Instituto Tecnológico de Tijuana

T12: Método pop en pilas.

Materia:

Estructura de Datos

Profesor(a):

Ray Brunett Parra Galaviz

Alumno(a):

Jiménez Mayoral Gloria Alejandra – 17212146

Fecha:

26 de septiembre de 2018

Pila: método pop.

```
1  #include <iostream>
2  #include<stdlib.h>
3  using namespace std;
```

Las líneas **1** y **2** implementan las librerías: iostream, para poder utilizar operaciones de entrada y salida. Y stdlib para poder utilizar funciones como system("CLS") para limpiar la pantalla. La línea **3** implementa el namespace para el uso de las palabras reservadas: cout (para imprimir datos) y cin (para ingresar datos).

```
4  class Pila
5  {
6      int arraypi[5], index;
7      public:
8      Pila()
9      {
10         index = -1;
11     }
12     void push(int x)
13     {
14         if(index>5)
15         {
16             system("CLS");
17             cout<<"Pila llena!! No puedes agregar mas elementos"<<endl;
18             return;
19         }
20         arraypi[++index]=x;
21         cout<<"Elemento "<<x<<" ingresado"<<endl;
22     }
```

A partir de la línea **4** hasta la línea **34**, se crea la clase Pila que contiene un arreglo con el tamaño de la pila de tipo entero y un índice del mismo tipo. Dentro de la clase, se crea un constructor y el método que se utilizará, ambos públicos. En el constructor (línea **8**) se inicializa el valor del índice en -1 para considerarlo como un inicio cero. El método push se utilizará para introducir elementos a la pila.

```
24         void pop()
25         {
26             if(index<0)
27             {
28                 system("CLS");
29                 cout<<"Pila vacia!! No puedes sacar elementos."<<endl;
30                 return;
31             }
32             cout<<"Elemento "<<arraypi[index--]<<" eliminado"<<endl;
33         }
```

En la línea **24** se crea el método **pop** de tipo void que funcionará para eliminar el último elemento de la pila. Es decir, el último elemento introducido, será el primero en ser eliminado. Por esto, las pilas se conocen como estructuras LIFO.

El método cuenta con una condicional para cuando el índice indique que no hay elementos para mostrar y que la pila está vacía. En caso de que la condición se cumpla, se imprimirá en pantalla un mensaje de que no se pueden sacar datos porque la pila está vacía. Y pasando de la condición, el método **pop** eliminará el elemento último disminuyendo el índice que señala la posición del último elemento.

En la línea **35** del código, comienza la función principal main, en la cual se declaran variables de tipo entero para ser utilizadas dentro de un método switch. Y también el objeto para el uso de la clase Pila. La función switch (línea **43**) funciona como un menú y cuenta con dos opciones a elegir por el usuario: el número uno con el método push de la clase. Y el número dos con el método **pop** para eliminar elementos, donde simplemente el objeto utiliza la función de pop y se ejecuta imprimiendo en pantalla el elemento eliminado. Y finalmente, el número tres, para terminar el programa y salir de éste. El menú se repetirá hasta que la opción del usuario sea terminar.

```
35  int main()
36  {
37      int op, op2;
38      Pila pil;
39      do
40      {
41          cout<<"1.Push(ingresar un elemento) \n2.Pop(quitar un elemento) \n3.Salir \nTu opcion es:  ";
42          cin>>op;
43          switch(op)
44          {
45              case 1:
46                  system("CLS");
47                  cout<<"Ingresa un elemento: ";
48                  cin>>op2;
49                  system("CLS");
50                  pil.push(op2);
51              break;
52              case 2:
53                  system("CLS");
54                  pil.pop();
55              break;
56              case 3:
57              break;
58          }
59      }while(op!=3);
60      return 0;
61  }
```