



Lập Trình Hướng Đối Tượng

CHƯƠNG 5: HÀM BẠN (FRIEND)

TRẦN VIỆT KHÁNH

NỘI DUNG

1

LỚP DẪN XUẤT

2

LỚP FRIEND

3

SỰ KHÁC NHAU GIỮA LỚP FRIEND VÀ LỚP DẪN XUẤT
(DELIVER)

Định nghĩa lớp dẫn xuất

- Một lớp được xây dựng thừa kế một lớp khác được gọi là lớp dẫn xuất (lớp con); lớp được dùng để xây dựng lớp dẫn xuất được gọi là lớp cơ sở (lớp cha).
- Một lớp dẫn xuất ngoài các thành phần riêng của lớp đó, còn được thừa kế các thành phần của các lớp cơ sở có liên quan

Cách xây dựng lớp dẫn xuất

- Giả sử ta đã xây dựng lớp A, B. Để xây dựng lớp C kế thừa public các lớp A và B, ta viết theo mẫu sau:

```
class C: public A, public B
{
    private:
        //Cac thuoc tinh
    public:
        //Cac phuong thuc
};
```

Tương tự, ta thay từ khóa public bằng private để C kế thừa private từ A và B

Cách xây dựng lớp dẫn xuất

Tính chất:

Kế thừa theo kiểu public thì tất cả các thành phần public của lớp cơ sở sẽ là thành phần public của lớp dẫn xuất. Kế thừa theo kiểu private thì tất cả các thành phần public của lớp cơ sở sẽ là thành phần private của lớp dẫn xuất.

Thừa kế các thuộc tính

Các thuộc tính của lớp cơ sở sẽ được thừa kế trong lớp dẫn xuất. Như vậy thì tập thuộc tính của lớp dẫn xuất sẽ gồm: các thuộc tính mới khai báo trong định nghĩa lớp dẫn xuất và các thuộc tính của lớp cơ sở. Tuy nhiên, trong lớp dẫn xuất không cho phép truy nhập đến các thuộc tính private của lớp cơ sở.

Lớp Bạn (Friend)

Mục đích:

Friend được xây dựng để khắc phục điểm yếu **lớp dẫn xuất không thể truy cập tới các biến private của lớp cơ sở.**

Một friend có thể là một hàm, một mẫu hàm, hoặc hàm thành viên, hoặc một lớp hoặc một mẫu lớp, trong trường hợp này, toàn bộ lớp và tất cả thành viên của nó là friend. Hàm friend trong C# của một lớp được định nghĩa bên ngoài phạm vi lớp đó, nhưng nó có quyền truy cập tất cả thành viên private và protected của lớp đó. Ngay cả khi các nguyên mẫu cho hàm friend xuất hiện trong định nghĩa lớp, thì các hàm friend không là các hàm thành viên.

Tính chất của hàm friend

Friend của một class có thể là thành viên của 1 class khác

Friend của 1 class có thể là thành viên của class khác hoặc tất cả các class trong cùng 1 chương trình. Như là 1 GLOBAL FRIEND

Friend có thể access private hoặc protected của class được khai báo là Friend.

Friends không phải là một thành viên vì vậy không có con trỏ "this".

Friend có thể khai báo ở bất cứ đâu (public, protected or private section) trong một class.

Khai báo và sử dụng

Để khai báo một hàm dạng hàm friend của một lớp, đặt trước nguyên mẫu hàm đó trong định nghĩa lớp với từ khóa friend trong C#, như sau:

```
class Person
{
    private:
        string name;

    public:
        friend void DisplayName( Person person); //hàm hiển thị tên
        void setName( string name );
};
```

Để khai báo tất cả hàm thành viên của lớp Employee là dạng friend của lớp Person, đặt một khai báo sau trong định nghĩa của lớp Person:

```
class Person
{
    friend class Employee; // Employee là friend của person
};
```

Sự khác nhau giữa Friend (Lớp Bạn) và Deliver (Lớp Dẫn xuất)

Trong lớp dẫn xuất không cho phép truy nhập đến các thuộc tính private của lớp cơ sở.

Vậy nếu bạn muốn tạo quan hệ giữa 2 class A và B. Sao cho B có thể truy cập các private hay protected của A nhưng bạn không muốn xây dựng các phương thức set hoặc get để tránh các class khác ngoài B cũng truy cập được thì làm thế nào?. Đó là truy cập có chọn lọc và từ khóa **Friend** sẽ giúp bạn làm điều này.

Ví dụ

```
class Person
{
    private:
        string name;

    public:
        //Hàm DisplayName không phải thành viên của person mà là Friend. Có thể gọi ở bất cứ đâu trong chương trình
        friend void DisplayName( Person person);
        void setName( string name );

        // phân định nghĩa ham setname
        void Person::setName( string name )
        {
            name = name;
        }
        // Ghi chú: DisplayName() không là ham thành viên của bất cứ class nào.
        void DisplayName( Person person )
        {
            /* Bởi vì, hàm DisplayName() là hàm friend của Person, do vậy nó có thể
            trực tiếp truy cập bất cứ thành viên nào của class này */
            Console.WriteLine("Tên của bạn là: " + person.name);
        }
};

// hàm main của chương trình
int main( )
{
    Person person;

    // set tên cho person không sử dụng hàm thành viên
    person.setName("Đỗ Trung Quân");

    // sử dụng hàm friend để in ra tên.
    DisplayName( person);

    return 0;
}
```

FAQs





XIN CẢM ƠN !

www.lytc.edu.vn