

Estructura de Datos 20/21

Práctica 4

El simulador de colas

Objetivos

- Programar una solución a un problema de simulación utilizando una cola como principal estructura de datos
- Reforzar la habilidad de aprender a aprender

Descripción del laboratorio

Imaginad que estáis haciendo cola (en un tiempo atrás, antes del COVID-19, esto era una realidad) para comprar entradas para ver una película en un cine. En general, podemos pensar que cuantas más ventanillas abiertas, más rápido avanzaremos en la cola. El propietario del cine quiere que sus clientes no hagan mucha cola, pero, por otro lado, no quiere contratar a más cajeros y cajeras de los que sean necesarios. Ahora mismo solo tiene a una persona en ventanilla, y el cine se está haciendo popular. El propietario del cine nos ha pedido que le hagamos una simulación para calcular cuántos cajeros tendría que contratar para que, en hora punta, el tiempo medio de procesamiento de cada cliente (desde que llega hasta que sale con el ticket) no sea superior a (por ejemplo) 2 minutos. ¿Cuántos cajeros necesita contratar para alcanzar este objetivo, sin pasarse?

Tareas a realizar

Antes de comenzar, hacemos las siguientes aclaraciones / simplificaciones:

- Tenemos una única cola de clientes, y la primera persona que llega a la cola, es la primera en ser atendida
- Los clientes llegan cada 15 N, donde N es una unidad de tiempo.
- Si hay una ventanilla disponible, el procesamiento del cliente comienza tan pronto el cliente se acerca a la ventanilla. No hay tiempo de espera. De lo contrario, se tiene que esperar a que haya una ventanilla disponible – es decir, se haya procesado al anterior cliente
- El tiempo que podemos considerar para procesar a cada cliente, desde que llega a la ventanilla hasta que compra la entrada y se marcha, es de 120 N.
- El número máximo de ventanillas es 10, porque no hay más espacio físico en el cine.
- La simulación se hará con 100 clientes.

Con esta información, las tareas a realizar son:

- 1) Implementar una cola (**ListCola**<T>), que es una estructura de datos que nos facilitará hacer la simulación, como una lista simplemente enlazada.
 - a. Definir una interfaz (**ColaInterfaz.java**) con los siguientes tres métodos. Recordad que la interfaz es genérica:
 - i. `public boolean isEmpty();`
 - ii. `public void insertar (T x);`
 - iii. `public T treure();`

- b. Implementar la clase **NodeCola.java**, que tendrá un elemento de tipo T, y un puntero al siguiente Nodo

```
public class ListCua<T> implements ...{
    //atributos
    private NodeCua<T> cabeza;
    private NodeCua<T> cola;
    //constructora crea una cola
    public ListCua()
    {...}
    // true si la cola está vacía.
    public boolean isEmpty()
    {...}
    //Añade un nodo a la cola
    public void insertar (T x)
    {...}
    //devuelve (la cabeza, o la cola?) de la lista, y elimina el nodo
    public T treure()
    {...}
}
```

- 2) Una vez tenemos la cola, modelar los clientes del cine.
- a. Para cada cliente, nos interesa su tiempo de llegada y de salida.
 - b. También nos interesará saber el tiempo total que ha estado haciendo cola, que es la diferencia entre tsalida – tentrada
 - c. Implementar la clase **Cliente.java**
- 3) Con estas dos clases ya estamos preparados para implementar el método simulación del fichero Java que os proporcionamos.
- a. La simulación creará una cola de clientes y un vector de cajeros.
 - b. El número máximo de cajeros es 10, y el de clientes, 100
 - c. Después, la simulación determinará cuánto tiempo se tarda en atender a todos los clientes si únicamente hay un cajero. Posteriormente, se procesa la misma cola con dos cajeros, con tres, etc.
 - d. Al final, compararemos el tiempo medio de compra de entradas según el número de cajeros

A continuación os proporcionamos del método principal del programa en la clase **Practica4.java**:

```

public static void Simulation() {

    clientCinema client;

    ListCua<clientCinema> cuaDeClients = new ListCua<clientCinema>();

    int [] caixerTime = new int [MAX_CASHIERS];

    int tempsTotal, tempsMitja, tempsDeSortida;

    //aquest bucle és per fer la simulació amb tots els caixers
    for (int cashier = 0; cashier < MAX_CASHIERS; cashiers++){

        /*inicializar tcajero para cada simulación. Cargar la cola de Clientes
        en cada iteración. El tiempo total también se resetea a cada
        iteración*/

        while (!cuaDeClients.isEmpty()){
            /*procesamos todos los clientes de la cola con los cajeros de
            aquella simulación (1, 2, etc.) */

            /*procesar los clientes significa sacar el cliente de la cola y
            comprobar el tiempo de llegada. Si el tllegada > tcajero, el cliente se
            puede procesar directamente, y el tiempo de salida será el tiempo de
            entrada más lo que tarde en comprar la entrada. Si tllegada < tcajero, se
            tendrá que esperar, y su tiempo de salida será lo que tarde en
            comprar la entrada, más el tiempo del cajero / ventanilla – que es un
            acumulado de procesar anteriores clientes. El tiempo de proceso de
            cada cliente es constante.*/
            /*Actualizar el tcajero con el tiempo de salida del cliente. Y
            guardarse un contador de tiempo total para poder hacer la media*/
        }
        tempsMitja = tempsTotal / NUM_CLIENTS; //constante = 100;
        //imprimir resultados
    }
}

```

Según nuestra simulación, para atender a 100 clientes y que éstos no tengan que esperar más de 2 minutos (120 segundos) se necesitan 8 cajeros. No hacen falta más; al contrario, tendremos cajeros que estarán esperando. En nuestra simulación, N son segundos.

A continuación os mostramos los resultados, para que tengáis una guía.

Nombre de caixers: 1 Temps mitjà en ser atesos (en segons): 5317

Nombre de caixers: 2 Temps mitjà en ser atesos (en segons): 2325

Nombre de caixers: 3 Temps mitjà en ser atesos (en segons): 1332

Nombre de caixers: 4 Temps mitjà en ser atesos (en segons): 840

Nombre de caixers: 5 Temps mitjà en ser atesos (en segons): 547

Nombre de caixers: 6 Temps mitjà en ser atesos (en segons): 355

Nombre de caixers: 7 Temps mitjà en ser atesos (en segons): 219

Nombre de caixers: 8 Temps mitjà en ser atesos (en segons): 120

Nombre de caixers: 9 Temps mitjà en ser atesos (en segons): 120

Nombre de caixers: 10 Temps mitjà en ser atesos (en segons): 120

Entrega

Fecha: ver tarea correspondiente en el campus virtual

Material: Un fichero ZIP con el proyecto IntelliJ IDEA y un documento RTF con la estrategia / algoritmo implementado.

Criterios (generales) de evaluación

- Práctica no entregada o entregada fuera de plazo = 0
- Entrega parcial, ≤ 3
- Entrega completa, entre 4 y 10. Se valorará el diseño y claridad del código, comentarios, etc.