

TAREA PROCESUAL ESTRUCTURA DE DATOS HITO 3

PRESENTACION DE PRACTICAS

NOMBRE: JOEL REYNALDO
APELLIDO: CONDORI TUMIRI

1. ¿A que se refiere cuando se habla de estructura de datos?

R.- nos permiten, como desarrolladores, organizar la información de manera eficiente, y diseñar la solución correcta para un determinado problema.

2. ¿Cuales son los TIPOS DE ESTRUCTURA QUE EXISTE?

R.-LISTAS ENLAZADAS, PILAS, COLAS, ARRAYS,

3. ¿Apoyandose en el link adjunto, explique, por que son utiles las estructuras de datos?

R.- es una forma de organizar los datos desde un ordenador, este nos permitirá realizar operaciones muy sencillas

4. ¿Qué es Pila ?

R.- es una lista ordenada que permite almacenar y recuperar datos, siendo el modo de acceso a sus elementos de tipo LIFO(LAST IN, FIRST OUT) ULTIMO EN ENTRAR PRIMERO EN SALIR

5. ¿Qué es STACK en java, una STACK será lo mismo que una PILA?

R.- STACK .- es una clase de las llamadas de tipo LIFO ósea que STACK es lo mismo que PILA

• 6. ¿Qué es TOPE en una PILA?

- R.- Una colección de datos a los cuales se les puede acceder mediante un extremo, que se conoce generalmente como tope

7. ¿Que es MAX en una PILA?

- R.- Es la máxima cantidad de elementos que puede tener almacenada una pila

• 8. ¿A que se refiere los metodos esVacia() y esLLena() en una PILA?

- R.- esVacia() es cuando la pila no contiene elementos almacenados, de forma que para realizar cambios, primero habrá que almacenar información en la pila
- El método esLLena() es cuando el tope de la pila es igual al max, impidiendo poder seguir llenando la pila.

9. ¿Que son los metodos estaticos en java?

R.- Un método estático es un método que tiene sentido invocarla sin crear previamente ningún objeto

10. ¿A traves de un grafico, muestre los metodos minimos que deberia de tener una PILA?

R.-

```
13 usages
public class PilaCliente
{
    9 usages
    private int tope;
    4 usages
    private int max;
    3 usages
    private Cliente[] pilita;

    4 usages
    public PilaCliente(int max)
    {
        this.max = max;
        this.tope = 0;
        this.pilita = new Cliente[this.max+1];
    }

    //tope es la cantidad de elementos que tiene una pila
    //max es la cantidad maxima que tiene una pila
    10 usages
    public boolean estaVacia()
    {
        if(this.tope == 0)
        {
            return true;
        }
        else
        {
            return false;
        }
    }
}
```

```
public boolean estaLleno()
{
    if(this.tope == this.max)
    {
        return true;
    }
    else
    {
        return false;
    }
}

public int numeroElementos()
{
    return this.tope;
}

10 usages
public void insertar(Cliente cliente)
{
    if(!estaLleno())
    {
        this.tope=this.tope+1;
        this.pilita[this.tope] = cliente;
    }
    else
    {
        System.out.println("La pila esta llena");
    }
}
```

```
public Cliente eliminar()
{
    Cliente elementoEliminado= null;
    if(!estaVacia())
    {
        elementoEliminado = this.pilita[this.tope];
        this.tope--;
    }
    else
    {
        System.out.println("La pila esta vacia");
    }
    return elementoEliminado;
}

3 usages
public void mostrar(){
    Cliente elem=null;
    if(estaVacia()){
        System.out.println("La pila esta vacia");
    } else {
        System.out.println("Elementos de la pila");
        PilaCliente aux = new PilaCliente(this.max);
        while(!estaVacia()){
            elem = this.eliminar();
            aux.insertar(elem);
            elem.mostrarDatos();
        }
        vaciar(aux);
    }
}

3 usages
public void vaciar(PilaCliente a){
    while (!a.estaVacia()) {
        insertar(a.eliminar());
    }
}
```


21 usages

```
public class Cliente
{
    4 usages
    private String nombre;
    4 usages
    private String apellido;
    4 usages
    private String direccion;
    4 usages
    private int edad;
    4 usages
    private String genero;

    5 usages
    public Cliente(String nombre, String apellido, String direccion, int edad, String genero)
    {
        this.nombre = nombre;
        this.apellido = apellido;
        this.direccion = direccion;
        this.edad = edad;
        this.genero = genero;
    }

    public String getNombre() { return nombre; }

    public String getApellido() { return apellido; }

    public String getDireccion() { return direccion; }

    1 usage
    public int getEdad() { return edad; }

    2 usages
    public String getGenero() { return genero; }

    public void setNombre(String nombre) { this.nombre = nombre; }

    public void setApellido(String apellido) { this.apellido = apellido; }
```

1 usage

```
public void setDireccion(String direccion) { this.direccion = direccion; }
```

```
public void setEdad(int edad) { this.edad = edad; }
```

```
public void setGenero(String genero) { this.genero = genero; }
```

3 usages

```
public void mostrarDatos()
{
    System.out.println("Nombre: " + this.nombre);
    System.out.println("Apellido: " + this.apellido);
    System.out.println("Direccion: " + this.direccion);
    System.out.println("Edad: " + this.edad);
    System.out.println("Genero: " + this.genero);
}
}
```

11. Crear las clases necesarias para la pila de clientes

```
public class Main
{
    public static void main(String[] args) {
        Cliente cliente1 = new Cliente( nombre: "Carlos", apellido: "Marcelo", direccion: "Calle 1", edad: 20, genero: "Masculino");
        Cliente cliente2 = new Cliente( nombre: "Lineth", apellido: "Santa", direccion: "Calle 2", edad: 25, genero: "Femenino");
        Cliente cliente3 = new Cliente( nombre: "Otavio", apellido: "Suarez", direccion: "Calle 3", edad: 30, genero: "Masculino");
        Cliente cliente4 = new Cliente( nombre: "Ana", apellido: "Marta", direccion: "Calle 4", edad: 35, genero: "Femenino");
        Cliente cliente5 = new Cliente( nombre: "Juan", apellido: "Gutierrez", direccion: "Calle 5", edad: 40, genero: "Masculino");

        PilaCliente pila = new PilaCliente( max: 5);
        pila.insertar(cliente1);
        pila.insertar(cliente2);
        pila.insertar(cliente3);
        pila.insertar(cliente4);
        pila.insertar(cliente5);
        pila.mostrar();
    }
}
```

```
"C:\Program Files\Java\jdk-18.0.2.1\bin\java.exe" -javaagent:C:\Users\PC\AppData\Local\JetBrains\Toolbox\apps\IDEA-U\ch-0\221.5921.22\lib\idea
```

Elementos de la pila

Nombre: Juan

Apellido: Gutierrez

Direccion: Calle 5

Edad: 40

Genero: Masculino

Nombre: Ana

Apellido: Marta

Direccion: Calle 4

Edad: 35

Genero: Femenino

12. Determinar cuantos CLIENTES son mayores de 20 años

```
public static void mayoresCientaEdad(PilaCliente pila, int edadMayor){  
    int contador = 0;  
    Cliente elem = null;  
    if(pila.estaVacia()){  
        System.out.println("La pila esta vacia");  
    } else {  
        while(!pila.estaVacia()){  
            elem = pila.eliminar();  
            if(elem.getEdad() > edadMayor){  
                contador++;  
            }  
        }  
    }  
    System.out.println("La cantidad de clientes con mas de " + edadMayor + " son: " + contador);  
}
```

"C:\Program Files\Java\jdk-18.0.2.1\bin\java.exe" -javaagent:C:\Users\PC\AppData

La cantidad de clientes con mas de 20 son: 4

Process finished with exit code 0

13. Mover el k-esimo element al final de la pila

```
usage
public static void moverKesimo(PilaCliente pila, Cliente valorTope){
    PilaCliente aux= new PilaCliente( max: 10);
    Cliente elem = null;
    while(!pila.estaVacia()){
        elem = pila.eliminar();
        if(elem != valorTope){
            aux.insertar(elem);
        }
    }
    pila.vaciar(aux);
    pila.insertar(valorTope);
    pila.mostrar();
}
```

Edad: 30

Genero: Femenino

Nombre: Otavio

Apellido: Suarez

Direccion: Calle 3

Edad: 30

Genero: Masculino

Nombre: Carlos

Apellido: Marcelo

Direccion: Calle 1

Edad: 20

Genero: Masculino

14. Cambiar la direccion de algunos CLIENTES de la PILA

```
1 usage
2
3 public static void asignarDireccion(PilaCliente pila, String nuevaDireccion){
4     Cliente elem = null;
5     if(pila.estaVacia()){
6         System.out.println("La pila esta vacia");
7     } else {
8         while(!pila.estaVacia()){
9             elem = pila.eliminar();
10            if(elem.getGenero().equals("Femenino")){
11                elem.setDireccion(nuevaDireccion);
12                elem.mostrarDatos();
13            }
14        }
15    }
16    pila.mostrar();
17 }
```

C:\Program Files\Java\jdk-18.0.2.1\bin\java.exe -javaagent:C:\Users\PC\AppData\Local\JetBrains\Tool

```
Nombre: Ana
Apellido: Marta
Direccion: Ceja
Edad: 35
Genero: Femenino
Nombre: Lineth
Apellido: Santa
Direccion: Ceja
Edad: 25
Genero: Femenino
La pila esta vacia
```

15. Mover ITEMS de la PILA

usage

```
public static void reordenaPila(PilaCliente pila){
    PilaCliente aux= new PilaCliente( max: 10);
    Cliente valorExtraidoPila=null;
    while(!pila.estaVacia()){
        valorExtraidoPila = pila.eliminar();
        if(valorExtraidoPila.getGenero().equals("Femenino")){
            valorExtraidoPila.mostrarDatos();}
        else{
            aux.insertar(valorExtraidoPila);
        }
    }
    pila.vaciar(aux);
    pila.mostrar();
}
```

"C:\Program Files\Java\jdk-18.0.2.1\bin\java.exe" -javaagent:C:\Users\PC\AppData

Nombre: Ana

Apellido: Marta

Direccion: Calle 4

Edad: 35

Genero: Femenino

Nombre: Lineth

Apellido: Santa

Direccion: Calle 2

Edad: 25

Genero: Femenino

Elementos de la pila

Nombre: Juan

Apellido: Gutierrez

Direccion: Calle 5

Edad: 40

Genero: Masculino

Nombre: Otavio

Apellido: Suarez

Direccion: Calle 3

Edad: 30

Genero: Masculino

Nombre: Carlos

Apellido: Marcelo

Direccion: Calle 1

Edad: 20

Genero: Masculino

Process finished with exit code 0