REPORTE DE PRÁCTICA

Nombre de la práctica: Eliminación de elementos negativos de una pila utilizando una pila auxiliar.

Objetivo de la práctica: Generar la estructura de datos pila y sus operaciones básicas.

Nombre del alumno: Elías Roblero Pérez.

Matricula: 183414

Fecha: 25 de enero del 2019

INTRODUCCION

-Descripción del tema:

El tema del que trata esta práctica es sobre una estructura muy importante que son las pilas.

Las pilas son estructuras de datos lineales, como los arreglos, ya que los componentes ocupan lugares sucesivos en la estructura y cada uno de ellos tiene un único sucesor y un único predecesor, con excepción del último y del primero, respectivamente. Una pila se define formalmente como una colección de datos a los cuales se puede acceder mediante un extremo, que se conoce generalmente como tope.

Representa una estructura lineal de datos en la que se puede agregar o quitar elementos únicamente por uno de los dos extremos. En consecuencia, los elementos de una pila se eliminan en orden inverso al que se insertaron; es decir, el último elemento que se mete en la pila es el primero que se saca. Debido a esta característica, se le conoce como estructura LIFO (Last-Input, First-Output: el último en entrar es el primero en salir).

La definición anterior es la que nos otorga Osvaldo Cairó, en su libro Estructura de datos, libro muy famoso y utilizado.

-¿Qué explicará el reporte? El presente reporte explica el funcionamiento de una pila utilizada para solucionar un problema planteado, el cual es eliminar solamente los elementos negativos que se ingresan al principio del programa, en este reporte de describen detalladamente los pasos que se siguieron para construir el

programa, para lo cual se crearon 2 clases, una que contiene la clase principal que es el Main, y en la segunda las operaciones que se realizarán en la pila.

Además de describir cada una de las operaciones desde agregar elementos a la pila hasta eliminar elementos, y todas las sentencias e instrucciones que requerí para que el programa funcionará adecuadamente.

DESARROLLO

Como punto a resaltar primeramente, el lenguaje que se utilizó para realizar esta práctica, es el lenguaje de programación Java, lenguaje que es orientado a objetos y que actualmente estamos aprendiendo y utilizando. Ya para desarrollar el programa, los pasos son los siguientes:

Lo primero que se realizó fue crear 2 clases, una clase publica llamada Pila que contiene el Main, y otra clase publica llamada PilaOperaciones que como su nombre lo dice contiene todas las operaciones que se realizarán como por ejemplo verificar si la pila está vacía, agregar elementos a la pila comúnmente llamado push y eliminar elementos de la pila llamado pop.

Se crearon 2 clases para hacer las operaciones separadas, porque en el Main es donde se ingresarán los datos y es lo importante; y después mandarlas a llamar o más bien instanciarlas a esa clase.

Al principio en la clase pública PilaOperaciones está declarado un arreglo llamado pila que es privado de tipo entero, que es la forma en la que se implementó la pila en este programa, y una variable privada de tipo entero llamada tope, que juega un papel importante en el funcionamiento de una pila, que llevará la cuenta de los números, cuando se eliminan elementos disminuye y cuando se agregan elementos aumenta:

```
public class PilaOperaciones{
    private int pila[];
    private int tope;
```

Posteriormente ahí mismo en la clase PilaOperaciones está realizado un constructor de tipo público llamado igual a la clase en la que está, los constructores son un método especial de una clase que se llama automáticamente siempre que se declara un objeto de esa clase.

Su función es inicializar el objeto y sirve para asegurarnos que los objetos siempre contengan valores válidos.

El constructor que está en la clase PilaOperaciones recibe como parámetro de tipo entero la variable capacidad que representa el tamaño máximo de elementos que contendrá la pila. Dentro ese constructor se crea un objeto nuevo de tipo arreglo llamado pila, para que se pueda utilizar en esa clase, ya que ella simulará el funcionamiento de una pila. Y de en ese constructor a la variable tope se le asigna el valor -1, que indica que una pila no tiene elementos y conforme el desarrollo del programa irá aumentado o disminuyendo dependiendo los elementos que se agreguen, con el constructor se crea la pila vacía:

```
public PilaOperaciones(int capacidad){
        pila = new int[capacidad];
        tope = -1;
}
```

A continuación se realizan líneas de código las cuales realizan las operaciones básicas de una pila, las cuales son verificar si la pila está vacía, agregar elementos a la pila, y eliminar elementos de la pila.

Así se puede observar en el método público de tipo boolean llamado isEmpty, es de tipo boolean porque retorna un valor de verdad, ya sea falso o verdadero, que nos ayudará para comprobar si la pila está o no está vacía. Si la pila no tiene elementos retornará el tope con valor de -1. Si lo anterior es falso no retornaría nada obviamente:

```
//pila vacia
    public boolean isEmpty(){
        return tope== -1;
    }
```

Luego de esto, se creó la siguiente operación de agregar elementos a la pila, para ello se realizó la construcción de un método público de tipo void llamado push, que recibe un parámetro de tipo entero identificado con la variable i, que será el número que se desea agregar a la pila.

Dentro de ese método se evalúa con la sentencia if, si el tope+1 es menor que pila.length (el tamaño del arreglo), se le sumará 1 a la pila en la posición tope y será asignado a la variable i.

El Length en java es utilizado para determinar el tamaño de un arreglo ya sea vector o matriz, siendo ésta una variable pública que está presenta en cualquier arreglo, al hacer una llamada a esta variable siempre se retornará una entero (int):

```
//Agregar a la pila
    public void push(int i){

        if(tope+1 <pila.length){
        pila[++tope] = i;
    }
```

Luego de esto igualmente en la clase PilaOperaciones se realiza la última operación de la pila que es eliminar elementos de ella. Para ello se crea el método pop, que es público y entero, en ese método se utiliza la sentencia if para evaluar el método isEmpty anteriormente realizado. Si es verdadero que haga un retorno de cero, y que a la pila le quite uno en la posición tope, es decir irá disminuyendo el tope de la pila conforme los elementos se eliminen:

```
//Eliminar de la pila
    public int pop(){
        if(isEmpty())
          return 0;
        return
        pila[tope--];
    }
```

Todas estas operaciones realizadas serán utilizadas en la clase Pila, la clase principal que contiene el método Main, que es dónde hace que funcione el programas, para ello se realizó lo siguiente:

En la parte de hasta arriba en la clase Pila, se importó la librería: import java.util.Scanner; que nos ayuda para que podamos ingresar datos que nosotros queramos y que no están previamente definidos. Después se creó el main con la instrucción conocida: public static void main(String[] args).

Posteriormente se creó un objeto teclado de tipo scanner:

```
 Scanner teclado = new Scanner(System.in);
```

Que nos ayudará para poder utilizar el objeto teclado en el desarrollo y ejecución del programa y esté reconocido.

Luego se crearon igualmente en el MAIN, 2 pilas de tipo PilaOperaciones, la primera que será utilizada para poder almacenar los números negativos que sean ingresados y la segunda pila para almacenar únicamente los números positivos, como lo podemos ver en la siguiente instrucción:

```
PilaOperaciones miPila = new PilaOperaciones(20);
PilaOperaciones miPila2 =  new PilaOperaciones(20);
```

Ambas con un tamaño de aproximadamente 20 datos el cual podrá ser la máxima cantidad de elementos en ambas pilas.

Posteriormente declaré 3 variables:

```
int numero;
int capacidad;
int respuesta;
```

La primera variable que es numero, representará como su nombre lo dice todos los números que ingresará el usuario. La variable capacidad representará el tamaño máximo de elementos que el usuario quiera que tenga la pila. Y la variable respuesta que será utilizada hasta el final del programa para ver si el usuario quiere o no seguir realizando más operaciones.

Dentro de una sentencia do-while fue donde puse que se ingresaran la cantidad máxima de los datos que quiera el usuario que tenga la pila y asigné a la variable capacidad con el objeto teclado:

```
do{
 System.out.print("\n-Ingrese el numero total de elementos que tendra la
pila: ");
capacidad=teclado.nextInt();
```

Ahí dentro del do-while hice un for que servirá para poder ingresar los números que el usuario quiera ya sean negatitvos o positivos.

En el for se declara el contador i que empezará desde cero, hasta que el contador sea menor a la capacidad de elementos que tendrá la pila, y que irá incrementándose de uno en uno.

Adentro de ese for se evalúa el numero ingresado por el usuario con la variable numero, utilizando una sentencia if, si el numero es menor a cero, es decir si es negativo se guardará en la primer pila utilizando el método push, el cual es invocado y que agregará el número el cual es pasado como parámetro.

Si el elemento es mayor a cero se guardará en la segunda pila igualmente invocando al método push, y se le pasa como parámetro el número que es positivo:

```
for(int i=0;i<capacidad;i++){
            System.out.print("("+(i+1)+")"+" Ingrese un numero: ");
            numero = teclado.nextInt();
            if(numero<0){
            miPila.push(numero);}
            else{
                miPila2.push(numero);
            }

        }
```

Después de esto se realiza la operación de eliminar elementos de la pila, para ello se utiliza una sentencia if-else, que evalúa si la pila está o no está vacía. Si la pila no está vacía se agrega una estructura while que mientras la pila no esté vacía irá eliminando e imprimiendo en pantalla los elementos negativos que han sido agregados en la primera pila, esta acción de ir eliminado se logra gracias al método pop, el cual es invocado en esta sección de código. En lo que respecta en la parte del Sino (else) se imprime en pantalla usando la conocida instrucción System.out.println, un mensaje que menciona que no se han agregado elementos negativos a la pila y por lo tanto no se ha eliminado nada:

```
//Eliminacion de elementos de la pila
        if(!miPila.isEmpty()){
        while(!miPila.isEmpty()){
            System.out.println("-Elemento eliminado: "+miPila.pop());

        }}
        else{
            System.out.println("*NO SE AGREGARON ELEMENTOS NEGATIVOS A LA
PILA*");
        }
```

Por último se vuelve a realizar una estructura while, que efectuará mientras la pila no esté vacía la impresión de los elementos positivos que anteriormente se agregaron, esto se lleva a cabo gracias al método pop, el cual es invocado dentro de esta sentencia:

```
System.out.println("Los elementos positivos que se agregaron fueron: ");
        while(!miPila2.isEmpty()){

            System.out.println(miPila2.pop());
        }
```

Y a manera propia agregué una sección de código ya hasta al final de la clase Pila, cuyo propósito es preguntar al usuario si quiere seguir realizando más operaciones, en esta parte es utilizada la variable respuesta que hasta al principio es declarada, y es igualada al objeto de tipo Scanner el teclado.nextInt; si la respuesta es 1 se realizará todo lo anterior desde ingresar tamaño de pila hasta lo último, ya que todo el proceso y las líneas de código están inmersas en una estructura do-while. Si su respuesta es no, se debe ingresar otro número diferente de 1 y se cierra el programa:

```
System.out.println("_____________________________________________________");

        System.out.println("Si desea realizar otro proceso mas escriba 1.");
        System.out.println("Si su respuesta es No escriba otro numero
diferente");
        System.out.print("Opcion: ");
        respuesta = teclado.nextInt();

System.out.println("_____________________________________________________");


    }while(respuesta==1);
 }
}
```

Y así finaliza la ejecución de esta programa cuyo objetivo era almacenar datos en una pila, eliminar los elementos negativos y al final mostrar cuales fueron.