

Práctica 1: Introducción a Hilos Cómputo Concurrente 2024-2

Kassandra Mirael ♥

2024

Tiempo estimado de realizacion: 3 - 5hrs

Objetivo: Entender los conceptos básicos del uso de hilos en Java, así como algunas diferencias entre hilos y procesos. Además de las Interfaces Runnable, Callable.

Introducción

Has llegado a la base de los Diamond Dogs, una central petrolera que sirve como cubierta, llevarás a cabo tu capacitación para que puedas solucionar problemas concurrentes en el futuro. En esta capacitación se te asignarán distintos problemas que deberás resolver individualmente.

Especificaciones

Se te incluye un proyecto en maven, en el cual hay múltiples clases, sigue estas instrucciones para poder completar tu capacitación básica.

Introducción y jugando con Hilos

Modifica el código de *Hilos.java*, realizando lo siguiente:

- Genera una estructura de datos (en donde después guardarás los hilos).
- Agrega 10 hilos a esta estructura de datos mediante un *for* (no te olvides de inicializarlos).
- Finalmente, haz *join* a cada hilo con un *for* o *foreach*.
- Agrega una captura de pantalla a tu reporte sobre como quedó tu código al final.

Contadores compartidos

Para este ejercicio se verán distintas implementaciones de un contador tú debes hacer algunas

y algunas otras ya vendrán hechas, sigue las instrucciones que viene en cada archivo para que puedas realizarlas correctamente.

El objetivo final de todo este conjunto de ejercicios es que el contador final obtenga el valor primero en 20000 y luego de 30000. (Solo en los 2 últimos tienes permitido usar *Synchronized*)

Extendiendo de Thread

Esta es la versión más simple, pues lo único que hacemos es extender de la clase Thread, realmente hay más contras que Pros (tiene una consistencia baja), pero tú tendrás que decidir cuando usarlo y cuando no. Una vez extendido, solo hay que sobrescribir el método `run()`, en el cual va toda la implementación de lo que queremos que haga el hilo.

Runnable

En esta versión se implementa la clase Runnable, su único Método es `run()`, dentro de `run()` se ejecuta el comportamiento de los hilos. Podemos hacer un comportamiento casi genérico para todos los hilos ejecutados, por lo que si queremos que los hilos hicieran tareas muy específicas, no nos beneficia esta manera, pero sí se puede llegar a hacer.

Implementando Runnable dentro de Thread

Por otra parte, se puede implementar la clase Runnable dentro del mismo Thread, de esta manera se puede asignar el comportamiento de cada hilo un tanto más manual y más acorde a lo que queramos hacer. Por ejemplo, si quisiéramos que un hilo sumara los números pares del 0 al 100 y el otro los números impares, podríamos usar esta manera.

Callable

Esta es la versión más diferente a las anteriores, debido a que no usamos la interfaz Runnable. En este primer ejemplo se te enseñará cómo se usa y de qué manera. (Más adelante se harán modificaciones, pero por el momento es suficiente con lo básico).

La principal diferencia es que el método `run()` no devuelve nada, mientras `call()` si lo hace, devuelve un tipo genérico de dato.

Contador Compartido: Synchronized

Antes de empezar, piensa qué es lo que hace Synchronized, analizando lo que paso en los ejercicios anteriores, ¿Que significará?...

Posterior a eso, realiza una breve investigación de lo que hace y escríbelo de forma breve.

Descomenta el siguiente bloque de código y comenta el que no lo estaba, ¿Notas alguna diferencia?

Contador Compartido: Durmiendo Hilos

Finalmente, es hora de divertirnos un poco más, en los casos anteriores a Synchronized, se espera que este en un rango de valores, dada la concurrencia que existe, en este caso, sera más estricto, tu objetivo principal, es que 2 y 3 hilos obtengan valores de 20,000 y 30,000 (Margen de error de 50 a lo sumo).

Esto lo harás con un método visto en la práctica pasada, el cual es: *Thread.sleep()*;

Cuestionario

- ¿Porque se utiliza Interrupted Exception en el método main?
- ¿Para que sirve el método Join?
- ¿Qué pasa si no le hacemos Join a los hilos?
- ¿Cuáles son las ventajas de implementar Runnable contra extender de Thread?
- ¿Cuál es la diferencia de implementar Runnable contra Callable?
- ¿Se puede predecir el orden en el que se imprime el mensaje de la clase Hilos?
- En el archivo *Hilos2.java*, ¿Qué pasa si sacamos la instancia de la clase “h” de t1, es decir, poner h antes de declarar t1?
- Escribe que variables son locales (variables que estan en memoria del hilo) y que variables son compartidas de cada archivo y el por qué. Puedes tomarle captura al código y encerrar en un recuadro dichas variables.
- Contesta las preguntas de la sección de Synchronized
- ¿Cómo podriamos darle un comportamiento diferente a los hilos?
- Escribe lo aprendido sobre esta práctica, así como tus conclusiones. También comparte si tuviste dificultades y los descubrimientos o alguna cosa de interés.

Para esta practica no se usan patrones de Diseño, ni UML, son más archivos sueltos

Por ultimo...



DIAMOND DOGS