

Productor-consumidor: Sincronización con colas en Kotlin

[crayon-6a9d613c65ffc515978061/]

Este código implementa un patrón de productor-consumidor utilizando hilos en Kotlin. Explicación del código:

- Se crea una cola (queue) que se utiliza para almacenar elementos. En este caso, se almacenan números enteros.
- Se crea un hilo de productor (producerThread) que agrega elementos a la cola. En este ejemplo, agrega los números del 1 al 5.
- El hilo del productor sincroniza la cola y agrega elementos a ella. Después de agregar un elemento, simula un retraso con `Thread.sleep(1000)` para simular la producción de elementos.
- Se crea un hilo de consumidor (consumerThread) que retira elementos de la cola.
- El hilo del consumidor sincroniza la cola y verifica si está vacía. Si la cola está vacía, el consumidor entra en un estado de espera utilizando `queue.wait()`. Esto permite que el consumidor espere hasta que el productor agregue elementos a la cola.
- Cuando el productor agrega elementos y notifica al consumidor, el consumidor se despierta y retira elementos de la cola utilizando `queue.poll()`.
- Los hilos del productor y el consumidor se inician con `producerThread.start()` y `consumerThread.start()`.
- Se espera a que ambos hilos terminen con `producerThread.join()` y `consumerThread.join()`.
- Finalmente, se imprime «Fin del programa» cuando ambos hilos han terminado de ejecutarse.