

TAD Cola

Práctica 9 TAD ColaCircular

Objetivo

El objetivo de esta práctica es que el alumno refuerce sus conocimientos acerca de las estructuras de datos FIFO (first in, first out) a través de la implementación del TAD ColaCircular.

Descripción general

Una cola circular es aquella en la que ninguno de sus elementos apunta a null, el último nodo se conecta al nodo inicial para formar un círculo. De esta forma todos sus nodos se encuentran conectados. Se implementará esta estructura de datos haciendo uso de nodos y utilizando solamente una referencia al último nodo agregado, de esta forma el inicio de la cola siempre se puede obtener como el siguiente del último nodo.

El TAD ColaCircular trabajará con objetos e implementará la interfaz `InterfazCola` buscando que el tiempo de ejecución de las funciones sea constante. Además se incluirán las funciones *ultimo* y *rotar*.

Material

El material de esta práctica consta de los siguientes archivos:

- **InterfazCola.class** interfaz para el tipo abstracto de datos Cola.
- **Nodo.class** clase que implementa los nodos que sirven para crear una cola.
- **ColaCircular.java** clase que implementará una cola circular.
- **PruebaColaCircular.class** programa para probar la clase `ColaCircular`.
- Documentación:
 - `InterfazCola.html` documentación de la interfaz `InterfazCola`.
 - `Nodo.html` documentación de la clase `Nodo`.

Desarrollo

1. Descargar los archivos `InterfazCola.class`, `Nodo.class`, `ColaCircular.java`, `PruebaColaCircular.class` en el directorio donde se va a desarrollar la práctica.

2. Leer la documentación y revisar el material de la práctica. Recordar que la firma de los métodos proporcionados y de los requeridos en el desarrollo de la práctica no podrán ser modificados.
3. Completar el código en la clase `ColaCircular`, esto requiere:
 - a. Implementar la interfaz `InterfazCola`. Al agregar el primer elemento, éste es a su vez el último elemento. Cuando se agrega un nuevo elemento, este se convierte en el último elemento de la cola circular. Si se intenta eliminar un elemento de una cola circular vacía, no hace nada. El iterador de la cola circular no debe soportar la operación *remove*.
 - b. Incluir el método **`ultimo()`** que devolverá el último objeto de la cola circular sin modificar la cola. Si la cola está vacía entonces devolverá `null`.
 - c. Incluir el método **`rotar()`** que permitirá “girar” la cola una posición haciendo que ahora el último nodo sea el que era el nodo inicial.
4. Probar la implementación de la clase ejecutando el programa `PruebaColaCircular`. Si el programa funciona adecuadamente se verán los siguientes mensajes:

```
Creando una cola circular con 6 elementos...
```

```
¿Esta vacia la cola circular? false  
Tamano de la cola circular: 6  
Primer elemento: Practica 1  
Ultimo elemento: Practica 6
```

```
Iterando sobre la cola circular...
```

```
Practica 1  
Practica 2  
Practica 3  
Practica 4  
Practica 5  
Practica 6
```

```
Rotando una vez la cola circular...
```

```
Ahora el primer elemento es: Practica 2  
Y el ultimo elemento es: Practica 1
```

```
Iterando sobre la cola circular...
```

```
Practica 2  
Practica 3  
Practica 4  
Practica 5
```

Practica 6

Practica 1

Eliminando un elemento...

Tamano de la cola circular: 5

Ahora el primer elemento es: Practica 3

Vaciando la cola circular...

¿Esta vacia la cola circular? true

5. Escribir una tabla con la complejidad de los métodos programados en la clase ColaCircular.

Operación	Tiempo de ejecución
constructor	
estaVacia	
vaciar	
tamano	
agregar	
tomar	
eliminar	
ultimo	
rotar	
iterador	