ESTRUCTURA DE DATOS

PATRICIO ALFREDO QUISPE CONDORI

SIS15240990

WILLIAM RODDY BARRA PAREDES

¿QUE ES ESTRUCTURA DE DATOS?

Estructura de datos es aquello que nos permite, como desarrolladores, organizar la información de manera eficiente, y en definitiva diseñar la solución correcta para un determinado problema.

TIPOS DE ESTRUCTURA

En programación estructurada se uttilizan tres tipos de estructuras:

SECUENCIALES: Aquellas que se ejecutan una después de otra siguiendo el orden en que se han escrito.

DECISIÓN: Que permiten omitir parte del código o seleccionar el flujo de ejecución de entre dos o mas alternativas.

ITERATIVAS: Se utilizan para repetir la ejecución de cierta parte del programa.

PILA Una pila se define formalmente como una colección de datos a los cuales se puede acceder mediante un extremo.

TOPE Numero de elementos que tiene la pila.

MAX Numero máximo de elementos que soporta la pila.

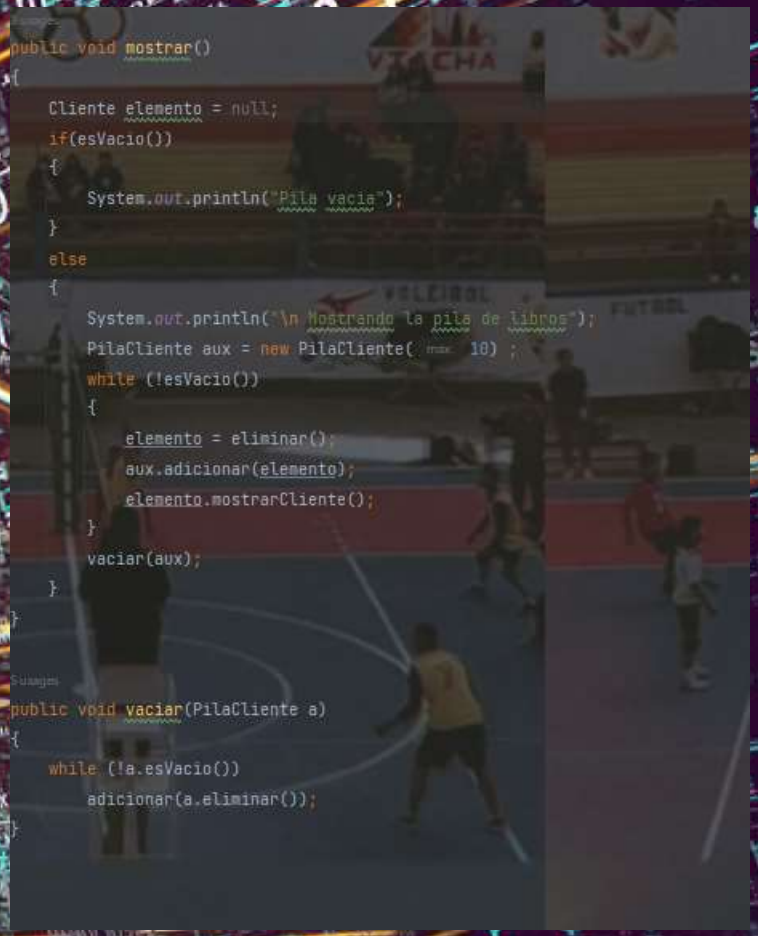
STACK Un objeto de la clase Stack es una pila. Permite almacenar objetos y luego recuperarlos en el orden inverso en el cual se insertan.

CLASE CLIENTE


PilaCliente.java
Cliente.java
main.java
package
public class Cliente {
private String nombres;
private String apellidos;
private int edad;
private String direccion;
private String genero;
public Cliente(String nombres, String apellidos, int edad, String direccion, String genero){
this.nombres = nombres;
this.apellidos = apellidos;
this.edad = edad;
this.direccion = direccion;
this.genero = genero;
}
public String getNombres() {
return nombres;
}
public void setNombres(String nombres) {
this.nombres = nombres;
}
public String getApellidos() {
return apellidos;
}
public void setApellidos(String apellidos) {
this.apellidos = apellidos;
}
public int getEdad() {
return edad;
}


PilaCliente.java
Cliente.java
main.java
package
}
public int getEdad() {
return edad;
}
public void setEdad(int edad) {
this.edad = edad;
}
public String getDireccion() {
return direccion;
}
public void setDireccion(String direccion) {
this.direccion = direccion;
}
public String getGenero() {
return genero;
}
public void setGenero(String genero) {
this.genero = genero;
}
public void mostrarCliente()
{
System.out.println(" >> Cliente");
System.out.println("NOMBRE: " + this.getNombres());
System.out.println("APELLIDO: " + this.getApellidos());
System.out.println("EDAD: " + this.getEdad());
System.out.println("DIRECCION: " + this.getDireccion());
System.out.println("GENERO: " + this.getGenero());
}
}

PILACLIENTE



```
public void mostrar()
{
    Cliente elemento = null;
    if(esVacio())
    {
        System.out.println("Pila vacia");
    }
    else
    {
        System.out.println("\n Mostrando la pila de libros");
        PilaCliente aux = new PilaCliente( max: 10) ;
        while (!esVacio())
        {
            elemento = eliminar();
            aux.adicionar(elemento);
            elemento.mostrarCliente();
        }
        vaciar(aux);
    }
}

Sugerir
public void vaciar(PilaCliente a)
{
    while (!a.esVacio())
        adicionar(a.eliminar());
}
```

CLASE CLIENTE

PilaCliente.java | Cliente.java | main.java | package

```java
package HIT03;

public class main {
    public static void main(String[]args){
        Cliente client1 = new Cliente( nombres: "Mariana", apellidos: "Palacios", edad: 18, direccion: "Villa Jardin", genero: "Femenino");
        Cliente client2 = new Cliente( nombres: "Alfredo", apellidos: "Quispe", edad: 19, direccion: "Viacha", genero: "Masculino");
        Cliente client3 = new Cliente( nombres: "Marcos", apellidos: "Mollo", edad: 25, direccion: "Galpon", genero: "Masculino");
        Cliente client4 = new Cliente( nombres: "Aleli", apellidos: "Valeriano", edad: 22, direccion: "San Juanito", genero: "Femenino");
        Cliente client5 = new Cliente( nombres: "Abzalon", apellidos: "Valeriano", edad: 25, direccion: "Vinto CBBA", genero: "Masculino");


        PilaCliente Clientes = new PilaCliente( max: 10);

        Clientes.adicionar(client1);
        Clientes.adicionar(client2);
        Clientes.adicionar(client3);
        Clientes.adicionar(client4);
        Clientes.adicionar(client5);

        Clientes.mostrar();
    }
```

12. Determinar cuántos **CLIENTES** son mayores de 20 años.

- El método deberá llamarse **mayoresCiertaEdad(Pila, edadMayor)**
- El método debe ser creado en la clase **MAIN** como un método estático.
- El método recibe 2 parámetros
  - La Pila de Clientes
  - El valor de la edad.
- Adjuntar los siguientes
  - El **código** del método que resuelve el problema.
  - Una **imagen** de la salida de la consola.

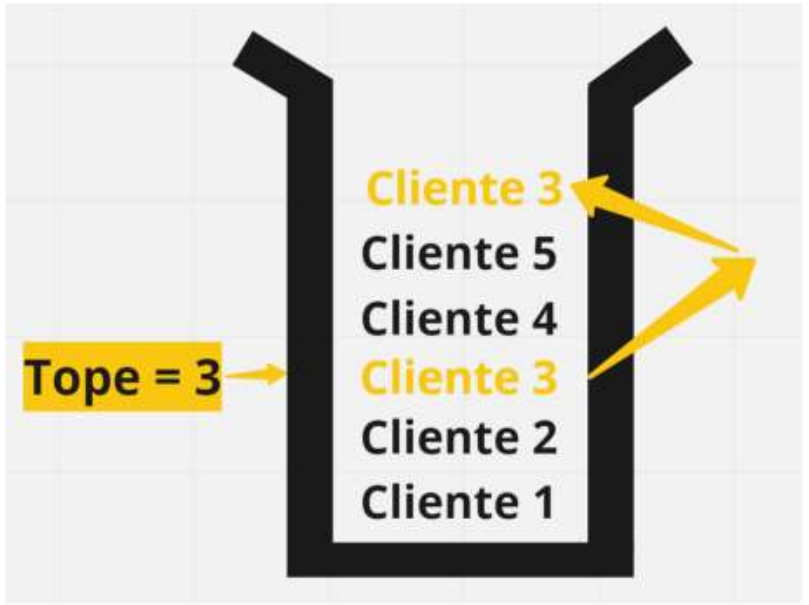
```java
//
public static void ClienteMayorDeEdad(PilaCliente  pila, int cantidad)
{
    PilaCliente  aux = new PilaCliente( max: 10) ;
    Cliente Clienteeliminado = null;
    int mayores = 0;

    while (!pila.esVacio())
    {
        Clienteeliminado = pila.eliminar();

        if (Clienteeliminado.getEdad() > cantidad)
        {
            mayores = mayores + 1;
        }
        aux.adicionar(Clienteeliminado);
    }
    pila.vaciar(aux);
    System.out.println("Edad mayor a 50 es: " + mayores);
}
```

```
main
"C:\Program Files\Java\jdk1.8.0_111\bin\java.exe" ...
Edad mayor a 20 es: 4
```

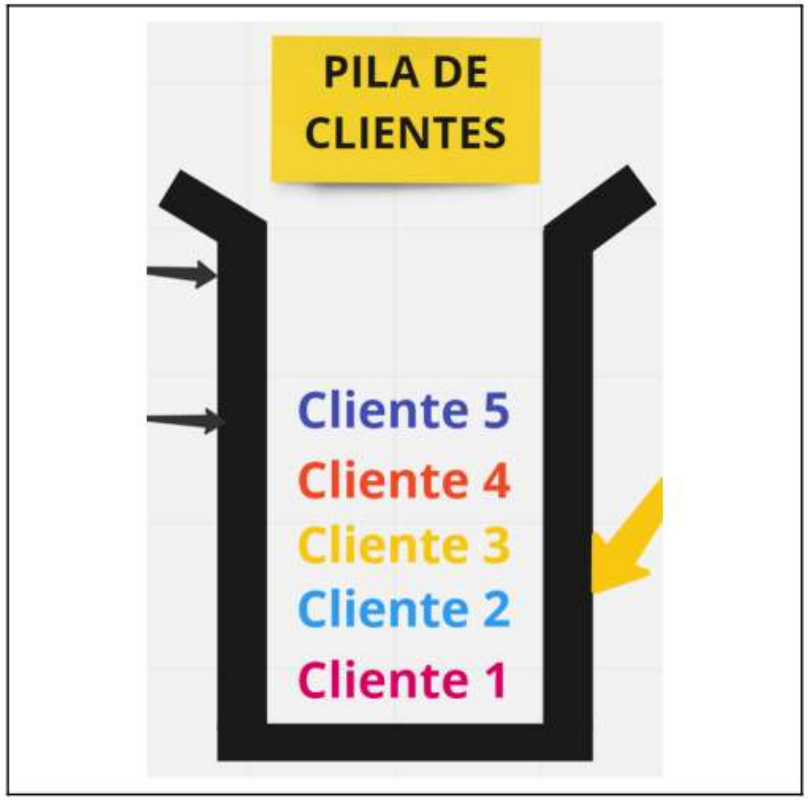
13. Mover el **k-ésimo** elemento al final de la pila.



- El método deberá llamarse **kEsimoPosicion(Pila, valorTope)**
- El método debe ser creado en la clase **MAIN** como un método estático.
- El método recibe 2 parámetros
  - La Pila de Clientes
  - El valor(int) de la posición que moverá al final de la pila.
- Adjuntar los siguientes
  - El **código** del método que resuelve el problema.
  - Una **imagen** de la salida de la consola.


//
public static void KesimoPosicion(PilaCliente pila, int valorTope) {
PilaCliente aux = new PilaCliente(max: 10);
Cliente valor = null;
while (pila.esVacio() == false) {
if (pila.NroElem() != valorTope) {
aux.adicionar(pila.eliminar());
} else {
valor = pila.eliminar();
}
}
pila.vaciar(aux);
pila.adicionar(valor);
pila.mostrar();
}
Clientes
NOMBRE: Marcos
APELLIDO: Mollo
EDAD: 25
DIRECCION: Galpon
GENERO: Masculino
Clientes
NOMBRE: Sebastian
APELLIDO: Huanca
EDAD: 60
DIRECCION: Caranavi
GENERO: Masculino
Clientes
NOMBRE: Abzalon
APELLIDO: Valeriano
EDAD: 25
DIRECCION: Vinto CBBA
GENERO: Masculino
Clientes
NOMBRE: Aleli
APELLIDO: Valeriano
EDAD: 22
DIRECCION: San Juanito
GENERO: Femenino
Clientes
NOMBRE: Alfredo
APELLIDO: Quispe
EDAD: 19
DIRECCION: Viacha
GENERO: Masculino
Clientes
NOMBRE: Mariana
APELLIDO: Palacios
EDAD: 18
DIRECCION: Villa Jardin
GENERO: Femenino
Process finished with exit code 0

14. Cambiar la dirección de algunos **CLIENTES** de la **PILA**.



- El método deberá llamarse **asignaDireccion(Pila, nuevaDireccion)**
- El método debe ser creado en la clase **MAIN** como un método estático.
- El método recibe 2 parámetros
  - La Pila de Clientes
  - El valor(String) de la nueva dirección.
- **Cambiar la dirección del cliente** siempre y cuando el género sea **FEMENINO**.
- Adjuntar los siguientes
  - El **código** del método que resuelve el problema.
  - Una **imagen** de la salida de la consola.

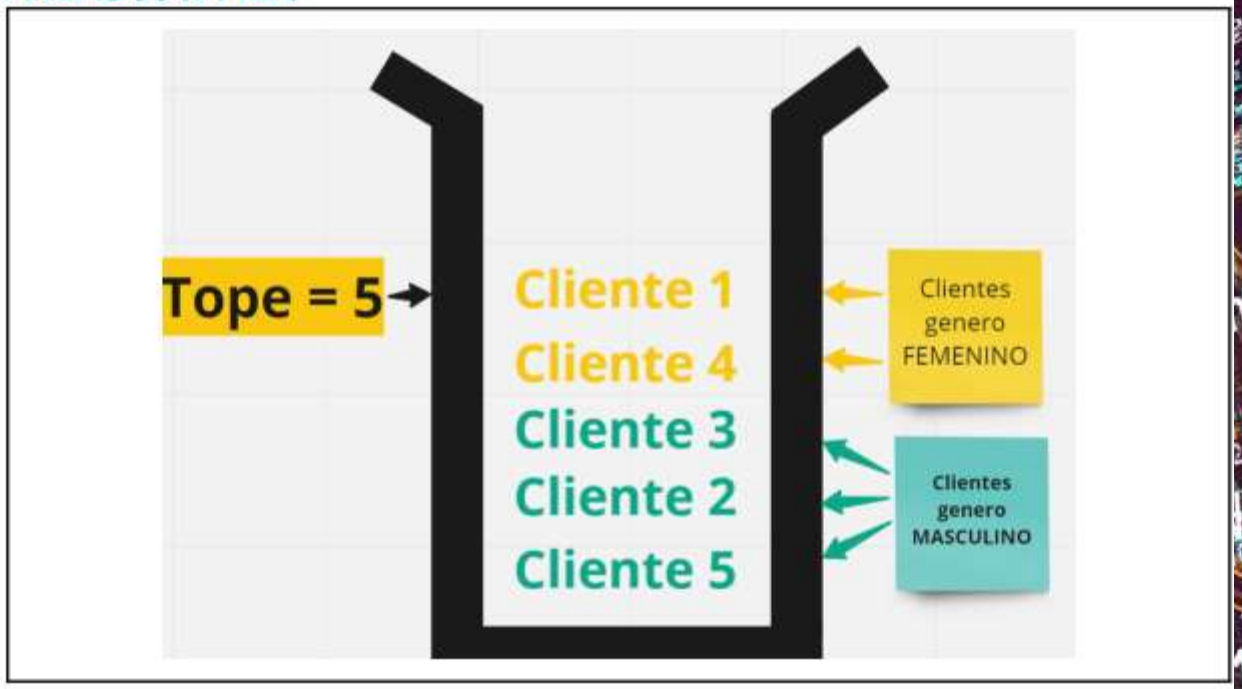
```java
//
public static void AsignaDireccion(PilaCliente pila, String nuevaDireccion){
    PilaCliente aux = new PilaCliente( max: 10);
    Cliente valor = null;

    while (!pila.esVacio()) {
        valor = pila.eliminar();
        if (valor.getGenero() == "Femenino"){
            valor.setDireccion(nuevaDireccion);
            aux.adicionar(valor);
        } else {
            aux.adicionar(valor);
        }
    }
    pila.vaciar(aux);
    pila.adicionar(valor);
    pila.mostrar();
}
```

```
 Clientes
NOMBRE: Mariana
APELLIDO: Palacios
EDAD: 18
DIRECCION: Rusia
GENERO: Femenino

 Clientes
```

15. Mover **ÍTEMS** de la **PILA.**



- El método deberá llamarse **reordenaPila**(Pila)
- El método debe ser creado en la clase **MAIN** como un método estático.
- El método recibe 1 parámetro
  - La Pila de Clientes
- Mover a la base todos los **clientes del género masculino** y los del género **femenino moverlos al final**.
- Adjuntar los siguientes
  - El **código** del método que resuelve el problema.
  - Una **imagen** de la salida de la consola.


//
public static void ReordenaPila(PilaCliente pila){
PilaCliente aux = new PilaCliente(max: 10);
Cliente valorExtraidoPila = null;
while (!pila.esVacio()) {
valorExtraidoPila = pila.eliminar();
if (valorExtraidoPila.getGenero() == "Femenino"){
valorExtraidoPila.mostrarCliente();
} else {
aux.adicionar(valorExtraidoPila);
}
}
pila.vaciar(aux);
pila.mostrar();
}