EJERCICIOS PROGRAMACIÓN MULTIHILO II

1. Implementa un programa que reciba a través de sus argumentos una lista de ficheros de texto y cuente el número de caracteres que hay en cada fichero.

   Modifica el programa para que se cree un hilo por cada fichero a contar. Muestra lo que se tarda en contar cada fichero en la forma secuencial, y a continuación empleando hilos. Para calcular el tiempo que tarda en ejecutarse un proceso podemos usar el método `System.currentTimeMillis()` de la siguiente manera:

   ```
   long t_comienzo, t_fin;
   t_comienzo = System.currentTimeMillis();
   Proceso();
   t_fin = System.currentTimeMillis();
   long tiempoTotal = t_fin – t_comienzo;
   System.out.println("El proceso ha tardado: " + tiempototal +
   " miliseg");
   ```

2. Implementa una clase *Contador*, con un atributo entero y 3 métodos (incrementa, decrementa, y getValor)

   Definir 2 tipos de hilo (hiloInc, hiloDec), uno para decrementar y otro para incrementar dicho contador (ejecutarán un bucle de X iteraciones, en las que se modifique el valor del contador según corresponda, y a continuación un Thread.sleep), que cuenten con un atributo interno de tipo Contador.

   En el main, crear 2 hilos, uno de cada clase, que trabajen sobre un mismo objeto Contador.

3. Crea una clase *Saldo*, con un atributo que indique el saldo, y el constructor en el que se dará un valor inicial al saldo. Contendrá también varios métodos:
   - Uno para obtener el saldo (incluir sleep)
   - Otro para modificarlo (incluir sleep)
   - Otro que realice un ingreso. Recibe una cantidad y se la añade, informando por pantalla de quién ha realizado ese ingreso y la cantidad resultante después del ingreso.

   Crear otra clase Thread, que realice ingresos de saldo desde el run.

   En el *main*, crear un objeto compartido Saldo por todos los hilos. Crear 3 hilos, cada uno con un nombre. Esperar a la finalización de todos los hilos para mostrar el valor final del saldo.

4. Crear una clase Cuenta, con un atributo saldo y 3 métodos:
   - Uno que devuelva el importe del saldo
   - Otro que reste al saldo el importe a retirar de la cuenta (modifica el saldo)
   - Otro que realice las comprobaciones para verificar que se puede efectuar la retirada (el importe final debe ser 0 o superior), enviando en caso contrario el aviso correspondiente.

   Posteriormente, crear otra clase denominada RetiradaDeCajero, que extienda de Thread. Cada hilo que se cree de esta clase, recibirá un nombre concreto al ser creado, y trabajará sobre un objeto Cuenta. Dentro de su método run(), se intentará realizar un número X de retiradas.

   En el main, crear 2 hilos, uno para cada titular de la cuenta (Pepe, María).

   Observaciones: el objeto Cuenta, será compartido por varios hilos.