

Trabajo Práctico Nº 04

Tema: La Clase Pila

1) Mostrar lo que se escribe por los siguientes segmentos de código. (P es una pila de elementos enteros: X, Y y Z son variables enteras)

<p>a.</p> <pre>X = 3; Y = 5; Z = 2; P = new Pila(); P.meter (X); P.meter (4); X = P.sacar (); P.meter (Y); P.meter (3); P.meter (Z); X = P.sacar (); P.meter (2); P.meter (X); while (! P.estaVacia()) do { X = P.sacar(); System.out.println(X); }</pre>	<p>b.</p> <pre>Y = 1; P = new Pila(); P.meter (5); P.meter (7); X = P.sacar (); X = X + Y; P.meter (X); P.meter (Y); P.meter (2); Y = P.sacar (); X = P.sacar (); while (! P.estaVacia ()) do { Y = P.sacar (); System.out.println(X); } System.out.println("X = "+ X); System.out.println("Y = "+ Y);</pre>
---	--

2) Mostrar lo que se escribe por los siguientes segmentos de código. P1 y P2 son pilas de elementos enteros: I ,J Y K son variables enteras.

<p>a.</p> <pre>P1 = new Pila (); P2 = new Pila (); for (i= 1; i<= 10; i++) P1.meter (i); while (! P1.estaVacia ()) do { I = P1.sacar (); if (I% 2 == 0) P2.meter (I); } while (! P2.estaVacia ()) { I = P2.sacar(); System.out.println(I); }</pre>	<p>b.</p> <pre>I =1; P1 = new Pila (); P2 = new Pila (); while (I * I < 50) { J = I * I; P1.meter(J); I ++; } for (I = 1; I<= 5; I++) { J = P1.sacar(); P2.meter(J); } I = P1.sacar (); for (J = 1; J <= I; J++) { K = P2.sacar(); P1.meter(K); } while (! P1.estaVacia()) { I = P1.sacar(); System.out.println(I); }</pre>
---	--

Trabajo Práctico Nº 04

3) Usar la siguiente información para los ejercicios a, b, c y d: Una pila P es un objeto que contiene dos atributos: elementos es un array de char de dimensión 5 y CAB un entero. C es una variable de tipo carácter. Para cada ejemplo de los que siguen, mostrar el resultado de la operación sobre la pila. Si ocurre desbordamiento o desbordamiento negativo, comprobar la caja correspondiente; si no mostrar el nuevo contenido del array, CAB y C (Nota: algunos valores del array pueden no estar en la pila).

a. P

'A'	'B'	'C'	'D'	'E'
[0]	[1]	[2]	[3]	[4]

CAB = 2 y C = 'F'

P.meter(C); Desbordamiento?..... Desbordamiento Negativo?.....

P

[0]	[1]	[2]	[3]	[4]

CAB = y C =

b. P

'S'	'T'	'U'	'V'	'W'
[0]	[1]	[2]	[3]	[4]

CAB = 3 y C = 'A'

P.meter(C); Desbordamiento?..... Desbordamiento Negativo?.....

P

[0]	[1]	[2]	[3]	[4]

CAB = y C =

c. P

'B'	'C'	'D'	'F'	'Q'
[0]	[1]	[2]	[3]	[4]

CAB = 0 y C = 'X'

P.meter(C); Desbordamiento?..... Desbordamiento Negativo?.....

P

[0]	[1]	[2]	[3]	[4]

CAB = y C =

d. P

'X'	'Y'	'Z'	'A'	'B'
[0]	[1]	[2]	[3]	[4]

CAB = 4 y C = 'C'

P.meter(C); Desbordamiento?..... Desbordamiento Negativo?.....

P

[0]	[1]	[2]	[3]	[4]

CAB = y C =

4) Utilizando la clase Pila implementada solo con array, escribir un programa que permita meter y sacar elementos de la pila. Cada vez que se realice una operación deberá informar cuantos elementos hay en la pila y cuantos elementos le faltan para llegar al máximo.

5) Usar los métodos sacar, meter, estaVacia para hacer lo siguiente:

Trabajo Práctico N° 04

- a- Poner I (una variable entera) como el tercer elemento de la pila, dejando la pila sin sus dos elementos superiores.
- b- Dado un entero N poner I como el Nésimo elemento de la pila, dejando la pila sin sus N elementos superiores.
- c- Dado un entero N, poner I como el Nésimo elemento de la pila, dejando la pila sin cambiar. (Nota: usar una segunda pila).
- d- Poner I igual al último elemento de la pila, dejando la pila vacía.
- e- Poner I igual al último elemento de la pila, dejando la pila sin cambiar.

6) Agregar a la clase Pila el método *cima* que retorne el elemento cima sin cambiar la pila.

7) Agregar a la clase Pila un método que retorne una copia de una Pila.

8) Realice un programa que dado el ingreso de una palabra como cadena de caracteres, permita visualizarla en forma inversa.

9) a. Escribir una versión del método *meter* que cumpla con la siguiente especificación:

boolean meter (nuevoelemento)

Añade un nuevo elemento a la pila, si la pila no esta llena retorna false, caso contrario retorna verdadero.

10) Escribir la clase Cadena, implementada con un arreglo de caracteres, con los siguientes métodos:

- a) Cadena(String c) : el constructor recibe un String y almacena los caracteres del mismo en el arreglo.
- b) esPalindromo : retorna verdadero si la cadena si la secuencia de caracteres se lee igual de izquierda a derecha y viceversa

Ejemplo: ABLE WAS I ERE I SAW ELBA.

11) Realice un programa que lea una expresión aritmética y determine si tiene correctamente colocados los separadores (), {}, []. Obtener la expresión como un String.

Ej.:

$[(a = b) * c]$ → correcto
 $\{(c-d) * [(c + a) / t]\}$ → incorrecto

Nota: puede utilizar una pila para registrar los diferentes tipos de agrupación separadores. En cualquier momento que encuentre un signo de estos abriendo la expresión lo mete en la pila y cada vez que encuentre un signo terminal examina la pila. Si los signos coinciden, continúa testeando, caso contrario la expresión no será correcta.

12) Un estacionamiento de autos contiene una sola línea, la cual puede guardar hasta 10 autos. Existe solamente una entrada/salida al estacionamiento en uno de los extremos de la línea. Si un usuario llega a retirar su auto que no esta cerca de la salida, todos los autos que están bloqueando su salida son retirados, el auto del usuario sale, y los otros autos quedan en el orden que estaban originalmente.

Escriba un programa que procese un grupo de entradas. Cada entrada contiene una 'e' para llegada o 's' para salida y una placa con el número de patente. Se asume que los autos llegan y salen en el orden especificado en la entrada. El programa debe mostrar un mensaje cada vez que sale o entra un auto. Cuando un auto llega, el mensaje debe especificar si hay lugar o no para el auto. Si no hay lugar, el auto saldrá sin entrar en el estacionamiento. Cuando un auto ha estado dentro del estacionamiento y sale, el mensaje debe incluir el número de veces que fue movido el auto para permitir que otros autos salieran.

13) Se desea implementar la clase Doble Pila de enteros. Una doble pila, es una pila en la que se pueden insertar y extraer elementos por ambos extremos. Esta doble pila contendrá como máximo 200 elementos. Los elementos a almacenar están comprendidos entre 500 y 1000. Suponiendo que valor es el elemento a guardar o extraer sabemos que:

- cuando **500 <= valor <= 750** los elementos se ingresan y extraen por un extremo.
- cuando **750 < valor <= 1000** los elementos se ingresan y extraen por el otro extremo.

Trabajo Práctico Nº 04

En una determinada posición de la doble pila nunca habrá más de un elemento, es decir, la cantidad total de elementos de la doble pila no excederá al máximo.

13.a. Realice el esquema gráfico de la doble pila.

13.b. Codificar la clase DoblePila que incluya la declaración de atributos y las operaciones básicas para esta clase (constructor, meter, sacar, estaLlena, estaVacía).

13.c. Escribir un programa, que pruebe la clase DoblePila.

14) Para los ejercicios siguientes tener en cuenta las siguientes explicaciones:

Una expresión aritmética se compone de operadores (+, -, *, /, etc) y operandos (constantes o variables numéricas). Por lo general una expresión aritmética se escribe en notación infija, que es la notación utilizada por las personas. Sin embargo las computadoras utilizan otros tipos de notaciones, postfija y prefija.

Notación	Orden	Ejemplo	Ejemplo
Infija	Operando1 Operador Operando2	A + B	A * B + C
Postfija	Operando1 Operando2 Operador	A B +	A B * C +
Prefija	Operador Operando1 Operando2	+ A B	+ * A B C

La principal ventaja de las notaciones postfija y prefija es que no requieren paréntesis para indicar precedencia de operaciones. Para transformar una expresión en notación infija a postfija o a prefija se utilizan pilas. A continuación se da el algoritmo para convertir de infija a postfija:

1. Obtener caracteres de la expresión y repetir los pasos 2 a 4 para cada carácter.
2. Si es operando, pasarlo a la expresión postfija
3. Si es operador:
 - 3.1 Si pila esta vacía, meterlo en la pila. Repetir a partir de 1.
 - 3.2 Si la pila no esta vacía:
 - Si la prioridad del operador leído es mayor que la prioridad del operando cabeza de la pila, meterlo en la pila y repetir a partir de 1.
 - Si la prioridad del operador es menor o igual que la prioridad del operador de la cabeza, sacar cabeza de la pila y pasarlo a la expresión postfija, volver a 3.
4. Si es paréntesis derecho:
 - 4.1 Sacar cabeza de pila y pasarlo a postfija
 - 4.2 Si nueva cabeza es paréntesis izquierdo, suprimir elemento cabeza
 - 4.3 Si cima no es paréntesis izquierdo, volver a 4.1
 - 4.4 Volver a partir de 1.
5. Si quedan elementos en la pila pasarlos a la expresión postfija
6. Fin del algoritmo.

El algoritmo para convertir de infija a prefija es muy parecido.

14.1 Escribir un programa que lea por teclado un String que representa una expresión aritmética infija y devuelva otro String que representa la expresión aritmética en postfija.

14.2 Diseñar el algoritmo para convertir una expresión aritmética en notación infija a notación prefija. Escribir un programa que lea por teclado un string que representa una expresión aritmética infija y devuelva otro String que representa la expresión aritmética en prefija.

14.3 Diseñar el algoritmo para convertir una expresión aritmética en notación postfija a notación prefija. Escribir un programa que lea por teclado un String que representa una expresión aritmética postfija y devuelva otro String que representa la expresión aritmética en prefija.

Trabajo Práctico N° 04

Para evaluar una expresión aritmética en prefija o postfija se utiliza una pila. A continuación se describe el algoritmo para evaluar una expresión en notación postfija.

La expresión es una cadena de longitud n .

1. Obtener un carácter
2. Si es operando, meterlo en la pila
3. Si es un operador, sacar dos elementos de la pila, realizar la operación y meter el resultado en la pila.
4. Repetir 1 a 3 para los n caracteres de la expresión.

14.4 Escribir un programa que evalúe una expresión en notación postfija. Tener en cuenta que la cadena de caracteres se componen de caracteres y deberá utilizar funciones para convertir de carácter a numérico los operandos.

14.5 Escribir un programa que evalúe una expresión en notación prefija. Tener en cuenta que la cadena de caracteres se componen de caracteres y deberá utilizar funciones para convertir de carácter a numérico los operandos.

15. Un mapa de rutas se puede representar mediante una matriz simétrica MM de $N \times N$ elementos, donde los valores 1 a N representan los pueblos/ciudades que aparecen en el mapa.

Los elementos de la matriz son tales que $MM(i,j) = 0$ si no hay conexión directa entre el pueblo i y el pueblo j , y su distancia es d .

Con esta representación del mapa queremos escribir un programa que simule el mapa descrito y que tenga como entrada dos pueblos (origen, destino) entre los que no hay conexión directa; decida si hay camino que pase por los pueblos del mapa y determine la distancia de ese camino. Utilizar una pila para ir almacenando los pueblos que van formando el camino recorrido y poder volver atrás si se alcanza un pueblo desde el que no se puede proseguir la ruta y probar con otra ruta.

16. Escribir un método para determinar si una secuencia de caracteres de entrada es de la forma:

X & Y

Donde **X** es una cadena de caracteres, e **Y** es la cadena inversa, siendo **&** el carácter separador.

17. Escribir un método para determinar si una secuencia de caracteres de entrada es de la forma:

A # B # C # ...

Donde cada una de las cadenas **A**, **B**, **C**, ... son de la forma **X & Y**, que a su vez estarán separadas por el carácter **#**.

18. Un Pila de Object permite utilizar la misma clase para trabajar con objetos enteros, reales, cadenas, etc., por medio del uso del polimorfismo.

```
class Pila {
    final int maxpila = 100;    // atributos
    Object[] elementos;
    int cima;

    Pila()    // Constructor de la clase Pila
    {
        elementos = new Object[maxpila];
        cima = -1 ;
    }
    public boolean estaVacia()    // verifica si la pila esta vacia
    {
        return (cima == -1);
    }
    public boolean estaLlena()    // verifica si la pila esta llena
    {
```

Trabajo Práctico N° 04

```
    return (cima == maxpila-1);
}
public void meter(Object e) // agrega un elemento en la pila
{
    cima++;
    elementos[cima]=e;
}
public Object sacar() // saca un elemento del tope de la pila
{
    int aux;
    aux= elementos[cima];
    cima--;
    return (aux);
}
}
```

Pruebe la siguiente clase con objetos de diferente clase (Integer, Float, Char, etc.)