

Programación multihilo en Java

Concurrencia y sincronización de hilos: simulación hincha globos y pincha globos

Hacer un programa multihilo que utilice los mecanismos de sincronización necesarios para que se ejecute una simulación de hinchado de globos según las especificaciones siguientes:

- Se deben de crear las clases siguientes:
 - `HinchaGlobos`, que implementará la funcionalidad necesaria para crear cinco hilos que se encarguen de hinchar globos.
 - `PinchaGlobos`, que implementará la funcionalidad necesaria para crear 1 hilo que se encargue de pinchar globos.
 - `Deposito`, que representará un depósito donde colocar un máximo de 10 globos e implementará la funcionalidad necesaria para crear un hilo que se encargue de reponer los globos cuando sea necesario.
 - `Globo`, que representará globos que se puedan hinchar y pinchar.
 - `Main`, que contiene el método `main` desde que se instancia al resto de clases y se inicia la simulación que se describe a continuación.
- En relación con la clase `Deposito`:
 - Se creará una sola instancia de esta clase:
 - Podrá contener un máximo de 10 globos, pero el hilo reponedor dispone de una cantidad ilimitada de globos que podrá colocar en el depósito.
 - El hilo reponedor llenará el depósito siempre que tenga acceso a él y disponga de espacio para ello.
 - Sólo podrá haber 3 globos hinchándose a la vez, cada uno por un único `HinchaGlobos` que se encargará de retirar el globo del depósito cuando explote o cuando lo pinchen.
 - El `PinchaGlobos` podrá explotar cualquiera de los globos que se están hinchando cuando consiga acceder al depósito.
 - Los `Hinchaglobos`, `PinchaGlobos` y el hilo reponedor competirán por el acceso al depósito.
- En relación con la clase `Globo`:
 - El globo se entrega con un volumen inicial de 0 y se podrán hinchar hasta alcanzar un volumen máximo de 5. Una vez superado dicho volumen, estallarán.
 - Cada vez que se hincha un globo su volumen aumentará en una unidad.
 - Los globos pueden ser pinchados mientras se están hinchando.
 - Los `HinchaGlobos` y `PinchaGlobos` competirán por el acceso a cada globo individual para efectuar sobre él la operación que tienen asignada.
- En relación con los hilos `HinchaGlobos`:
 - Cada uno se nombrará con HG seguido de un número que lo identifique.

- Cada uno se asignará a sí mismo un globo del depósito que hinchará cada vez que obtenga acceso a él hasta que estalle o sea pinchando, en cuyo caso será responsable de retirarlo tan pronto como pueda del depósito.
- Si ya hubiera tres globos hinchándose, esperará hasta que haya uno disponible que no esté asignado.
- Cada vez que realiza un hinchado de un globo deberá esperar un mínimo de 1 segundo para efectuar el siguiente hinchado de ese mismo globo, teniendo en cuenta que durante ese tiempo el [PinchaGlobos](#) podrá acceder a él.
- No se establece un tiempo máximo de espera entre dos hinchados consecutivos.
- En relación con el hilo [PinchaGlobos](#):
 - Se llamará PG.
 - Intentará pinchar uno de los globos que se está hinchando, elegido de forma aleatoria.
 - Para explotar el siguiente globo debe esperar un mínimo de 10 segundos.
 - No es responsable de retirar los globos que pincha.
 - Debe considerar la posibilidad de que no haya globos que pinchar.
- Se escribirá un mensaje cada vez que:
 - Se entregue un globo (ej.: GLOBO 5 ENTREGADO A HG3).
 - Se hinche un globo indicando el nuevo volumen (ej.: GLOBO 5 VOLUMEN 5)
 - Estalle un globo por superar el volumen máximo (ej.: GLOBO 5 ESTALLA)
 - Un PG pinche un globo (ej.: GLOBO 5 PINCHADO POR PG)

El programa deberá finalizar de forma ordenada cuando el usuario lo decida y ofrecer la posibilidad de pausar y reanudar la simulación.