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:

Debido a unas inundaciones, un puente peatonal situado sobre un río ha quedado dañado. Para evitar su colapso, sólo se permite un peso máximo sobre él. Se ha estipulado que dicho peso corresponde al de dos adultos, al de tres niños, o al de un adulto y un niño. En el caso de que haya varias personas esperando, los niños tendrán preferencia de paso.

PROYECTO 1.

Tiempo estimado: 40 minutos.
Puntos: 4

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

- **`Puente`**. La clase `Puente` mantendrá el estado de ocupación del puente e implementará los siguientes métodos:
  - **`EntraAdulto`**. Que deberá ser invocado por los adultos cuando quieren acceder al puente .
  - **`SaleAdulto`**. Que deberá ser invocado por los adultos al salir del puente.
  - **`EntraNino`**. Que deberá ser invocado por los niños cuando quieren entrar en el puente.
  - **`SaleNino`**. Que deberá ser invocado por los niños al salir del puente.
- **`Adulto`**. Representará cada uno de los adultos 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 puente usando la clase `Puente`, permanecerá en el puente una cantidad de tiempo aleatoria de entre 3 y 5 segundos y saldrá del puente.
- **`Nino`**. Representará cada uno de los niños mediante un hilo. El hilo se creará implementando el *interface* `Runnable`. El hilo pondrá un mensaje de inicio indicando su identificador, intentará acceder al puente usando la clase `Puente`, permanecerá en el puente una cantidad de tiempo aleatoria de entre 2 y 5 segundos y saldrá del puente.

- **`Generador`**. Contendrá el método `main` y será quién comience la ejecución. Debe lanzar, de forma aleatoria, adultos o niños a intervalos de tiempo de entre 1 a 3 segundos. La frecuencia de llegada de adultos será del 60% y la de niños 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 `Puente`, mediante las primitivas de Java `wait(), notify()` y/o `notifyAll()`.

PROYECTO 2.

Tiempo estimado: 20 minutos.
Puntos: 3

Se modificará el *Proyecto2* para que la clase `Puente` 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 niños. Al finalizar, cada niño deberá devolver el tiempo que ha empleado en cruzar, y generador pondrá un mensaje final indicando el tiempo total de ocupación del puente por los niños.

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