UNIVERSIDAD DE MÁLAGA | uma.es

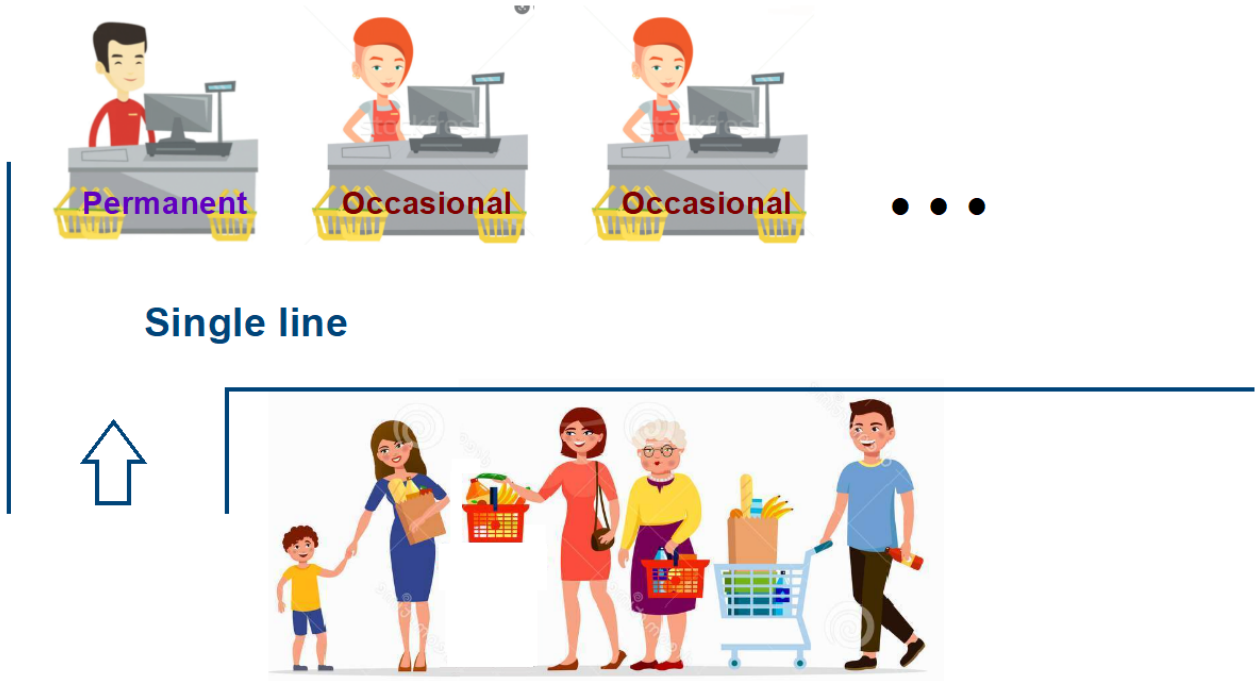
Dpto. de Lenguajes y Ciencias de la Computación

Programación de Sistemas y Concurrencia

Primer examen ordinario.
Curso 2021-2022
21 de Junio de 2022.

Descripción del Sistema

Un **Supermercado** quiere atender a sus **Clientes** dándoles el mejor sevicio. Para ello, el **Supermercado** dispone de un **Cajero** permanente así como de cajeros ocasionales que se añaden cuando hace falta hay muchos clientes esperando para pagar.



Así, cuando llega un nuevo **Cliente**, si el número total de **Cliente**s que están esperando para pagar es mayor que 3 veces el número de **Cajero**s activo en ese momento, se crea un nuevo **Cajero** ocasional. Por ejemplo, si hay 6 **Cliente**s esperando y dos **Cajero**s (el permanente y uno ocasional), y llega un nuevo **Cliente** (el séptimo) se crearía un nuevo **Cajero** ocasional ya que 7 > 3*2. Cuando un **Cajero** ocasional ve que ya no hay ningún **Cliente** esperando, termina su trabajo automáticamente. Supón que puede haber cualquier número de **Cajero**s en el **Supermercado** y que tanto los **Cajero**s como los **Cliente**s se implementan como objetos de la clase **Thread**. Además, cuando el **Supermercado** cierra, todos los **Cliente**s que están dentro del **Supermercado** deben ser atendidos y, finalmente, cuando ya no hay **Cliente**s todos los **Cajero**s que están aún trabajando, incluyendo el permanente, deben terminar. Para modelar este comportamiento, se proporciona un esqueleto con las siguientes clases:

- **Cliente**. Es una hebra que modela el comportamiento de un **Cliente**. El método *run()* de esta clase 'entra' en el Supermercado y cuando paga su compra, termina. **Esta clase está ya implementada.**

- **Cajero**. Es una hebra que modela el comportamiento de los **Cajeros** (tanto del permanente como de los ocasionales). Proporciona la funcion *numCajeros()* that devuelve el número de **Cajero**s que están en ese momento en el **Supermercado**. La clase **Cajero** tiene un constructor con un parámetro booleano que permite crear **Cajero**s permanentes u ocasionales. El método *run()* termina cuando no hay clientes a los que cobrar. **Esta clase está ya implementada.**
- **Supermercado**. Esta clase modela el recurso compartido. Cuando se crea este objeto, también se crea el **Cajero** permanente (ya está implementado). La clase Supermercado proporciona los siguiente métodos que deben ser implementados:
    - `public void fin()`: Esta función la llama el programa principal **Driver** cuando el **Supermercado** no debe admitir a nuevos **Cliente**s (simboliza el cierre del supermercado), aunque los **Cliente**s que ya están en el **Supermercado** deben ser atendidos. Observa que si se ha cerrado el supermercado y no hay **Cliente**s en su interior todos **Cajero**s tienen que terminar su ejecución.
    - `public void nuevoCliente()`: Cada **Cliente** utiliza este método para simular que llega al **Supermercado**. Si el **Supermercado** está cerrado, el cliente simplemente se va (el método termina). Si el **Supermercado** está abierto, y el **Cajero** permanente está desocupado (dormido), el **Cliente** debe despertarlo; en otro caso, si el número de **Cliente**s es mayor que 3 veces el número de **Cajero**s activos en ese momento, debe crear un nuevo **Cajero** ocasional. En cualquier caso, el **Cliente** tiene que *esperar* a que lo atienda cualquier **Cajero** antes de salir del método. Para crear un nuevo **Cajero** ocasional debe llamarse al constructor de la clase **Cajero** (`new` Cajero (**this**, **false**)), e iniciar su ejecución (start).
    - `public boolean permanenteAtiendeCliente(int id)`: el **Cajero** permanente llama a esta función para atender a un cliente. Si hay clientes esperando, atiende a uno de ellos, y devuelve **true**. Si no hay **Cliente**s, entonces el **Cajero** debe esperar a que llegue un nuevo **Cliente**, o a que cierre el **Supermercado**. El método devuelve **false** solo cuando no hay **Clientes** y el **Supermercado** está cerrado.
    - `public boolean ocasionalAtiendeCliente(int id)`: los **Cajero**s ocasionales llaman a esta función para atender a un cliente. Si no hay más clientes, devuelve **false** para indicar que ya puede irse porque no hace falta; si hay algún cliente esperando, lo atiende y devuelve **true**.
      Finalmente, ten en cuenta que los **Cajero**s atienden a los **Clientes** *en cualquier orden*.
- La clase **Driver** contiene a la función *main()* que crea el **Supermercado** y, de forma progresiva, también a los **cliente**s; finalmente, cierra el supermercado cuando ha 'terminado la jornada'.

Nota que el ejercicio tiene dos condiciones de sincronización:

1. **CS-Permanente**: El **Cajero** permanente tiene que esperar, si no hay clientes y el supermercado está aún abierto.
2. **CS-Cliente**. Cada **Cliente**, que ha entrado en el **Supermercado**, tiene que esperar hasta que algún **Cajero** lo atienda

Desarrolla dos implementaciones de la clase Supermecado de los dos tipos siguientes: :
Tipo 1: semáforos **generales** (5 pts.).
Tipo 2: métodos sincronizados o locks (5 pts.).

En el cv se proporcionan las clases para que implementes el sistema. Observa que se ha creado una interfaz Supermercado con las funciones que debes implementar con semáforos o monitores.

Una posible salida del sistema con 30 clientes que entran dos veces al supermercado podría ser:

Cajero permanente espera
Llega cliente 0. Hay 1
Cajero permanente atiende a un cliente. Quedan 0
Cajero permanente espera
Llega cliente 1. Hay 1
Cajero permanente atiende a un cliente. Quedan 0
Cajero permanente espera
Llega cliente 0. Hay 1
Cajero permanente atiende a un cliente. Quedan 0
Llega cliente 2. Hay 1
Llega cliente 4. Hay 2
Llega cliente 1. Hay 3
Cajero permanente atiende a un cliente. Quedan 2
Llega cliente 3. Hay 3
Llega cliente 2. Hay 4
Se crea un cajero nuevo 1
El nuevo cajero 1 comienza a servir a un cliente.
Cajero permanente atiende a un cliente. Quedan 3
Cajero 1 atiende a un cliente: 2
Cajero permanente atiende a un cliente. Quedan 1
Llega cliente 3. Hay 2
Llega cliente 5. Hay 3
Llega cliente 4. Hay 4
Cajero 1 atiende a un cliente: 3
Cajero permanente atiende a un cliente. Quedan 2
Cajero 1 atiende a un cliente: 1
Cajero permanente atiende a un cliente. Quedan 0
Cajero permanente espera
No hay clientes. Cajero 1 termina: 0
Llega cliente 6. Hay 1
Cajero permanente atiende a un cliente. Quedan 0
Llega cliente 5. Hay 1
Llega cliente 6. Hay 2
Cajero permanente atiende a un cliente. Quedan 1
Llega cliente 7. Hay 2
Cajero permanente atiende a un cliente. Quedan 1
Llega cliente 9. Hay 2
Llega cliente 10. Hay 3
Llega cliente 8. Hay 4
Se crea un cajero nuevo 2
El nuevo cajero 2 comienza a servir a un cliente.
Cajero permanente atiende a un cliente. Quedan 3
Cajero 2 atiende a un cliente: 2
Llega cliente 7. Hay 3
Llega cliente 11. Hay 4
Cajero permanente atiende a un cliente. Quedan 3
Llega cliente 13. Hay 4
Llega cliente 12. Hay 5
Cajero 2 atiende a un cliente: 4
Llega cliente 10. Hay 5
Cajero permanente atiende a un cliente. Quedan 4
Llega cliente 9. Hay 5
Llega cliente 8. Hay 6
Llega cliente 14. Hay 7
Se crea un cajero nuevo 3
El nuevo cajero 3 comienza a servir a un cliente.
Cajero 2 atiende a un cliente: 6
Llega cliente 11. Hay 7
Cajero permanente atiende a un cliente. Quedan 6
Cajero 3 atiende a un cliente: 5
Cajero 2 atiende a un cliente: 4
Cajero permanente atiende a un cliente. Quedan 3
Cajero 3 atiende a un cliente: 2
Cajero 2 atiende a un cliente: 1
Cajero permanente atiende a un cliente. Quedan 0
No hay clientes. Cajero 3 termina: 0
Llega cliente 13. Hay 1
Llega cliente 16. Hay 2
Llega cliente 15. Hay 3
Llega cliente 17. Hay 4
Cajero permanente atiende a un cliente. Quedan 3
Cajero 2 atiende a un cliente: 2
Llega cliente 12. Hay 3
Cajero permanente atiende a un cliente. Quedan 2
Cajero 2 atiende a un cliente: 1
Llega cliente 15. Hay 2
Llega cliente 14. Hay 3
Cajero 2 atiende a un cliente: 2
Llega cliente 16. Hay 3
Cajero permanente atiende a un cliente. Quedan 2
Llega cliente 18. Hay 3
Cajero 2 atiende a un cliente: 2
Cajero permanente atiende a un cliente. Quedan 1
Llega cliente 19. Hay 2
Llega cliente 17. Hay 3
Cajero 2 atiende a un cliente: 2
Llega cliente 22. Hay 3
Cajero permanente atiende a un cliente. Quedan 2
Llega cliente 20. Hay 3
Llega cliente 21. Hay 4
Llega cliente 23. Hay 5
Llega cliente 18. Hay 6
Cajero 2 atiende a un cliente: 5
Cajero permanente atiende a un cliente. Quedan 4
Llega cliente 19. Hay 5
Cajero 2 atiende a un cliente: 4
Cajero permanente atiende a un cliente. Quedan 3
Cajero 2 atiende a un cliente: 2
Llega cliente 22. Hay 3
Cajero permanente atiende a un cliente. Quedan 2
Llega cliente 20. Hay 3
Cajero 2 atiende a un cliente: 2
Llega cliente 24. Hay 3
Cajero permanente atiende a un cliente. Quedan 2
Llega cliente 23. Hay 3
Cajero 2 atiende a un cliente: 2
Llega cliente 25. Hay 3
Llega cliente 21. Hay 4
Cajero permanente atiende a un cliente. Quedan 3
Cajero 2 atiende a un cliente: 2
Cajero 2 atiende a un cliente: 1
Cajero permanente atiende a un cliente. Quedan 0
Llega cliente 24. Hay 1
Llega cliente 27. Hay 2
Cajero permanente atiende a un cliente. Quedan 1
Llega cliente 26. Hay 2
Cajero 2 atiende a un cliente: 1
Llega cliente 29. Hay 2
Llega cliente 28. Hay 3
Cajero permanente atiende a un cliente. Quedan 2
Cajero 2 atiende a un cliente: 1
Cajero permanente atiende a un cliente. Quedan 0
No hay clientes. Cajero 2 termina: 0
Llega cliente 25. Hay 1
Llega cliente 27. Hay 2
Llega cliente 29. Hay 3
Llega cliente 26. Hay 4
Se crea un cajero nuevo 4
El nuevo cajero 4 comienza a servir a un cliente.
Cajero permanente atiende a un cliente. Quedan 3
Supermercado cerrado!!!.
Supermercado cerrado. Me voy 28
Cajero 4 atiende a un cliente: 2
Cajero permanente atiende a un cliente. Quedan 1
Cajero permanente atiende a un cliente. Quedan 0
No hay clientes. Cajero 4 termina: 0
Cajero permanente termina: 0