

### Hoja de trabajo No. 3

**Realizar:** Programa de implementación de pilas.

**Realizarse:** en forma **INDIVIDUAL**.

**Fecha de entrega:** miércoles 29 de julio

**Objetivos:**

- Implementación de las operaciones de una cola.
- Implementación de colas con listas circulares.
- Implementación de colas con arreglos.

**Programa a realizar:**

Se simulación de atención a clientes en las colas de un banco.<sup>1</sup> Considere un banco con cuatro ventanillas. Un cliente entra en al banco en una hora específica  $t_1$  (en minutos desde la apertura del banco) y desea llevar a cabo una transacción en cualquier ventanilla. Se espera que la transacción tome el tiempo necesario  $t_2$  (minutos) para ser completada. Si una de las ventanillas esta libre, la transacción del cliente será atendida inmediatamente. Así el tiempo de permanencia del cliente en el banco será  $t_2$  (minutos).

Pero si no hay ninguna ventanilla libre, el cliente debe escoger la ventanilla que tenga la cola con menor cantidad de clientes y esperar hasta ser atendido. Así el tiempo de permanencia del cliente en el banco será  $t_2$  (minutos) más el tiempo que estuvo en la cola (en minutos).

Su programa deberá calcular el tiempo promedio de permanencia de los clientes en el banco. Tendrá cuatro colas, con un máximo de 10 clientes por cola. Debe generarse un conjunto de parejas de números ( $t_1, t_2$ ) que representan a cada cliente;  $t_1$  es un número entre 1 y 480 que indica el tiempo de llegada del cliente al banco, en minutos desde que el banco abrió. El tiempo  $t_2$  indica la duración de la transacción en minutos y es un número entre 1 y 30. Debe generarse los datos para 50 clientes.

El programa utiliza una **lista** de eventos, ordenada por el tiempo del evento. Inicialmente se guardan en esta fila los eventos de la llegada de los clientes (ordenados por el tiempo de llegada,  $t_1$ ). De esta lista de eventos se retira, ordenadamente por el tiempo, cada evento y se procesa de la siguiente forma:

- Evento de llegada: se coloca el cliente ( $t_1, t_2$ ) en la cola más corta. Si es el único cliente en esa cola, se coloca un evento de salida, en la lista de eventos para salir en  $t_2$ , indicando la cola del cliente.
- Evento de salida: se retira el cliente de la cola en que se encuentra, se calcula el tiempo que ha permanecido en el banco. El siguiente cliente en la cola (si existe alguno) es ahora atendido en la ventanilla y se agrega un evento salida (para esta cola) en la lista de eventos, indicando que el cliente saldrá  $t_2$  más tarde.

El programa se termina cuando la lista de eventos está vacía. Al procesar cada evento de llegada se muestran los clientes, ( $t_1, t_2$ ) que están en la cola a la que es asignado el cliente que llegó. Si ya se ha excedido el límite de clientes en todas las colas se muestra un mensaje y se retiran todos los eventos de llegada de la lista de eventos.

Al finalizar el programa se indica el promedio de tiempo de permanencia de los clientes en el banco.

**Diseño:**

- Debe tenerse una interfaz genérica para las operaciones de la cola.
- Debe construir una clase abstracta que implemente la interfaz y tenga los métodos y variables de instancia comunes a todas las implementaciones de la cola.
- Debe construir una clase que extienda la clase abstracta y que implemente la cola usando listas circulares.
- Debe construir una clase que extienda la clase abstracta y que implemente la cola usando arreglos.

**Tareas:**

- Construir el programa que simule el banco y que emplee la implementación de la cola con listas circulares.

---

<sup>1</sup> Tomado del libro "Estructuras de datos en Pascal.", Aaron M. Tenenbaum y Moshe J. Augenstein



- b. El programa debe permitir que se cambie la declaración de las colas para usar la implementación de colas con arreglos y que funcione sin otros cambios.
- c. Debe dejar evidencia de todo el desarrollo en el repositorio de Subversion.
- d. Debe implementar casos de prueba, como mínimo para cada operación de la cola, en ambas implementaciones (listas circulares y arreglos). Estos casos de prueba también deben estar en el repositorio de Subversion.
- e. El programa debe tener excepciones, tanto para las operaciones de la cola, como para situaciones que se prevean en la simulación del banco.
- f. Elaborar un video con la operación de su programa. Colóquelo en un sitio externo al Sakai y solo mande su enlace.

Debe subir a Sakai todos los productos elaborados en los incisos a (junto a las interfaces y clases que implementan las colas) y d, y los enlaces a su repositorio de Subversion y al video producido.

**Calificación:** su programa debe funcionar para ser calificado.

Aspecto	Puntos
Interfaz y clases de implementación de colas. Deben estar estructuradas como se indica en la sección de diseño en este documento.	20
Documentación generada con Javadoc	5
Casos de prueba: debe existir como mínimo uno por cada operación de las colas en ambas implementaciones. Usar JUnit.	20
Uso del repositorio: existen más de tres versiones guardadas, la última versión es igual a la colocada en el SAKAI	20
Video que muestre el funcionamiento completo del programa (casos usuales, protección de errores de usuario)	10
Funcionamiento del programa	25
<b>TOTAL:</b>	<b>100</b>