

# Hàm xây dựng và hàm hủy

Trong bài học hôm nay chúng ta sẽ cùng tìm hiểu về 2 hàm đặc biệt trong c# đó là hàm xây dựng (Constructor) và hàm hủy (Destructor)

## 1. Hàm xây dựng

Hàm xây dựng mặc nhiên

Hàm xây dựng có tham số

## 2. Hàm hủy

# 1. Hàm xây dựng (Constructor )

- Constructor là một phương thức đặc biệt được gọi tự động tại thời điểm đối tượng được tạo ra.
- Mục đích của hàm xây dựng dùng để khởi tạo dữ liệu cho dữ liệu thành viên.
- Constructor phải trùng tên với tên lớp và không có kiểu trả về kể cả kiểu void.
- Trong c# có 2 loại hàm xây dựng đó là
  - Hàm xây dựng mặc nhiên
  - Hàm xây dựng có đối số

# Hàm xây dựng mặc nhiên

- Một constructor không có đối số được gọi là constructor mặc định. Nó được gọi tại thời điểm tạo đối tượng.
- Nếu bạn không cung cấp hàm tạo cho lớp của mình, C# sẽ tạo một hàm theo mặc định để khởi tạo đối tượng và đặt các biến thành viên thành các giá trị mặc định tùy thuộc vào kiểu dữ liệu của nó.

# Hàm xây dựng mặc nhiên

- Liệt kê giá trị mặc định của một số kiểu dữ liệu hay sử dụng sau:

Kiểu dữ liệu	Giá trị mặc định
bool	false
float	0.0F
int	0
byte	0
decimal	0M
long	0L
char	'\0'

# Ví dụ

```
namespace ConsoleApp1
{
    class People
    {
        int old;
        string name;
        double height;
        public People()
        {
            Console.WriteLine("Goi ham xay dung mac nhien");
            Console.WriteLine("Name" + name);
            Console.WriteLine("Old: " + old);
            Console.WriteLine("height: " + height);
        }
    }
    class Program
    {
        static void Main(string[] args)
        {
            People p = new People();
            Console.ReadKey();
        }
    }
}
```

# Hàm xây dựng mặc nhiên

- Lưu ý: nếu chúng ta không tạo bất kỳ hàm xây dựng nào thì chương trình sẽ tự tạo cho chúng ta hàm xây dựng mặc nhiên. Chúng ta chỉ có 1 cách duy nhất để khởi tạo đối tượng.

# Hàm xây dựng có tham số

- Một constructor có tham số được gọi là hàm xây dựng có tham số.
- Nó được sử dụng để cung cấp các giá trị khác nhau cho các đối tượng riêng biệt

# Hàm xây dựng có tham số

```
namespace ConsoleApp1
{
    class People
    {
        int old;
        string name;
        double height;
        public People()
        {
            Console.WriteLine("\n---Goi ham xay dung mac nhien---");
        }
        public People(int old, string name, double height)
        {
            Console.WriteLine("\n---Goi ham xay dung co 3 tham so---");
            this.old = old;
            this.name = name;
            this.height = height;
        }
        public People(int old, string name)
        {
            Console.WriteLine("\n---Goi ham xay dung co 2 tham so---");
            this.old = old;
            this.name = name;
        }
        public void Show()
        {
            Console.WriteLine("Old: " + old + ",\nName: " + name + ",\nHeight: " + height);
        }
    }
}
```



# Hàm xây dựng có tham số

```
class Program
{
    static void Main(string[] args)
    {
        People p = new People();
        p.Show();
        People p1 = new People(20, "Nguyen Van A", 180);
        p1.Show();
        People p2 = new People(18, "Nguyen Van A");
        p2.Show();
        Console.ReadKey();
    }
}
```

# Hàm xây dựng có tham số

- Như vậy nếu chúng ta định nghĩa bao nhiêu hàm xây dựng, thì chúng ta chỉ có bao nhiêu cách khởi tạo đối tượng. Ở ví dụ trên, chúng ta chỉ định nghĩa 3 hàm xây dựng, vậy nên chúng ta chỉ có 3 cách khởi tạo đối tượng.
- Nếu chúng ta định nghĩa hàm xây dựng, thì chúng ta không sử dụng được hàm khởi tạo mặc nhiên.

## 2. Hàm hủy

- Một hàm hủy hoạt động ngược lại với hàm tạo, Nó phá hủy các đối tượng của các lớp.
- Nó chỉ có thể được định nghĩa một lần trong một lớp.
- Giống như các hàm xây dựng, hàm hủy được gọi tự động.

## 2. Hàm hủy

```
namespace ConsoleApp1
{
    class People
    {
        public People()
        {
            Console.WriteLine("\n---Goi ham xay dung mac nhien---");
        }
        ~People()
        {
            Console.WriteLine("\n---Goi ham huy---");
        }
    }
    class Program
    {
        static void Main(string[] args)
        {
            People p = new People();
            Console.ReadKey();
        }
    }
}
```

## 2. Hàm hủy

- Hàm hủy là duy nhất cho lớp của nó, tức là không thể có nhiều hơn một hàm hủy trong một lớp.
- Hàm hủy không có kiểu trả về và có cùng tên với tên lớp
- Hàm hủy được phân biệt với một hàm xây dựng vì ký hiệu ~trước tên của nó.
- Hàm hủy không chấp nhận bất kỳ tham số nào và không được sửa đổi hàm hủy

## 2. Hàm hủy

- Hàm hủy không thể được định nghĩa trong Cấu trúc. Hàm hủy chỉ được sử dụng với các lớp.
- Hàm hủy không thể bị overloaded hoặc kế thừa. (overload là gì mình sẽ cùng tìm hiểu trong một bài khác)
- Hàm hủy được gọi khi chương trình thoát.