

NGÔN NGỮ LẬP TRÌNH C++

Chương 01.1. Cấu trúc chung của chương trình C++

Chương 01.2. Các kiểu dữ liệu cơ bản trong C++

Chương 01.3. Khai báo. Biểu thức. Khối lệnh

Chương 01.4. Vào/ra dữ liệu với C++

Chương 01.5. Các lệnh điều khiển chương trình

Chương 01.6. Mảng và chuỗi ký tự

Chương 01.7. Kiểu cấu trúc và kiểu liệt kê

Chương 01.8. Con trỏ

Chương 01.9. Hàm trong C++

Chương 01.1: Cấu trúc chung của chương trình C++

I. Giới thiệu về ngôn ngữ C++

II. Các phần tử cơ bản của ngôn ngữ C++

III. Cấu trúc chung của một chương trình C++ viết trên DOS

IV. Cấu trúc chung của một chương trình C++ viết trên Linux

I. Giới thiệu về ngôn ngữ C++

1. Lịch sử phát triển của ngôn ngữ C++

2. Tại sao ngôn ngữ C++ thông dụng?

3. Trình biên dịch Borland C++

1. Lịch sử phát triển của ngôn ngữ C++

- ◆ Năm 1973 ngôn ngữ lập trình C ra đời với mục đích ban đầu là để viết hệ điều hành Unix trên máy tính mini PDP. Sau đó C đã được sử dụng rộng rãi trên nhiều loại máy tính khác nhau và đã trở thành một ngôn ngữ lập trình có cấu trúc rất được ưa chuộng.
- ◆ Để đưa tư tưởng lập trình hướng đối tượng vào C, năm 1980 nhà khoa học người Mỹ B. Stroustrup đã cho ra đời một ngôn ngữ C mới có tên ban đầu là “C có lớp”, sau đó đến năm 1983 thì gọi là C++. Ngôn ngữ C++ là một sự phát triển cao của C. Trong C++ không chỉ đưa vào tất cả các khái niệm, công cụ của lập trình hướng đối tượng mà còn đưa vào nhiều khả năng mới cho hàm.

2. Tại sao ngôn ngữ C++ thông dụng?

- ◆ Mặc dù tư tưởng lập trình hướng đối tượng đã được đưa vào nhiều ngôn ngữ lập trình nhưng C++ vẫn là ngôn ngữ lập trình hướng đối tượng thông dụng bởi vì: C++ là ngôn ngữ kế thừa và mở rộng từ ngôn ngữ C (một ngôn ngữ cấu trúc rất được ưa chuộng). Vì có sự kế thừa nên tất cả các chương trình viết trên C đều chạy được trên C++.
- ◆ C++ có những đặc điểm tốt hơn C
 - Quản lý tên hàm đã được mở rộng thông qua cơ chế chồng hàm *function overloading*.

2. Tại sao ngôn ngữ C++ thông dụng?

- Tư tưởng phân vùng các biến namespaces cho phép quản lý các biến được tốt hơn.
- Tính hiệu quả
- Các phần mềm xây dựng trở nên dễ hiểu hơn
- Hiệu quả sử dụng của các thư viện
- Khả năng sử dụng lại mã thông qua templates
- Quản lý lỗi
- Cho phép xây dựng các phần mềm lớn hơn

3. Trình biên dịch C++

- ◆ Trên DOS hoặc Windows:
 - Borland C++ 3.1: Việc sử dụng Borland C++ 3.1 trên DOS giống như Turbo Pascal 7.0. Tất cả các thao tác mở, đóng tệp, soạn thảo chương trình, biên dịch và chạy thử chương trình giống như Turbo Pascal.
 - Visual C++: Tạo một project kiểu Win32 console application.
 - Borland C++ 5.5 Free Command-line Compiler
- ◆ Trên Linux:
 - Dùng trình biên dịch g++

II. Các phần tử cơ bản của ngôn ngữ C++

1. Bộ ký tự

2. Từ khoá

3. Các tên tự đặt

4. Các tên chuẩn

5. Dấu chấm phẩy

6. Lời chú thích

1. Bộ ký tự của ngôn ngữ C++

- ◆ Mọi ngôn ngữ lập trình đều được xây dựng trên một bộ ký tự nào đó. Các ký tự được ghép lại với nhau để tạo thành các từ. Các từ lại được liệt kết với nhau theo một quy tắc nào đó để tạo thành các câu lệnh. Một chương trình bao gồm nhiều câu lệnh diễn đạt một thuật toán để giải một bài toán nào đó.
- ◆ Bộ ký tự của ngôn ngữ C++ gồm có các ký tự sau:
 - 26 chữ cái hoa: A, B, C, ..., Z và 26 chữ cái thường: a...z
 - 10 chữ số: 0, 1, 2, ..., 9
 - Các ký hiệu toán học: + - * / =) (

1. Bộ ký tự của ngôn ngữ C++

- Ký tự gạch nối _
- Các dấu chấm câu và các ký tự đặc biệt khác: . , ; : [] ? ! \ & | % # \$
- Dấu cách là một khoảng trống dùng để ngăn cách giữa các từ.

Chú ý: Khi viết chương trình ta không được sử dụng các ký tự không có trong tập ký tự trên.

2. Từ khoá

- ◆ Từ khoá là những từ của riêng C++. Chúng thường được sử dụng để khai báo các kiểu dữ liệu, để viết các toán tử và các câu lệnh.
- ◆ Các từ khoá của C++ gồm có:

asm	_asm	__asm	auto	break	case
cdecl	_cdecl	__cdecl	char	class	const
continue	_cs	__cs	default	delete	do
double	_ds	__ds	else	enum	_es
__es	_export	__export	extern	far	_far

2. Từ khoá

- ◆ Các từ khoá của C++ gồm có:

`__far` `_fastcall` `__fastcall` `float` `for` `friend`
`goto` `huge` `_huge` `__huge` `if` `inline`
`int` `interrupt` `_interrupt` `__interrupt` `_loadds` `__loadds`
`long` `near` `_near` `__near` `new` `operator`
`pascal` `_pascal` `__pascal` `private` `protected` `public`
`register` `return` `_saveregs` `__saveregs` `_seg` `__seg`
`short` `signed` `sizeof` `_ss` `__ss` `static`
`struct` `switch` `template` `this` `typedef` `union`
`unsigned` `virtual` `void` `volatile` `while`

3. Các tên tự đặt

- ◆ Tên dùng để xác định các đại lượng khác nhau trong chương trình như tên hằng, tên biến, tên hàm, tên con trỏ, tên cấu trúc, tên tệp, tên nhân,...
- ◆ Quy tắc đặt tên: Tên là một dãy ký tự có thể là chữ cái, chữ số hoặc dấu gạch nối song ký tự đầu tiên phải là chữ cái hoặc dấu gạch nối. Tên không được đặt trùng với từ khoá.
- ◆ Một số ví dụ về tên đặt sai:

3XYZ_7 R#3

F(x) case

Al pha

4. Tên chuẩn

- ♦ Tên chuẩn là các tên đã được đặt trình biên dịch đặt. Tên chuẩn có thể là tên hằng, tên các hàm.

Ghi nhớ: + Các từ khoá, tên tự đặt, tên chuẩn phân biệt chữ hoa chữ thường, nghĩa là viết hoa, viết thường là khác nhau.

Ví dụ: Tên AB khác với tên ab

- + Riêng từ khoá, tên chuẩn luôn luôn dùng chữ thường.

5. Dấu chấm phẩy

- ◆ Dấu chấm được dùng để ngăn cách giữa các câu lệnh. Dấu chấm phẩy thường đặt ở cuối câu lệnh và không thể thiếu được.

Ví dụ:

```
float x;
```

```
x = 10.5;
```

```
x = 2*x - 2.5;
```


IV. Cấu trúc chung của một chương trình C++

```
//Khai báo sử dụng thư viện chương trình con, thư viện lớp
#include<iostream>
#include<stdio.h>
using namespace std;
.....
//Mô tả lớp đối tượng
.....
//Khai báo các hàm (chương trình con)
//== =chương trình chính= = =
int main()
{
    //Khai báo các biến, hằng, kiểu dữ liệu, đối tượng
    .....
    //Các lệnh của chương trình
    .....
    cout<<endl;
    return 0;
}
//== =Định nghĩa các hàm== = =
.....
```

Tương đương với
USES trong PASCAL

Tương đương với
BEGIN trong PASCAL

Thân chương trình
chính

Tương đương với
END trong PASCAL

IV-Các bước lập trình

B1: Soạn thảo chương trình

- Sử dụng một trình soạn thảo văn bản dạng text (ASCII), để soạn chương trình, ghi thành tệp .cpp
- Trên DOS/Windows: Notepad++
- Trên Linux: vim

B2: Biên dịch chương trình

- Sử dụng trình biên dịch C++ để dịch chương trình C++ sang ngôn ngữ máy, tạo ra tệp .exe
- Trên DOS/Windows: bcc32
- Trên Linux: g++

B3: Chạy thử chương trình

- Từ giao diện hệ điều hành cho chạy chương trình, nhập vào dữ liệu mẫu và kiểm tra kết quả.

Kết nối máy chủ Linux

- 1) Sử dụng chương trình kết nối: PuTTY
Android: JuiceSSH
- 2) Địa chỉ máy chủ Linux: `dse.vnua.edu.vn`
- 3) Tài khoản:
 - Username: mã sv
 - Password: ngày sinh (dd/mm/yy)
- 4) Các bước kết nối máy chủ Linux

Kết nối máy chủ Linux

4) Các bước kết nối máy chủ Linux

B1: Chạy PuTTY

B2: Tạo kết nối và ghi lại với một tên nào đó

B3: Kết nối: kích đúp vào tên đã ghi để kết nối tới máy chủ Linux => Lần đầu tiên kích Yes => Xuất hiện màn hình đăng nhập:

Login as: Mã SV

Password: Ngày sinh

B4: Đổi mật khẩu, gõ lệnh passwd ↵

Một số lệnh Linux

- 1) Xem thư mục tài khoản: `ls` ↵
- 2) Đổi tên tệp: `mv TenHienTai TenMoi` ↵
- 3) Xóa tệp: `rm TenTep` ↵
- 4) Soạn thảo chương trình: `vim tentep.cpp` ↵

Ghi tệp: gõ `:w` rồi ấn Enter

Thoát khỏi vim, ấn `:q` và Enter

- 5) Biên dịch chương trình: `g++ tentep.cpp` ↵
- 6) Chạy thử chương trình: `a.out` ↵

Một số lệnh Linux

7) Xem nội dung tệp bài tập: `cat`

hoặc `less`

Để thoát khỏi lệnh `less`, gõ `q`

8) Thoát khỏi máy chủ Linux: gõ lệnh `exit` hoặc `logout`