Examen_Practicas_PCD_1718_V2.pdf

Info_sw

Programación Concurrente y Distribuida

3º Grado en Ingeniería Informática

Escuela Técnica Superior de Ingeniería
Universidad de Huelva

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 una mina se extrae mineral que se amontona mediante una cinta transportadora. El mineral es retirado del montón por dos cargadoras pequeñas, que son capaces de retirar 1 Tm (Tonelada métrica), y una cargadora grande que puede retirar 3 Tm. Si no hay suficiente mineral, las cargadoras esperarán a que la cinta aporte más mineral para retirarlo. Dado que la cargadora grande es más eficientes que las pequeñas, ésta tendrá prioridad para retirar el mineral, cuando hay suficiente.

PROYECTO 1.

Tiempo estimado: 40 minutos.
Puntos: 4

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

- **`Monton`**. La clase Monton gestionará la cantidad de mineral que hay en el montón, que inicialmente tendrá 4 Tm. La evolución del montón dependerá de los siguientes métodos:
  - **`CargaPoco`**. Que deberá ser invocado por las cargadoras pequeñas cuando quieran retirar mineral del montón.
  - **`CargaMucho`**. Que deberá ser invocado por las cargadoras grandes cuando quieren retirar mineral del montón. Las cargadoras grandes retiran todo la cantidad de una sola vez, si hay disponible, si no, deben esperar a que haya suficiente.
  - **`Rellena`**. Que deberá ser invocado por la cinta cuando aporta mineral al montón.
- **`Pequenya`**. Representará cada uno de las cargadoras pequeñas mediante un hilo. El hilo se creará heredando de la clase `Thread`. El hilo pondrá un mensaje de inicio indicando su identificador, intentará retirar mineral del montón usando la clase `Monton`, y esperará una cantidad de tiempo aleatoria de entre 1 y 3 segundos (tiempo del transporte) para volver a cargar. Las cargadoras harán 10 cargas y finalizarán.
- **`Grande`**. Representará la cargadora grande mediante un hilo. El hilo se creará implementando el `interface Runnable`. El hilo pondrá un mensaje de inicio

indicando su identificador, intentará retirar mineral del montón usando la clase `Monton`, y esperará una cantidad de tiempo aleatoria de entre 2 y 4 segundos (tiempo del transporte) para volver a cargar. La cargadora hará 7 cargas y finalizará.

- **Cinta**. Representará la cinta mediante un hilo. El hilo rellenará el montón a intervalos de tiempo de entre 1 y 5 segundos. La cantidad de mineral que aporta cada vez será de entre 2 y 5 Tm.
- **`Mina`**. Contendrá el método `main` y será quién comience la ejecución. Debe lanzar, los hilos de las cargadoras y de la cinta. Esperará a que las cargadoras acaben e interrumpirá la cinta para acabar la aplicación.

El control de la concurrencia y la sincronización se realizará en la clase `Monton`, 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 `Monton` controle la concurrencia mediante `ReentrantLocks` y `Conditions`.

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

PROYECTO 3.

Tiempo estimado: 15 minutos.

Puntos: 2

Se modificará el *Proyecto1* para controlar el montón mediante un **semáforo general**.

Se deberá eliminar del Proyecto 1 la clase `Monton` y sustituirlo por un semáforo. Para este proyecto, la cargadora grande no tendrá prioridad, y podrá retirar el mineral poco a poco según esté disponible, compitiendo por él con las pequeñas.

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