Hàm ảo – Hàm thuần ảo

Nguyễn Khắc Huy

Nội dung

- Hàm ảo
- Hàm thuần ảo
- Hàm hủy ảo
- Bài tập
- Đa kế thừa

Hàm ảo

- Con trỏ đối tượng trong kế thừa:
  - ✓ Truy xuất đối tượng bằng con trỏ => linh động.
  - ✓ Truy xuất đối tượng kế thừa bằng con trỏ lớp cơ sở.
  - ✓ Kiểu con trỏ quyết định phương thức được gọi
    - ➔ **liên kết tĩnh.**
  - ✓ Đối tượng kế thừa truyền vào hàm nhận tham số kiểu cơ sở.
    - ➔ **Đối tượng kế thừa có thể đóng vai trò đối tượng cơ sở.**

```
A     obj;
A     *p;
p = &obj;
p = new A;
```

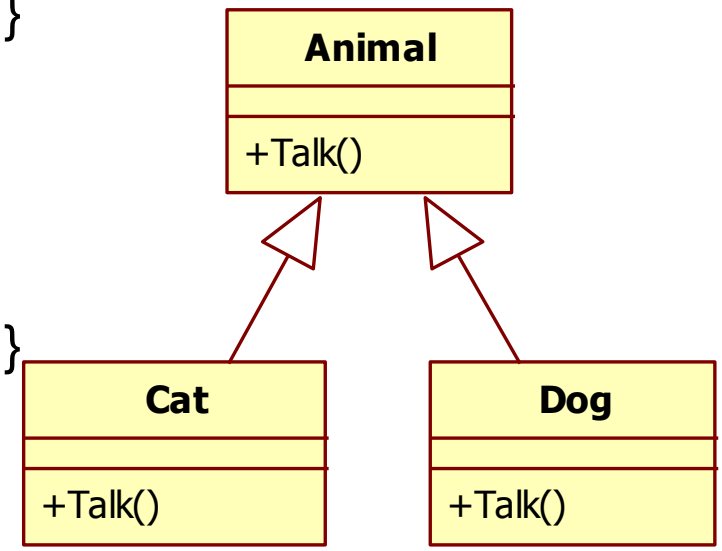
```
// B kế thừa A.
B     obj;
A     *p = &obj;
p->func();
```

```
// B kế thừa A.
void func(A obj) { }
B     obj;
func(obj);
```

Hàm ảo

- Ví dụ:

```
class Animal
{
public:
        void talk() {   cout << “Don’t talk!”;   }
};
class Cat: public Animal
{
public:
        void talk() {   cout << “Meo meo!”;   }
};
class Dog: public Animal
{
public:
        void talk() {   cout << “Gau gau!”;   }
};
```



Hàm ảo

- Ví dụ:

```
void giveATalk(Animal *p)
{
        p->talk();
}

void main()
{
        Cat     c;
        Dog     d;

        giveATalk(&c);
        giveATalk(&d);
}
```

```
void main()
{
        Animal a;
        Cat    c;
        Dog    d;

        Animal  *p;
        p = &a;
        p->talk();

        p = &c;
        p->talk();

        p = &d;
        p->talk();
}
```

Animal talks!!

Animal talks!!

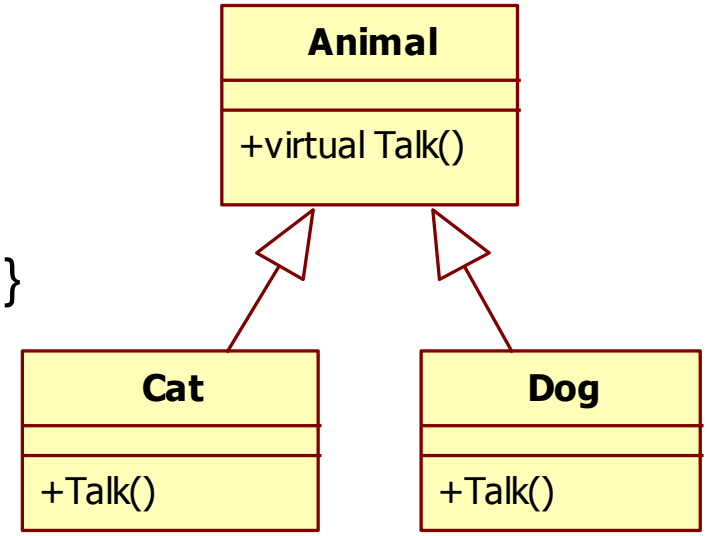
Animal talks!!

Animal talks!!

Hàm ảo

- Ví dụ:

```
class Animal
{
public:
        virtual void talk() {   cout << “Don’t talk!”;   }
};
class Cat: public Animal
{
public:
        void talk() {   cout << “Meo meo!”;   }
};
class Dog: public Animal
{
public:
        void talk() {   cout << “Gau gau!”;   }
};
```



Hàm ảo

- Ví dụ:

```
void giveATalk(Animal *p)
{
        p->talk();
}

void main()
{
        Cat     c;
        Dog     d;

        giveATalk(&c);
        giveATalk(&d);
}
```

```
void main()
{
        Animal  a;
        Cat     c;
        Dog     d;

        Animal  *p;
        p = &a;
        p->talk();

        p = &c;
        p->talk();

        p = &d;
        p->talk();
}
```

Animal talks!!

Animal talks!!

Animal talks!!

Animal talks!!

Animal talks!!

Hàm ảo

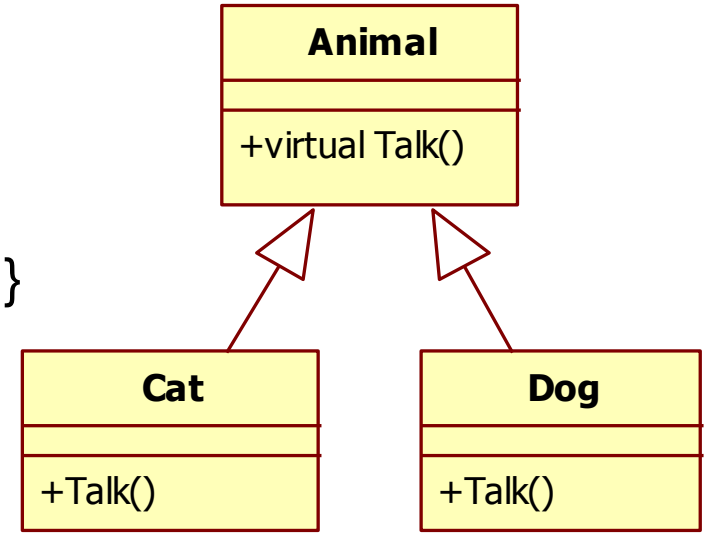
- Khái niệm hàm ảo:
  - ✓Một phương thức của lớp.
  - ✓Mang tính ảo.
    - ➔ Chuyển lời gọi hàm cho đúng đối tượng con trỏ đang trỏ đến.
    - ➔ Liên kết động.
  - ✓Chỉ có ý nghĩa khi gọi thông qua con trỏ.
  - ✓Khai báo hàm ảo trong C++:

    virtual <Chữ ký hàm>;

Hàm ảo

- Ví dụ:

```
class Animal
{
public:
        virtual void talk() {   cout << “Don’t talk!”;   }
};
class Cat: public Animal
{
public:
        void talk() {   cout << “Meo meo!”;   }
};
class Dog: public Animal
{
public:
        void talk() {   cout << “Gau gau!”;   }
};
```



Hàm ảo

- Ví dụ:

```
void giveATalk(Animal *p)
{
        p->talk();
}

void main()
{
        Cat     c;
        Dog     d;

        giveATalk(&c);
        giveATalk(&d);
}
```

```
void main()
{
        Animal  a;
        Cat     c;
        Dog     d;

        Animal  *p;
        p = &a;
        p->Talk();

        p = &c;
        p->Talk();

        p = &d;
        p->Talk();
}
```

Animal talks!!

Cat talks!!

Cat talks!!

Dog talks!!

Dog talks!!

Hàm ảo

- Sử dụng hàm ảo để làm gì?
  - ✓Gọn gàng, đơn giản, uyển chuyển, linh động.
    - ➔ Chương trình có tính dễ mở rộng, nâng cấp.

```
void giveATalk(Animal *p)
{
        p->talk();
}
```

```
void giveATalk(Animal obj, int
iType)
{
        if (iType == 0)
        {
                Cat   c = (Cat)obj;
                c.talk();
        }
        else if (iType == 1)
        {
                Dog   d = (Dog)obj;
                d.talk();
        }
}
```

Nội dung

- Đa kế thừa
- Hàm ảo
- Hàm thuần ảo
- Hàm hủy ảo
- Bài tập

Hàm thuần ảo

- Có một số hàm ảo không thể cài đặt hoặc không có ý nghĩa khi cài đặt trong lớp cơ sở.

```
class Animal
{
public:
        virtual void talk() {   cout << “Don’t talk!”;   }
};
```

Biến thành hàm thuần ảo!!

Hàm thuần ảo

- Khái niệm hàm thuần ảo:
  - ✓Hàm ảo chỉ có khai báo mà không có cài đặt.
  - ✓Phần cài đặt do lớp kế thừa đảm nhận.
  - ✓Khai báo trong C++:

    virtual <Chữ ký hàm> **= 0**;
- Lớp trừu tượng (abstract class):
  - ✓Lớp chứa hàm thuần ảo.
  - ✓Không thể tạo đối tượng từ lớp trừu tượng.
  - ✓Chỉ dùng để kế thừa.

Hàm thuần ảo

- Ví dụ:

```
class Animal
{
public:
        virtual void talk() = 0;
};
class Cat: public Animal
{
public:
        void talk() {   cout << “Meo meo!”;   }
};
class Dog: public Animal
{
public:
        void talk() {   cout << “Gau gau!”;   }
};
```

```
void main()
{
    Animal  a;                    // Sai.
    Animal  *p = new Animal; // Sai.
    Animal  *q = new Cat;      // Đúng.

    q->talk();
}
```

Cat talks!!

Nội dung

- Đa kế thừa
- Hàm ảo
- Hàm thuần ảo
- Hàm hủy ảo
- Bài tập

Hàm hủy ảo

- Ví dụ:

```
class GiaoVien
{
private:
        char      *m_strHoTen;
public:
        ~GiaoVien() { delete m_
};
class GVCN : public GiaoVi
{
private:
        char      *m_strLopCN;
public:
        ~GVCN() { delete m_strLopCN; }
};
```

```
void main()
{
    GiaoVien  *p1 = new GiaoVien;
    delete p1;

    GVCN     *p2 = new GVCN;
    delete p2;

    GiaoVien  *p3 = new GVCN;
    delete p3;
}
```

~GiaoVien()

~GVCN()
~GiaoVien()

~GiaoVien()

Hàm hủy ảo

- Dr. Guru khuyên:
  - ✓**Hàm hủy của lớp phải luôn là hàm ảo.**
    - ➔ Chuyển lời gọi đến hàm hủy của lớp kế thừa.

```
class GiaoVien
{
private:
    char  *m_strHoTen;
public:
    virtual ~GiaoVien() { delete m_strHoTen; }
};

GiaoVien  *p3 = new GVCN;
delete p3;
```

~GVCN()
~GiaoVien()

Tóm tắt

- Đa kế thừa

- Hàm ảo:
  - ✓Chuyển lời gọi hàm đến đúng đối tượng.
  - ✓Chỉ có ý nghĩa khi gọi từ con trỏ.
- Hàm thuần ảo:
  - ✓Hàm ảo chỉ có khai báo mà không có cài đặt.
  - ✓Lớp kế thừa đảm nhận việc cài đặt.
  - ✓Lớp có chứa hàm thuần ảo ➔ lớp trừu tượng
  - ✓Lớp trừu tượng chỉ dùng để kế thừa.
- Hàm hủy ảo:
  - ✓Hàm hủy phải luôn luôn là hàm ảo.

Nội dung

- Đa kế thừa
- Hàm ảo
- Hàm thuần ảo
- Hàm hủy ảo
- Bài tập

Bài tập

- Bài tập 9.1:

```
class A
{ public:
    [yyy] void f1() { cout << “Good morning.\n”; f2(); }
    [zzz] void f2() { cout << “Good afternoon.\n”; }
};
class B: public A
{ public:
    void f1() { cout << “Good evening.\n”; f2(); }
    void f2() { cout << “Good night.\n”; }
};
void main()
{
    A   *pObj = new B;
    pObj->f1();
}
```

Cho biết những gì xuất hiện trên màn hình trong các trường hợp:

a) [yyy] trống, [zzz] trống.

b) [yyy] trống, [zzz] virtual.

c) [yyy] virtual, [zzz] trống.

d) [yyy] virtual, [zzz] virtual.

Bài tập

- Bài tập 9.2:

  Có 2 loại hình: đường thẳng và hình chữ nhật.

  - Đường thẳng: biểu diễn bởi hai điểm đầu cuối.

  - Hình chữ nhật: biểu diễn bởi hai điểm trên trái và dưới phải.

  Giả sử có danh sách các hình thuộc 2 loại trên. Viết chương trình xuất thông tin của từng hình trong danh sách đó.

  Sau đó, giả sử có thêm loại hình mới là hình tròn.

  - Hình tròn: biểu diễn bởi tâm và bán kính.

  Khi đó, chương trình sẽ phải được chỉnh sửa như thế nào?

Bài tập

- Bài tập 9.3:

  Tốc độ chạy của các động vật cho bởi bảng sau:

Động vật	Tốc độ
Báo	100km/h
Linh dương	80km/h
Sư tử	70km/h
Chó	60km/h
Người	30km/h

Viết chương trình cho phép so sánh tốc độ chạy giữa một cặp động vật bất kỳ thuộc nhóm trên.

Thêm vào con ngựa chạy 60km/h, chương trình sẽ thay đổi thế nào?