

Chương 4. Kế thừa lớp đối tượng

Nội dung

#2

- Khái niệm về tính kế thừa
 - Thiết kế lớp kế thừa
 - Thiết lập và hủy trong kế thừa
- ## Đặt vấn đề

#3

Giả sử đã xây dựng lớp CDate hoàn chỉnh

Cần xây dựng ứng dụng tính tiền lãi của một ngân hàng thành lập ngày 14/3/1997

Cần xây dựng ứng dụng quản lý sinh viên có thuộc tính ngày tháng năm sinh (sinh viên phải từ 17 tuổi trở lên)

Đặt vấn đề

Cách 1: Sửa lại lớp CDate cho phù hợp với các yêu cầu của lớp CDate trong ứng dụng trên

Sửa lại hàm kiểm tra Ảnh hưởng đến các chương trình khác có sử dụng lớp CDate ở dạng tổng quát.

Đặt vấn đề

Cách 2: Xây dựng lớp CDate mới độc lập với lớp CDate Tốn nhiều công sức.

Cách 3: Sao chép lớp CDate để tạo lớp CDate mới và sau đó sửa lại theo yêu cầu của chương trình Khó khăn do thực hiện thủ công khi mở rộng, cập nhật, ...

Đặt vấn đề

#6

Cần có cơ chế cho phép khai báo lớp CDate mới là lớp CDate cũ với 1 số các sửa đổi bổ sung.

Đặt vấn đề

#7

Tương tự cho chương trình đánh caro, cờ tướng trên máy tính. Mỗi quân cờ được xem

như 1 điểm ký tự (CDiemKT) nhưng mỗi quân cờ có những đặc điểm khác nhau. Do vậy cần sử dụng lớp CDiemKT bổ sung và sửa đổi một số phần chứ không phải tốn công sức để xây dựng lại từ đầu.

Khái

niệm  #8

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.

Khái niệm

#9

● Kế thừa thể hiện khả năng tái sử dụng các

lớp đã được định nghĩa.

- Có thể định nghĩa lớp đối tượng mới dựa trên 1 hay nhiều lớp đối tượng đã có sẵn.
- Lớp có sẵn được gọi là lớp cơ sở (based class) và lớp kế thừa được gọi là lớp dẫn xuất (derived class)

Khái

niệm  #10

* * * *

A B

C

- - * - - * - - * - - * - - + * + + + * + * + * *

* + + + * +

A B

* tính chất chung

- tính chất của A + tính chất của B

+ + + + + + + + + + +

- - - - -

Ký

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

VD: Lớp ngày cho ngân hàng và sinh

viên  #12

Khai

báo  #13

```
class TênLớpCha
```

```
{
```

Thuộc tính và phương thức của lớp

```
cha }
```

```
class TênLớpDẫnXuất :
```

TênLớpCha {

Thuộc tính và phương thức bổ sung
của lớp dẫn xuất

}

Khai

báo  #14

Có 2 cách để định nghĩa hành động bổ sung
cho phương thức đã có sẵn ở lớp cha trong lớp
dẫn xuất (phương thức lớp dẫn xuất trùng tên

với phương thức lớp cha)

- Dùng từ khóa **new**
- Dùng từ khóa **virtual** và **override**

Khai báo – Dùng từ khóa

new  #15

```
class COSO
{
    protected kiểu data1;
    protected kiểu data2;
```

```
public void Method1()
{
}
public void Method2()
{
}
```

| | |
|---|--|
| <pre> } class DANXUAT : COSO { private kiểu data3; public new void </pre> | <pre> Method1() {} public void Method4() {} } </pre> |
|---|--|

Khai báo – Dùng virtual & override

#16

| | |
|---------------------------|--|
| <pre> class COSO { </pre> | <pre> protected kiểu data1; protected kiểu data2; public virtual void </pre> |
|---------------------------|--|

```

Method1() {}
public virtual void
Method2() {}
}
class DANXUAT : COSO
{
    private kiểu data3;

```

```

public override void
Method1() {}
public void Method4()
{}

```

```

}
```

Ví

dụ  #17

Viết chương trình nhập xuất nhân viên, biết rằng gồm 2 loại nhân viên: Nhân viên biên chế và nhân viên hợp

đồng. Thông tin của nhân viên gồm: Mã số, Họ tên.

- Nhân viên biên chế có thông tin riêng là bậc lương. ▪

Nhân viên hợp đồng có thông tin riêng là số giờ làm.

Ví

dụ  #18

Ta có cây kế thừa sau:

VD dùng từ khoá new

#19

```
class CNHANVIEN  
{ protected int maso;
```

```
protected string 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);
}
}
```

Ví dụ – Dùng từ khoá new

#20

```
class CBIENCHE : CNHANVIEN
{
```

```
private float hesoluong;  
public new void Nhap()  
{ 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); }  
}
```

Ví dụ – Dùng từ khoá new

#21

```
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); }  
}
```

Ví dụ – Dùng virtual &

override #22

```
class CNHANVIEN
```

```
{ protected int maso; protected string hoten;
```

```
public virtual 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 virtual void Xuat()
```

```
{ Console.WriteLine("Ma so: {0}\nHo ten: {1}", maso, hoten);
```

```
}
```

```
}
```

Ví dụ – Dùng virtual & override

#23

```
class CBIENCHE : CNHANVIEN
{
    private float hesoluong;
    public override void Nhap()
    { base.Nhap();
        Console.Write("Nhap he so luong: ");
        hesoluong =
float.Parse(Console.ReadLine()); }
    public override void Xuat()
    { base.Xuat();
        Console.WriteLine("He so luong: " +
hesoluong); }
}
```

Ví dụ – Dùng virtual & override

#24

```
class CHOPDONG : CNHANVIEN
```

```
{ private float sogio;
```

```
    public override void Nhap()
```

```
    { base.Nhap();
```

```
        Console.Write("Nhap so gio lam viec: ");
```

```
        sogio = float.Parse(Console.ReadLine());
```

```
    }
```


```
    public override void Xuat()
```

```
    { base.Xuat();
```

```
        Console.WriteLine("So gio lam viec: " +  
sogio); }
```


}

Ví dụ - Sử dụng phương thức trong

Main()  #25

```
static void Main(string[] args)
{ CBIENCHE nvbc = new CBIENCHE();
  nvbc.Nhap();
  CHOPDONG nvhd = new
  CHOPDONG(); nvhd.Nhap();
  Console.WriteLine("\nNhan vien bien che:
  "); nvbc.Xuat();
  Console.WriteLine("\nNhan vien hop dong:
```

```
"); nvhd.Xuat();  
}
```

Phạm vi kế thừa #26

Có 3 phạm vi kế thừa:

- public
- protected
- private

Lưu ý: Nếu không nói rõ là phạm vi kế thừa gì, chúng ta ngầm định đó là kế thừa **public**

Phạm vi kế
thừa  #27

- **public**: thành phần public & protected của lớp cơ sở là thành phần public & protected của lớp dẫn xuất.
- **protected**: thành phần public & protected của lớp cơ sở là thành phần protected của

lớp dẫn xuất.

- **private**: thành phần public & protected của lớp cơ sở là thành phần private của lớp dẫn xuất.

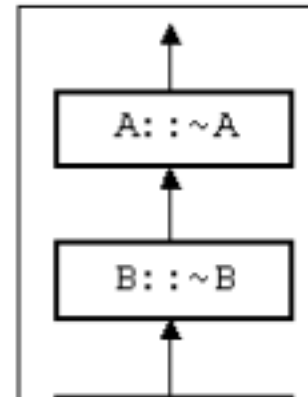
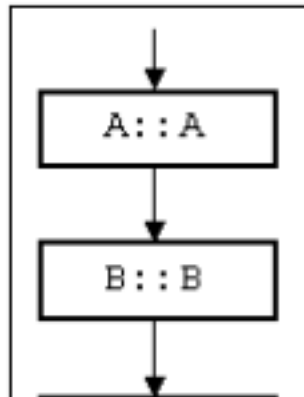
Phương thức thiết lập & huỷ trong kế thừa #28

- Khi khởi tạo đối tượng:
 - Phương thức thiết lập của **lớp cha** sẽ được gọi trước
 - Sau đó mới là phương thức thiết lập của **lớp**

con.

- Khi hủy đối tượng:
 - Phương thức hủy của **lớp con** sẽ được gọi trước
 - Sau đó mới là phương thức hủy của **lớp cha**.

Phương thức thiết lập & hủy trong kế thừa



A

B

C

Phương thức thiết lập & huỷ trong kế thừa

#30

Trong phương thức thiết lập của lớp dẫn xuất, chúng ta có thể chỉ định phương thức thiết lập nào của lớp cơ sở sẽ được gọi thực hiện. **Nếu**

không chỉ định, phương thức thiết lập mặc định của lớp cơ sở sẽ được gọi.

Phương thức thiết lập & huỷ trong kế thừa

```
class A
```

```
{
```

```
    public A() {}
```

```
    public A(int) {}
```

```
}  
class B : public A  
{  
    public B(int) //Thực hiện A()  
    {}  
}
```

Phương thức thiết lập & huỷ trong kế
thừa

#32

```
class A  
{  
    public A() {}
```



```
public A(int){}
}
class B : public A
{
    public B(int) : base(int) //Thực hiện
    A(int) {}
}
```

Bài tập

#33

Thiết kế chương trình quản lý các hoạt động sau trong một Viện khoa học: nhà khoa học, nhà quản lý và NV phòng thí nghiệm. Các thao tác phân bổ liệu của các hoạt động trên:

- Nhà khoa học: hỏi thăm, khám sinh, bằng cấp, chúc vui, số báo

baùo ñaõ coâng boá, soá ngaøy coâng trong thaùng, baäc löông

- Nhaø quaûn lý hoï teân, naêm sinh, baêng caáp, chöùc vuiï, soá ngaøy coâng trong thaùng, baäc löông
- NV phoøng thí nghiệãm: hoï teân, naêm sinh, baêng caáp, löông trong thaùng. Thöïc hieän caùc yeâu caàu sau:
- Caùc phöông thöùc thieát läp ñeã nhaäp lieäu, bieát raêng nhaân vieân phoøng thí nghiệãm laõnh löông khoaùn, coøn löông cuûa nhaø khoa hoïc vaø nhaø quaûn lý baêng soá ngaøy coâng trong thaùng * baäc löông.
- Xuaát döõ lieäu ra maøn hình
- In toaùng löông ñaõ chi traû cho töøng loaïi ñoái tööïng.

FAQs

