



Temas

- Destruidores



Métodos

Destructores

Destructores

- Un destructor es una función que se ejecuta automáticamente cuando se destruye un objeto, por ejemplo cuando un objeto es local a un método y se termina la ejecución de dicho método.
- El destructor por si mismo no destruye el objeto, pero es útil para liberar memoria dinámica que se había reservado al crear el objeto, o para cerrar un archivo que se abrió al crear el objeto, etc.
- Un destructor en C++ tiene las siguientes características:
 - **Tiene el mismo nombre de la clase,**
pero precedido por el caracter ~.
 - **No tiene definido un valor de retorno**

Destructores

- Una clase puede tener solamente un destructor, el cual no debe recibir parámetros ni tener valor de retorno.
- Si el programador no crea un destructor C++ crea uno "vacío" que se encarga de realizar algunas acciones necesarias al destruir un objeto.

Por ejemplo:

```
class Punto
{
    public: ~Punto( )
    {
        cout << "destructor Punto llamado \n";
    }
    //
};
```

Destructores

Ejemplo completo: El siguiente ejemplo muestra la declaración de una clase (la clase Pila) y una aplicación que hace uso de ella.

```
#include <iostream.h>
const int SIZE = 100;
//Definición de la clase Pila
class Pila {
    int pila[SIZE];
    int tope;
public:
    Pila(); // constructor
    ~Pila(); // destructor
    void push(int i);
    int pop();
};
// constructor de la clase pila
Pila::Pila()
{
    tope = 0;
    cout<<"Se inicializa la pila"<<endl; // este letro se utiliza solo para entender el ejemplo
} // CONTINUA
```

Destructores

```
// destructor de la clase pila
Pila::~~Pila()
{
    cout<<"se destruye la pila"<<endl; // este letrero se utiliza solo para hacer más claro el ejemplo
}
void Pila::push(int valor)
{
    if (tope == SIZE)
    {
        cout<<"La pila esta llena"<<endl;
        return;
    }
    pila[tope] = valor;
    tope++;
}
int Pila::pop()
{
    if (tope == 0)
    {
        // CONTINUA
```

Destructores

```
cout<<"No hay nada en la pila"<<endl;
    return -1;
}
tope--;
return pila[tope];
}
int main()
{
    Pila a, b;
    a.push(1);
    b.push(2);
    a.push(3);
    b.push(4);
    cout<<a.pop()<<endl;
    cout<<a.pop()<<endl;
    cout<<b.pop()<<endl;
    cout<<b.pop()<<endl;
    return 0;
}
```