|  |  |
| --- | --- |
| **Título** | DESARROLLO DE APLICACIONES MULTIPLATAFORMA |
| **Módulo** | PROGRAMACIÓN DE SERVICIOS Y PROCESOS |
| **Nombre del alumno/a** | JOSE MURCIA BELMONTE |

El docente debe marcar con una X si es:

* Actividad evaluable x
* Actividad no evaluable
* Información de la tarea

Hilos

Realiza un programa en java que use hilos independientes, en clases diferentes. Un hilo imprime números pares del 1 al 10 y la suma de dichos números, y otro hilo que imprime números impares del 1 al 10 y la suma de dichos números. Cree dos instancias (hilos) de cada uno y muestre la salida

* Tiempo estimado de realización

5 horas.

* Objetivos de la tarea

Indicar los resultados de aprendizaje que se trabajan en la tarea. Indicados en el RD correspondiente del título.

RA1. Desarrollar aplicaciones compuestas por varios procesos reconociendo y  
aplicando principios de programación paralela.

RA2. Desarrollar aplicaciones compuestas por varios hilos de ejecución analizando y  
aplicando librerías específicas del lenguaje de programación

* Formato e instrucciones de entrega

La entrega se realiza desde el aula del módulo del campus virtual, en las fechas indicadas.

El formato de archivo para adjuntar la tarea será el proyecto entero comprimido. Nombrando el archivo de tarea según la siguiente estructura: Apellido1\_Nombre\_Act.X (Unidad X). Ejemplo: Apellido1\_Nombre\_Act.1 (Unidad 1)

Recursos necesarios para la realización

**Orientaciones**

*Para la realización de la actividad tiene que hacer con DIFERENTES HILOS ejecutándose en paralelo.*

**Recursos**

Eclipse

* CORRECCIÓN POR PARTE DEL DOCENTE

NOTA (de 0 a 10).

COMENTARIO DEL DOCENTE. Mínimo 3 líneas.

CRITERIOS EVALUATIVOS DE LA ACTIVIDAD. Indicar los criterios marcados en la ficha de actividad): Enumerar y escribir resultados de aprendizaje.

* RESOLUCIÓN

El ejercicio resuelve el problema de la suma de números pares e impares en dos hilos distintos. Para su resolución se ha optado por no utilizar el método JOIN(). El motivo es que en el momento de la resolución de este ejercicio no se había explicado y, pese a que sé que no es la mejor resolución posible, quería enseñar el sistema que utilicé en aquel momento para resolver el problema del orden de sumandos y resultados.

Dicho lo cual, cuando genero una instancia de cada objeto le asigno un valor (true / false) para indicar si es par o impar. Cuando se ejecuta se lanzan ambos hilos con el método *start()* lo que, unido al método *run()* de la clase CounterWithThreads generará dos hilos distintos. Posteriormente cada instancia llamará al método *counterManagement()* el cual se encarga de hacer la suma y mostrar el resultado. Aquí dentro, se ha forzado a los bucles a recorrer los números pares o los impares, según sea el caso, para evitar duplicar cálculos. A su vez, para evitar que los resultados se muestren desordenados (recordemos que se está ejecutando en multihilo) se almacenan los números por los que va pasando cada bucle y el resultado dentro de un String. Esto logra que a la hora de mostrar los resultados estos sean limpios y ordenados.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

En las capturas anteriores se puede apreciar como al ser multihilo, aunque salgan ordenados los datos, el resultado mostrado cambia de una ejecución a otra.