

BÀI GIẢNG CẤU TRÚC DỮ LIỆU - GIẢI THUẬT CHƯƠNG 1- MỞ ĐẦU

TS. NGUYỄN ĐÌNH DƯƠNG
BỘ MÔN KHMT - KHOA CNTT

Email: duongnd@hcmut.edu.vn

Ngày 15/02/2021

Nội dung

Giới thiệu tổng quan

- 1.1 Một số khái niệm cơ bản
- 1.2 Ngôn ngữ lập trình C++
- 1.3 Thư viện STL C++
- 1.4 Bài tập

Trao đổi

Nội dung

Giới thiệu tổng quan

- 1.1 Một số khái niệm cơ bản
- 1.2 Ngôn ngữ lập trình C++
- 1.3 Thư viện STL C++
- 1.4 Bài tập

Trao đổi

1. Giới thiệu tổng quan

1. 1. Một số khái niệm cơ bản

Đặt vấn đề

- Chương trình máy tính là một dãy các câu lệnh để xử lý thông tin và đem lại kết quả mong muốn cho người sử dụng. Tuy nhiên, các thông tin lấy từ thực tế thường là các thông tin trừu tượng và không thể biểu diễn một cách trực tiếp trong máy tính.
- Câu hỏi: Vậy thông tin lấy từ thực tế, được biểu diễn như thế nào trong máy tính?



1. Giới thiệu tổng quan

1. 1. Một số khái niệm cơ bản

- **Cấu trúc dữ liệu:** cách thức tổ chức sắp xếp dữ liệu để biểu diễn thông tin trên máy tính: Danh sách (list), Ngăn xếp (stack), Hàng đợi (queue), Cây (tree), Đồ thị (graph), ...

KDLTT (kiểu dữ liệu trừu tượng): là kiểu dữ liệu phức hợp bao gồm

- các đối tượng và
- các phép toán trên các đối tượng

class trong C++:

- data members,
- member functions



1. Giới thiệu tổng quan

1. 1. Một số khái niệm cơ bản

- **Giải thuật** (thuật toán): một thủ tục xác định bao gồm một dãy hữu hạn các bước cần thực hiện để thu được đầu ra cho một đầu vào cho trước của bài toán.

1. Giới thiệu tổng quan

1. 1. Một số khái niệm cơ bản

- **Giải thuật** (thuật toán): một thủ tục xác định bao gồm một dãy hữu hạn các bước cần thực hiện để **thu được đầu ra** cho **một đầu vào** **cho trước** của bài toán.
- **Đặc trưng của thuật toán:**
 - Đầu vào (Input): Thuật toán nhận dữ liệu vào từ một tập nào đó.
 - Đầu ra (Output): Với mỗi tập các dữ liệu đầu vào, thuật toán đưa ra các dữ liệu tương ứng với lời giải của bài toán.
 - Chính xác (Precision): Các bước của thuật toán được mô tả chính xác.
 - Hữu hạn (Finiteness): Thuật toán cần phải đưa được đầu ra sau một số hữu hạn (có thể rất lớn) bước với mọi đầu vào.
 - Đơn trị (Uniqueness): Các kết quả trung gian của từng bước thực hiện thuật toán được xác định một cách đơn trị và chỉ phụ thuộc vào đầu vào và các kết quả của các bước trước.
 - Tổng quát (Generality): Thuật toán có thể áp dụng để giải mọi bài toán có dạng đã cho.
- Cấu trúc dữ liệu và giải thuật là hai thuật ngữ luôn đi kèm với nhau. Cấu trúc dữ liệu là đầu vào và cũng là đầu ra của giải thuật

1. Giới thiệu tổng quan

1. 1. Một số khái niệm cơ bản

- Về cấu trúc dữ liệu ta quan tâm đến các vấn đề cơ bản sau:
 - Cách cài đặt cấu trúc dữ liệu đó;
 - Cách thực hiện các thao tác cơ bản với cấu trúc dữ liệu đó: Tạo mới; Thêm 1 phần tử; Xóa 1 phần tử; Tim 1 phần tử ...
- Về giải thuật ta quan tâm đến các vấn đề cơ bản sau:
 - Tư tưởng của giải thuật;
 - Nội dung của thuật toán và cách cài đặt thuật toán đó;
 - Đánh giá độ phức tạp về thời gian và không gian.

1. Giới thiệu tổng quan

1. 1. Một số khái niệm cơ bản

Biểu diễn thuật toán

- Ngôn ngữ tự nhiên
- Lưu đồ
- Giả mã
- Ngôn ngữ lập trình

1. Giới thiệu tổng quan

1. 1. Một số khái niệm cơ bản

Biểu diễn thuật toán - Ngôn ngữ tự nhiên

Ví dụ 1.1 (Tìm ước chung lớn nhất (UCLN))

- ① (Input) Nhập a và b : Số tự nhiên
- ② Nếu $b \neq 0$ thì chuyển sang bước 3, nếu không thì bỏ qua bước 3, đi làm bước 4
- ③ Đặt $r = a \% b$; Đặt $a = b$; Đặt $b = r$; Quay trở lại bước 2
- ④ (Output) Kết luận ước số chung lớn nhất phải tìm là giá trị của a .
Kết thúc thuật toán.

1. Giới thiệu tổng quan

1. 1. Một số khái niệm cơ bản

Biểu diễn thuật toán - Ngôn ngữ tự nhiên

Ví dụ 1.1 (Tìm ước chung lớn nhất (UCLN))

- ① (Input) Nhập a và b : Số tự nhiên
- ② Nếu $b \neq 0$ thì chuyển sang bước 3, nếu không thì bỏ qua bước 3, đi làm bước 4
- ③ Đặt $r = a \% b$; Đặt $a = b$; Đặt $b = r$; Quay trở lại bước 2
- ④ (Output) Kết luận ước số chung lớn nhất phải tìm là giá trị của a .
Kết thúc thuật toán.

- **Ưu điểm:** Đơn giản, không cần kiến thức về về cách biểu diễn (mã giả, lưu đồ,...)
- **Nhược điểm:**
 - Dài dòng, không cấu trúc
 - Đôi lúc khó hiểu, không diễn đạt được thuật toán.

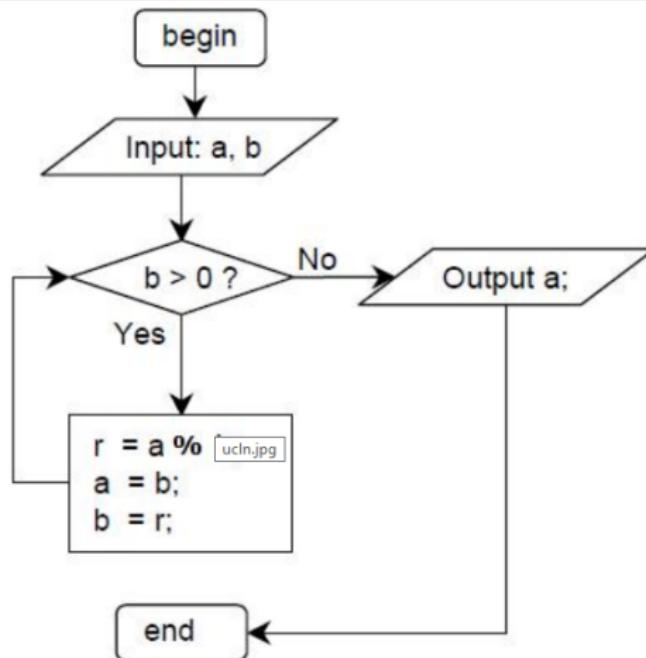


1. Giới thiệu tổng quan

1. 1. Một số khái niệm cơ bản

Biểu diễn thuật toán - Sơ đồ khối

Ví dụ 1.2 (Tìm ước chung lớn nhất (UCLN))



1. Giới thiệu tổng quan

Biểu diễn thuật toán - Giả mã

- Mô tả bậc cao của một thuật toán
- Cấu trúc rõ ràng hơn ngôn ngữ tự nhiên
- Không chi tiết như mã nguồn
- Được ưa thích trong biểu diễn giải thuật
- Ẩn đi các khía cạnh thiết kế chương trình

1. 1. Một số khái niệm cơ bản

Algorithm *arrayMax(A, n)*

Input array *A* of *n* integers

Output maximum element of *A*

currentMax $\leftarrow A[0]$

for *i* $\leftarrow 1$ to *n* – 1 do

 if *A*[*i*] > *currentMax* then

currentMax $\leftarrow A[i]$

return *currentMax*

1. Giới thiệu tổng quan

Biểu diễn thuật toán - Giả mã

- Luồng điều khiển
 - if...then...[else...]
 - while...do ...
 - repeat...until...
 - for...do...
 - Lùi đầu dòng thay thế các dấu ngoặc
- Khai báo phương thức
Algorithm method (arg [, arg...])

1. 1. Một số khái niệm cơ bản

Input...

Output...

- Gọi phương thức: var.method (arg [, arg...])
- Trả về giá trị: return biến_thức
- Các biểu thức
 - ←: Gán
 - =: so sánh bằng

1. Giới thiệu tổng quan

1. 1. Một số khái niệm cơ bản

Chương trình là một thể hiện của Giải thuật trong một ngôn ngữ lập trình nào đó

1. Giới thiệu tổng quan

1. 2. Ngôn ngữ lập trình C++

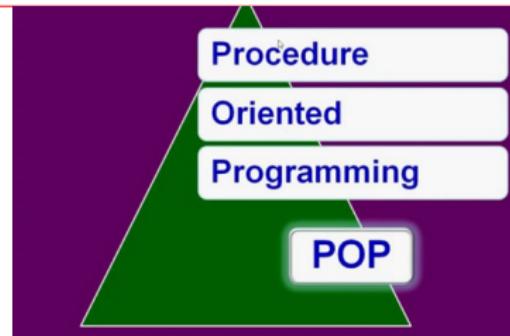
- Ngôn ngữ lập trình C++ là ngôn ngữ được phát triển dựa trên ngôn ngữ lập trình C.
- Về cơ bản, cú pháp của C++ giống với cú pháp của C. Tuy nhiên nó có một số mở rộng sau đây:
 - Nhập, xuất dữ liệu (cout, cin)
 - Hàm có đối mặc định, hàm có đối tham chiếu
 - Nạp chồng hàm (hay tải bội hàm – overload function)
 - Hàm mẫu
 - Lớp → xây dựng các chương trình HĐT
- Tốc độ thực thi nhanh.
- Có nhiều sách, tài liệu tham khảo trên internet về cấu trúc dữ liệu và giải thuật được viết bằng C/C++.

1. Giới thiệu tổng quan

1. 2. Ngôn ngữ lập trình C++

Lập trình hướng thủ tục (POP)

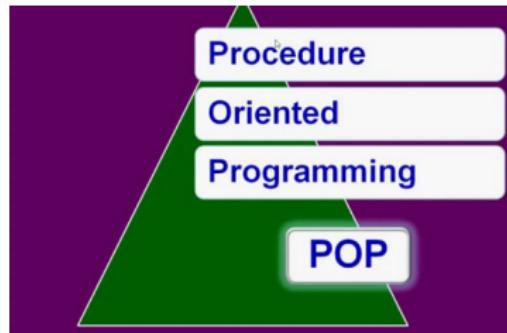
- Lập trình hướng thủ tục (POP) là kỹ thuật lập trình chia nhỏ một chương trình lớn thành các chương trình con (còn được gọi là các hàm).
- Mỗi hàm sẽ đảm nhiệm một chức năng khác nhau trong hệ thống.
- Mục đích của việc này là để đơn giản hóa cấu trúc của chương trình, thuận tiện cho việc kiểm tra, sửa đổi và thực thi một cách hiệu quả.
- Các ngôn ngữ lập trình hướng thủ tục: C, Pascal



1. Giới thiệu tổng quan

Lập trình hướng thủ tục (POP)

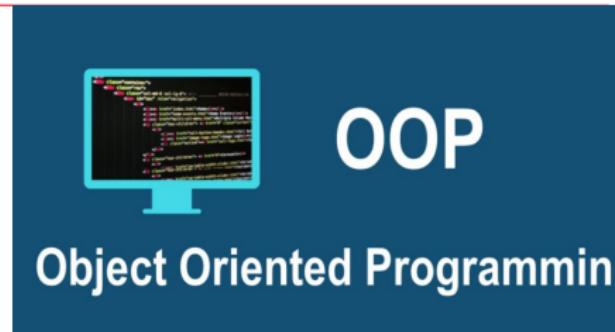
- Chỉ tập trung vào việc phát triển các hàm, ít chú trọng đến dữ liệu
- Dữ liệu của hệ thống di chuyển từ hàm này qua hàm khác, được dùng chung giữa các hàm.
- Tuân theo hình thức tiếp cận top-down khi thiết kế chương trình
- Dùng con trỏ hoặc biến toàn cục để liên kết các hàm với nhau



1. Giới thiệu tổng quan

Lập trình hướng đối tượng (OOP)

- Lập trình hướng đối tượng (OOP) là kỹ thuật lập trình dựa trên “công nghệ đối tượng”, tạo ra các đối tượng trong code để trừu tượng hóa các đối tượng thực tế trong thế giới thực.
- Đối tượng trong OOP có thuộc tính (attribute) và phương thức (method).
- Các đối tượng có thể tương tác qua lại lẫn nhau.
- Các ngôn ngữ lập trình hướng đối tượng: C++, Java, Python.

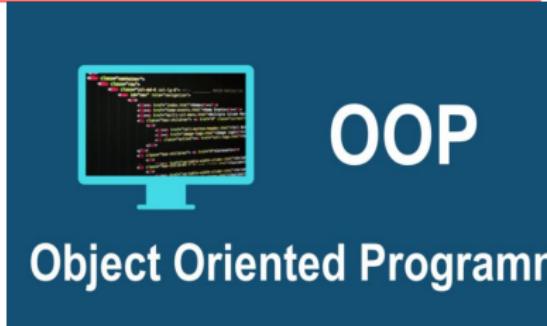


1. Giới thiệu tổng quan

1. 2. Ngôn ngữ lập trình C++

Tính chất lập trình hướng đối tượng (OOP)

- **Encapsulation** - tính đóng gói: các phương thức và dữ liệu có mối quan hệ với nhau được lưu vào một lớp để thuận tiện cho việc quản lý, sử dụng. Chỉ có phương thức nội tại của chính đối tượng mới có thể thay đổi trạng thái nội tại của nó.



- **Abstraction** - tính trừu tượng: chỉ tập trung vào những thuộc tính và phương thức cần thiết cho việc giải quyết vấn đề trong lập trình và bỏ qua các thông tin không quan trọng.
- **Inheritance** - tính kế thừa: các đối tượng “con” có thể thừa hưởng các đặc tính có sẵn từ đối tượng “cha” mà không cần định nghĩa lại (tùy theo ngôn ngữ lập trình).
- **Polymorphism** - tính đa hình: các đối tượng không cùng một lớp, khi tiếp nhận cùng một thông điệp thì sẽ phản hồi theo những cách khác nhau.

1. Giới thiệu tổng quan

1. 2. Ngôn ngữ lập trình C++

So sánh giữa POP và OOP

- *Trọng tâm:* OOP chú ý vào dữ liệu hơn là thuật toán. POP chú ý đến việc xây dựng các hàm và thuật toán hơn là dữ liệu.
- *Sự phân chia:* OOP chia nhỏ chương trình thành các đối tượng. POP chia nhỏ chương trình thành các hàm con.
- *Chế độ truy cập:* Các từ khóa phạm vi truy cập trong OOP được chia thành Public, Private, Protected và Default. POP không có thành phần này.
- *Hướng tiếp cận khi thiết kế chương trình:* OOP tiếp cận từ dưới lên. POP tiếp cận từ trên xuống.
- *Quá trình thực thi:* OOP cho phép các chức năng chạy cùng một lúc. POP cho phép các hàm và chức năng chạy lần lượt.

1. Giới thiệu tổng quan

1. 2. Ngôn ngữ lập trình C++

So sánh giữa POP và OOP

- *Truy cập dữ liệu:* OOP hạn chế truy cập dữ liệu giữa các đối tượng. POP cho phép dữ liệu tự do di chuyển trong hệ thống và các hàm có thể chia sẻ dữ liệu cho nhau.
- *Bảo mật:* OOP ẩn dữ liệu trong chế độ Public, Protected và Private nên có bảo mật cao. POP không có chế độ ẩn dữ liệu, độ an toàn thấp.
- *Thêm mới dữ liệu:* Hoạt động này có thể được thực hiện dễ dàng với các đối tượng trong OOP, còn POP thì khó hơn.
- *Nạp chồng/Duplicate:* OOP hỗ trợ nạp chồng các hàm, hàm tạo và toán tử còn POP thì không.
- *Ứng dụng:* OOP có thể áp dụng trong xây dựng các chương trình có độ phức tạp cao. POP chỉ nên được dùng với chương trình đơn giản.

1. Giới thiệu tổng quan

1. 3. Thư viện STL C++

Giới thiệu về thư viện chuẩn C++

- STL: Standard Template Library
- STL chính là một thư viện chứa những template của cấu trúc dữ liệu cũng như thuật toán được xây dựng một cách tổng quát nhất, nhằm hỗ trợ cho người dùng trong quá trình lập trình.
- Thư viện STL giúp thực hiện toàn bộ các công việc như vào ra dữ liệu, quản lý mảng động, xây dựng sẵn những cấu trúc dữ liệu cơ bản (ngăn xếp, hàng đợi, tập hợp,...) và bao gồm cả các giải thuật cơ bản như sắp xếp, tìm min - max, tính tổng, thậm chí là tìm ước chung lớn nhất,...

1. Giới thiệu tổng quan

1. 3. Thư viện STL C++

Các thành phần của thư viện STL

- **Containers Library:** Thư viện chứa các cấu trúc dữ liệu mẫu như vector, stack, queue, deque, set, map,...
- **Algorithm Library:** Chứa các thuật toán viết sẵn để thao tác với dữ liệu.
- **Iterator Library:** Là các biến lặp, sử dụng để truy cập, duyệt các phần tử dữ liệu của các containers. Về cơ bản, nó giống như các biến chạy trên dữ liệu nhưng truy cập vào địa chỉ của dữ liệu.
- **Numeric Library:** Chứa các hàm toán học.
- **Ví dụ:** thư viện `<iostream>` hay `<string>`
- **Khai báo:**

```
using namespace std;  
# include <Tên_thư_viện>
```

1. Giới thiệu tổng quan

1. 4. Bài tập

- ① Xây dựng lớp biểu diễn các điểm trong mặt phẳng với một câu tử không đối, một câu tử có đối đầy đủ, hai phương thức nhập và in tọa độ của điểm lên màn hình.
- ② Xây dựng lớp biểu diễn một đoạn thẳng (biết đoạn thẳng được xác định bởi tọa độ điểm đầu và điểm cuối). Với các câu tử không đối, có đối đầy đủ, phương thức nhập, in tọa độ hai đầu mút, tính độ dài đoạn thẳng.
- ③ Xây dựng lớp biểu diễn các thí sinh, biết mỗi thí sinh bao gồm các thông tin: Số báo danh, Họ tên, năm sinh, giới tính, điểm toán, điểm lý, điểm hóa. Lớp có các câu tử, các phương thức nhập, in, lấy tổng điểm, lấy điểm từng môn
- ④ Thiết lập lớp PhanSo để biểu diễn khái niệm phân số với hai thành phần dữ liệu tử số, mẫu số và các hàm thành phần cộng, trừ, nhân, chia hai phân số, các hàm thành phần xuất, nhập, định giá trị cho phân số. Viết chương trình cho phép nhập vào hai phân số, in ra kết quả các phép toán cộng, trừ, nhân, chia hai phân số kể trên.
- ⑤ Xây dựng lớp biểu diễn khái niệm số phức với hai thành phần dữ liệu thực, ảo và các hàm thành phần xuất, nhập, định giá trị cho số phức, cộng, trừ, nhân, chia hai số phức. Viết chương trình cho phép nhập vào hai số phức, in ra kết quả các phép toán cộng, trừ, nhân, chia hai số phức kể trên.

1. Giới thiệu tổng quan

1. 4. Bài tập

- ⑥ Xây dựng lớp biểu diễn các đối tượng dãy số với các phương thức hàm tạo, hàm in, hàm thêm một phần tử vào dãy, hàm xóa một phần tử của dãy, hàm tìm kiếm một phần tử có trong dãy không nếu có trả lại vị trí của phần tử đó trong dãy.
- ⑦ Xây dựng lớp biểu diễn các đối tượng là các sinh viên (các thuộc tính, phương thức do sv tự xác định)
- ⑧ Xây dựng lớp biểu diễn các vector trong không gian n chiều có các phương thức toán tử: +, - hai vector, tích vô hướng hai véc tơ, đổi dấu, », «.
- ⑨ Xây dựng lớp biểu diễn các đa thức với các phương thức toán tử: +, -, * hai đa thức, tính giá trị đa thức, », «.
- ⑩ Xây dựng lớp biểu diễn các ma trận có các phương thức toán tử: +, - ,* hai ma trận, », «.

Nội dung

Giới thiệu tổng quan

- 1.1 Một số khái niệm cơ bản
- 1.2 Ngôn ngữ lập trình C++
- 1.3 Thư viện STL C++
- 1.4 Bài tập

Trao đổi

TRAO ĐỔI