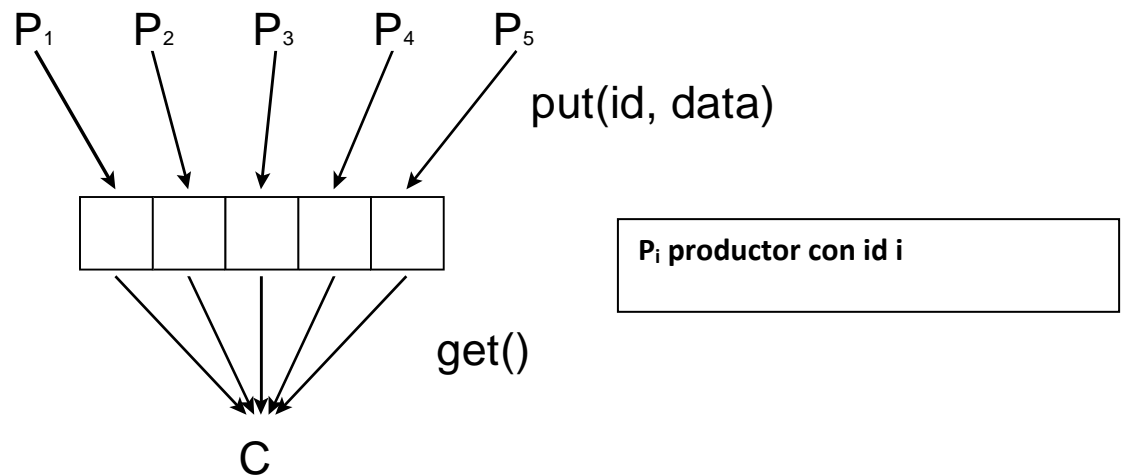


Monitores/Locks

Descripción del sistema

Modelemos un sistema donde hay un Consumidor (**Consumer**) y varios Productores (**Producers**) (**NUM_PRODUCERS=5**). El **Buffer** contiene tantas celdas como productores y cada productor con identificador **id** puede producir (**put**) datos sólo en el **id** de celda del **Buffer**. Por su parte, el consumidor podrá consumir (**get**) de cualquier celda del **Buffer** en cuanto su Productor le haya escrito algún dato. Además, los datos generados por los Productores **deben consumirse en el mismo orden** en que se producen.



Las entidades que participan en este programa son:

- **Consumer.** Una clase que hereda de Thread y modela un consumidor. Un consumidor espera algún dato listo para ser consumido, y consume los datos en el mismo orden en que se generan. Un dato debe ser consumido una sola vez.
- **Producer.** Una clase que hereda de Thread y modela un productor. Cuando el Productor con identificación **id** produce un nuevo dato en la posición **id** del array, no puede volver a producir de nuevo otro dato hasta que se haya consumido el anterior.
- **Table.** Esta clase modela los recursos compartidos, es decir, el **Buffer**. Hay una celda por Productor y la celda en la posición **id** solo puede ser tocada por el Consumidor y la **id** del Productor. En esta clase debes programar estas dos funciones:
 - **void put(int id, int data):** el productor con identificación **id** usa este método para producir un número entero en **Table**, en la posición **id**. El Productor debe esperar

si la celda no está vacía, es decir, si el Consumidor aún no ha consumido el entero anterior.

- **int** `get()`: el consumidor usa este método para consumir un número entero disponible de **Table**. Si no hay datos disponibles, entonces tiene que esperar. Los datos deben consumirse en el mismo orden en que se generan. Cuando el Consumidor consume el número entero en la celda **id**, el productor de esa celda puede colocar otro número entero en esa posición. Por supuesto, el consumidor debe consumir el mismo entero solo una vez.

Debe desarrollar una implementación de **Table** con Monitores o Locks.

Un ejemplo de salida.

```
3 has put 81
    Consumer takes 81 from 3
1 has put 76
3 has put 26
0 has put 60
    Consumer takes 76 from 1
1 has put 59
2 has put 93
    Consumer takes 26 from 3
3 has put 95
4 has put 68
    Consumer takes 60 from 0
0 has put 31
    Consumer takes 59 from 1
1 has put 31
    Consumer takes 93 from 2
2 has put 85
    Consumer takes 95 from 3
3 has put 80
    Consumer takes 68 from 4
4 has put 13
    Consumer takes 31 from 0
...

```

