

NGÔN NGỮ LẬP TRÌNH C/C++

Nguyễn Hải Châu
Khoa Công nghệ thông tin
Trường Đại học Công nghệ
(Bài giảng tuần 10)

1

Nội dung

- Đối tượng trong C++ (tiếp)
 - Đối của phương thức
 - Con trỏ this

2

Ví dụ: Phương thức nhapsl()

```
void DIEM::nhapsl()  
{  
    cout << "\n Nhập hoành do (cot) va tung do  
    (hang) cua diem:" ;  
    cin >> x >> y ;  
    cout << "\n Nhập ma mau cua diem: " ;  
    cin >> m ;  
}
```

3

Con trỏ this

```
void DIEM::nhapsl()  
{  
    cout << "\n Nhập hoành do (cot) va tung do  
    (hang) cua diem:" ;  
    cin >> this->x >> this->y ;  
    cout << "\n Nhập ma mau cua diem: " ;  
    cin >> this->m ;  
}
```

4

Ví dụ

- DIEM d1;
- d1.nhapsl();
- Khi đó:
 - this = &d1; và do đó:
 - this → x chính là d1.x
 - this → y chính là d1.y
 - this → m chính là d1.m
- Tham số truyền cho đối con trỏ this chính là địa chỉ của đối tượng đi kèm với phương thức trong lời gọi phương thức

5

Các đối khác của phương thức

```
void DIEM::doan_thang(DIEM d2, int mau)  
{  
    int mau_ht;  
    mau_ht = getcolor();  
    setcolor(mau);  
    line(this->x, this->y, d2.x, d2.y);  
    setcolor(mau_ht);  
}
```

- Xem thêm ví dụ trang 227-229

6

Hàm tạo/Cấu tử (constructor)

- Hàm tạo (hay cấu tử) là một phương thức đặc biệt để khởi tạo đối tượng
- Cách viết hàm tạo:
 - Tên của hàm tạo: Tên của hàm tạo bắt buộc phải trùng với tên của lớp
 - Không khai báo kiểu cho hàm tạo
 - Hàm tạo không có kết quả trả về
- Trong một lớp có thể có nhiều hàm tạo (trùng tên nhưng khác số lượng đối)

7

Ví dụ hàm tạo

```
class DIEM_DH {
private:
    int x, y, m ;
public:
    // Hàm tạo không đối: Khởi tạo x = 0, y = 0, m = 1
    DIEM_DH()
    {
        x = y = 0; m = 1;
    }
}
```

8

```
// Hàm tạo này xây dựng bên ngoài định nghĩa lớp
DIEM_DH(int x1, int y1, int m1 = 15) ;
// Các phương thức khác
};
// Xây dựng hàm tạo bên ngoài định nghĩa lớp
DIEM_DH::DIEM_DH(int x1, int y1, int m1) ;
{
    x = x1; y = y1; m = m1;
}
```

9

Sử dụng hàm tạo trong khai báo

```
DIEM_DH d; // Gọi tới hàm tạo không đối.
// Kết quả d.x = 0, d.y = 0, d.m = 1

DIEM_DH u(300, 100, 5); // Gọi tới hàm tạo có đối.
// Kết quả u.x = 300, u.y = 100, d.m = 5
```

10

Sử dụng hàm tạo trong cấp phát

```
DIEM_DH *r = new DIEM_DH ;
//Gọi tới hàm tạo không đối
// Kết quả r → x = 0, r → y = 0, r → m = 1

DIEM_DH *q = new DIEM_DH(40, 20, 4);
// Gọi tới hàm tạo có đối
// Kết quả q → x = 40, q → y = 20, q → m = 4
```

11

Dùng hàm tạo biểu diễn đối tượng hằng

- Tên_lớp(danh sách tham số) ;
- Ví dụ:

```
DIEM_DH(234, 123, 4) // Biểu thị một đối tượng kiểu DIEM_DH
// có các thuộc tính x = 234, y = 123, m = 4
```

12

Lớp không có hàm tạo, hàm tạo mặc định

- Lớp không có hàm tạo: Chương trình dịch cung cấp một hàm tạo mặc định không làm gì cả
- Lớp có ít nhất một hàm tạo: Chương trình dịch sẽ không cung cấp hàm tạo mặc định, mọi câu lệnh xây dựng đối tượng mới sẽ gọi đến hàm tạo đã có

13

Hàm hủy/Hủy tử (destructor)

- Hàm hủy là một phương thức của lớp có chức năng ngược với hàm tạo
- Hàm hủy được gọi trước khi giải phóng một đối tượng để thực hiện giải phóng bộ nhớ
- Hàm hủy mặc định: Do chương trình dịch tạo ra và không làm gì cả

14

Quy tắc viết hàm hủy

- Mỗi lớp chỉ có một hàm hủy viết theo các quy tắc sau:
 - Kiểu của hàm: Hàm hủy cũng giống như hàm tạo là hàm không có kiểu, không có giá trị trả về.
 - Tên hàm: Tên của hàm hủy gồm một dấu ngã (đứng trước) và tên lớp:
~Tên_lớp
 - Đối: Hàm hủy không có đối

15

Ví dụ hàm hủy

```
class DT
{
private:
    int n; // Bac da thuc
    double *a; // Tro toi vung nho...
public:
    ~DT()
    {
        this -> n = 0;
        delete this -> a;
    }
};
```

16

Tự đọc ở nhà: Các hàm inline

Từ trang 253 đến 257 trong giáo trình

17

Định nghĩa các phép toán cho lớp

- Đối với mỗi lớp ta có thể sử dụng lại các kí hiệu phép toán thông dụng (+, -, *, ...) để định nghĩa cho các phép toán của lớp
- Sau khi được định nghĩa các kí hiệu này sẽ được dùng như các phép toán của lớp theo cách viết thông thường.
- Cách định nghĩa này được gọi là phép chồng toán tử

18

Cách định nghĩa phép toán cho lớp

- **Tên hàm toán tử:** Gồm từ khoá operator và tên phép toán.
- Ví dụ:
 - operator+ (định nghĩa chồng phép toán +)
 - operator- (định nghĩa chồng phép toán -)
- Các đối của hàm toán tử: Số lượng các đối bằng số ngôi của phép toán
- Thân hàm toán tử viết như các hàm thông thường

19

Ví dụ

```
struct PS
{
    int a;          //Tử số
    int b;          //Mẫu số
};
PS operator+(PS p1, PS p2);    // p1 + p2
PS operator-(PS p1, PS p2);    // p1 - p2
PS operator*(PS p1, PS p2);    // p1 * p2
PS operator/(PS p1, PS p2);    // p1/p2
```

20

Cách dùng hàm toán tử

- Cách thứ nhất: Dùng như các hàm thông thường
- Ví dụ:

```
PS p, q, u, v ;
u = operator+(p, q) ;           // u = p + q
v = operator-(p, q) ;           // v = p - q
```
- Cách thứ hai: Theo cách viết của C++

```
u = p + q;
v = p - q;
```

21

Bài tập

- Thiết kế lớp vector:
 - Thể hiện cấu trúc dữ liệu của vector 3 chiều
 - Viết các phương thức thực hiện các phép toán trên vector: Cộng, trừ hai vector, nhân vector với một số, chuẩn hóa vector, tích vô hướng của hai vector 3 chiều

22