

Programación Concurrente y de Tiempo Real*

Grado en Ingeniería Informática

Asignación de Prácticas Número 3

Se le plantean a continuación un conjunto de ejercicios sencillos de programación sobre herencia y hebras, que debe resolver de forma individual como complemento a la segunda tercera sesión práctica. Para cada uno, debe desarrollar un programa independiente que lo resuelva. **Documente todo su código con etiquetas (será sometido a análisis con javadoc)**. Si lo desea, puede también agrupar su código en un paquete de clases, aunque no es obligatorio.

1. Ejercicios

1. Utilice la clase `Punto.java` que tiene disponible en la carpeta de la práctica para escribir una clase `Poligono.java`.

2. Utilizando la clase anterior `Poligono`, escriba, mediante uso de herencia por extensión de clase, una clase `Triángulo.java`, que represente a objetos de este tipo. Dótele de nuevos métodos propios de la clase, y sobrescriba algunos de los heredados. Escriba ahora un programa que haga uso de todo lo anterior llamado `UsaTodo.java`, y que cree objetos de todas las clases anteriores, y haga uso de sus métodos. En particular, el programa debe mostrar la compatibilidad de tipos de la superclase con la subclase. Haga lo propio para clases `Pentagono.java`, `Cuadrado.java` y `Hexagono.java`.

3. Utilizando herencia de la clase `Thread`, genere una condición de concurso¹ sobre una variable común n (valor inicial 0) entre dos hilos que respectivamente incrementen y decrementen el mismo número de veces a n . Active ambos hilos concurrentemente utilizando una co-rutina y compruebe que, aunque el valor teórico final debe ser cero, en la práctica no tiene por qué ser así. Guarde su código en `Hilo.java` y `UsaHilo.java`. Escriba una corta tabla de prueba junto con el análisis de lo ocurrido en un documento en `tabla.pdf` (**recuerde que debe desarrollar el documento con WriteLatex**) donde recogerá el número de iteraciones que realizaron los hilos y el valor final obtenido para n .

4. Se desea realizar el escalado de una vector de número enteros de 10^8

*©Antonio Tomeu

¹Una condición de concurso sobre un recurso compartido se produce cuando dos o más *threads* acceden al recurso de forma concurrente **sin control de exclusión mutua**.

componentes². Escriba un programa secuencial `escalaVector.java` que haga el trabajo de escalado. Ahora, escriba una versión paralela multihebrada y llámela `escalaVectorParalelo.java`. Escriba una tabla (**recuerde que debe desarrollar el documento con WriteLatex**) de análisis `tablaCPU.java` que deberá recoger de forma aproximada los picos de uso máximo de la CPU (en tanto por ciento) como una función del tamaño del vector (10^4 , 10^5 , 10^6 ...) y del tipo de procesamiento empleado.

5. Descargue ahora la clase `Cuenta_Banca2.java`, disponible en la carpeta de código **Ejemplos de Clases en Java** del Tema 2 del curso. Simule ahora utilizando hilos mediante implementación de la interfaz `Runnable` a una red de cajeros automáticos, donde cada cajero realiza una operación de ingreso o de reintegro. Provoque ahora una condición de concurso de los hilos contra una instancia de la clase anterior, de forma que la suma neta de las operaciones de todos ellos sea igual a 0. En esta situación, el saldo inicial de la cuenta debería haber permanecido constante. Compruebe que no tiene por qué ser así. Llame a su clase `Cajero.java` y escriba una clase que simule la ejecución de los cajeros `Usa_Cajero.java`. **Nota: no es necesario que suba el fichero Cuenta_Banca.java.**

2. Procedimiento y Plazo de Entrega

Se ha habilitado una tarea de subida en *Moodle* que le permite subir cada fichero que forma parte de los productos de la práctica de forma individual en el formato original. Para ello, suba el primer fichero de la forma habitual, y luego siga la secuencia de etapas que el propio *Moodle* le irá marcando. Recuerde además que:

- Los documentos escritos que no sean ficheros de código deben generarse **obligatoriamente** utilizando Latex, a través del editor WriteLatex, disponible en la nube. Tiene a su disposición en el Campus Virtual un manual que le permitirá desarrollar de forma sencilla y eficiente documentos científicos de alta calidad. Puede encontrar el citado manual en la sección dedicada a Latex en el bloque principal del curso virtual. El url de WriteLatex es: <https://www.writelatex.com/>
- No debe hacer intentos de subida de borradores, versiones de prueba o esquemas de las soluciones. *Moodle* únicamente le permitirá la subida de los ficheros por **una sola vez**.
- La detección de plagio (copia) en los ficheros de las prácticas, o la subida de ficheros vacíos de contenido o cuyo contenido no responda a lo pedido con una extensión mínima razonable, invalidará plenamente la asignación, sin perjuicio de otras acciones disciplinarias que pudieran corresponder.

²Es posible que la máquina virtual necesite más memoria de la habitual. Recuerde que puede modificar la memoria de trabajo de la máquina virtual con el *flag* `-XmsYYYm`, donde *YYY* es la cantidad de memoria en MB.

- El plazo de entrega de la práctica se encuentra fijado en la tarea de subida del Campus Virtual.
- Entregas fuera de este plazo adicional no serán admitidas, salvo causa de fuerza mayor debidamente justificadas mediante documento escrito.
- Se recuerda que la entrega de todas las asignaciones de prácticas es recomendable, tanto un para un correcto seguimiento de la asignatura, como para la evaluación final de prácticas, donde puede ayudar a superar esta según lo establecido en la ficha de la asignatura.
- Se recuerda que la entrega de todas las asignaciones de prácticas es obligatoria, y requisito indispensable para poder superar las prácticas de la asignatura, y por tanto a la asignatura en sí misma.