



Unidad 7 – Concurrency

Lanzar “Hola mundo Thread”

Crear:

- una clase que herede de Thread que escriba en pantalla “Hola mundo Thread”
- una clase que implemente Runnable que escriba en pantalla “Hola mundo Runnable”.

El main, una vez que haya ambos threads, debe escribir en pantalla “Fin de main”. Verificar que en múltiples ejecuciones del main, se escriban los mensajes en diferente orden.

Yield

Repetir el ejercicio anterior pero invocando al método yield() en el thread y en el Runnable antes de escribir en pantalla. Verificar que la mayoría de las veces se escribe “Fin de main”.

Join

Lanzar un thread que escriba el mensaje “Thread 1” 5 veces. Utilizar el método Thread.sleep(1000) para la pausa. Lanzar simultáneamente otro thread similar que escriba “Thread 2”. Hacer que el main escriba “Fin del main” luego de que ambos threads hayan terminado (utilizar el método join()).

Interrupciones

Lanzar un thread que escriba el valor de un contador (y lo incremente). Hacer que el main interrumpa al thread luego de 5 segundos y escriba el mensaje “Fin del main” luego de que el thread haya terminado. Verificar que el thread no siga escribiendo en pantalla.

CVS y Threads

Lanzar un thread que dé de alta 20 alumnos en un archivo CSV (separado por comas) de alumnos, con el siguiente formato: *NOMBRE,APELLIDO,DNI*

Hacer una clase Ordenador, que tome por parámetro la lista de alumnos a ordenar y otro que indique el tipo de ordenamiento (orden por nombre, por dni, por apellido) y en base a ese parámetro decida con que “Comparator” debe ordenar.

Lanzar otro thread que lea todo el archivo, ordene los alumnos por NOMBRE, y los escriba en pantalla. Dos minutos después, deberá leer de vuelta el archivo y ordenar por APELLIDO, y en otros dos minutos, ordenar por DNI.

Carrera de caballos

Se desea crear un simulador en tiempo real de una carrera de caballos. Para esto es necesario crear un programa java que simule una carrera. Al iniciar la carrera deben comenzar a “correr” 5 caballos en simultáneo. El primero que llega a los 1000 metros es el ganador.

Al finalizar la carrera el programa tiene que mostrar 2 listados:

- El listado de caballos ordenado por nombre y mostrando su posición de llegada

Programación 2

Ingeniería en Informática



- El listado de caballos ordenado por orden de llegada a la meta

Para simular la velocidad del caballo, hay que hacer que el caballo avance a intervalos regulares (sugerido 50ms) una cantidad de metros aleatoria entre su velocidad mínima y máxima (expresada en metros) dada en el siguiente cuadro:

Caballo	Velocidad mínima	Velocidad máxima
Centella	11	14
Tornado	6	24
Suspiro	12	13
Relámpago	10	21
Pegaso	7	22

Sincronización - Sumar y restar

Crear la clase Contador con los métodos void incrementar() que incrementa en uno al contador, void decrementar() que lo decrementa en uno y int dameEstadoActual() quede devuelva su estado actual.

Lanzar concurrentemente 6 threads, 3 que incrementen 1000 veces a un contador y otros 3 que lo decrementen 1000 veces al mismo contador. Tienen que hacerlo de la forma más concurrente posible. Luego de que finalicen los 6 threads, el main debe mostrar el estado actual del contador. Verificar que el resultado sea 0.