



TAREA FINAL HITO4 ESTRUCTURA DE DATOS

ESTUDIANTE: SAÚL ESCOBAR SERRANO

¿A QUE SE REFIERE CUANDO SE HABLA DE ESTRUCTURA DE DATOS?

La estructura de datos se refiere a la forma en que los datos se organizan, almacenan y manipulan en un sistema informático. Es un concepto fundamental en la ciencia de la computación y se utiliza para representar la forma en que se organiza la información dentro de una computadora o en una estructura de almacenamiento de datos.

¿QUE SIGNIFICA FIFO?

FIFO es un acrónimo de "First In, First Out", que en español significa "primero en entrar, primero en salir". Es un principio o estrategia utilizado en la gestión de datos o en la planificación de tareas.

En el contexto de las estructuras de datos, una cola es un ejemplo común de implementación de la estrategia FIFO. Los elementos se agregan al final de la cola y se eliminan del principio. Cada vez que se retira un elemento de la cola, el siguiente elemento en la cola se convierte en el primero en ser procesado.

¿MUESTRA LA DIFERENCIA ENTRE LIFO Y FIFO?

La diferencia entre LIFO (Last In, First Out) y FIFO (First In, First Out) radica en el orden en que los elementos son procesados o eliminados de una estructura de datos.

LIFO (Last In, First Out):

- También conocido como el principio "último en entrar, primero en salir".
- El último elemento que se agrega a la estructura es el primero en ser procesado o eliminado.
- Se asemeja a una pila de objetos físicos, donde el último objeto que se coloca encima es el primero en ser retirado.

FIFO (First In, First Out):

- También conocido como el principio "primero en entrar, primero en salir".
- El primer elemento que se agrega a la estructura es el primero en ser procesado o eliminado.
- Se asemeja a una cola de personas esperando su turno, donde la primera persona que llega es la primera en ser atendida.

¿QUÉ ES UNA COLA?

En el contexto de estructuras de datos, una cola es una estructura de datos lineal que sigue el principio FIFO (First In, First Out), lo que significa que el primer elemento que se agrega a la cola es el primero en ser eliminado. Es similar a una cola de personas esperando su turno, donde la persona que llega primero es la primera en ser atendida.

En una cola, los elementos se agregan al final de la cola (operación conocida como "enqueue" o "push") y se eliminan del principio de la cola (operación conocida como "dequeue" o "pop"). Esto garantiza que los elementos se procesen en el mismo orden en que fueron agregados.

¿QUÉ ES QUEUE EN JAVA, UNA QUEUE SERÁ LO MISMO QUE UNA COLA?

En Java, Queue es una interfaz que forma parte del framework de colecciones (`java.util.Queue`). Representa una estructura de datos de cola en la programación orientada a objetos y define un conjunto de métodos para manipular la cola.

En este contexto, una Queue en Java es esencialmente lo mismo que una cola en términos de estructura de datos. Implementa el principio FIFO (First In, First Out), lo que significa que los elementos se agregan al final de la cola y se eliminan del principio.

La interfaz Queue en Java proporciona métodos como `add`, `offer` para agregar elementos a la cola, y métodos como `remove`, `poll` para eliminar y obtener el primer elemento de la cola. Además, la interfaz Queue también incluye métodos adicionales para consultar y manipular la cola, como `peek`, `isEmpty`, `size`, entre otros.

¿QUÉ ES INI O REAR EN UNA COLA?

En una cola, "INI" o "REAR" son términos comúnmente utilizados para referirse al extremo posterior de la cola, es decir, al lugar donde se agregan los nuevos elementos.

"INI" se refiere al inicio o frente de la cola, mientras que "REAR" se refiere al final o parte posterior de la cola. Estos términos pueden variar dependiendo del contexto y las convenciones utilizadas en un lenguaje de programación o en una implementación específica de una cola.

. ¿QUÉ ES FIN O FRONT EN UNA COLA?

En una cola, "FIN" o "FRONT" son los términos utilizados comúnmente para referirse al extremo frontal de la cola, es decir, el lugar donde se eliminan los elementos de la cola.

"FIN" o "FRONT" representa la posición del primer elemento en la cola, también conocido como el elemento más antiguo o el siguiente elemento a ser eliminado.

Cuando se agregan elementos a una cola, se colocan en el extremo posterior (también llamado "REAR" o "INI"). A medida que se eliminan elementos de la cola, se toman del extremo frontal (FIN o FRONT). Por lo tanto, el extremo frontal siempre apunta al próximo elemento que será eliminado de la cola.

¿A QUE SE REFIERE LOS MÉTODOS ESVACIA() Y ESLLENA() EN UNA COLA?

Los métodos `esVacia()` y `esLlena()` se refieren a métodos utilizados en una implementación de cola para verificar el estado de la cola en cuanto a si está vacía o llena.

esVacia(): Este método se utiliza para verificar si la cola no contiene elementos, es decir, si está vacía. Retorna un valor booleano que indica si la cola está vacía o no. Si la cola está vacía, el método `esVacia()` devuelve `true`, de lo contrario, devuelve `false`.

esLlena(): Este método se utiliza para verificar si la cola ha alcanzado su capacidad máxima y está llena. No todas las implementaciones de cola tienen una capacidad máxima definida, pero en aquellas que sí la tienen, este método se utiliza para determinar si la cola ha alcanzado su capacidad máxima. Al igual que `esVacia()`, el método `esLlena()` también devuelve un valor booleano. Si la cola está llena, devuelve `true`, de lo contrario, devuelve `false`.

MÉTODOS ESVACIA() Y ESLLENA() EN UNA COLA

```
public boolean esVacia() {  
    if (ini == 0 && fin == 0)  
        return true;  
    return false;  
}
```

1 usage

```
public boolean esLlena() {  
    if (fin == max)  
        return true;  
    return false;  
}
```

¿QUÉ SON LOS MÉTODOS ESTÁTICOS EN JAVA?

Los métodos estáticos en Java son métodos que se definen en una clase como parte de su definición, en lugar de estar asociados con instancias individuales de la clase.

No requieren una instancia: Los métodos estáticos se pueden llamar directamente utilizando el nombre de la clase seguido del operador punto, sin necesidad de crear un objeto o instancia de la clase que los contiene.

No pueden acceder a variables de instancia: Los métodos estáticos no pueden acceder directamente a variables de instancia (variables no estáticas) ni utilizar la palabra clave "this" dentro de ellos, ya que no están asociados con una instancia particular de la clase.

Acceden a variables estáticas: Los métodos estáticos pueden acceder y manipular variables estáticas (variables declaradas con la palabra clave "static") y otros métodos estáticos de la misma clase.

Se pueden llamar directamente desde otras clases: Los métodos estáticos se pueden invocar directamente desde otras clases sin la necesidad de importar la clase que los contiene, siempre y cuando la clase y el método estén accesibles y visibles en el contexto en el que se hace la llamada.

¿A TRAVÉS DE UN GRÁFICO, MUESTRE LOS MÉTODOS MÍNIMOS QUE DEBERÍA DE TENER UNA COLA?

Cliente	Cola_de_Cliente	Main_Cliente
Cliente (String, String, int, String, String, String)	Cola_de_Cliente ()	Main_Cliente ()
tipo String	ini int	ordenarnos_Por_Edad_Mayores (Cola_de_Cliente, int) void
nombre String	fin int	moviendo_Cliente_sentrados_de_la_Colas (Cola_de_Cliente, String, Cola_de_Cliente) void
apellidos String	clientes Cliente[]	convirtiendo_de_Golden_a_Vip (Cola_de_Cliente, String, String) void
pais String	max int	main (String[]) void
edad int	nroElementos () int	
genero String	adicionar (Cliente) void	
setTipo (String) void	esVacia () boolean	
setPais (String) void	esLlena () boolean	
getPais () String	getIni() int	
muestraCliente () void	getFin() int	
getNombre () String	mostrar () void	
getApellidos () String	vaciar (Cola_de_Cliente) void	
setApellidos (String) void	eliminar () Cliente	
getTipo() String		
getEdad () int		
setEdad (int) void		
getGenero () String		
setGenero (String) void		
setNombre (String) void		