



Emilio Gomez Breschi

19300100

4D1

Sonia Erika Ibáñez de la
Torre Desarrollo de Software

10/Octubre/2021

Programación Orientada a Objetos

¿Que es un constructor y para que es utilizado?

En programación, los constructores son funciones o métodos que permiten realizar tareas de instanciación de objetos. Cuando un objeto es creado a partir de una clase, se llama al constructor que se encargará de inicializar los atributos del objeto, como así también cualquier llevar a cabo cualquier otra función necesaria.

(c++, 2017)

Sintaxis del constructor , en la clase y cuando se saca de la

clase. Class FooClass

```
{  
  
    public:  
  
        FooClass(/*Parámetros de entrada opcionales */); //Constructor };  
  
    FooClass::FooClass()  
  
    {  
  
    //Líneas de código del constructor  
  
    }
```

```
class Box {  
public:  
    // Default constructor  
    Box() {}  
  
    // Initialize a Box with equal dimensions (i.e. a cube)  
    explicit Box(int i) : m_width(i), m_length(i), m_height(i) // member init list  
    {}  
  
    // Initialize a Box with custom dimensions  
    Box(int width, int length, int height)  
    : m_width(width), m_length(length), m_height(height)  
    {}  
  
    int Volume() { return m_width * m_length * m_height; }  
  
private:  
    // Will have value of 0 when default constructor is called.  
    // If we didn't zero-init here, default constructor would  
    // leave them uninitialized with garbage values.  
    int m_width{ 0 };  
    int m_length{ 0 };
```

```
int m_height{ 0 };  
};  
(corob-msft, 2019)
```

¿Cuántos constructores se pueden tener en una clase y bajo que condiciones?

Puede definir tantos constructores sobrecargados como sea necesario para personalizar la inicialización de varias maneras. Normalmente, los constructores tienen accesibilidad pública para que el código fuera de la definición de clase o la jerarquía de herencia pueda crear objetos de la clase . Pero también puede declarar un constructor como `protected` o `private` .

(Steve, 2009)

Que es la ambigüedad en la sobrecarga de constructores

Un `int` se puede convertir tanto en `double` como en `unsigned`; ambas son consideradas conversiones de "ampliación" y tienen el mismo rango. Como norma general , cada vez que sobrecarga y una de las sobrecargas es de tipo integral , es una buena idea sobrecargar también en `int`, para asegurarse de que obtenga lo que desea, y evitar ambigüedades, cuando alguien intenta a pasa una constante integral.

`private:`

`t m_Value;`

`unsigned int m_Length;`

(Munro, 2011)

Ventajas y desventajas al utilizar constructores .- Tener el mismo nombre de la clase que inicializa

Ventajas:

- Puede definirse inline o fuera de la declaración de la clase
- No devuelve valores
- Puede admitir parámetros como cualquier otra función
- Puede existir más de un constructor, e incluso no existir

Desventajas:

Los constructores tienen el mismo nombre que la clase, no retornan ningún valor y no pueden ser heredados. Además deben ser públicos, no tendría ningún sentido declarar un constructor como privado, ya que siempre se usan desde el exterior de la clase, ni tampoco como protegido, ya que no puede ser heredado.

(Cuenca, 2004)

Ejemplo:

```
#include <iostream.h>

class pareja
{
public:
    //Constructor
    pareja(int a2, int b2);
    //Funciones miembros de la clase "pareja"
    void Lee(int &a2, int &b2);
    void Guardar(int a2, int b2);
private:
    //Datos miembros de la clase "pareja"
    int a, b;
}

pareja::pareja(int a2, int b2)
{
    a = a2;
    b = b2;
}

void pareja::Lee(int &a2, int &b2)
{
    a2 = a;
    b2 = b;
}

void pareja::Guarda(int a2, int
b2) {
    a = a2;
    b = b2;
```

```

}
int main(int argc, char *argv[])
{
    pareja par1(12, 32);
    int x, y;
    par1.Lee(x, y);
    cout << "Valor de par1.a: " << x << endl;
    cout << "Valor de par2.b: " << x << endl;
    return 0;
}
(Coronado, 2018)

```

Bibliografía

c++. (2017). Recuperado el 10 de Octubre de 2021, de <https://cpp.dokry.com/3992/sintaxis-del-constructor-de-c.html>

corob-msft. (27 de Diciembre de 2019). *Microsoft*. Recuperado el 10 de Octubre de 2021, de <https://docs.microsoft.com/es-es/cpp/cpp/constructors-cpp?view=msvc-160>

Coronado, S. P. (29 de Noviembre de 2018). *EcuRed*. Recuperado el 10 de Octubre de 2021, de [https://www.ecured.cu/index.php?title=Constructor_\(C%2B%2B\)&action=history](https://www.ecured.cu/index.php?title=Constructor_(C%2B%2B)&action=history)

Cuenca, J. D. (4 de Junio de 2004). *mailxmail*. Recuperado el 10 de Octubre de 2021, de <http://www.mailxmail.com/curso-programacion-c/constructores>

Munro. (15 de Septiembre de 2011). *uwenku*. Recuperado el 10 de Octubre de 2021, de <http://es.uwenku.com/question/p-orcsmazj-bh.html>

Steve. (25 de Febrero de 2009). *uwenku*. Recuperado el 10 de Octubre de 2021, de <http://es.uwenku.com/question/p-sfxuqynw-bd.html>