

# Tính kế thừa trong C#

## Tính kế thừa trong C#

Kế thừa là một trong 4 khía cạnh của lập trình hướng đối tượng, nó là khả năng cho phép chúng ta định nghĩa ra một lớp mới dựa trên một lớp khác có sẵn, kế thừa giúp cho việc mở rộng code - bảo trì trở nên dễ hơn.

Tính kế thừa trong c# cho phép lập trình viên tạo ra một lớp mới kế thừa một lớp đã tồn tại. Nghĩa là lớp được kế thừa sẽ có tất cả những thuộc tính và phương thức với quyền truy cập không phải private của lớp cha.

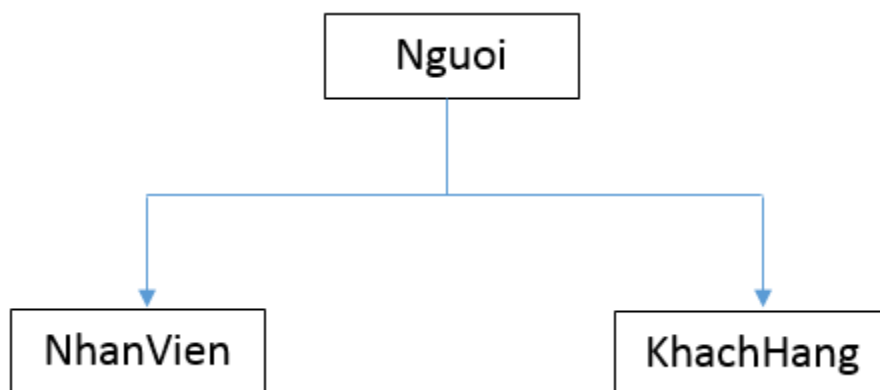
- Lớp cơ sở là lớp mà được lớp khác kế thừa.
- Lớp kế thừa là lớp kế thừa lại các thuộc tính, phương thức từ lớp cơ sở.

C# chỉ hỗ trợ đơn thừa kế (nghĩa là một lớp chỉ được phép kế thừa một lớp khác).

Ví dụ, lớp NhanVien mô tả chung (thuộc tính) về nhân viên trong một công ty (chức năng, nhiệm vụ, văn bằng ...), kể cả các

ứng xử (phương thức) của nhân viên. Từ đó kế thừa lại xây dựng nên lớp mới cho nhân viên lễ tân NhanVienLeTan, nhân viên bán hàng NhanVienBanHang ..., nhưng lớp mới đã kế thừa lại các thông tin của lớp cơ sở và thêm vào những đặc tính riêng của nó

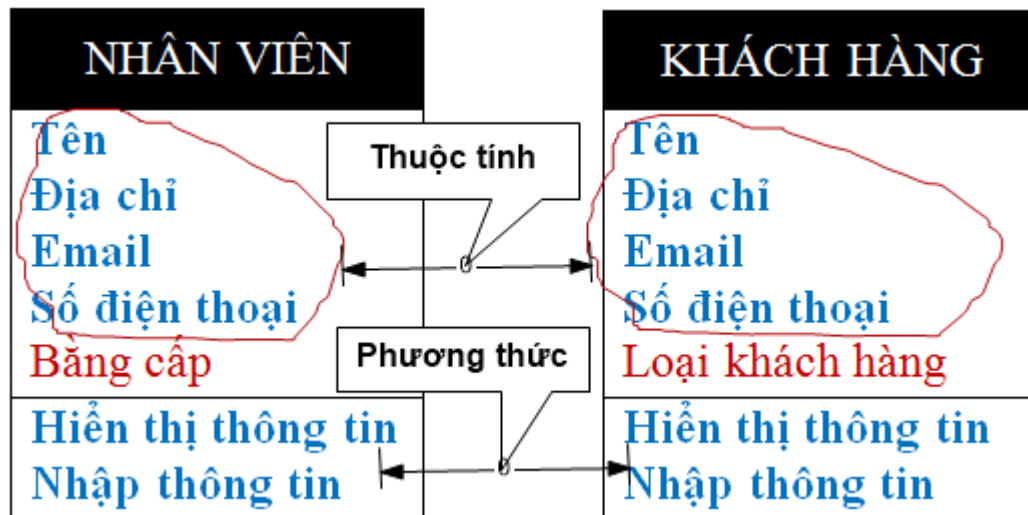
Hình bên dưới là một ví dụ về tính kế thừa trong c#, cụ thể khi tạo lớp NhanVien và lớp KhachHang, chúng ta sẽ kế thừa lớp Nguoi.



## Tính kế thừa trong C# – Mục đích sử dụng

Mục đích của tính kế thừa là để tái sử dụng những thuộc tính và phương thức phổ biến trong một lớp mà không cần phải tạo ra chúng. Xem ví dụ bên dưới.

Yêu cầu tạo 2 lớp nhân viên và khách hàng có các thuộc tính và phương thức như hình



Chúng ta nhận thấy rằng giữa 2 lớp này có 4 thuộc tính (tên, địa chỉ, email, số điện thoại) và 2 phương thức (hiển thị thông tin, nhập thông tin) giống nhau. Vì vậy, chúng ta sẽ tạo ra một lớp mới trong những thuộc tính và phương thức này.

NGƯỜI
Tên
Địa chỉ
Email
Số điện thoại
Hiển thị thông tin
Nhập thông tin

NHÂN VIÊN
Bảng cấp
Hiển thị thông tin
Nhập thông tin

KHÁCH HÀNG
Loại khách hàng
Hiển thị thông tin
Nhập thông tin

## Tính kế thừa trong C# – Cài đặt kế thừa

Tại lớp được kế thừa, chèn dấu hai chấm sau tên lớp và theo sau dấu hai chấm là lớp cha (lớp cơ sở). Lúc này lớp được kế thừa (lớp dẫn xuất) sẽ thừa kế những thuộc tính và phương thức không phải private (non-private) của lớp cha. Bên dưới là cú pháp được sử dụng để kế thừa trong C#.

# Tên\_Lớp\_Con : Tên\_Lớp\_Chả

Ví dụ: Cho phép lớp NhanVien kế thừa lớp Nguoi

```
class NhanVien:Nguoi
{
}
```

## Tính kế thừa trong C# – Phương thức khởi tạo

Tại lớp cha (lớp cơ sở) có định nghĩa phương thức khởi tạo (constructor) có tham số thì tại các lớp dẫn xuất (lớp con hay lớp được kế thừa) **phải gọi phương thức khởi tạo của lớp cha**.  
Bên dưới là một ví dụ

### Xây dựng lớp Nguoi

```
using System;

using System.Collections.Generic;

using System.Linq;
```

```
using System.Text;

namespace ConsoleApplication.lesson05
{
    class Nguoi
    {
        // Khai báo thuộc tính

        public string maso;

        public string hoten;

        public string gioitinh;

        // Phương thức khởi tạo có 4 tham số

        public Nguoi(string maso, string hoten, string
gioitinh)

        {

            this.maso = maso;

            this.hoten = hoten;

            this.gioitinh = gioitinh;
```

```
    }  
  
    }  
  
}
```

## **Xây dựng lớp NhanVien kế thừa lớp Nguoi**

```
using System;  
  
using System.Collections.Generic;  
  
using System.Linq;  
  
using System.Text;  
  
namespace ConsoleApplication.lesson05  
{  
  
    class NhanVien:Nguoi  
    {  
  
        private string bangcap;  
  
        public NhanVien(string maso, string hoten,  
string gioitinh, string bangcap)  
  
            : base(maso, hoten, gioitinh)
```

```
{  
    this.bangcap = bangcap;  
}  
  
}
```