



Prácticas de Programación Concurrente y Distribuida

3º Curso de Grado en Ingeniería Informática

PRÁCTICA 4

Condición de Sincronización en JAVA

El objetivo de la práctica es conocer la forma de sincronizar métodos de objetos concurrentes y permitir que los hilos puedan esperar por ciertas condiciones y señalar la ocurrencia de las mismas.

Se tomará como base la pila en su implementación tras la práctica 3 (`PilaLenta`) una vez sincronizados los métodos `Apila` y `Desapila`.

1. Se deberá modificar el método `Apila`, para que cuando un productor intente apilar un elemento, con la pila llena, espere a que haya un hueco para introducirlo. Cada vez que quede un hueco libre en la pila, todos los productores que esperen el hueco intentarán colocar su elemento. Los que no lo consigan volverán a esperar un nuevo hueco. Este intento de apilar el elemento se realizará tres veces, transcurridos los cuales se desistirá del intento de apilar, se lanzará la excepción de que la pila está llena y el hilo finalizará.
2. Se deberá modificar el método `Desapila` para que tenga un comportamiento similar al descrito para `Apila` pero cuando la pila esté vacía.

Con objeto de hacer un mejor seguimiento de las acciones realizadas por los hilos se mostrarán mensajes de consola indicando las acciones que estos realizan. La salida gráfica se mantiene como en la práctica 3.

Para comprobar el correcto funcionamiento del método `Apila` se lanzarán cuatro hilos productores y un hilo consumidor que actuarán de la siguiente forma:

- Los productores tratarán de apilar 15 elementos cada uno a intervalos de tiempo aleatorio entre 1 y 3 segundos.
- El consumidor deberá extraer 15 elementos a intervalos de tiempo aleatorios entre 1 y 3 segundos. Cuando el consumidor haya finalizado de extraer todos sus elementos deberá hacer que los hilos productores que estén esperando hueco en la pila dejen de hacerlo y terminen su ejecución.