

PRÁCTICA 5

Para implementar el TAD Cola y el TAD Pila, podemos seguir los siguientes pasos:

TAD Cola

Definir una nueva clase llamada "Cola" que tenga un atributo de tipo "LinkedList" para almacenar los elementos de la cola.

Implementar el método "encolar" que añade un elemento al final de la cola utilizando el método "addLast" de la clase "LinkedList".

Implementar el método "desencolar" que elimina y devuelve el primer elemento de la cola utilizando el método "removeFirst" de la clase "LinkedList".

Implementar el método "estaVacia" que devuelve true si la cola está vacía o false en caso contrario utilizando el método "isEmpty" de la clase "LinkedList".

Implementar el método "estaLlena" que devuelve el número de elementos de la cola utilizando el método "capacidad" de la clase "LinkedList".

Aquí está el código para la clase "Cola":

```
import java.util.LinkedList;

public class Cola {
    private LinkedList<Integer> lista;
    private int capacidad;

    public Cola(int capacidad) {
        this.capacidad = capacidad;
        lista = new LinkedList<Integer>();
    }

    public boolean encolar(int elemento) {
        if (lista.size() < capacidad) {
            lista.addLast(elemento);
            return true;
        } else {
            return false;
        }
    }

    public int desencolar() {
        if (!lista.isEmpty()) {
            return lista.removeFirst();
        } else {
            return -1; // Opcional: lanzar una excepción o mostrar un
mensaje de error
        }
    }
}
```

```

    }

    public boolean estaVacia() {
        return lista.isEmpty();
    }

    public boolean estaLlena() {
        return lista.size() == capacidad;
    }
}

```

TAD Pila

Definir una nueva clase llamada "Pila" que tenga un atributo de tipo "LinkedList" para almacenar los elementos de la pila.

Implementar el método "apilar" que añade un elemento al final de la pila utilizando el método "addLast" de la clase "LinkedList".

Implementar el método "desapilar" que elimina y devuelve el último elemento de la pila utilizando el método "removeLast" de la clase "LinkedList".

Implementar el método "estaVacia" que devuelve true si la pila está vacía o false en caso contrario utilizando el método "isEmpty" de la clase "LinkedList".

Implementar el método "estaLlena" que devuelve el número de elementos de la pila utilizando el método "capacidad" de la clase "LinkedList".

Aquí está el código para la clase "Pila":

```

import java.util.LinkedList;

public class Pila {
    private LinkedList<Integer> lista;
    private int capacidad;

    public Pila(int capacidad) {
        this.capacidad = capacidad;
        lista = new LinkedList<Integer>();
    }

    public boolean apilar(int elemento) {
        if (lista.size() < capacidad) {
            lista.addFirst(elemento);
            return true;
        } else {
            return false;
        }
    }
}

```

```

    public int desapilar() {
        if (!lista.isEmpty()) {
            return lista.removeFirst();
        } else {
            return -1; // Opcional: lanzar una excepción o mostrar un
mensaje de error
        }
    }

    public boolean estaVacia() {
        return lista.isEmpty();
    }

    public boolean estaLlena() {
        return lista.size() == capacidad;
    }
}

```

Para probar que ambas clases funcionan correctamente, podemos escribir un programa de prueba que cree una pila y una cola, realice varias operaciones sobre ellas y compruebe que los resultados son los esperados. Aquí hay un ejemplo de programa de prueba:

```

public class Prueba_Pila_Cola {
    public static void main(String[] args) {
        // Prueba de la clase Cola
        Cola cola = new Cola(5);
        cola.encolar(1);
        cola.encolar(2);
        cola.encolar(3);
        cola.encolar(4);
        cola.encolar(5);
        System.out.println("Elementos de la cola:");
        while (!cola.estaVacia()) {
            System.out.println(cola.desencolar());
        }
        cola.desencolar();

        // Prueba de la clase Pila
        Pila pila = new Pila(5);
        pila.apilar(1);
        pila.apilar(2);
        pila.apilar(3);
        pila.apilar(4);
        pila.apilar(5);
        System.out.println("Elementos de la pila:");
        while (!pila.estaVacia()) {

```

```
        System.out.println(pila.desapilar());
    }
    pila.desapilar();
}
}
```