

I. Ejercicios UD 2. PROGRAMACIÓN DE HILOS

1. Crea un nuevo proyecto en Eclipse. Crea una clase llamada HiloFor, que herede de thread, y en su método run() imprima 10 veces el mensaje "Estoy dentro del hilo". Crea la clase DentroFuera que cree un objeto HiloFor, lo lance con start() y después imprima 10 veces el mensaje "Estoy fuera del hilo". Ejecuta y analiza el resultado, ¿es el que esperabas?
2. Continuamos ahora el ejercicio anterior creando varios hilos.
Crea una nueva clase llamada HiloC, que herede de thread, que tenga un atributo entero int c. En el constructor, se le pasa el valor de c como parámetro, e imprime el mensaje "Creando hilo c", donde c es valor pasado por parámetro.
En su método run() imprimirá 5 veces el mensaje "Hilo c línea x", donde x es el número de línea. Crea la clase VariosHilos que, en un bucle, cree y lance 5 objetos HiloC, pasando a cada uno los valores 1, 2, 3, 4 y 5 respectivamente. Antes de finalizar el main, imprimirá el mensaje "Todos los hilos creados".
Ejecuta varias veces. Comenta el resultado. ¿Es siempre el mismo? ¿Cuándo se imprime la sentencia "Todos los hilos creados"? ¿Por qué? ¿Cuántos hilos hay en ejecución?
3. Crea dos nuevos hilos llamados HiloTic e HiloTac. El primero debe visualizar en pantalla un bucle infinito con la palabra "TIC", y el segundo con la palabra "TAC". Crea la clase HiloReloj, que en su método main cree e inicie los hilos anteriores.
Introduce después de la sentencia de escritura de cada hilo un retardo de 1 segundo. Utiliza el método sleep(long mils), donde se indica el tiempo en milisegundos. Este método arroja excepciones InterruptedException, por lo que debes añadir un bloque try-cath para capturarla.
4. Repite el ejercicio 2 utilizando Runnable, cambiando las clases HiloC y VariosHilos por HiloCR y VariosHilosR (R de Runnable).
5. Recupera las clases HiloC y VariosHilos del ejercicio 2. Llámalas ahora HiloJoin y PruebaJoin. Utilizando el método join() realiza las modificaciones necesarias para que la ejecución sea:

```
Creando Hilo 1
Hilo 1 línea 1
Hilo 1 línea 2
Hilo 1 línea 3
Hilo 1 línea 4
Hilo 1 línea 5
Creando Hilo 2
Hilo 2 línea 1
Hilo 2 línea 2
...
Hilo 5 línea 5

Todos los hilos creados
```
6. Implemente un programa en java que use hilos independientes. Un hilo imprime números pares del 1 al 10, y otro hilo imprime números impares del 1 al 10. Cree dos instancias (hilos) de cada uno y muestre la salida. Realice el programa utilizando herencia y otro con la interfaz.

7. Crear un programa que lance 10 hilos de ejecución donde a cada hilo se le pasará la base y la altura de un triángulo, y cada hilo ejecutará el cálculo del área de dicho triángulo informando de qué base y qué altura recibió y cuál es el área resultado.
8. Crear un programa que permita realizar el fichaje de entrada a los trabajadores de una empresa. Para ello, al llegar, deberán registrar su nombre y su hora de entrada (datos que se le pedirán al usuario). Teniendo en cuenta que la hora de entrada es a las 9:00, se les mostrará un mensaje tras su fichaje indicando si han llegado a tiempo o se han retrasado.
9. Crear un programa que nos muestre los números de un intervalo, teniendo un retardo entre uno y otro de 1'5 segundos.
10. Crear un programa que tenga almacenadas las estaciones del año en un array. El usuario pasará una de las 4 estaciones, y en función de la elegida, se mostrarán los meses que le corresponden, con una escala de tiempo entre ellos.
11. Escribe una clase llamada SuperMarket que implemente el funcionamiento de N cajas de un supermercado. Los M clientes del supermercado estarán un tiempo aleatorio comprando y con posterioridad seleccionarán de forma aleatoria en qué caja posicionarse para situarse en su cola correspondiente. Cuando les toque el turno serán atendidos procediendo al pago correspondiente e ingresando en la variable Resultados del supermercado. Se deben crear tantos threads como clientes haya y los parámetros M y N se deben pasar como argumentos al programa. Para simplificar la implementación, el valor de pago de cada cliente puede ser aleatorio.