Creamos un nuevo proyecto.

Bucle for i: tiene ejecución secuencial, ósea que tras la primera vuelta se da la otra ,etc. hasta terminar.

Lo que vamos a hacer es lanzar otro proceso entremedias

Proceso

Para el procesos tenemos la clase thread, este implementa una interfaz runeable, este a su vez obliga a implementar el método run

Botón derecho >autoimplent >methods> implemnto run

El quantum es una unidad de tiempo que se le da aun proyecto, y pasa otro, que es fifo

Botón dercho references en vez de 88000 caracateres ponemos mas

Hay qeponer star y no run, porque sino seria una ejecución concurrente y no secuencial

Teien eel método sleep, que es un método estatico, esto significa que sellama desde la clase y no desde el objeto

La clase también puede utilizar el sleep por el thread

Proyecto lmbok es un proyecto que te permite agilizar los getters y setter y constructores, es una librería

Botón derecho

Build path

Configure build path

Libreries

Add externally jar

Doble click en lombok

Este busca los archivos de eclipse para añadirles lombok, si ya esta instalado sale con una pimienta

El sleep hace que pierda la cpu

Para que se haga una zona sincronizada sse tiene que hacer con un objeto comn compartido

Para ello ponemos synchronized(contador){…}

Ahora sucede otra cosa, que se queda el primero casi siempre entre losresultados, esto es porque en la zona sincronizada se queda el primer proceso, y hasta que no acabe no entrara otro, pero para que entre otro también tiene que perder la cpu, por lo que probabilidad es baja de que acabe el 1º proceso

La zona sincronizada tiene que ser lo más pequeñas posibles

Si ponemos el contador 1000 y sale 1001 procesos: cuando el proceso nº 1000 sale, y el otro puede que se quede bloqueado al principio antes de entrar en la sincornizacion, llegando a ser 1001 procesos

Solución si cuando cojes la cpu,

synchronized(contador){

if(contador.getContador()<numero){…}}

if(contador.getContador()<numero){

try { sleep…}}}

tambien cambiar do while por while, pero pasar que un proceso se quede al inicio, por lo que sale el mismo error

Ejercicio 1.Se desea que entre cuatro procesos se contabilice sincronizadamente y en orden hasta 20000. Mostrar un listado según van contando mostrando el nombre de proceso que contabiliza y el número en el que se llegan. Cuando lleguen a 5000 individualmente cada proceso finalizará. Por otro lado previamente el proceso principal habrá creado al iniciar el programa un nuevo proceso llamado Cronometro que se encargará de contabilizar los segundos que pasan desde que inicia la aplicación hasta que finaliza (aproximadamente, con hacer un Thread.sleep(1000), incrementamos una unidad en un contador que supuestamente mide los segundos suficiente). Cuando todos los procesos hayan finalizado el proceso padre matará al cronómetro y mostrara cuántos segundos han transcurrido.