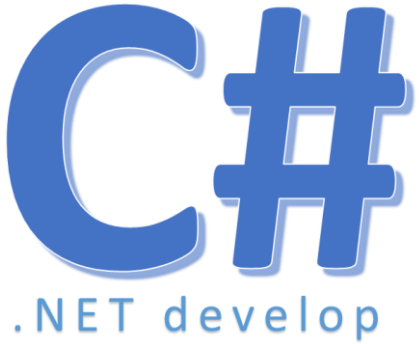
CHƯƠNG 5
KẾ THỪA - INHERITANCE



Nội dung

- Khái niệm kế thừa
- Xây dựng lớp cơ sở
- Xây dựng lớp dẫn xuất
- Từ khóa new

Kế thừa

Đặt vấn đề

Ví dụ 1: Xây dựng lớp **ngày** trong ứng dụng tính tiền lãi của một ngân hàng thành lập ngày 14/3/1997

Giả sử đã xây dựng lớp **CNGAY**

- Cách 1: Sửa lại lớp CNGAY cho phù hợp với các yêu cầu của lớp **CNGAYNH** trong ứng dụng trên → Sửa lại hàm kiểm tra → Ảnh hưởng đến các chương trình khác có sử dụng lớp CNGAY ở dạng tổng quát.
- Cách 2: Xây dựng lớp CNGAYNH độc lập với lớp CNGAY → Tốn nhiều công sức.
- Cách 3: Sao chép lớp CNGAY để tạo lớp CNGAYNH và sau đó sửa lại lớp CNGAYNH theo yêu cầu của chương trình → Khó khăn do thực hiện thủ công khi mở rộng, cập nhật, ...

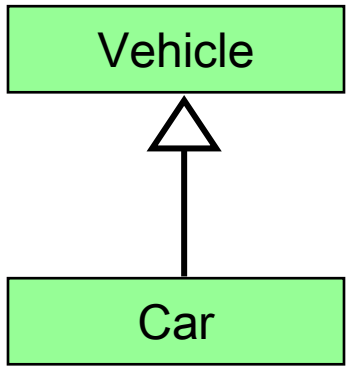
➔ Cần có cơ chế cho phép khai báo lớp CNGAYNH là lớp CNGAY với 1 số các sửa đổi bổ sung.

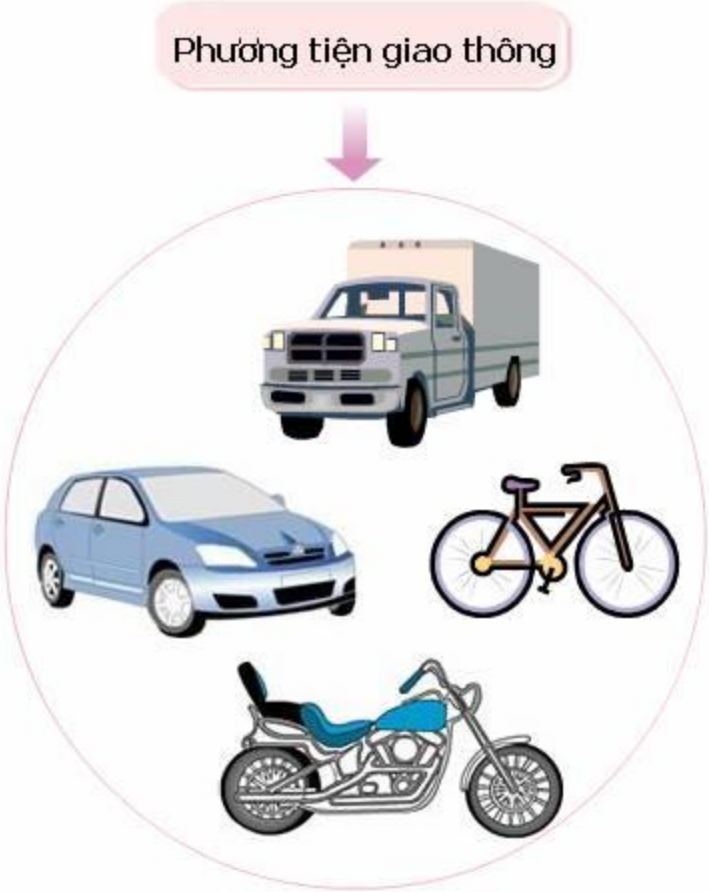
Kế thừa

Các đối tượng có cùng chung một số đặc điểm, hành vi được nhóm lại với nhau

- Xe đạp
- Xe máy
- Xe hơi
- Xe tải

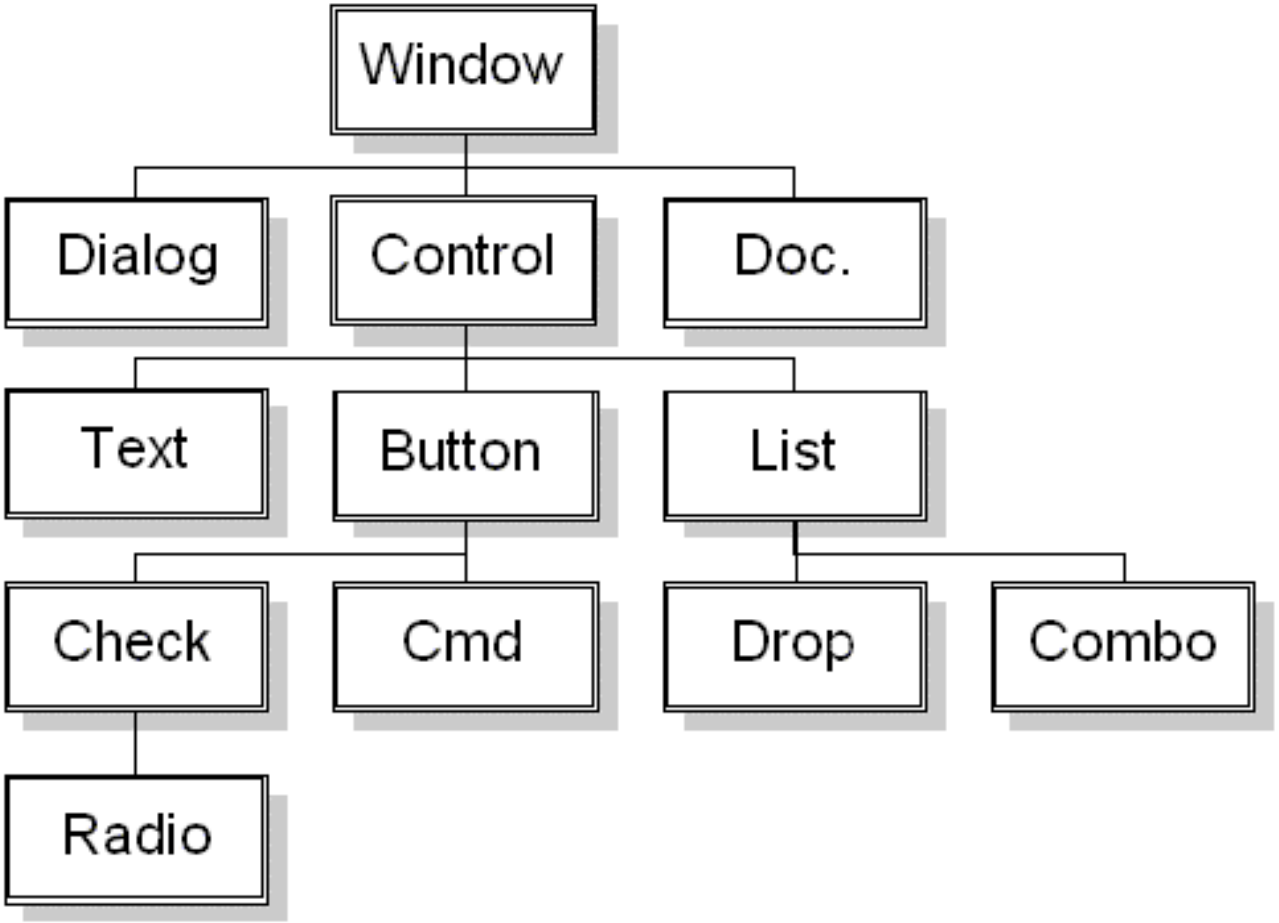
➔ Phương tiện giao thông





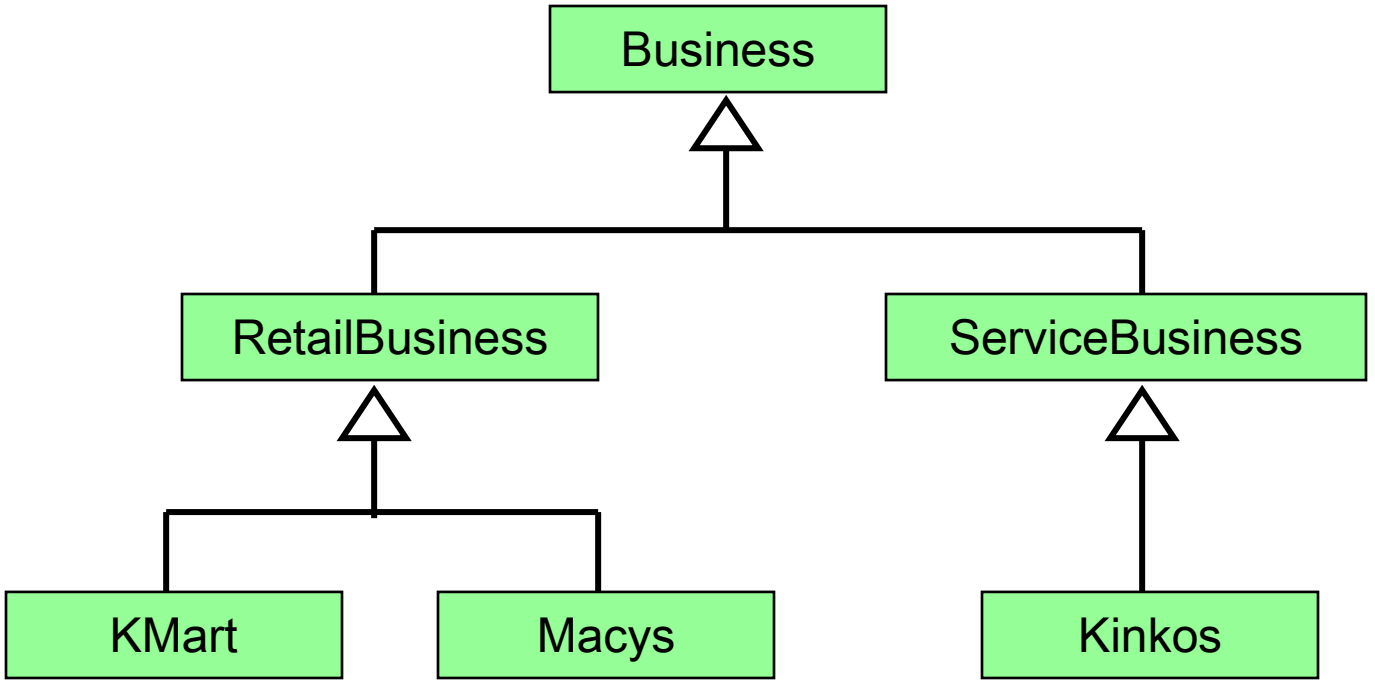
Kế thừa

Ví dụ: Windows form



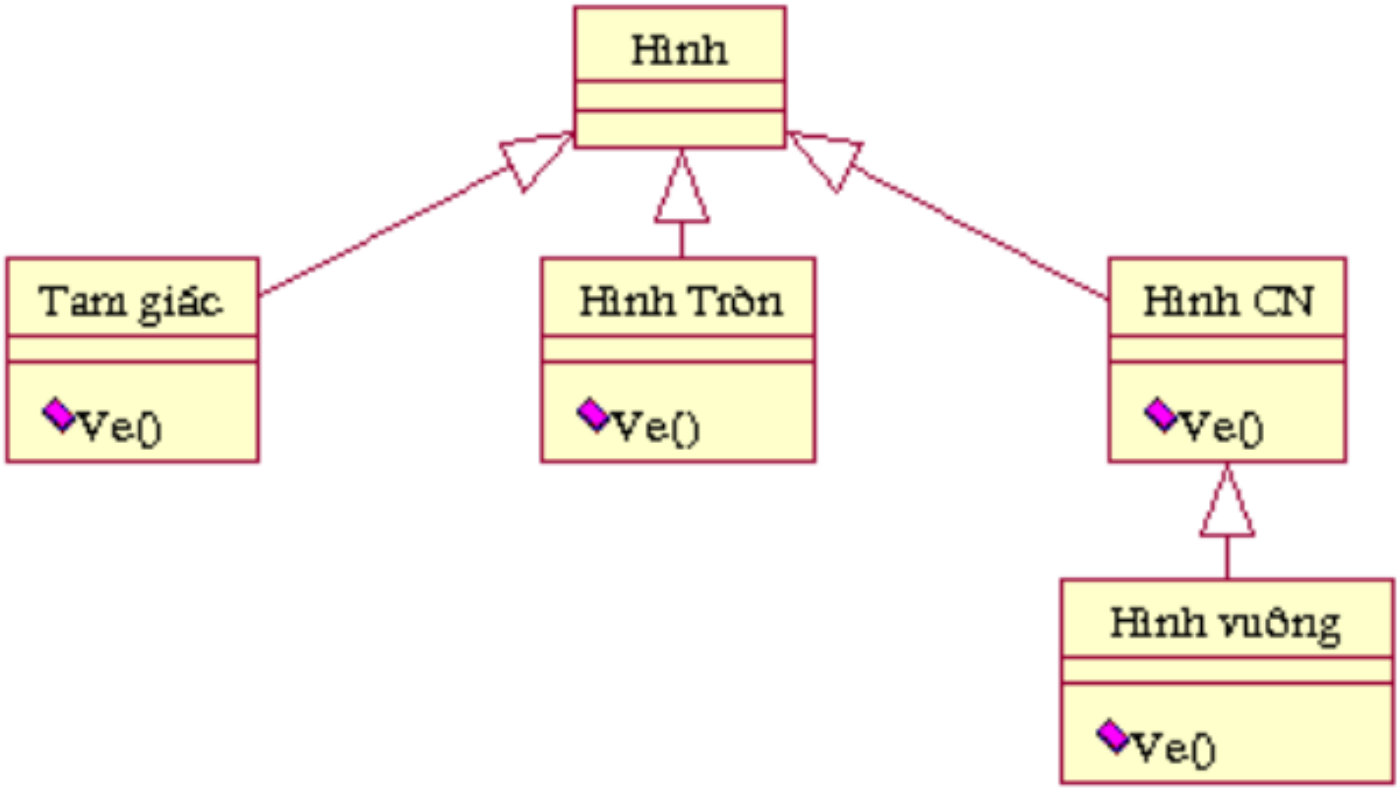
Kế thừa

- Ví dụ: Một lớp con có thể là lớp cha của các lớp khác



Kế thừa

- Ví dụ: Lớp kế thừa hình học



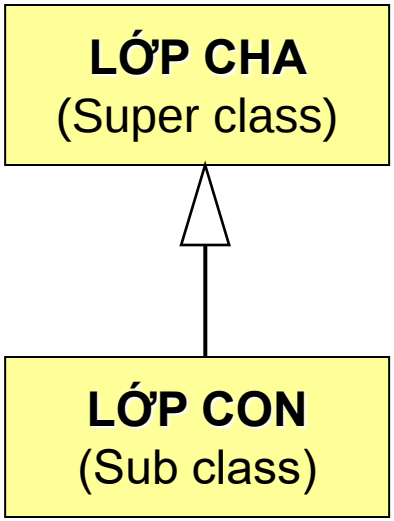
Kế thừa

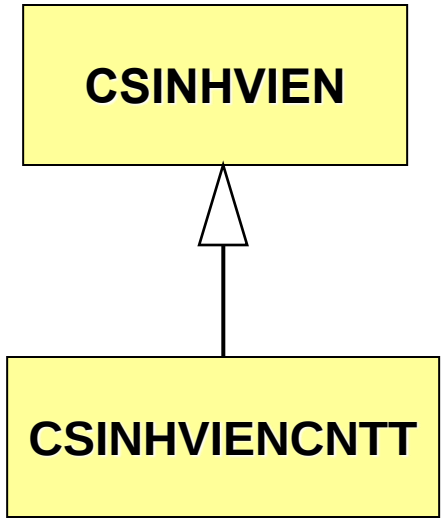
- Tạo ra các lớp mới từ việc sử dụng lại những thành phần của lớp đã có.
- Kế thừa cho phép khai báo 1 lớp B là 1 lớp dẫn xuất từ lớp A. Khi đó B sẽ có tất cả các thuộc tính và đặc điểm của A, ngoài ra B có thể có thêm những thuộc tính và những hành động mới.
- Lớp có sẵn được gọi là lớp **cơ sở** (based class) và lớp được kế thừa được gọi là lớp **dẫn xuất** (derived class)

Kế thừa

Lớp cơ sở
(Base class)

Lớp dẫn xuất
(Derived class)





Kế thừa

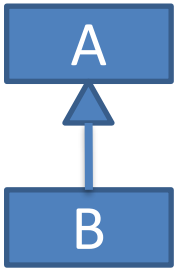
- Lợi ích
  - Nhất quán
  - Thuận tiện
  - Tái sử dụng code

Kế thừa

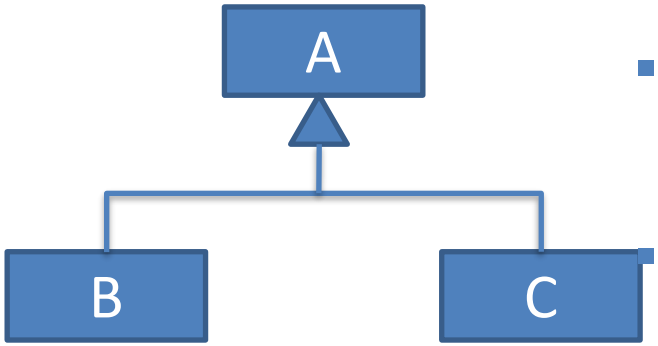
Trong 1 mô tả dự án thực tế làm sao phát hiện được mô hình lớp mà ở đó có tính kế thừa?

Cần nắm được khái niệm
Tổng quát hóa? chuyên biệt hóa?

Kế thừa



- A: Là trường hợp tổng quát của B
- B: Là trường hợp đặc biệt của A



- A: Là trường hợp tổng quát của B và C
- B, C: Là trường hợp đặc biệt của A

Ví dụ:

"***Sơn dương*** *là một loài động vật,* ***đại bàng*** *cũng là một loài động vật* ", thì có nghĩa là: "*Sơn dương và đại bàng là những loại động vật chuyên biệt, chúng có những đặc điểm chung của động vật và ngoài ra chúng có những đặc điểm phân biệt nhau*". Và như vậy, ***động vật*** là tổng quát hóa của **sơn dương** và ***đại bàng***; ***sơn dương*** và ***đại bàng*** là chuyên biệt hóa của ***động vật.***

Khai báo Kế thừa

- Lớp cơ sở (base class): làm cơ sở để các lớp khác kế thừa
- Lớp dẫn xuất (derived class): kế thừa đặc điểm của lớp cơ sở
- Khai báo
  - class **LopDanXuat** : **LopCoSo**
  - Ví dụ:
    - class XeDap : PhuongTienGiaoThong
    - class XeMay : PhuongTienGiaoThong

Constructor

- Không được kế thừa
- Lớp con truy cập bằng từ khóa **base**

```
class HinhHoc
 {
    ...
    public HinhHoc(double chuVi, double dienTich)
    {
       ChuVi = chuVi;
       DienTich = dienTich;
    }
 }
 class HinhTron : HinhHoc
 {
   ...
    public HinhTron(double chuVi, double dienTich) : base(chuVi, dienTich)
    {
    }
 }
```

Từ khóa base

- Dùng để truy cập đến thành phần của lớp cơ sở
- Mức truy cập:
  - public
  - protected
- Khai báo

```
class <baseName>{
    ...
<access_modifier>  <return_type> <Base_Method> (list_of_argument){
    }
}
class <derivedName> : <baseName>{
        base.<Base_Method>( ... );
}
```

Từ khóa new

- Dùng để khai báo phương thức ở lớp dẫn xuất khi đã có phương thức cùng tên ở lớp cơ sở.

```
class CNHANVIEN
{
   protected int maso;
   protected string hoten;
   public CNHANVIEN()
   {          maso = 0; hoten = "";       }
   public void Nhap()
   {   Console.Write("Nhap ma so nhan vien: ");
      maso = int.Parse(Console.ReadLine());
      Console.Write("Nhap ho ten nhan vien: ");
      hoten = Console.ReadLine();
   }
   public void Xuat()
   {    Console.WriteLine("Ma so: {0}\nHo ten: {1}", maso, hoten);   }
}
```

Từ khóa new

```
class CBIENCHE : CNHANVIEN
  {
     private float hesoluong;
     public CBIENCHE(): base() //dùng từ khóa base() để sử dụng lại constructor của lớp CNHANVIEN
     {          hesoluong = 0;       }
     public new void Nhap() // Vì lớp CNHANVIEN có  phương thức cùng tên Nhap() nên ta dùng new
     {
        base.Nhap();
        Console.Write("Nhap he so luong: ");
        hesoluong = float.Parse(Console.ReadLine());
     }
     public new void Xuat()
     {
        base.Xuat();
        Console.WriteLine("He so luong: " + hesoluong);
     }
  }
```

Từ khóa new

```
class CHOPDONG : CNHANVIEN
{  private float sogio;
   public new void Nhap()
   {  base.Nhap();
      Console.Write("Nhap so gio lam viec: ");
      sogio = float.Parse(Console.ReadLine());
   }
   public new void Xuat()
   {  base.Xuat();
      Console.WriteLine("So gio lam viec: " + sogio);
   }
}
```

FAQs