EJERCICIOS PROGRAMACIÓN MULTIHILO

1. Crea 2 hilos en Java que realicen lo siguiente:
   El primero de ellos deberá visualizar en pantalla en un bucle infinito la palabra "TIC". El otro hará lo mismo con la palabra "TAC". Dentro de los bucles emplea el método sleep para generar un retardo e intentar que las 2 palabras aparezcan de manera ordenada. ¿Es esto posible? ¿O se duplican siempre en algún punto?

2. Crea un hilo cuya única funcionalidad sea visualizar el mensaje "Hola Mundo". Crea un programa Java que visualice el mensaje anterior 5 veces creando para ello 5 hilos diferentes usando la clase creada anteriormente. Modifica el mensaje en el hilo para incluir el identificador del hilo.

3. Crea una clase que implemente la interfaz Runnable cuya única funcionalidad sea visualizar el mensaje "Hola Mundo" seguido de una cadena que se recibirá en el constructor y seguido del identificador del hilo. Crea un programa Java que visualice el mensaje anterior creando para ello 5 hilos diferentes. Luego haz que el hilo espere un tiempo proporcional a su identificador, usando para ello el método sleep.

4. Crea un hilo que en su método run() muestre un “NO” hasta un máximo de 30 veces. En el método principal (main) tras ejecutar el start() de tu hilo, mostrará “YES” hasta un máximo de 30 veces.
   Como salida obtendrás una serie alternativa de NOes YESes ya que una vez iniciada la ejecución del thread, el tiempo de la CPU se reparte entre todos los procesos y threads del sistema, con lo cual se intercalan instrucciones del método main() con instrucciones del método run().

   Posible salida:

   ```
   YES YES YES YES YES YES YES YES YES YES YES YES NO NO NO NO NO NO NO NO NO NO NO NO NO NO NO NO
   NO NO NO NO NO YES YES YES YES YES YES YES YES YES YES YES YES YES YES YES YES YES YES
   ```

5. Vamos a hacer un programa similar al anterior pero instanciando 2 threads.
   En el HiloA mostrará el mensaje NO, hasta un maximo de 30 veces
   En el HiloB mostrará 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”. Por ejemplo.

6. Hacer un programa que calcule el factorial y un número de la sucesión de Fibonacci. Cada uno de los cálculos tiene que ser realizado por un hilo independiente. Las clases que implementen cada hilo tienen que llamarse Factorial y Fibonacci. Implementar los procesos de 2 maneras diferentes:
   a) Heredando de la clase Thread
   b) Implementando la interface Runnable

   *Recuerda que factorial(0)=1 y para todo n>0, factorial(n)=n*factorial(n-1).*

   Recuerda que fib(0)=0, fib(1)=1 y para todo n>1, fib(n)=fib(n-1)+fib(n-2).

7. Hacer un programa que calcule los factoriales del 5 al 14 modificando la clase Factorial del ejercicio anterior, para que se visualicen los mensajes:
   empieza el proceso … cálculo del factorial de: XX

acabó el proceso ….. el factorial de: XX es XXXX

En el método main declarar un vector o array de 10 hilos y lanzarlos a ejecución.

8. 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 de la siguiente manera:

```
System.currentTimeMillis()
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");
```