

Inheritance – Kế thừa

CSC10003 – Phương pháp lập trình hướng đối tượng

Nội dung thực hành lần này sẽ tập trung phân tích và hướng dẫn một số vấn đề về tính kế thừa trong lập trình hướng đối tượng

1. Tính chất kế thừa trong lập trình hướng đối tượng

- ✚ Kế thừa là một tính chất quan trọng trong lập trình hướng đối tượng, có khả năng định nghĩa một lớp đối tượng dựa trên một hoặc nhiều lớp đối tượng khác được định nghĩa trước đó
- ✚ Kế thừa giải quyết vấn đề về tái sử dụng, trùng lặp thông tin.
- ✚ Lớp A kế thừa từ lớp B, khi đó:
 - Lớp B là lớp cha hay lớp cơ sở
 - Lớp A là lớp kế thừa, lớp con hay lớp dẫn xuấtLớp A thừa kế đầy đủ thuộc tính và phương thức của lớp cha
- ✚ Lớp con được thừa kế đầy đủ những tính chất được định nghĩa trong lớp cha
 - **Lớp con sẽ có đầy đủ thuộc tính và phương thức được định nghĩa trong lớp cha**
- ✚ Lớp con là một trường hợp đặc biệt của lớp cha, lớp cha là tổng quát hóa của lớp con
 - **Lớp con có thể định nghĩa thêm thuộc tính và phương thức riêng**
- ✚ Khai báo một lớp kế thừa theo cấu trúc sau:

```
class <Tên lớp con> : <Loại kế thừa> <Tên lớp cha>
{
    private:
        <Khai báo thành phần riêng>

    public:
        <Khai báo thành phần chung>

};
```

- ✚ Ví dụ về tính kế thừa:
 - Chó và mèo đều là động vật, đều có các bộ phận trên cơ thể, đều có hành động ăn, uống, ngủ. Chó khác mèo ở chỗ tiếng kêu khác nhau. Vậy chó mèo kế thừa từ lớp động vật.
 - Trong trường học sẽ có nhiều giáo viên: giáo viên chủ nhiệm, giáo viên thỉnh giảng, ... Tất cả giáo viên đều có chung thông tin là Họ tên, mức lương, thâm niên giảng dạy. Giáo viên chủ nhiệm sẽ có thêm thông tin về lớp chủ nhiệm, có thêm hoạt động sinh hoạt chủ nhiệm
 - Cài đặt ví dụ cho trường hợp kế thừa lớp giáo viên

```
class GiaoVien
{
    string HoTen;
    int ThamNienGiangDay;
    int MucLuong;

public:
    GiaoVien();
    GiaoVien(string hoten, int thamnien, int mucluong);
    void TinhLuong();
};

//Lớp giáo viên chủ nhiệm kế thừa từ lớp giáo viên
//Lớp giáo viên chủ nhiệm là trường hợp đặc biệt
//Lớp giáo viên chủ nhiệm kế thừa lớp giáo viên với tầm vực public
//Lớp giáo viên kế thừa thuộc tính và phương thức của lớp giáo viên
class GiaoVienCN : public GiaoVien
{
    string LopChuNhiem; //Thuộc tính riêng chỉ có GVCN mới có

public:
    //Phương thức riêng chỉ có GVCN mới có
    GiaoVienCN(string hoten, int thamnien, int mucluong, string lopcn);
    void SinhHoatLopCN();
};
```



- ✚ Lớp con có thể sử dụng nạp chồng hàm để cài đặt lại phương thức có cách xử lý khác so với lớp cha, mặc dù vẫn có chung ý nghĩa
 - Phương thức cần cài đặt lại cần phải giống y hệt với khai báo của lớp cha
 - Có thể gọi lại phương thức ở lớp cha để xử lý nếu cần thiết
 - Ví dụ sau cài đặt hàm tính lương cho giáo viên và giáo viên thỉnh giảng, giáo viên thỉnh giảng kế thừa từ lớp giáo viên với cách tính lương khác với lớp giáo viên

```
class GiaoVien
{
    int MucLuong;
public:
    float TinhLuong()
    {
        return MucLuong * 100000;
    }
};
```

```
class GiaoVienTG : public GiaoVien
{
    //Thuộc tính riêng của lớp con
    int SoMonThinhGiang;

public:
    float TinhLuong()
    {
        //Gọi lại phương thức tính lương của lớp cha
        return GiaoVien::TinhLuong() * SoMonThinhGiang * 40000;
    }
};
```

2. Tầm vực kế thừa

- Tầm vực protected: nếu lớp cha khai báo thuộc tính và phương thức với tầm vực protected thì chỉ những lớp con kế thừa từ lớp cha đó mới có thể truy xuất sử dụng lại.

Tầm vực của thuộc tính và phương thức	Chỉ truy xuất sử dụng bên trong lớp	Truy xuất sử dụng từ lớp con kế thừa	Truy xuất sử dụng bên ngoài lớp
Public	X	X	X
Protected	X	X	
Private	X		

- Lớp con kế thừa từ lớp cha sẽ có tầm vực kế thừa với 3 hình thức kế thừa:
 - Kế thừa public
 - Kế thừa protected
 - Kế thừa private

- Bảng chuyển đổi tầm vực của thuộc tính và phương thức theo 3 hình thức kế thừa:

Tầm vực của thuộc tính và phương thức	Kế thừa public	Kế thừa protected	Kế thừa private
Public	public	protected	private
Protected	protected	protected	private
Private	private	private	private

- ✚ Dựa trên bảng chuyển đổi tầm vực, có thể thấy rằng:
 - Lớp con kế thừa public từ lớp cha thì giữ nguyên tầm vực thuộc tính và phương thức của lớp cha
 - Lớp con kế thừa protected từ lớp cha thì thuộc tính và phương thức public ở lớp cha khi chuyển xuống con xài thì sẽ chuyển sang tầm vực protected
 - Lớp con kế thừa private từ lớp cha thì chuyển tất cả tầm vực sang private

3. Constructor và Destructor trong kế thừa

- ✚ Do có lớp con kế thừa lớp cha, constructor và destructor của cả 2 sẽ được gọi khi tạo lập hay hủy đối tượng. Tuy nhiên, cần lưu ý thứ tự gọi trong kế thừa
 - ✚ **Khi gọi constructor, lớp cha sẽ thực thi trước, sau đó mới đến lớp con**
 - ✚ **Khi gọi destructor, lớp con sẽ thực thi trước, sau đó mới đến lớp cha**
 - ✚ **Lớp nào có thuộc tính kiểu con trở thì tự lớp đó xử lý**
 - ✚ Nếu lớp cha và lớp con đều có nhiều constructor, trong đó ta có thể quy định constructor nào của lớp cha sẽ được gọi trước khi gọi constructor của lớp con.
 - **Nếu không quy định, hàm dựng mặc định của lớp cha sẽ được gọi!!!**
 - ✚ Ví dụ cài đặt quy định gọi constructor ở lớp cha của lớp con:

```
class GiaoVien
{
    string HoTen;
    int ThamNienGiangDay;
    int MucLuong;

public:
    //Default constructor
    GiaoVien()
    {
        HoTen = "";
        ThamNienGiangDay = 0;
        MucLuong = 0;
    }

    //Constructor có nhiều tham số
    GiaoVien(string hoten, int thamniengd, int mucluong)
    {
        HoTen = hoten;
        ThamNienGiangDay = thamniengd;
        MucLuong = mucluong;
    }
}
```

```
//Copy constructor
GiaoVien(const GiaoVien& gv)
{
    HoTen = gv.HoTen;
    ThamNienGiangDay = gv.ThamNienGiangDay;
    MucLuong = gv.MucLuong;
}
};

class GiaoVienTG : public GiaoVien
{
    //Thuộc tính riêng của lớp con
    int SoMonThinhGiang;


public:
    //Constructor có nhiều tham số
    //Quy định gọi constructor có nhiều tham số ở lớp cha khi gọi thực thi
    GiaoVienTG(string hoten, int thamniengd, int mucluong, int somontg) :
        GiaoVien(hoten, thamniengd, mucluong)
    {
        SoMonThinhGiang = somontg;
    }

    //Copy constructor quy định gọi copy constructor ở lớp cha khi thực thi
    //Copy trước những cái kế thừa có sẵn, xử lý copy thuộc tính riêng
    GiaoVienTG(const GiaoVienTG &gvtg) : GiaoVien(gvtg)
    {
        SoMonThinhGiang = gvtg.SoMonThinhGiang;
    }
};
```


4. Quan hệ kế thừa IS-A trong UML

 Quan hệ IS-A thể hiện cho sự kế thừa, lớp A có quan hệ IS-A với lớp B thì:

- A là một trường hợp đặc biệt của B
- A cùng loại với B
- A kế thừa B

 Ví dụ một số đối tượng có quan hệ IS-A

- Con mèo là một loại động vật (lớp con mèo kế thừa từ lớp động vật)
- Hình vuông là trường hợp đặc biệt của hình chữ nhật
- Tài khoản vip là một loại tài khoản

 Nguyên tắc tìm quan hệ kế thừa: xem xét lớp đó có những điểm chung nào? Lớp đó có phải là một loại của lớp khác không? Có là trường hợp đặc biệt không?

- Ký hiệu UML cho quan hệ kế thừa IS-A: hình mũi tên với đầu tam giác chỉ vào lớp cha
- Ví dụ: lớp con mèo kế thừa từ lớp động vật



5. Bài tập thực hành

Bài tập 1: Hãy xây dựng chương trình C++ để hệ thống hóa nhân sự của một trường đại học. Trường đại học này có đội ngũ gồm: giảng viên, trợ giảng, nghiên cứu viên và chuyên viên. Tất cả đội ngũ nhân sự đều có thông tin về họ tên, ngày tháng năm sinh, mã số nhân sự. Mỗi vai trò cần phải được cung cấp thêm thông tin và có cách tính lương dựa trên tính chất công việc, cụ thể:

- Giảng viên cần phải được cung cấp thông tin về học hàm, học vị, số năm giảng dạy và danh sách tên tất cả các môn học đang giảng dạy. Trợ giảng cần cung cấp thông tin về số môn trợ giảng. Giảng viên và trợ giảng được tính lương theo công thức:

$$\text{lương giảng viên} = (\text{số môn giảng dạy} * \text{số năm giảng dạy} * 0.12) * 20000$$

$$\text{lương trợ giảng} = (\text{số môn trợ giảng} * 0.3) * 18000$$

- Nghiên cứu viên được cung cấp thông tin về danh sách tất cả mã dự án nghiên cứu đang tham gia, và chuyên viên phải cung cấp thông tin về danh sách tất cả mã dự án giáo dục đã hoàn thành. Mỗi dự án đều có mã dự án bắt đầu bằng một chữ cái kèm theo 2 chữ số, ví dụ một số mã dự án: M01, X92, ... và nghiên cứu viên còn có thêm thông tin về số năm kinh nghiệm nghiên cứu, chuyên viên có thông tin về số năm kinh nghiệm làm việc. Nghiên cứu viên và chuyên viên được tính lương theo công thức:

$$\text{lương nghiên cứu viên} = (\text{số năm KN nghiên cứu} * 2 + \text{số dự án có mã bắt đầu bằng chữ D}) * 20000$$

$$\text{lương chuyên viên} = (\text{số năm KN làm việc} * 4 + \text{số dự án có mã bắt đầu bằng chữ T}) * 18000$$

Sinh viên dựa trên mô tả bài tập, tự đề xuất xây dựng các class kèm với các thuộc tính và phương thức cần thiết. Sinh viên chú ý tầm vực của các class và của các thuộc tính và phương thức phải hợp lý và thỏa các tính chất hướng đối tượng. Ngoài ra sinh viên cung cấp thêm cách thức nhập xuất cho bài tập này, ở hàm main, sinh viên cài đặt cho phép nhập vào vector các nhân sự của trường đại học, hỗ trợ các cách thức nhập xuất để người dùng nhập vào đúng loại

nhân sự cần tạo và tính tổng lương của tất cả nhân sự mà trường đại học này cần phải trả, đồng thời vẽ sơ đồ lớp UML cho bài tập này.

Bài tập 2: Viết chương trình C++ tự đề xuất khai báo class cùng với các thuộc tính và phương thức mà sinh viên nghĩ class đó có thể có đồng thời vẽ sơ đồ lớp UML dựa vào các quan hệ sau:

- Hình vuông – hình chữ nhật
- Thư viện – sách

Bài tập 3: Phân tích ý nghĩa của sơ đồ UML sau:

