

Prácticas Concurrencia y Distribución (18/19)

Arno Formella, Anália García Lourenço, Hugo López Fernández, David Olivieri

semana del 21 al 27 de enero

Las prácticas de este curso están organizadas en actividades semanales para mantener un esfuerzo constante a lo largo del curso. Las entregas (¡semanales!) se realizan mediante la herramienta FaiTIC que estará configurada para permitir la subida de ficheros durante las horas de prácticas en el aula de informática.

Las entregas consisten en el fichero según lo pedido en la actividad que se haya publicado **que se sube a la plataforma durante las clases presenciales de prácticas** de cada semana. Dichas entregas sirven al mismo tiempo como **testigo de asistencia** a clases prácticas. No obstante el profesorado de prácticas puede usar otras medidas adicionales para monitorizar dicha asistencia.

Las entregas consisten en la subida de un **único** fichero simple o de tipo archivo (.zip, .rar, .tgz, etc.). Dichos ficheros **siempre** deberían tener nombres que se forman de la siguiente manera:

`Apellido1_Apellido2_Nombre_Grupo.Extensión`

donde

- el `Apellido1`, `Apellido2`, y `Nombre` se entienden como tales,
- `Grupo` es uno de CDI1 hasta CDI6,
- `Extensión` según tipo de fichero que puede ser solamente un fichero `.java` con el código fuente, un solo fichero como archivo que contiene la documentación y el código adicional necesario.

*Ficheros con nombres que no cumplen con esta nomenclatura serán simplemente **ignorados**. No existirá la entrega de ficheros fuera de estos plazos.*

1. Práctica 1: Introducción a la concurrencia en Java

Objetivos: Adquirir conocimientos básicos sobre la forma como está implementada la concurrencia en Java.

Material adicional: Son de especial interés los siguientes enlaces

- <http://docs.oracle.com/javase/7/docs/>
- <http://docs.oracle.com/javase/8/docs/>

- <http://docs.oracle.com/javase/tutorial/essential/concurrency/index.html>
- <http://www.stack.nl/~dimitri/doxygen/index.html>
- <http://en.wikipedia.org/wiki/Markdown>

Requisito general a todos los programas en todos los programas concurrentes es: **siempre termina el programa y todos sus subprocesos/hilos con un mensaje como *Program of exercise X has terminated***, es decir, todos los componentes del programa concurrente terminan correctamente su ejecución.

Las preguntas que aparecen intercaladas en los anunciados tienen como objetivos: animar a la reflexión y al auto-aprendizaje, servir como ejemplos de posibles preguntas en la fase de evaluación (examen), fundamentan la base para los breves informes que se puedan entregar para las actividades.

1. Examina en el manual y con ejemplos las dos formas que provee Java para crear un hilo: la clase `Thread` y la interfaz `Runnable`.
¿Hay alguna diferencia de funcionamiento entre ambas formas? ¿A nivel de diseño, cuál te parece preferible, y por qué?
2. Utiliza la forma con `Runnable` para crear un programa que cree y ejecute tantos hilos como se le indica via línea de comando
3. Utiliza tu programa del apartado anterior y aumenta para
 - que cada hilo imprima en pantalla un mensaje como *Hello world, I'm a java thread number X*.
 - y después de uno o varios segundos (puedes usar un segundo argumento via línea de comando), un mensaje como *Bye, this was thread number X*,

¿Las salidas del programa reflejan lo que has esperado?