

## Tính kế thừa trong C# – Overriding cho method

Là một tính năng cho phép lớp con định nghĩa lại phương thức của lớp cha. Phương thức được khai báo ở lớp con phải cùng tên, cùng tham số mà lớp cha đã khai báo.

Trong C#, để thực hiện điều này thì những phương thức cho phép lớp con overriding phải chỉ định từ khoá **virtual**. Và tại những lớp con phải chỉ định từ khoá **override** khi muốn định nghĩa lại xử lý cho phương thức. Hãy xem xét ví dụ sau:

### Tại lớp cha (lớp Người)

```
using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ConsoleApplication.lesson05

{

    class Người
```

```
{

    // Khai báo thuộc tính

    public string maso;

    public string hoten;

    public string gioitinh;

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

    {

        this.maso = maso;

        this.hoten = hoten;

        this.gioitinh = gioitinh;

    }

    public virtual void Nhap()

    {

        Console.Write("Nhap ma so: ");

        maso = Console.ReadLine();

    }

}
```

```
        Console.Write("Nhap ho ten: ");

        hoten = Console.ReadLine();

        Console.Write("Nhap gioi tinh: ");

        gioitinh = Console.ReadLine();

    }

    public virtual void HienThi()

    {

        Console.WriteLine("Ma so: {0}, Ho ten: {1},  
Gioi tinh: {2}", maso, hoten, gioitinh);

    }

}

}
```

### **Tại lớp con (lớp NhanVien)**

```
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;
        }

        public override void Nhap()
        {

            base.Nhap(); // Gọi phương thức của lớp
Nguoi

            // Thêm xử lý cho thuộc tính bằng cấp
        }
    }
}
```

```
        Console.Write("Nhap thong tin bang cap: ");

        bangcap = Console.ReadLine();

    }

    public override void HienThi()

    {

        base.HienThi();

        Console.WriteLine("Bang cap: {0}", bangcap);

    }

}

}
```

## Tính kế thừa trong C# – Từ khoá Sealed

Một lớp được chỉ định với từ khoá sealed là một lớp không cho phép kế thừa. Một phương thức được chỉ định với từ khoá sealed là một phương thức không cho overriding.

### Cú pháp khai báo lớp sealed

```
sealed class Class_Name
```

```
{
```

```
    //body of the class
```

```
}
```

### Cú pháp khai báo phương thức sealed

```
access_modifier sealed override return_type Method_Name
```

```
{
```

```
    //body of the method
```

```
}
```

### Ví dụ khai báo phương thức sealed

```
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;

}

// Phương thức này không được phép overriding
ở lớp con

public sealed override void Nhap()

{

    base.Nhap(); // Gọi phương thức của lớp Nguoi

    // Thêm xử lý cho thuộc tính bằng cấp

    Console.Write("Nhập thông tin bang cap: ");

    bangcap = Console.ReadLine();

}

public override void HienThi()

{
```

```
base.HienThi();  
  
Console.WriteLine("Bang cap: {0}", bangcap);  
  
}  
  
}  
  
}
```

## Tính kế thừa trong C# – Bài tập thực hành

Xây dựng chương trình quản lý nhân sự trong một công ty. Chương trình phải có các chức năng như: quản lý thông tin nhân viên (mã nhân viên, tên nhân viên, giới tính, bằng cấp, ngày sinh) và thông tin khách hàng (mã khách hàng, tên khách hàng, giới tính, ngày sinh, loại khách hàng).

Chương trình cho phép người dùng lựa chọn thao tác thông qua menu

- Thêm nhân viên (nhập: 1)
- Thêm khách hàng (nhập: 2)
- Hiển thị thông tin nhân viên (nhập: 3)
- Hiển thị thông tin khách hàng (nhập: 4)



- Thống kê khách hàng (nhập: 5)
- Thoát chương trình (nhập: 6)

### **Khi người dùng nhập "1":**

- Yêu cầu người dùng nhập mã nhân viên , tên nhân viên, giới tính, ngày sinh và bằng cấp.
- Giới tính chỉ được phép nhập nam hoặc nữ.
- Bằng cấp chỉ được phép nhập trung cấp, cao đẳng hoặc đại học.
- Ngày sinh phải có độ dài là 10

### **Khi người dùng nhập "2":**

- Yêu cầu người dùng nhập mã khách hàng, tên khách hàng, giới tính, ngày sinh và loại khách hàng.
- Giới tính chỉ được phép nhập nam hoặc nữ.
- Loại khách hàng chỉ được phép nhập mới, thành viên hoặc vip.
- Ngày sinh phải có độ dài là 10

**Khi người dùng nhập “3”:** Hiển thị tất cả danh sách nhân viên hiện có trong công ty.

**Khi người dùng nhập “4”:** Hiển thị tất cả danh sách khách hàng hiện có của công ty.

**Khi người dùng nhập “5”:** Thống kê khách hàng dựa theo loại khách hàng. Chương trình sẽ hiển thị số lượng khách hàng mới, thành viên, vip.

**Khi người dùng nhập “6”:** Chương trình sẽ hiển thị thông báo xác nhận “Bạn có muốn thoát chương trình không (c/k)?”. Nếu nhập “c”, sẽ thoát chương trình.