

TIN HỌC ĐẠI CƯƠNG

Bài 3: Hàm và nhập/xuất dữ liệu

Nội dung chính

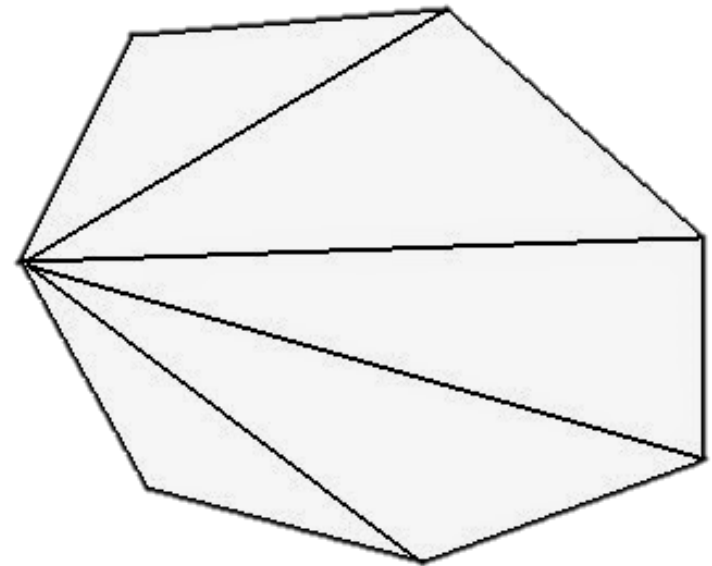
1. Phân rã bài toán
2. Hàm (function)
3. Một số kiểu dữ liệu thường dùng
4. Nhập và xuất dữ liệu
5. Bài tập

Phần 1

Phân rã bài toán

Phân rã bài toán

- Ý tưởng:
 - Một bài toán lớn có thể phân rã thành các bài toán nhỏ hơn (các thuật toán con)
 - Việc giải bài toán lớn = phối hợp lời giải các bài toán con với nhau
- Ví dụ: tính diện tích một đa giác lồi
 - Chia đa giác thành các tam giác con
 - Tính diện tích từng tam giác con
 - Lấy tổng diện tích các tam giác con



Phân rã bài toán (vấn đề)

- Giải phương trình bậc 2:
 - Chia delta thành 3 trường hợp (âm, 0, dương)
 - Giải riêng rẽ từng trường hợp một
- Hầu hết các bài toán phức tạp đều được chia thành các chức năng con
 - Hệ thống menu của các phần mềm là một ví dụ điển hình của việc chia phần mềm thành các chức năng con
- Phân rã bài toán là một trong những nguyên tắc rất quan trọng của lập trình, đối với ngôn ngữ C++, chúng ta chia chương trình thành các chương trình con (hàm)
- Những bài toán không phân rã được thường là những bài rất khó

Phần 2

Hàm (function)

Chương trình C/C++ gồm nhiều hàm

```
1  #include <iostream>
2  using namespace std;
3
4  double KhoangCach(double x1, double y1, double x2, double y2) {
5      return sqrt((x1 - x2) * (x1 - x2) + (y1 - y2) * (y1 - y2));
6  }
7
8  int main() {
9      double x1, y1, x2, y2;
10
11     cout << "X1 = "; cin >> x1;      // nhập x1
12     cout << "Y1 = "; cin >> y1;      // nhập y1
13     cout << "X2 = "; cin >> x2;      // nhập x2
14     cout << "Y2 = "; cin >> y2;      // nhập y2
15
16     cout << "KC = " << KhoangCach(x1, y1, x2, y2);
17
18     return 0;
19 }
```

Hàm (function)

- Hàm:

- Vay mượn khái niệm hàm của toán học (thường trả về kết quả)
- Là chương trình con thực thi một thuật toán nào đó
- Thực chất là một đoạn chương trình được đặt tên

- Cú pháp:

```
<kiểu kết quả> <tên hàm>(<các tham số>) {  
    // nội dung thực hiện thuật toán  
}
```

- Ví dụ:

```
int dientich(int dai, int rong) {  
    return dai * rong;  
}
```


Hàm (function)

- Gọi hàm: gọi thông qua tên và tham số

```
int n = dientich(30,40);
```

- Viết thành hàm thì có lợi gì?

- Hiện thực hóa ý tưởng phân rã bài toán thành các bài toán con
- Viết một lần, gọi mọi nơi
- Nếu có sai hoặc chưa tối ưu thì chỉ cần chỉ sửa ở một chỗ
- Có thể lại dùng trong các bài khác
- Bản thân các hàm toán học được sử dụng trong các bài trước là hàm do các lập trình viên khác viết ra, chúng ta chỉ sử dụng lại

Kinh nghiệm: hàm không nên dài quá 20 dòng (cả hàm nên nằm hoàn toàn trong một trang màn hình)

Phần 3

Một số kiểu dữ liệu thường dùng

Một số kiểu dữ liệu thường dùng

- Đã sử dụng trong bài trước: **nguyên** (int), **thực** (double, float), **logic** (bool)
- Dùng khi khai báo biến, hàm số,...
`double abc; // biến abc kiểu số thực`
`bool xyz; // biến xyz kiểu logic`
- **Hỏi**: tại sao cần định nghĩa “kiểu dữ liệu”?
- **Đáp**: nếu không biết kiểu của biến thì các phép toán thực hiện có thể không chính xác
 - Phép chia lấy dư chỉ thực hiện được với số nguyên
 - Kiểu logic không thể cộng với nhau

Kiểu nguyên (int)

- Dùng để lưu trữ số nguyên không quá lớn (trong khoảng từ ~ âm 2 tỉ đến dương 2 tỉ)
- Một biến kiểu int chiếm 4 byte (32 bit) trong bộ nhớ
- Nếu muốn thao tác các số lớn hơn, dùng kiểu “long long”
 - Biến kiểu “long long” dài 8 byte (64 bit)
 - Có thể chứa số lớn đến khoảng 9 tỉ tỉ
- Các phép tính cơ bản với số nguyên:
 - Các phép toán số học: cộng (+), trừ (-), nhân (*), chia lấy thương (/), lấy số dư (%)
 - Các phép toán đặc biệt: tăng 1 đơn vị (++), giảm 1 đơn vị (--)
 - Các phép so sánh giá trị: bằng (==), khác (!=), lớn hơn (>), nhỏ hơn (<), lớn hơn hoặc bằng (>=), nhỏ hơn hoặc bằng (<=)

Kiểu nguyên (int)

```
int a = 11;
int b = 3;
cout << (a+b); // in ra 14
cout << (a-b); // in ra 8
cout << (a*b); // in ra 33
cout << (a/b); // in ra 3
cout << (a%b); // in ra 2
cout << ++a; // tăng a lên 1, in ra 12
cout << b++; // in ra 3, b tăng lên 4
cout << --b; // giảm b đi 1, in ra 3
cout << a--; // in ra 12, giảm a đi 1
```

Kiểu thực (float, double)

- Dùng để lưu trữ các số thực
 - Kiểu double có độ chính xác cao hơn kiểu float nhưng tốn nhiều bộ nhớ hơn
 - Biến kiểu float dài 4 byte (32 bit)
 - Biến kiểu double dài 8 byte (64 bit)
 - Làm việc trên double (thường) nhanh hơn làm việc trên float
- Các phép tính cơ bản:
 - Các phép toán số học: cộng (+), trừ (-), nhân (*), chia (/)
 - Các phép so sánh giá trị
 - Nhiều hàm toán học bổ sung (khai báo thư viện <cmath>): fabs, sqrt, pow, floor/ceil, exp, log, log10,...

Kiểu logic (bool)

- Lưu trữ các giá trị đúng/sai (true/false)
- Sử dụng trong các tình huống luận lý:
 - Là kết quả của các phép so sánh: `>`, `>=`, `<`, `<=`, `==`, `!=`
 - Các phép toán logic: và (`&&`), hoặc (`||`), đảo (`!`), xor (`^`)
 - Sử dụng khi ra quyết định (sẽ học ở chương 4)

Phần 4

Nhập và xuất dữ liệu

Xuất dữ liệu

- Xuất dữ liệu thông qua biến `cout` (thường sẽ ghi ra màn hình)
 - Sau khi in xong, con trỏ sẽ dừng lại ở ngay sau phần vừa in để tiếp tục chờ lệnh in mới
 - Có thể thực hiện liên tục nhiều lệnh xuất dữ liệu liên tiếp
 - Việc trình bày ra màn hình đôi khi khá quan trọng
- Ví dụ:

```
cout << "hello!!!"; // in chuỗi "hello!!!"
cout << abc; // in ra giá trị của abc
cout << 5+6; // tính giá trị 5+6 và in ra
cout << "A = " << a; // in "A = ", sau đó in giá trị a
cout << endl; // chuyển con trỏ xuống dòng mới
cout << endl << endl; // chuyển con trỏ xuống 2 dòng
```

Nhập dữ liệu

- Nhập dữ liệu thông qua biến **cin** (thường sẽ nhập từ bàn phím)
 - Máy tính đợi người dùng bấm <enter> sau đó sẽ phân tích xem người dùng nhập gì
 - Ngoại trừ một số tình huống đặc biệt, nhập dữ liệu luôn ghi vào một biến nào đó, nếu người dùng nhập không đúng định dạng số liệu có thể gây lỗi
 - Rất cẩn thận khi nhập nhiều dữ liệu cùng một lúc
- Ví dụ:

```
cin >> a;           // nhập dữ liệu vào biến a
cin >> a >> b;      // nhập dữ liệu vào a và tiếp vào b
```

Phần 5

Bài tập

Bài tập (1/2)

1. Nhập 2 số thực a và b . Tính a^b và in ra màn hình
 - Gợi ý: sử dụng hàm `pow` trong thư viện `cmath`
2. Tương tự như bài trên, nhưng tính $\sqrt[b]{a}$
3. Nhập số nguyên dương n . Hãy tính và in ra màn hình giá trị số $P = 1 + 2 + \dots + n$
4. Nhập tọa độ 2 điểm trên mặt phẳng $M(x_1, y_1)$ và $N(x_2, y_2)$, tính và in ra màn hình độ dài đoạn MN , các giá trị tọa độ đều là số nguyên
5. Tính và in ra màn hình diện tích tam giác ABC biết vị trí của 3 đỉnh trên mặt phẳng $A(x_1, y_1)$, $B(x_2, y_2)$ và $C(x_3, y_3)$, các giá trị tọa độ đều là số thực

Bài tập (2/2)

6. Nhập một số nguyên a , hãy tính số X là tích số của chữ số đầu và chữ số cuối của a
- Ví dụ: $a = 24681357$, $X = 14$ (2×7)
7. Cho $f(x) = a_0 + a_1x + a_2x^2 + a_3x^3 + a_4x^4$. Nhập vào từ bàn phím các giá trị x , a_0 , a_1 , a_2 , a_3 và a_4 là các số thực, sau đó tính và in ra $f(x)$ tương ứng
8. Nhập số thực X , hãy tính và in ra màn hình N là số chính phương nhỏ nhất lớn hơn X
- Ví dụ:
- | | |
|-----------|-----------|
| $X = 1.5$ | $N = 4$ |
| $X = -3$ | $N = 0$ |
| $X = 99$ | $N = 100$ |
| $X = 64$ | $N = 81$ |