

# **Chương 10**

## **ĐA XẠ-PHƯƠNG THỨC ẢO**

# 1. MỤC TIÊU

- Hiểu được cơ chế hoạt động của phương thức ảo.
- Ứng dụng được phương thức ảo.
- Thi cao học đề thi hay hỏi phần này.
- Phỏng vấn xin việc người ta cũng rất hay hỏi.

## 2. VÍ DỤ DẪN NHẬP

- Hãy thực hiện đoạn chương trình dưới đây và cho biết kết quả của việc chạy chương trình trong bốn trường hợp:
- *Trường hợp 1:* **XXXX** là khoảng trắng,  
**YYYY** là khoảng trắng.
- *Trường hợp 2:* **XXXX** là virtual,  
**YYYY** là khoảng trắng.
- *Trường hợp 3:* **XXXX** là khoảng trắng,  
**YYYY** là virtual.
- *Trường hợp 4:* **XXXX** là virtual,  
**YYYY** là virtual.

## 2. VÍ DỤ DẪN NHẬP (tiếp)

```
11. #include <iostream.h>
12. class A
13. {
14.     public:
15.         XXXX void Sketchy()
16.         {
17.             cout<<"\n A`sSketchy() ";
18.             Sketchy(-1);
19.         }
20.         YYYY void Sketchy(int num)
21.         {
22.             cout<<"\n A's Sketchy
                (int)"<< num;
23.         }
24. } ;
```

## 2. VÍ DỤ DẪN NHẬP (tiếp)

```
11.class B:public A
12.{
13.    public:
14.        void Sketchy()
15.        {
16.            cout<<"\nB`s Sketchy() ";
17.            Sketchy(-2);
18.        }
19.        void Sketchy(int num)
20.        {
21.            cout<<"\n B's Sketchy
                (int)"<< num;
22.        }
23.};
```

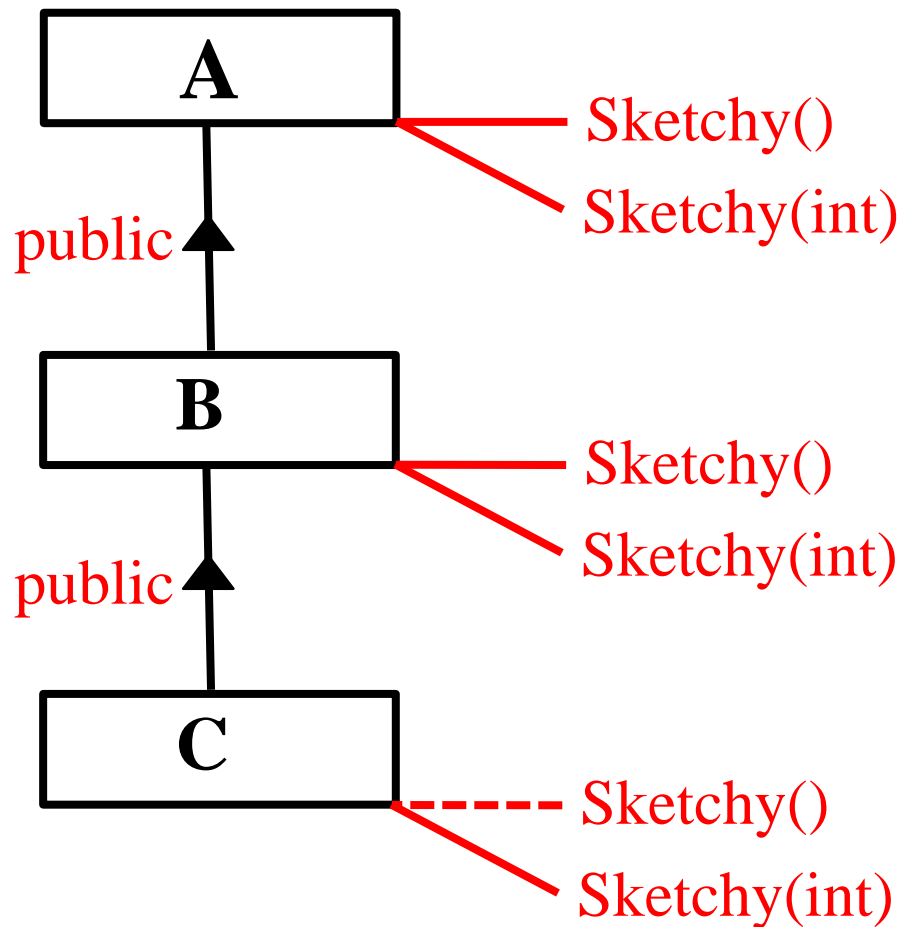
## 2. VÍ DỤ DẪN NHẬP (tiếp)

```
1. class C:public B
2. {
3.     public:
4.         void Sketchy(int num)
5.         {
6.             cout<<"\nC's Sketchy
7.             (int) "<<num;
8.         }
9. };
10. void Curious(A* wacky)
11. {
12.     wacky->Sketchy();
13.     ((C*)wacky)->Sketchy(123);
14. }
```

## 2. VÍ DỤ DẪN NHẬP (tiếp)

```
1. void main()  
2. {  
3.     A* inky = new B;  
4.     inky->Sketchy();  
5.     inky->Sketchy(23);  
6.     Curious(inky);  
7.     B* pinky = new C;  
8.     pinky->Sketchy();  
9.     pinky->Sketchy(46);  
10.    Curious(pinky);  
11. }
```

## 2. VÍ DỤ DẪN NHẬP (tiếp)





## 2. VÍ DỤ DẪN NHẬP (tiếp)

- Kết quả trường hợp 01

## **2. VÍ DỤ DẪN NHẬP (tiếp)**

- Kết quả trường hợp 02

## **2. VÍ DỤ DẪN NHẬP (tiếp)**

- Kết quả trường hợp 03

## **2. VÍ DỤ DẪN NHẬP (tiếp)**

- Kết quả trường hợp 04

### 3. KHÁI NIỆM

- Khái niệm: Đa xạ là cơ chế **tầm vực động**, cho phép "*xác định*" đúng hành vi (*phương thức*) của đối tượng khi yêu cầu thực hiện.
- Việc "*xác định*" được thực hiện theo nguyên tắc tự nhiên: **đối tượng thuộc lớp nào sẽ gọi thực hiện phương thức của lớp đối tượng đó.**
- **Tầm vực động là cơ chế gọi thực hiện phương thức thông qua con trỏ đối tượng.**

## 4. CÚ PHÁP ĐA XẠ

```
11. class CCoSo
12. {
13.     private:
14.         ...
15.     protected:
16.         ...
17.     public:
18.         ...
19.         virtual KDL <TenPhuongThuc>
                                (<ThamSo>);
20. } ;
```

## 4. CÚ PHÁP ĐA XẠ

```
11. class CDanXuat:<Từ Khóa Dẫn Xuất>
                                CCoSo
12. {
13.     private:
14.         ...
15.     protected:
16.         ...
17.     public:
18.         ...
19.         KDL <TenPhuongThuc>
                                (<ThamSo>) ;
20. } ;
```

```
11. class CCoSo
12. {
13.     private:
14.         ...
15.     protected:
16.         ...
17.     public:
18.         ...
19.         virtual KDL <TenPhuongThuc>
                                   (<ThamSo>);
20. };
21. class CDanXuat:<Từ Khóa Dẫn Xuất>
                                   CCoSo
22. {
23.     private:
24.         ...
25.     protected:
26.         ...
27.     public:
28.         ...
29.         KDL <TenPhuongThuc>
                                   (<ThamSo>);
30. };
```



## 4. CÚ PHÁP ĐA XẠ (tiếp)

- Một phương thức được khai báo bắt đầu với từ khóa **virtual** thì được gọi là phương thức ảo và phương thức này được gọi thực hiện theo cơ chế đa xạ nếu lời gọi thực hiện phương thức được thông qua một con trỏ đối tượng.
- Các phương thức ở lớp dẫn xuất **cùng tên** và **cùng danh sách tham số đầu vào** thì cũng sẽ là phương thức ảo nếu ở lớp cơ sở **phương thức cùng tên** và **cùng tham số là phương thức ảo**.

## 5. BÀI TOÁN CHI TIẾT MÁY

- **Bài toán:** Một cái máy có nhiều chi tiết. Mỗi chi tiết máy có thể là chi tiết đơn hoặc là chi tiết phức.
  - + **Chi tiết đơn:** là chi tiết không chứa bên trong nó chi tiết khác. Thông tin của chi tiết đơn bao gồm: *mã số chi tiết, giá tiền*.
  - + **Chi tiết phức:** là chi tiết chứa bên trong nó nhiều chi tiết thành phần, mỗi một chi tiết thành phần này có thể là chi tiết đơn hoặc là chi tiết phức. Thông tin của chi tiết phức bao gồm: *mã số chi tiết, số lượng chi tiết thành phần, danh sách các chi tiết thành phần*. Giá tiền của chi tiết phức bằng tổng giá tiền của các chi tiết thành phần.

## 5. BÀI TOÁN CHI TIẾT MÁY (tiếp)

- **Yêu cầu:** Thiết kế các lớp thích hợp để thực hiện các yêu cầu sau:
  - + *Nhập các chi tiết cho máy.*
  - + *Tìm kiếm một chi tiết máy theo mã số.*
  - + *Tính tiền cho một chi tiết máy, hoặc cho máy.*
  - + *Xuất các chi tiết máy.*
  - + *Đếm số lượng chi tiết đơn có trong cái máy.*

## 6. BÀI TOÁN TÍNH TIỀN LƯƠNG

- **Bài toán:** Công ty ABC là công ty sản xuất kinh doanh thú nhồi bông. Công ty có nhiều nhân viên làm việc trong ba bộ phận khác nhau: *bộ phận quản lý, bộ phận sản xuất, bộ phận văn phòng*. Việc tính lương cho nhân viên dựa vào các yếu tố sau:

- + **Đối với nhân viên văn phòng**

Lương = Lương Cơ Bản + Số ngày làm việc \* 100.000 + Trợ Cấp

- + **Đối với nhân viên sản xuất**

Lương = Lương Cơ Bản + Số Sản Phẩm \* 2.000

- + **Đối với nhân viên quản lý**

Lương = Lương Cơ Bản \* Hệ số chức vụ + Thưởng.

## 6. BÀI TOÁN TÍNH TIỀN LƯƠNG (tiếp)

- + Ngoài ra công ty cần quản lý các thông tin về nhân viên của mình như: *họ tên, ngày sinh và các thông số trên* để tính lương cho từng nhân viên trong công ty.
- **Yêu cầu:** Thiết kế các lớp thích hợp để thực hiện các yêu cầu sau:
  - + *Nhập thông tin của các nhân viên để phục vụ cho việc tính lương.*
  - + *Thực hiện việc tính lương cho từng nhân viên.*
  - + *Xuất thông tin của các nhân viên.*
  - + *Tính tổng lương của công ty.*
  - + *Tìm kiếm một nhân viên theo họ tên.*