

Sử dụng những kiểu dữ liệu cơ sở trong chương trình

Phần c: Hàm và nhập xuất

Nhập môn lập trình

Trình bày: Nguyễn Sơn Hoàng Quốc

Email: nshquoc@fit.hcmus.edu.vn

Nội dung

- Các hàm thông dụng trong thư viện
- Hàm nhập xuất
- Các vấn đề tìm hiểu mở rộng kiến thức
- Thuật ngữ và bài đọc thêm tiếng Anh

CÁC HÀM THÔNG DỤNG CÓ SẴN TRONG THƯ VIỆN

cuu duong than cong . com

Hàm và thư viện hàm

- Khái niệm
 - Hàm là đoạn chương trình được **xây dựng sẵn** → hỗ trợ người lập trình sử dụng
 - Ví dụ: Hàm tính căn số, lũy thừa, trị tuyệt đối, logarit, ...
 - Tập hợp các hàm được xây dựng sẵn của NNLT thường được gọi là **thư viện hàm**.
 - Hệ thống thư viện hàm rất đa dạng → cần phải tra cứu thêm **tài liệu** hoặc **hệ thống giúp đỡ** của phần mềm hỗ trợ lập trình.

Ví dụ tính $F(x, y) = x + \sqrt{1 + y^2}$

```
#include <math.h>
#include <stdio.h>

void main()
{
    double x, y, Fxy;
    printf("x = ");
    scanf("%lf", &x);
    printf("y = ");
    scanf("%lf", &y);
    Fxy = x + sqrt(1 + y*y);
    printf("F(x, y) = %lf", Fxy);
}
```

```
#include <cmath>
#include <iostream>
using namespace std;
void main()
{
    double x, y, Fxy;
    cout << "x = ";
    cin >> x;
    cout << "y = ";
    cin >> y;
    Fxy = x + sqrt(1 + y*y);
    cout << "F(x, y) = ", Fxy);
}
```

Các hàm toán học

- Các hàm toán học đa số có tham số kiểu **double**, giá trị nhập vào và kết quả tính toán đều có kiểu **double**.
- Để sử dụng các hàm toán học, người lập trình cần ghi thêm vào đầu chương trình chỉ thị:
 - **#include** <math.h> đối với NNLT C
 - **#include** <cmath> đối với NNLT C++ chuẩn

Các hàm toán học

- Một số hàm toán học thông thường

Nguyên mẫu hàm	Công dụng
<code>double sqrt(double x);</code>	Tính \sqrt{x}
<code>double pow(double x, double y);</code>	Tính x^y ($x > 0$)
<code>double exp(double x);</code>	Tính e^x ($e \approx 2,71828$)
<code>double log(double x);</code>	Tính $\ln(x)$
<code>double log10(double x);</code>	Tính $\log_{10}(x)$
<code>int abs(int x);</code> <code>long labs(long x);</code> <code>double fabs(double x);</code>	Tính $ x $ (x kiểu <code>int</code>) Tính $ x $ (x kiểu <code>long</code>) Tính $ x $ (x kiểu <code>double</code>)

Các hàm toán học

- Một số hàm toán học thông thường

Nguyên mẫu hàm	Công dụng
<code>double cos(double x);</code> <code>double sin(double x);</code> <code>double tan(double x);</code>	Tính $\cos(x)$, $\sin(x)$, $\tan(x)$ (x tính theo radian, 1 radian bằng $\frac{180}{\pi}$ độ)
<code>double acos(double x);</code> <code>double asin(double x);</code> <code>double atan(double x);</code>	Tính $\cos^{-1}(x)$ Tính $\sin^{-1}(x)$ Tính $\tan^{-1}(x)$
<code>double floor(double x);</code> <code>double ceil(double x);</code>	Tính $[x]$ Tính $\lceil x \rceil$

Các hàm ký tự

- Để sử dụng các hàm ký tự liệt kê trong danh sách sau bằng cách dùng thư viện nhờ chỉ thị **#include <ctype.h>**

Nguyên mẫu hàm	Công dụng
<code>bool isupper(char ch);</code> <code>bool iswupper(wchar_t ch);</code>	Kiểm tra ch có phải là ký tự hoa?
<code>char toupper(char ch);</code> <code>wchar_t towupper(wchar_t ch);</code>	Trả về ký tự hoa tương ứng với ch
<code>bool islower(char ch);</code> <code>bool iswlower(wchar_t ch);</code>	Kiểm tra ch có phải là ký tự thường?
<code>char tolower(char ch);</code> <code>wchar_t towlower(wchar_t ch);</code>	Trả về ký tự thường tương ứng với ch

HÀM NHẬP XUẤT TRONG NGÔN NGỮ C/C++

Nhập xuất trong ngôn ngữ C

- Thư viện <stdio.h>
- Nhập: Sử dụng hàm `printf`
- Xuất : Sử dụng hàm `scanf`
- Ví dụ:

- Nhập số nguyên

```
int a;  
scanf("%d", &a);
```

- Xuất số nguyên

```
int a = 2;  
printf("%d", a);
```

Định dạng dữ liệu nhập xuất

- Đối với NNLT C
 - Nhập xuất số nguyên (kiểu **char**, **int**)
 - Có dấu dạng thập phân: %d hay %i
 - Không dấu:
 - Dạng thập phân: %u
 - Dạng thập lục phân: %x hay %X
 - Dạng bát phân: %o
 - Trường hợp nhập xuất số nguyên kiểu khác:
 - **short** (16-bit): %hd, %hi, %ho, %hu, %hx, %hX
 - **long**: %ld, %li, %lo, %lu, %lx, %lX

Ví dụ

- Viết chương trình nhập một số nguyên từ màn hình. Tính bình phương và xuất kết quả

```
1. #include <stdio.h>
```

```
2. void main()
```

```
3. {
```

```
4.     int x, binhPhuong;
```

```
5.     printf("Nhap x = ");
```

```
6.     scanf("%d", &x);
```

```
7.     binhPhuong = x * x;
```

```
8.     printf("Binh phuong = %d", binhPhuong);
```

```
9. }
```

Định dạng dữ liệu nhập xuất (Đối với NNLT C)

- Nhập xuất số thực chấm động (kiểu **float**)
 - Dạng viết thập phân: %f
 - Dạng viết số mũ
 - chữ e hay E thay cho cơ số 10
 - ví dụ: 1.2E-8: %e hay %E
- Trường hợp nhập xuất thực kiểu khác:
 - **double**: %lf, %le, %lE
 - **long double**: %Lf, %Le, %LE

Ví dụ minh họa

- Viết chương trình nhập vào độ dài hai cạnh của hình chữ nhật. Tính chu vi và xuất kết quả.

cuu duong than cong . com

cuu duong than cong . com

Định dạng dữ liệu nhập xuất (ngôn ngữ C)

- Ký tự đặc biệt: `\\` (dấu `\`) và `%%` (dấu `%`)
- Ký tự tab và ký tự xuống dòng: `\t`, `\n`
- Nhập xuất ký tự: `%c`
- Nhập xuất chuỗi ký tự: `%s`
- Xuất số thực theo độ chính xác:
 - Theo dạng `wid.pre`, sau `%` trước `f`
 - Ví dụ: `%9.2f`
 - Nghĩa là độ rộng ít nhất 9 ký tự (thêm khoảng trống vào nếu thiếu)
 - Nhiều nhất là 2 ký tự cho phần lẻ sau dấu chấm thập phân.

Định dạng dữ liệu nhập xuất (NNLT C++)

- Việc nhập xuất được thực hiện bởi các đối tượng đã được định nghĩa sẵn trong `<iostream>`:
 - `cin` kèm với toán tử `>>` (được gọi là extraction operator) để nhập dữ liệu.
 - `cout` kèm với toán tử `<<` (được gọi là insertion operator) để xuất dữ liệu.
- Được cung cấp hệ thống định dạng dữ liệu nhập xuất cho thiết bị nhập chuẩn và mở rộng cho các thiết bị nhập xuất khác như tập tin.

cuu duong than cong . com

Định dạng dữ liệu nhập xuất (NNLT C++)

- Thêm chỉ thị sau vào đầu chương trình:
`#include <iomanip>`
- Việc định dạng dữ liệu được thực hiện bằng các toán tử định dạng (manipulator).
 - **endl**: xuống dòng mới.
 - **setw(n)**: định độ rộng của dữ liệu xuất.
 - **left** và **right**: dùng chung với **setw(n)** để canh lề trái hay lề phải.
 - **setfill(ch)**: dùng chung với **setw(n)** để qui định ký tự ch được thêm vào thay vì dùng khoảng trắng.
 - **dec**, **oct**, **hex**: được dùng để qui định số nguyên (khi nhập xuất) được ghi theo dạng thập phân, bát phân, thập lục phân.
 - **setprecision(n)**: dùng để qui định độ chính xác khi in số thực.

Ví dụ

```
1. #include <iostream>
2. #include <iomanip>
3. using namespace std;
4. void main()
5. {
6.     int area=970, height=10, volume=9700;
7.     cout << setw(8) << "Area" << setw(10) << area << endl;
8.     cout << setw(8) << "H" << setw(10) << height << endl;
9.     cout << setw(8) << "Volume" << setw(10) << volume << endl;
10.}
```

Kết quả chạy chương trình

Area	970
H	10
Volume	9700

Ví dụ

```
1. #include <iostream>
2. #include <iomanip>
3. using namespace std;
4. void main()
5. {
6.     long n;
7.     cout << "n (hexadecimal) = ";
8.     cin >> hex >> n;
9.     cout << "Octal representation: " << oct << n << endl;
10.    cin >> n;
11.}
```

CÁC VẤN ĐỀ TÌM HIỂU MỞ RỘNG KIẾN THỨC NGHỀ NGHIỆP

cuu duong than cong . com

Đọc thêm

- Lịch sử phát triển dữ liệu cơ sở theo NNLT
- Chuẩn lưu trữ vật lý của các loại dữ liệu cơ sở
- Lỗi hỏng bảo mật trong mã nguồn
- Sự khác biệt, tương đồng giữa các NNLT

cuu duong than cong . com

THUẬT NGỮ VÀ BÀI ĐỌC THÊM TIẾNG ANH

cuu duong than cong . com

Thuật ngữ tiếng Anh

- **ASCII code**: mã ký tự theo chuẩn 1 byte. Bảng mã ASCII (American Standard Code for Information Interchange) có 256 ký tự (gồm cả ký tự thông thường và ký tự đặc biệt)
- **character**: ký tự nói chung
 - **wide character**: ký tự 16 bit
 - **wide string**: chuỗi ký tự gồm các ký tự 16 bit
- **constant**: hằng số
- **data type**: kiểu dữ liệu
- **floating-point, real data type**: số thực dấu chấm động, kiểu dữ liệu số thực
- **function library**: thư viện hàm
- **fundamental data type**: kiểu dữ liệu cơ bản, cơ sở
- **input**: nhập
 - **input data**: dữ liệu nhập

Thuật ngữ tiếng Anh

- **integral data type, integer**: kiểu dữ liệu nguyên
 - **long integer**: kiểu nguyên dài (32 bit)
- **operator**: toán tử, phép toán
 - **bit mask**: mặt nạ bit
 - **bit operator**: phép toán trên bit
 - **logical operator, boolean operator**: phép toán luận lý
- **ouput**: xuất
 - **ouput data**: dữ liệu xuất
- **overflow**: tràn số
- **unicode**: nói chung về ký tự unicode
- **variable**: biến (dùng trong lập trình)
 - **variable declaration**: khai báo biến

Bài đọc thêm tiếng Anh

- **Thinking in C**, Bruce Eckel, E-book, 2006.
- **Theory and Problems of Fundamentals of Computing with C++**, John R. Hubbard, Schaum's Outlines Series, McGraw-Hill, 1998.

cuu duong than cong . com