

## Trabajo Práctico N° 3

## Tema: Pilas

## 1. Realizar la traza de los siguientes segmentos de código.

<p>a) <b>int</b> X=3;  <b>int</b> Y=5;  <b>int</b> Z=2;  Pila P=<b>new</b> 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);  <b>while</b>(!P.estaVacia()) {      X=P.sacar();      System.out.println(X);  }</p>	<p>b) <b>int</b> Y=1;  Pila P=<b>new</b> Pila();  P.meter(5);  P.meter(7);  <b>int</b> X=P.sacar();  X+=Y;  P.meter(X);  P.meter(Y);  P.meter(2);  Y=P.sacar();  X=P.sacar();  <b>while</b>(!P.estaVacia()) {      Y=P.sacar();      System.out.println(Y);  }  System.out.println("X = " + X);  System.out.println("Y = " + Y);</p>
<p>c) Pila P1=<b>new</b> Pila();  Pila P2=<b>new</b> Pila();  <b>int</b> X;  <b>for</b>(<b>int</b> i=1; i&lt;=10; i++)      P1.meter(i);  <b>while</b> (!P1.estaVacia()) {      X=P1.sacar();      <b>if</b>(X%2 == 0)          P2.meter(X);      }  <b>while</b> (!P2.estaVacia()) {      X=P2.sacar();      System.out.println(X);  }</p>	<p>d) <b>int</b> I=1;  <b>int</b> J;  Pila P1=<b>new</b> Pila();  Pila P2=<b>new</b> Pila();  <b>while</b>(I * I &lt; 50) {      J=I * I;      P1.meter(J);      I++;  }  <b>for</b>(<b>int</b> i=1; i&lt;=5; i++) {      J=P1.sacar();      P2.meter(J);  }  I=P1.sacar();  <b>int</b> K;  <b>for</b>(<b>int</b> j=1; j&lt;=I; j++) {      K=P2.sacar();      P1.meter(K);  }  <b>while</b>(!P1.estaVacia()) {      I=P1.sacar();      System.out.println(I);  }</p>

2. Dado el objeto pila P que contiene dos atributos: un arreglo de caracteres de dimensión 5 y CIMA 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 el caso correspondiente; si no mostrar el nuevo contenido del arreglo, CIMA y C.

a)

P

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

CIMA = 4 y C = 'F'

**P.meter(C);**

¿Desbordamiento?.....

¿Desbordamiento Negativo? .....

P

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

CIMA = ... y C = ...

b)

Trabajo Práctico N° 3

## Tema: Pilas

P 

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

 CIMA = 3 y C = 'A'

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

P 

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

 CIMA = ... y C = ...

c)

P 

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

 CIMA = 0

**C=P.sacar();** ¿Desbordamiento?..... ¿Desbordamiento Negativo? .....

P 

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

 CIMA = ... y C = ...

d)

P 

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

 CIMA = 4 y C = 'B'

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

P 

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

 CIMA = ... y C = ...

e)

P 

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

 CIMA = -1

**C=P.sacar();** ¿Desbordamiento?..... ¿Desbordamiento Negativo? .....

P 

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

 CIMA = ... y C = ...

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

4. Utilizando la clase **Pila** implementada con arreglo únicamente (no utiliza índice cima), 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. Agregar a la clase **Pila** el método **elementoCima()** que retorne el elemento cima sin cambiar la **pila**. Nota: no olvidar que si bien la **pila** se implementa como un arreglo, no debería manejarse como tal.

6. Usar los métodos **sacar()**, **meter()**, **estaVacia()** para hacer las siguientes operaciones:

- Asignar a X el segundo elemento desde la parte superior de la **pila**, dejando la **pila** sin sus dos elementos de la parte superior.
- Asignar a X el segundo elemento desde la parte superior de la **pila**, sin modificarla.

Trabajo Práctico N° 3Tema: Pilas

---

- c) Desde un entero positivo N, asignar a X el N-ésimo elemento desde la parte superior de la pila, dejando la pila sin sus N elementos de la parte superior.
- d) Dado un entero positivo N, asignar a X el N-ésimo elemento desde la parte superior de pila, sin modificarla.
- e) Asignar a X el elemento fondo de la pila, dejando la pila vacía.
- f) Asignar a X el elemento fondo de la pila, sin modificarla.

7. Agregar a la clase Pila un método que retorne una copia exacta de una pila. Nota: no olvidar que si bien la pila se implementa como un arreglo, no debería manejarse como tal.

8. Implementar una clase Cadena que permita ingresar una cadena de caracteres (String) y que tenga el método esPalindromo() que retorna verdadero si la secuencia de caracteres se lee igual de izquierda a derecha y viceversa.

Ejemplo: ABLE WAS I ERE I SAW ELBA es palíndromo.

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

Ejemplo:

$[(a \% b) * c]$	→	correcto
$2 * (a + b)] / 2.5 + x$	→	incorrecto
$\{(c - d) * [(c + a / t]$	→	incorrecto

Nota: puede utilizar una pila para registrar los diferentes tipos de agrupación de 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.

10. Implementar una pila de autos, la cual puede guardar hasta 10 autos. En un main permitir al usuario ingresar autos y buscar un auto por número de patente y si se encuentra en la pila sacarlo y mostrar la información del mismo.