

Silvio Andrés Orozco Vizquerra
Carné 18282
Fecha 23/agosto/2018

Ejercicio 6

La funcionalidad total del programa debe ser que se cree un sistema que este encarga de la organización de los clientes y las cajas rápidas de 10 artículos o menos de un supermercado. Este deberá manejar una cola general y ciertas cantidades de cajas que puedan cambiar de tamaño. Este sistema debe cumplir con optimizar el tiempo de espera de las personas con 10 artículos o menos.

Clase driversuper:

- **Objetivo:** La función es ser el driver de nuestro programa. Aquí se interactuará con el usuario y se pedirá la información necesaria para ejecutar las funcionalidades del programa y de igual forma este será nuestro simulador que generara el paso del tiempo, al igual que los clientes nuevos generados aleatoriamente.
- **Metodos:**
 - `public static void main(String args[])` Para ejecutar las funcionalidades de las demás clases y de nuestro programa.

Clase Scanner:

- **Objetivo:** Ser una instancia para obtener datos del usuario.
- No tiene propiedades.
- **Métodos:**
 - `Scanner()` para instanciar el scanner.
 - `nextInt()` lee el siguiente int ingresado.
 - `Next()` lee el siguiente string ingresado.

Clase Random:

- **Objetivo:** Ser una instancia para obtener números aleatorios en el sistema en rangos específicos.
- No tiene propiedades.
- **Métodos:**
 - `Random()` para instanciar un random.
 - `nextInt()` Genera un int en cierto rango aleatoriamente.

Clase sistema:

- **Objetivo:** La principal función de esta clase es poder inicializar la cantidad de cajas dentro del super, al igual que la cola general. Este se encargará de llevar un orden en las cajas y optimizarlas para mejorar la condición de espera para los clientes.
- **Propiedades (Todas son privadas.):**
 - Cola colageneral. Este es un arreglo dinámico de tipo cola que almacena clientes y será la cola general del sistema.
 - Cola[] cajas. Este es un arreglo de colas que serán las cajas de cobro dentro de nuestro sistema, las cuales pueden atender clientes.
 - Int cantidadcajas. Estas es la cantidad de cajas abiertas por el sistema al inicializarse.
 - Int min; Almacena el paso del tiempo a través del simulador.

- Métodos:
 - Randint(min: int, max: int): Recibe 2 valores un mínimo y un máximo para devolver un valor aleatorio en un intervalo cerrado entre ambos. Devuelve un valor int.
 - Sistema (int cantcajas): Recibe los datos para instanciar un nuevo sistema de supermercado con el numero de cajas que desea abrise siendo el constructor de esta clase.
 - String Mostrarsistema (): Devuelve un valor string que muestra los estados de cada caja con los clientes y su cantidad de articulos respectivamente en orden.
 - Void operar (cliente cnuevo): Recibe un valor de un nuevo cliente para operar el sistema luego de una iteración y con ello, consigue resolver si algún cliente ya termino de pasar con sus articulos, si hay nuevos clientes por agregar o si se decide hacer un cambio de posición de clientes aleatoriamente para optimizar el sistema.

Clase cola:

- Objetivo: La principal función de esta clase es poder crear un arreglo dinámico de clientes que será una cola para que esta pueda cambiar de tamaño a lo largo del sistema. Al igual que será una instancia del sistema.
- Propiedades (Todas son privadas.):
 - Cliente[] arr; Es el arreglo de clientes que almacena.
- Métodos:
 - cola(int tam inicial): Recibe los datos para instanciar un nueva cola con su respectivo tamaño inicial. Por defecto, será 0.
 - Void Agregar(cliente elemento): Recibe un cliente para ser agregado a cierta cola y con ello ser atendido en algún momento.
 - Void Eliminar(int indice): Recibe el índice que se desea eliminar de nuestro arreglo para posteriormente eliminar dicho elemento de nuestro arreglo, y correr todos los demás un espacio.
 - Boolean modificarpaso(): Este método permite modificar a los clientes en el primer lugar de cada caja para comenzar a cobrar sus articulos y devuelve un valor boolean especificando si ya no tienen articulos en su carreta para poder cobrarle y que salgan de la cola.
 - Void cambiar(int indice1, int indice2, cliente cmenosarticulos, cliente cmasarticulos): A través de pedir los datos de 2 clientes y sus índices en la fila, se permite intercambiar sus lugares para optimizar el sistema y que el cliente con menor articulos pueda pasar más rápido en la cola.
 - Int consultar(int indice): Este método permite consultar la cantidad de articulos de algún cliente según su posición en la cola devolviendo el valor de articulos.
 - Cliente consultarcliente(int indice): Este método permite consultar que cliente se encuentra en algún índice de la cola, devolviendo el valor de este cliente.
 - Int Consultartamano(): Este método permite consultar el tamaño actual del arreglo de la cola para ser utilizado por el sistema.

Clase cliente:

- **Objetivo:** La principal función de esta clase es poder crear clientes que utilicen el sistema y puedan beneficiarse del mismo para comprar sus artículos.
- **Propiedades (Todas son privadas.):**
 - `Int articulos`; Este es la cantidad de artículos que el cliente desea comprar.
 - `Int static nodecliente`; Este es el valor de la cantidad de clientes que han utilizado el sistema.
- **Métodos:**
 - `cliente(int cantarticulos)`: Recibe los datos para instanciar un nuevo cliente con su respectiva cantidad de artículos para pasar a la facturación y ser parte de la cola.
 - `Int getarticulos()`; Este método devuelve el valor entero de la cantidad de artículos de un cliente.
 - `Boolean articulosmayora9()`; Este método devuelve un valor de verdadero o falso según se consulte si la cantidad de artículos es mayor a 9 o no.
 - `Boolean setarticulos0()`; Este método resta 1 artículo al cliente si se está siendo cobrado y devuelve un valor de verdadero cuando la cantidad de artículos es 0 y por tanto, se le puede cobrar para salir de la caja.

Extra:

Lo que hizo la persona para fallar épicamente fue lo siguiente:

La chica que trabaja en esta posición era muy buena y realizaba su trabajo con honradez. Sin embargo, cometió el error de iniciar una relación con un compañero de trabajo con quién la pasó muy bien por un par de meses. A pesar de ello, tiempo después este compañero la engañó fríamente y este al ser quien atendía las cajas, siempre recibía la mayor cantidad de personas con la mayor cantidad de artículos. La chica decía que solamente era necesaria una caja rápida y le daba todo el trabajo su ex. Por tanto, cuando el jefe se enteró de esto, tuvo que despedirla y pedir un sistema que optimizara los recursos y que no tomase decisiones personales y perjudicara a una persona y mucho menos al supermercado.