



Escuela Técnica Superior de Ingeniería

Prácticas de Programación Concurrente y Distribuida

3º Curso de Grado en Ingeniería Informática

PRÁCTICA 5

Condición de Sincronización en JAVA

El objetivo de la práctica es familiarizarse con los mecanismos de Java para la sincronización de hilos. Para ello, se resolverá el siguiente supuesto:

Una empresa de cromados dispone de un tanque para realizar el tratamiento. La empresa cromar dos tipos de piezas: llantas y parachoques. En el tanque se pueden cromar a la vez dos parachoques, o cinco llantas, aunque también es posible mezclar un parachoques con hasta tres llantas.

El proyecto para solucionar el enunciado contendrá las siguientes clases:

- **Tanque.** La clase `Tanque` mantendrá el estado de ocupación del tanque e implementará los siguientes métodos:
 - **Entrallanta.** Que deberá ser invocado por las llantas cuando quieren acceder al tanque.
 - **Salellanta.** Que deberá ser invocado por las llantas al salir del tanque.
 - **EntraPC.** Que deberá ser invocado por los parachoques cuando quieren entrar en el tanque.

- **SaltePC.** Que deberá ser invocado por los parachoques al salir del tanque.
- **Parachoque.** Representará cada uno de los parachoques mediante un hilo. El hilo se creará heredando de la clase `Thread`. El hilo pondrá un mensaje de inicio indicando su identificador, intentará acceder al tanque usando la clase `Tanque`, permanecerá en él durante 4 segundos y saldrá.
- **Llanta.** Representará cada uno de las llantas mediante un hilo. El hilo se creará implementando el *interface* `Runnable`. El hilo pondrá un mensaje de inicio indicando su identificador, intentará acceder al tanque usando la clase `Tanque`, permanecerá en él un tiempo aleatorio entre 2 y 3 segundos y saldrá.
- **Generador.** Contendrá el método `main` y será quién comience la ejecución. Debe lanzar, de forma aleatoria, llantas y parachoques a intervalos de tiempo entre 1 y 2 segundos. La frecuencia de llegada de las llantas será del 70% y la de parachoques del 30%. Deberá esperar a que finalicen todos los hilos para finalizar.

El control de la concurrencia y la sincronización se realizará en la clase `Tanque`, mediante las primitivas de Java `wait()`, `notify()` y/o `notifyAll()`.

Con objeto de visualizar la correcta evolución de los hilos, se deberá realizar la presentación gráfica en pantalla de su estado. Para tal fin, se deberá crear un *Frame* que contenga un *canvas*. El *canvas* deberá mostrar en cada momento el contenido del tanque y la cola de piezas que están esperando a ser cromadas. En la representación, cada elemento (llanta o parachoque) será identificado por su "id".