




Lập trình C++ 2



Nội dung chương trình C++2

- Giới thiệu Lập trình hướng đối tượng
- Xây dựng các lớp cơ bản
- Quá tải toán tử và phương thức
- Kế thừa và nạp chồng
- Con trỏ và cấp phát bộ nhớ động
- Hàm ảo và đa hình
- Các vấn đề khác



Bài 1: Lập trình hướng đối tượng và Lớp cơ bản

- Khái niệm lập trình hướng đối tượng
- Xây dựng các lớp cơ bản
- Private và public
- Phương thức Set và Get
- So sánh class và struct
- Bài tập

3



Lập trình thủ tục

- Giải bài toán với các cấu trúc điều khiển
 - Cấu trúc tuần tự
 - Cấu trúc rẽ nhánh: if, switch
 - Cấu trúc lặp: while, do-while, for
- Lập trình thủ tục (Procedural Programming)
 - Thiết kế top-down
 - Hàm (function)
 - Chương trình được chia thành các chức năng, là sự phối hợp của các chức năng

4

Ví dụ về lập trình thủ tục

- Tính diện tích và chu vi hình tròn

- float TinhChuvi(int bk);
- float TinhDientich(int bk);
- void main()
- {
- ...
- chuvi = TinhChuvi(bankinh);
- dtich = TinhDientich(bankinh);
- ...
- }

5

Lập trình hướng đối tượng - OOP

- Lập trình hướng đối tượng - OOP
(*Object-Oriented Programming*)
- Tại sao lại OOP ?
 - Dễ phát triển chương trình.
 - Dễ sử dụng lại những gì đã được xây dựng trước.
 - Hiện nay OOP được dùng rất phổ biến, đặc biệt để xây dựng các phần mềm lớn.

6

OOP là gì ?

- Lập trình hướng đối tượng
 - Thay vì bao gồm các chức năng, **một chương trình trong OOP bao gồm các đối tượng**
 - Một đối tượng gồm 2 thành phần: dữ liệu (data) và các phương thức (methods)
 - Các đối tượng giao tiếp, trao đổi với nhau thông qua những phương thức mà chúng định nghĩa

7

OOP là gì ?

- Đối tượng (object) là sự kết hợp giữa dữ liệu và các cách thức xử lý dữ liệu đó.
 - Ví dụ 1 - Đối tượng Hình tròn
 - Thuộc tính là bán kính hình tròn
 - Phương thức gồm tính diện tích, tính chu vi...
 - Ví dụ 2 - Đối tượng Sinh viên
 - Thuộc tính là mã sinh viên, điểm các môn học...
 - Phương thức gồm tính điểm trung bình, hiển thị thông tin...
- Lớp (class) được dùng để chỉ một tập hợp các đối tượng cùng kiểu. Đối tượng còn được gọi là thể hiện (instance) của lớp.

8

Ví dụ chương trình HĐT

Chương trình tính toán với hình tròn

```
#include <iostream.h>
```

Khai báo lớp Hinhtron

```
class Hinhtron
```

```
{
```

```
private:
```

```
    int bankinh;
```

```
public:
```

```
    void DatBankinh(int bk);
```

```
    float LayDientich();
```

```
    float LayChuvi();
```

```
};
```

Khai báo thuộc tính của hình tròn là bán kính. Từ khoá private để chỉ rằng dữ liệu bankinh chỉ được dùng riêng trong lớp Hinhtron

Khai báo các phương thức cơ bản trên hình tròn là: thiết đặt bán kính, tính diện tích, tính chu vi. Từ khoá public để chỉ rằng các phương thức có thể được gọi ở bất kỳ đâu.

9

Ví dụ chương trình HĐT

- Định nghĩa các phương thức

```
void Hinhtron::DatBankinh(int bk)
{
    bankinh = bk;
}
float Hinhtron::LayDientich()
{
    return 3.14*bankinh*bankinh;
}
float Hinhtron::LayChuvi()
{
    return 2*3.14*bankinh;
}
```

10

Ví dụ chương trình HĐT

- Sử dụng đối tượng hình tròn

```
void main()
```

```
{
```

```
Hinhtron x;
```

Khai báo đối tượng (biến) x
thuộc lớp Hinhtron

```
x.DatBankinh(5);
```

Gọi phương thức
DatBankinh để đặt giá trị
bán kính cho x

```
cout<<"Dien tich la: "<<x.LayDientich();
```

```
cout<<"Chu vi la: "<<x.LayChuvi();
```

Gọi phương thức tính toán
trên giá trị bán kính đã thiết
đặt cho x

```
}
```

11

Đặc điểm của OOP

- Che dấu dữ liệu (Data hiding)
 - Hàm main không thể biết dữ liệu **bankinh** của lớp Hình tròn và không thể truy cập vào biến này (từ khoá private).
=> **Không thể ra lệnh: *x.bankinh = 5;* trong main**
 - Hàm main gọi phương thức LayDientich mà không cần biết cách tính diện tích của lớp Hinhtron như thế nào.
- Đóng gói (Encapsulation)
 - Dữ liệu và các thao tác trên dữ liệu (phương thức) được đặt trong cùng một đối tượng.
 - Chú ý rằng phương thức **LayChuvi** và **LayDientich** không có đối số, nó sẽ tính toán với giá trị **bankinh** trong bản thân đối tượng x.

12

Đặc điểm của OOP

- **private** và **public** là 2 từ khoá chỉ phạm vi (scope) của dữ liệu/phương thức.
- Điều gì xảy ra nếu thay **private** bằng **public** ?
 - **public:**
 - `int bankinh;`
Trả lời: Có thể ra lệnh ***x.bankinh = 5;*** trong hàm main
=> Vi phạm tính che dấu dữ liệu của OOP !!!
- *Dữ liệu của lớp thường có phạm vi là private. Chỉ nên đặt public cho những dữ liệu chung mà sự thay đổi của nó không ảnh hưởng lớn tới đối tượng.*

13

Đặc điểm của OOP

- Điều gì xảy ra nếu thay **public** bằng **private** ?
 - **private:**
 - `void DatBankinh(int bk);`
 - `float LayChuvi();`
 - `float LayDientich();`
Trả lời: Không thể gọi *x.LayChuvi()* và *x.LayDientich()* ở hàm main
=> Không thể sử dụng được đối tượng !!!
- *Các phương thức của lớp thường có phạm vi là public để cho các đối tượng khác có thể sử dụng được. Đặt phương thức là private nếu như thực sự muốn nó chỉ được sử dụng trong phạm vi của lớp đó.*
- Khi khai báo **public:** hoặc **private:** thì tất cả các dữ liệu/phương thức phía sau đều có phạm vi như vậy.
- Nếu không chỉ rõ thì phạm vi mặc định là **private:**

14

Dữ liệu và phương thức

- Trong lớp chỉ nên có những dữ liệu mà các đối tượng cần đến để tính toán, thao tác; những dữ liệu thể hiện đặc tính của đối tượng.
 - Ví dụ 1: Lớp Hinhvuong có dữ liệu là độ dài một cạnh.
 - Ví dụ 2: Lớp Hinhchunhat có dữ liệu là chiều dài và chiều rộng.
- Phương thức của một lớp là những thao tác cần dùng trên dữ liệu.
 - Ví dụ 1: Lớp Hinhvuong có thao tác tính diện tích, tính chu vi, tính độ dài đường chéo...

15

Phương thức Đặt (Set) và Lấy (Get)

- Hầu hết mỗi dữ liệu của lớp đều có 2 phương thức để đặt giá trị và lấy giá trị. (*Tại sao ?*)
- Ví dụ lớp Hinhtron:
 - void Hinhtron::DatBankinh(int bk)
 - {
 - bankinh = bk;
 - }
 - int Hinhtron::LayBankinh()
 - {
 - return bankinh;
 - }

16

So sánh struct và class

- **struct** thường được dùng để chứa dữ liệu, không chứa phương thức.
- **class** chứa dữ liệu và các phương thức trên dữ liệu đó.
- Trong C++, có thể dùng từ khoá **struct** để định nghĩa lớp và phạm vi mặc định là public. Tuy nhiên cách này ít được sử dụng.

17

Xây dựng một chương trình HĐT

- **Bước 1:** Chỉ ra các đối tượng cần có trong chương trình và dữ liệu + phương thức của từng đối tượng. Xác định quan hệ giữa các đối tượng.
- **Bước 2:** Cài đặt lớp cho các đối tượng. Chỉ ra phạm vi của dữ liệu và phương thức. Định nghĩa phương thức.
- **Bước 3:** Sử dụng các đối tượng trong chương trình để giải bài toán.

18



Ôn lại

- Sự khác nhau giữa Lập trình hướng đối tượng và Lập trình thủ tục ?
- Đặc điểm của OOP ?
- Đối tượng là gì ?
- Lớp là gì ?
- Phân biệt public và private ?
- Phương thức Đặt (Set) và Lấy (Get) ?
- Sự khác nhau giữa struct và class ?

19



Bài tập

- Cài đặt lớp Hinhtron đã học. Thêm phương thức trả lại giá trị bán kính hình tròn.
- Xây dựng lớp Sinh viên với dữ liệu gồm tên sinh viên, mã sinh viên, điểm LT và điểm TH C++ 1. Tạo các phương thức cần thiết.

20