

Tarea para PSP02.

Enunciado.

Ejercicio 1):

Crea un programa en el que instances 2 threads (HiloA e HiloB) que en su método run() muestren:

En el HiloA el mensaje NO, hasta un máximo de 30 veces

En el HiloB el mensaje YES, hasta un máximo de 30 veces

En el programa principal mostrará un mensaje "Ejecución en HiloA" y se ejecutará el HiloA; luego mostrará un mensaje "Ejecución en HiloB" y se ejecutará el HiloB; y luego un mensaje "Ejecución en main".

Ejercicio 2):

Crea 3 hilos que se dormirán un tiempo comprendido entre 0 y 10 segundos (tiempo calculado aleatoriamente).

El método run() mostrará un mensaje similar a:

El hiloX va a estar dormido durante 7675 milisegundos. Luego lo dormiremos (esto deberá estar incluido en un bloque try {} catch, atrapar una excepción)

Y finalmente indicaremos: hiloX ha despertado

El tiempo que estará dormido el hilo lo calcularemos en el constructor mediante la fórmula:

`tiempoDormido = ((int) (Math.random() * 10000));`

Cuando lo ejecute comprueba que los hilos van despertando poco a poco.

Ejercicio 3):

A partir del código del documento MainEj3NoSincro y teniendo en cuenta la salida que obtengo tras su ejecución

```
Ya termine.
```

```
Esta es mi salida
```

```
Esta es mi salida
```

```
0 0 1 2 3 1 4 2 5 3 6 4 7 5 6 7
```

Haz los cambios que consideres necesarios para que el resultado de la ejecución de 2 threads en paralelo sea la secuencia de 1 a 7 sin repeticiones ni saltos de números, es decir, utilizando la sincronización quiero obtener algo similar a

```
run:

Esta es mi salida
0 1 2 3 4 5 6 7
Esta es mi salida
0 1 2 3 4 5 6 7
Ya termine.
```

Ejercicio 4):

Diseña un programa que simule la gestión de un punto de venta con tres terminales que acceden a un objeto Cajero que mantiene la cuenta de las localidades vendidas.

Clase Cajero → es el recurso compartido por los terminales (sección crítica)

- Un atributo localidades que contiene la cuenta de las localidades vendidas
- Un método sumarLocalidades que ira contando las localidades vendidas
- Un método mostrarLocalidades que nos dirá cuántas localidades se vendieron

El constructor simplemente inicializa el cajero dándole un nombre.

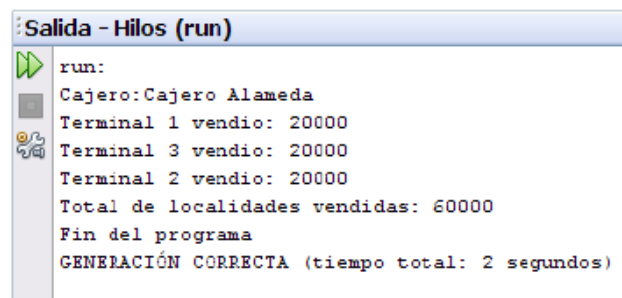
Clase Terminal → (que es un hilo), en su constructor se le pasará un objeto Cajero, que será al que accederá y su nombre. El número máximo de localidades que puede vender cada terminal es de 20000.

En su método run(): mientras tenga localidades las ira vendiendo y anotando la venta general de localidades y la suya propia.

Cuando acabe mostrara cuantas localidades vendió.

MainCajero → es el programa principal desde el que creamos un cajero, 3 terminales que trabajen con ese cajero, ponemos en funcionamiento los 3 terminales y cuando terminen mostramos un mensaje que nos indique cuantas localidades han vendido (debería mostrar 60000, en ejecución sincronizada).

Emplea la sincronización para obtener una salida similar a esta



```
run:
Cajero: Cajero Alameda
Terminal 1 vendio: 20000
Terminal 3 vendio: 20000
Terminal 2 vendio: 20000
Total de localidades vendidas: 60000
Fin del programa
GENERACIÓN CORRECTA (tiempo total: 2 segundos)
```

Criterios de puntuación. Total 10 puntos.

Actividad 1: 2,5 puntos

Actividad 2: 2,5 puntos

Actividad 3: 2,5 puntos

Actividad 4: 2,5 puntos

Nota: se valorará positivamente la gestión de excepciones y código limpio y ordenado.

Recursos necesarios para realizar la Tarea.

- IDE NetBeans/Eclipse.
- Contenidos de la unidad.
- Ejemplos expuestos en el contenido de la unidad.

Indicaciones de entrega.

Debes crear un proyecto de Java, y un paquete para cada ejercicio. Para entregarlos comprímelo todo en un fichero e indica que IDE has utilizados de los dos posibles.

Asegúrate que el nombre no contenga la letra ñ, tildes ni caracteres especiales extraños. Así por ejemplo la alumna **Begoña Sánchez Mañas para la primera unidad del MP de PSP**, debería nombrar esta tarea como...

sanchez_manas_begona_PSP01_Tarea