

## Problema clásico de sincronización y comunicación de procesos cooperativos

### Práctica 8

#### Enunciado:

Implementar en Java el problema de sincronización y comunicación de procesos que se describe a continuación.

#### Características de funcionamiento:

- 1.- Se tienen **dos procesos: *productor* y *consumidor***, que se tienen que sincronizar y comunicar compartiendo un recurso llamado búffer.
- 2.- Se debe usar **memoria compartida** como modelo de comunicación.
- 3.- Para permitir que el proceso productor y consumidor se comuniquen, se debe tener disponible un **búffer** de elementos que pueda ser llenado por el productor y vaciado por el consumidor.
- 4.- El búffer es de un tamaño definido (fijo), en este caso, será de 6 elementos.
- 5.- El proceso productor genera información (un número entero aleatorio del 0 al 15) que es utilizada por el proceso consumidor.
- 6.- El productor no puede producir un elemento si no hay espacio en el búffer, y deberá esperar hasta que se libere una posición en el búffer.
- 7.- El consumidor no puede tomar un elemento que aún no ha sido producido y debe de esperar hasta que se produzca un elemento en el búffer.
- 8.- El último elemento que se produce es el elemento que se debe de consumir (como una pila).

## Características de implementación:

- Crear una clase **Principal** que contenga la creación del búffer limitado, del hilo productor y del hilo consumidor.
- Crear una clase **Productor** que derive de Thread, que realice lo siguiente:
  - Se ejecute en hilo Productor de manera infinita
  - Se duerma un tiempo aleatorio
  - Se produzca un elemento entero y se introduzca en el búffer, invocando al método *introduceDato* de la clase Buffer
- Crear una clase **Consumidor** que derive de Thread y realice lo siguiente:
  - Se ejecute en hilo Consumidor de manera infinita
  - Se duerma un tiempo aleatorio
  - Se consuma un elemento del búffer, invocando al método *sacaDato* de la clase Buffer
- Crear una clase que se llame **Buffer**, en la cual se declara una estructura estática de 5 elementos que será el enlace entre el proceso productor y consumidor. La clase debe definir los siguientes métodos:

Método *introduceDato* que se encarga de verificar si es posible insertar un elemento en el búffer (de tipo entero). Si el búffer está lleno entonces el proceso productor debe esperar, en caso contrario se debe insertar el dato en el búffer.

Método *sacaDato*, se encarga de verificar que exista por lo menos un elemento en el búffer para ser consumido. Si hay un elemento lo consume, de lo contrario no puede realizar ninguna otra operación.