piscina` mantendrá el estado de ocupación del piscina e implementará los siguientes métodos:
  - **`EntraPiscina`**. Que deberá ser invocado por los usuarios cuando quieren acceder a la piscina .
  - **`SalePiscina`**. Que deberá ser invocado por los usuarios libres al salir de la piscina.
  - **`CogeMaterial`**. Que deberá ser invocado por los usuarios que han terminado el calentamiento y quieren usar las aletas y palas..
  - **`SueltaMaterial`**. Que deberá ser invocado por los usuarios cuando dejen libres las aletas y palas.
- **`Nadador`**. Representará cada uno de los usuarios del club 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 piscina usando la clase `Piscina`, realizará el calentamiento, durante un tiempo aleatoria de entre 1 y 2 segundos , tratará de conseguir el material, y lo usará durante un tiempo aleatorio de entre 2 y 3 segundos y saldrá de la piscina.

- **`Generador`**. Contendrá el método `main` y será quién comience la ejecución. Debe lanzar, nadadores a intervalos de tiempo de entre 1 a 2 segundos. Deberá esperar a que finalicen todos los hilos para finalizar.

El control de la concurrencia y la sincronización se realizará en la clase `Piscina`, 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 `Piscina` 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 6 hilos para lanzar los nadadores. Al finalizar, cada nadador deberá devolver el tiempo que ha nadado usando las aletas y las palas, y generador pondrá un mensaje final indicando el tiempo total de ocupación de todos los nadadores.

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