



Ejercicios UD_2

En estos ejercicios vamos a utilizar los atributos, métodos y constructores vistos hasta este momento.

1. Queremos crear una aplicación en la que a través de la creación de hilos solucionemos que un usuario nos introduzca un texto y el programa nos lo imprima en pantalla 10 veces. En el programa principal iremos indicando los pasos de la aplicación mediante mensajes en pantalla, p.ejem, *"Creamos hilo"*, *"Lanzamos ejecución del hilo"*. Lanza tres hilos y prueba su ejecución. Solucionarlo tanto con la clase Thread como con la implementación de Runnable.

```
> Creamos el hilo
> Lanzamos ejecucion del hilo
Termina programa principal
ejecutando
ejecutando
ejecutando
```

2. Crea una aplicación en la que se lancen dos hilos uno de ellos que sea un contador de números pares y otro de impares(del 1 al 100).Visualiza la prioridad de tus hilos. Cambia la prioridades de los hilos,¿que ocurre?
3. Implementar un programa que escriba un "hola mundo" por cada hilo de ejecución que se cree (seis es un número razonable) y que además indique desde que hilo se imprime. Ejecutalo varias veces para entender su ejecución.
Codifica de nuevo el ejercicio para hacer que cada uno espere un tiempo determinado por el usuario. ¿Qué diferencias se aprecian?.
Analiza en la documentación de Java (*javadoc*) el método join.
¿Que beneficios aporta el método join?
¿Utilizando el método join puedes mejorar la ejecución de este ejercicio?

4. Genera una aplicación en la que utilizaremos las cualidades de los hilos para obtener la media total de una array de 2000 posiciones. Utiliza los hilos para aprovechar las cualidades de la concurrencia y obtener el resultado mas rápido.
5. Vamos a tomar como base el trabajo de un supermercado, el trabajo que realizan las cajeras. Convertiremos la tarea de las cajeras en una acción concurrente donde atenderán a clientes, cada cliente tendrá un numero determinado de productos y el precio de cada producto. Indicaremos en que momento empieza a atender a un cliente, se sacara por pantalla una linea por cada producto(simulando una linea de ticket)y cuando acaba con ese cliente el total de la compra.

La cajera Cajera 1 va a comenzar con la compra del cliente Cliente 1
La cajera Cajera 2 va a comenzar con la compra del cliente Cliente 2
Cajera 1 procesando el producto 1 del cliente Cliente 1precio 2.7euros
Cajera 2 procesando el producto 1 del cliente Cliente 2precio 1.9euros
Cajera 1 procesando el producto 2 del cliente Cliente 1precio 4.0euros
Cajera 2 procesando el producto 2 del cliente Cliente 2precio 3.5euros
Cajera 1 procesando el producto 3 del cliente Cliente 1precio 2.8euros
Cajera 2 procesando el producto 3 del cliente Cliente 2precio 5.9euros
La cajera Cajera 2 finaliza el cobro Cliente 2 y el total de la compra es

6. Vamos a simular la famosa carrera de camellos de las atracciones de las ferias. Lanzamos varios camellos que van a disputar una carrera. Cada uno de ellos avanzara de forma diferente, dependiendo del valor que consigan , en nuestro caso a través de un Random. Por ahora vamos a hacer que la carrera no se termine hasta que todos los camellos lleguen a meta.
Si quisieras que siempre gane un camello en particular,¿cómo intentarías hacerlo?
7. Implementar un programa que lance cuatro threads, cada uno incrementará una variable contador de tipo entero, compartida por todos, 5000 veces y luego saldrá. ¿ Se obtiene el resultado correcto?