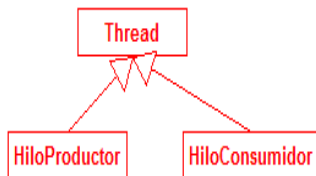


Los siguientes ejercicios utilizan hilos que se sincronizan: comparten un recurso en exclusión mutua y coordinan sus actividades mediante wait() notify().

Ejercicios hilos sincronizados que se coordinan para llevar a cabo una tarea

1. Importa a Eclipse el proyecto Productor Consumidor AL (el .zip) .

La clase Buffer es el recurso a compartir por los hilos productor y consumidor. Complétala tal como indican los comentarios. Será un recurso utilizado en exclusión mutua y sincronizado de tal manera que productor y consumidor coordinen sus actividades.



Añade al proyecto :

- x una clase HiloProductor cuyo constructor recibe como parámetros el nombre del hilo y el buffer compartido. La tarea que hace el hilo es generar 20 valores aleatorios entre 1 y 70 y cada valor lo inserta en el buffer. Después de cada inserción el hilo duerme 10 msg.
- x una clase HiloConsumidor cuyo constructor recibe como parámetros el nombre del hilo y el buffer compartido. La tarea que hace el hilo es borrar del buffer 10 elementos. Después de cada borrado el hilo duerme 30 msg.
- x una clase AppProductorConsumidor que contiene el main(). Incluye aquí el código necesario para crear un hilo productor y dos consumidores, los tres compartiendo el buffer.

```

Hilo Productor Añadido valor 54
Hilo Consumidor 1 Consumido valor 44
Hilo Consumidor 2 Consumido valor 54
Hilo Productor Añadido valor 24
Hilo Productor Añadido valor 40
Hilo Consumidor 2 Consumido valor 24
Hilo Productor Añadido valor 30
Hilo Consumidor 1 Consumido valor 40
Hilo Productor Añadido valor 27
Hilo Productor Añadido valor 5
Hilo Consumidor 2 Consumido valor 30
Hilo Productor Añadido valor 4
Hilo Consumidor 1 Consumido valor 27
Hilo Productor Añadido valor 20
Hilo Productor Añadido valor 13
Hilo Productor Añadido valor 5
Hilo Consumidor 1 Consumido valor 5
Hilo Productor Añadido valor 58
Hilo Consumidor 2 Consumido valor 4
Hilo Productor Añadido valor 37
Hilo Consumidor 1 Consumido valor 20
Hilo Productor Añadido valor 67
  
```

posible ejecución

2. Importa a Eclipse el proyecto Otra Cuenta Bancaria Sincronizada AL y complétalo.

La clase CuentaBancaria es el recurso que se compartirá. Los hilos que lo utilicen tendrán que coordinarse además para que no se pueda sacar si el balance de la cuenta es menor que la cantidad a reintegrar.

Siempre se puede ingresar (no hay condiciones que cumplir para el ingreso), solo se puede sacar si hay dinero suficiente (habrá que esperar si ocurre esto último). Cada vez que se ingresa se avisa para que los hilos que esperan a sacar puedan hacerlo.

La clase HiloDeposito representa al hilo que ingresará cantidades (100€) en la cuenta.

La clase HiloReintegro representa al hilo que ingresará cantidades (50€) en la cuenta.

La clase TestCuenta crea el recurso compartido y crea e inicia dos hilos (uno para ingresos, otro para reintegros) para iniciar la simulación. Hay que esperar a que terminen los hilos y mostrar al final el importe que ha quedado en la cuenta.

posible ejecución

```

Depositando 100.0
, nuevo balance es 100.0
Sacando 50.0
, nuevo balance es 50.0
Sacando 50.0
, nuevo balance es 0.0
Depositando 100.0
, nuevo balance es 100.0
Sacando 50.0
, nuevo balance es 50.0
Sacando 50.0
, nuevo balance es 0.0
Depositando 100.0
, nuevo balance es 100.0
Depositando 100.0
, nuevo balance es 200.0
Depositando 100.0
, nuevo balance es 300.0
Depositando 100.0
, nuevo balance es 400.0
Depositando 100.0
, nuevo balance es 500.0
Estoy en main - la cuenta tiene un saldo de 500.0
  
```