

Modelos de Programación II

Valentina Lopez

Modelo de Programación II

Fundación Universitaria del Areandina
Ingeniería de sistemas.
Agosto 2024
Bogotá D.C.

Resumen

Sustentación inicial de proyecto minibanco, eje 4 de ingeniería de sistemas, universidad Areandina; El cual será presentado por las personas anteriormente mencionadas, para la clase de ‘Modelos de Programación I’

Esta sustentación se centra en explicar el taller propuesto en el desarrollo práctico de una aplicación que utilice interfaces gráficas de usuario (GUIs). Esto implica una serie de pasos que van desde la identificación y comprensión del problema a resolver, el diseño detallado de la interfaz gráfica, la colaboración en equipo para seleccionar la mejor solución, hasta la construcción y refinamiento de la aplicación para satisfacer los requisitos establecidos

Como parte de la explicación de este, se debe confirmar que el video estará cargado en la plataforma YouTube.

Tabla de Contenido

Introducción	3
Objetivos	3
Objetivo general	3
Objetivos específicos	3
Desarrollo de la actividad	4
Parte 1 lista tipo pila - cola	4
Parte 2 Lista enlazada simple - Personas	6
Conclusiones	10
Referencias	10

Introducción

Las listas enlazadas son estructuras de datos que almacenan elementos de manera dinámica, sin necesidad de un tamaño fijo, y permiten una inserción y eliminación eficiente. En esta actividad se examinarán dos tipos: las pilas, que siguen el principio LIFO (último en entrar, primero en salir), y las colas, que siguen el principio FIFO (primero en entrar, primero en salir). El objetivo es implementar un programa para gestionar listas enlazadas simples, realizando operaciones como insertar, contar números pares, calcular promedios y eliminar elementos, para entender mejor su uso y funcionamiento.

Objetivos

Objetivo general

Desarrollar un código de programación en JAVA que implemente listas enlazadas simples, permitiendo la gestión de datos mediante operaciones de inserción, visualización, y manipulación de elementos en estructuras de tipo pila y cola.

Objetivos específicos

1. Implementación de lista enlazada que permita la inserción de # elementos.
2. Implementar funciones para así visualizar los datos en la lista creada.
3. Visualizar de manera ordenada los números presentes en la lista, así dando el entendimiento de la operación
4. Visualización de primer y ultimo dato ingresado a la lista y con ello tener acceso a la estructura

5. Creación de lista enlazada que almacene información específica de personas, permitiendo la visualización de datos.
6. Contabilizar cantidad de elementos en lista, poder eliminar y manipular los datos.

Desarrollo de la actividad

Parte 1 lista tipo pila - cola (Código)

```
//Parte 1 Lista tipo pila y cola

import java.util.LinkedList;
import java.util.Scanner;

public class PilaCola {
    private LinkedList<Integer> lista;

    public PilaCola() {
        lista = new LinkedList<>();
    }

    public void insertar(int elemento) {
        lista.addFirst(elemento);
    }

    public void mostrar() {
        System.out.println("Datos en lista: ");
        for (int num : lista) {
            System.out.print(num + " ");
        }
        System.out.println();
    }

    public int cantidadPares() {
        int count = 0;
        for (int num : lista) {
```

```

        if (num % 2 == 0) {
            count++;
        }
    }
    return count;
}

public double promedio() {
    if (lista.isEmpty()) return 0;
    double sum = 0;
    for (int num : lista) {
        sum += num;
    }
    return sum / lista.size();
}

public int ultimoDato() {
    return lista.isEmpty() ? -1 : lista.getFirst();
}

public static void main(String[] args) {
    PilaCola pilaCola = new PilaCola();
    Scanner scanner = new Scanner(System.in);

    System.out.print("Digite la cantidad de elementos que desea en la
lista: ");
    int n = scanner.nextInt();

    for (int i = 0; i < n; i++) {
        System.out.print("Ingrese numeros deseados: ");
        int elemento = scanner.nextInt();
        pilaCola.insertar(elemento);
    }

    pilaCola.mostrar();
    System.out.println("La cantidad de números pares: " +
pilaCola.cantidadPares());
    System.out.printf("El promedio es: %.2f%n", pilaCola.promedio());
    System.out.println("El último dato de la lista es: " +
pilaCola.ultimoDato());
}

```

```

        scanner.close();
    }
}

```

Prueba del código (Funcionamiento)

```

PS C:\Users\Valentina L> & 'C:\Program Files\Java\jdk-22\bin\java.exe' '--enable-preview' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\Valentina L\AppData\Local\Temp\vscodesws_8e77f\jdt_ws\jdt.ls-java-project\bin' 'PilaCola'
Digite la cantidad de elementos que desea en la lista: 10
Ingrese numeros deseados: 27
Ingrese numeros deseados: 85
Ingrese numeros deseados: 53
Ingrese numeros deseados: 50
Ingrese numeros deseados: 27
Ingrese numeros deseados: 68
Ingrese numeros deseados: 15
Ingrese numeros deseados: 34
Ingrese numeros deseados: 56
Ingrese numeros deseados: 32
Datos en lista:
32 56 34 15 68 27 50 53 85 27
La cantidad de números pares: 5
El promedio es: 44,70
El último dato de la lista es: 32

```

En esta parte podemos ver que la lista generada puede visualizar tanto en pila como en cola los datos ingresados, también puede mostrar la cantidad de números pares relacionados en la lista, el promedio de la lista y el último dato ingresado.

Parte 2 Lista enlazada simple - Personas

(Código)

```

//Parte 2 Lista enlazada simple personas

import java.util.Scanner;
class NodoPersona {
    String codigo;
    String nombre;
    String telefono;
}

```

```

    int edad;
    NodoPersona siguiente;

    public NodoPersona(String codigo, String nombre, String telefono, int
edad) {
        this.codigo = codigo;
        this.nombre = nombre;
        this.telefono = telefono;
        this.edad = edad;
        this.siguiente = null;
    }
}

class ColaPersona {
    private NodoPersona frente;
    private NodoPersona finalNodo;
    private int tamano;

    public ColaPersona() {
        this.frente = null;
        this.finalNodo = null;
        this.tamano = 0;
    }

    public void insertar(String codigo, String nombre, String telefono,
int edad) {
        NodoPersona nuevoNodo = new NodoPersona(codigo, nombre, telefono,
edad);
        if (finalNodo != null) {
            finalNodo.siguiente = nuevoNodo;
        }
        finalNodo = nuevoNodo;
        if (frente == null) {
            frente = nuevoNodo;
        }
        tamano++;
    }

    public void mostrar() {
        NodoPersona actual = frente;

```



```

        System.out.println("Los datos en la lista son: ");
        while (actual != null) {
            System.out.print("(" + actual.codigo + ", " + actual.nombre +
", " + actual.telefono + ", " + actual.edad + "), ");
            actual = actual.siguiente;
        }
        System.out.println();
    }

    public void eliminar() {
        if (frente != null) {
            frente = frente.siguiente;
            tamano--;
            if (frente == null) {
                finalNodo = null;
            }
        }
    }

    public int contar() {
        return tamano;
    }
}

public class MainColaPersona {
    public static void main(String[] args) {
        ColaPersona cola = new ColaPersona();
        Scanner scanner = new Scanner(System.in);

        System.out.print("¿Cuántas personas desea agregar en la lista? ");
        int n = scanner.nextInt();
        scanner.nextLine(); // Consumir el salto de línea

        for (int i = 0; i < n; i++) {
            System.out.print("Ingresa el código: ");
            String codigo = scanner.nextLine();
            System.out.print("Ingresa el nombre: ");
            String nombre = scanner.nextLine();
            System.out.print("Ingresa el teléfono: ");
            String telefono = scanner.nextLine();

```

```

        System.out.print("Ingresa la edad: ");
        int edad = scanner.nextInt();
        scanner.nextLine(); // Consumir el salto de línea
        cola.insertar(codigo, nombre, telefono, edad);
    }

    cola.mostrar();
    cola.eliminar();
    System.out.println("Después de eliminar el primer elemento:");
    cola.mostrar();
    System.out.println("Contar los elementos que quedan en la lista: "
+ cola.contar());

    scanner.close();
}
}

```

Prueba de código (Funcionamiento)

```

PS C:\Users\Valentina L> & 'C:\Program Files\Java\jdk-22\bin\java.exe' '--enable-preview' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Use
rs\Valentina L\AppData\Local\Temp\vscodesws_8e77f\jdt_ws\jdt.ls-java-project\bin' 'MainColaPersona'
¿Cuántas personas desea agregar en la lista? 4
Ingresa el código: 35
Ingresa el nombre: Valentina
Ingresa el teléfono: 310222567
Ingresa la edad: 24
Ingresa el código: 67
Ingresa el nombre: Brandon
Ingresa el teléfono: 310333678
Ingresa la edad: 22
Ingresa el código: 98
Ingresa el nombre: Veronica
Ingresa el teléfono: 310444789
Ingresa la edad: 25
Ingresa el código: 80
Ingresa el nombre: David
Ingresa el teléfono: 310555890
Ingresa la edad: 28
Los datos en la lista son:
(35, Valentina, 310222567, 24), (67, Brandon, 310333678, 22), (98, Veronica, 310444789, 25), (80, David, 310555890, 28),
Después de eliminar el primer elemento:
Los datos en la lista son:
(67, Brandon, 310333678, 22), (98, Veronica, 310444789, 25), (80, David, 310555890, 28),
Contar los elementos que quedan en la lista: 3
PS C:\Users\Valentina L> 

```

En la parte dos del código podemos ver que la lista de personas ingresadas con los datos requeridos de cada uno, se visualizan tanto en pila como en cola, adicional el dato inicial de la

lista, como quedaría si se elimina el primer dato ingresado y los datos que quedan en lista y su cantidad.

Conclusiones

Implementar listas enlazadas simples, como pilas y colas, ayuda a entender mejor las estructuras de datos en programación. Esta actividad muestra cómo estas listas permiten almacenar y manipular datos de manera flexible y eficiente. Al trabajar con inserciones, cálculos de promedios, conteo de números pares y gestión de información, se ha destacado la utilidad de las listas enlazadas y los conceptos LIFO y FIFO. El ejercicio también ha mejorado habilidades prácticas y de resolución de problemas, preparando mejor a los programadores para desafíos más avanzados.

Referencias

La Geekipedia De Ernesto. (2018, 4 mayo). Curso Java Intermedio #33 | Listas tipo pila en Java (Estructura de datos) [Vídeo]. YouTube.

 Curso Java Intermedio #33 | Listas tipo pila en Java (Estructura de datos)

Java Platform SE 8. (s. f.). <https://docs.oracle.com/javase/8/docs/api/>

Chio Code. (2023, 25 mayo). ¡Aprende Paso a Paso Como Usar Listas Enlazadas! [Vídeo]. YouTube.

 ¡Aprende Paso a Paso Como Usar Listas Enlazadas!