

Productor-consumidor con un constructor en el medio. (2 puntos)

Vamos realizar un programa de productores-consumidores con una fase intermedia de procesado. El proceso sería el siguiente:

- Productor:
 - Genera un nuevo producto, cualquier número.
 - Muestra por pantalla el producto generado.
 - Envía el producto a través de un Exchanger.
- Constructor:
 - Obtiene un producto de un productor a través de un Exchanger.
 - Construye el producto elevando al cuadrado el producto del productor.
 - Muestra por pantalla el producto obtenido y el construido con el nombre del hilo.
 - Envía el producto construido a través de un Exchanger.
- Consumidor:
 - Obtiene un producto construido de un constructor a través de un Exchanger.
 - Muestra por pantalla el producto con el nombre del hilo.

Entidades de interés:

- Hilo main:
 1. Prepara el ejecutor.
 2. Lanza las tareas.
 3. Después de un tiempo razonable finalizará el código.
- Productor, constructor y consumidor:
 - Son tareas dentro de un ejecutor.
 - No lanzan otras tareas, tendrán algún bucle interno hasta que se decida finalizar todo.

Para los intercambios sólo está permitido utilizar un Exchanger. Para todo lo demás pueden utilizar las herramientas que crean oportunas. Se podrán tener tantas entidades de cada tipo como se requiera.

Se valorará la calidad, legibilidad del código y el funcionamiento correcto del código. Se penalizarán los errores concurrentes. El código entregado deberá ser funcional para su evaluación.

AmURJCzon (3 puntos)

Procedemos a simular un almacen de productos. Vamos a tener una serie de tareas que se van a ir generando, por el encargado. Estas tarea serán procesadas por los empleados. Tendremos dos tipos tareas de obtención y de guardado.

Cada empleado obtendrá la siguiente tarea disponible. En caso de no existir tareas disponibles, esperará hasta que llegue una tarea. El almacen está compuesto

por una serie de estanterías, listas dentro de Java, que estarán por separado en exclusión mutua. Si la tarea es de guardado, se irá al almacén a almacenar el dato. Si la tarea es de obtención, irá al almacén a por el producto y lo dejará en la cola de salida. En la cola de salida está el repartidor, será la persona que obtendrá todos los paquetes de salida y los irá mostrando por pantalla. Para tener un funcionamiento correcto el encargado generará primero tareas de guardado, y después generará tareas de obtención. Debido a un robo reciente, se ha puesto una cámara de seguridad que se encargará cada 2 segundos de mostrar el tamaño de cada estantería del almacén.

Una vez que se envíen todos los paquetes, todo el mundo se marchará a su casa y se dará por concluida la jornada.

Entidades:

- Encargado y repartidor
 - Es un hilo independiente
- Empleado
 - Es una tarea de un ejecutor
 - Es una tarea que acabará cuando se finalicen todas las tareas.
- Cámara de seguridad
 - Es una tarea de un ejecutor
 - Es una tarea programada

Se puede utilizar todo lo visto durante el curso. Se busca utilizar el mayor número de herramientas de alto nivel conocidas, en los casos que sea posible.

Se valorará la calidad, legibilidad del código y el funcionamiento correcto del código. Se penalizarán los errores concurrentes. El código entregado deberá ser funcional para su evaluación.

Las cámaras inteligentes (1 punto)

La cámara de AmURJCzon además de mostrar los datos, los almacena en una dupla con el número de la estantería y el tamaño de la misma. Partiendo de una lista con este tipo de datos queremos obtener los siguientes datos:

- Las estadísticas del tamaño de las estanterías.
- Número máximo de elementos en cada estantería.
- Una lista con los valores de los tamaños por cada estantería.
- Las estanterías ordenadas por la suma de tamaños.

Se valorará la calidad, legibilidad del código y el funcionamiento correcto del código. Se penalizarán los errores concurrentes. El código entregado deberá ser funcional para su evaluación.