

Prácticas de hilos

Cree las siguientes clases de Java.

1. Cree una clase de Java que debe llamarse Recurso. Esta clase debe tener un método llamado "void usarRecurso(String nombre)". Este método debe escribir en consola:
El hilo "nombre" está usando el recurso "nombreRecurso"
El hilo "nombre" ha terminado de usar el recurso "nombreRecurso"
Entre los dos mensajes debe transcurrir un tiempo aleatorio entre 0 y 2 segundos.
El nombreRecurso se deberá pasar en el constructor.

En los siguientes casos, cuando se indique que el hilo XXX usará el recurso, se refiere a que ejecutará el método usarRecurso("Hilo XXX").

2. Usando la clase Recurso anterior simule la siguiente situación: 10 hilos desean usar un recurso, los 10 hilos comparten el mismo objeto Recurso. Sólo 2 hilos pueden usar el recurso a la vez.
3. Usando la clase Recurso anterior simule la siguiente situación:
Cree dos objetos de la clase Recurso: recurso1 y recurso2.
20 hilos desean usar los recursos. Primero usan el recurso1 y cuando terminan el recurso2.
El recurso1 sólo puede ser usado por 2 hilos a la vez.
El recurso2 sólo puede ser usado por 1 hilo cada vez.
4. Usando la clase Recurso anterior simule la siguiente situación:
Cree dos objetos de la clase Recurso: recurso1 y recurso2.
20 hilos desean usar los recursos. Intentan usar el recurso1, si no pueden van a usar el recurso2 o viceversa.
El recurso1 sólo puede ser usado por 2 hilos a la vez.
El recurso2 sólo puede ser usado por 1 hilo cada vez.

Resuélvalo usando el siguiente algoritmo: Una forma de solucionarlo es: Se necesitará un semáforo en el que esperarán todos los hilos y un contador para cada recurso que cuente cuántos hilos están usando cada recurso. Cuando un hilo inicie el uso de un recurso deberá actualizar el contador correspondiente. Cuando un hilo termine de usar el recurso deberá actualizar el contador correspondiente y mandar un release al semáforo para que entre el siguiente hilo. Si no está disponible el siguiente recurso que desea usar, debe volver a adquirir el semáforo.

Cada hilo debe llevar la cuenta de los recursos que ha usado y los que le quedan por usar.
¿Es necesario proteger cada uno de los contadores con un semáforo?

5. En el ejercicio anterior, ¿en qué situación se produce espera ocupada?
Una forma de resolver el ejercicio anterior anterior, evitando la espera ocupada, es añadiendo dos semáforos adicionales en los que se pondrán a esperar los hilos que sólo les quede por visitar uno de los recursos. Se debe añadir un contador en cada uno de los semáforos adicionales que lleve la cuenta de los hilos que están esperando en cada uno de los dos nuevos semáforos, para que cada hilo que libera uno de los recursos sepa si debe hacer un release sobre el semáforo principal o sobre uno de los semáforos nuevos. ¿Se deben proteger los nuevos contadores con semáforos? Razone su respuesta.
6. Usando dos semáforos y dos hilos simule una situación de deadlock.
7. Opcional: Resuelva el ejercicio 4 usando el algoritmo planteado en el ejercicio 5.