

Lập trình



Bộ môn: Điện tử & kỹ thuật máy tính

Viện: Điện tử Viễn Thông

Trường: ĐH Bách khoa Hà nội

Chương 1: Mở đầu

- 1.1 *Giới thiệu*
- 1.2 *Giới thiệu chung về kỹ thuật lập trình*
- 1.3 *Một số kỹ thuật lập trình cơ bản*
- 1.4 *Các bước phát triển chương trình*
- 1.5 *Lịch sử ngôn ngữ C/C++*

1.1 Giới thiệu

- *3 tín chỉ = 45 tiết*
- *Học trên lớp + thực hành*
- *Điểm=Điểm giữa kỳ (30%)+Điểm thi cuối kỳ(70%)*
- *Không làm bài tập về nhà: trừ điểm giữa kỳ.*
- *Thi: lập trình trên giấy.*
- *Bài tập lớn (không bắt buộc): cộng điểm giữa kỳ*
- *Mục đích môn học: trang bị cho sinh viên kiến thức cơ bản về C/C++ và cấu trúc dữ liệu.*

Nguyên tắc học tập

- *Tích cực, chủ động học thường xuyên*
- *Kết hợp giữa nghe giảng, tham khảo tài liệu và thực hành.*
- *Không nên học thuộc lòng, học chay*
- *Không nên mong đợi nhiều vào ôn tập*
- *Không nên dựa dẫm vào các bài tập mẫu trong sách*

Công cụ học tập

- *Máy tính PC*
- *Công cụ lập trình: **Visual C++ 6.0 (Visual Studio 6.0)**, Visual studio C++ .NET, Borland C++ Builder*
- *Nền ứng dụng: **Win32 Console Application***
- *Tài liệu tham khảo:*
 1. *GS Phạm Văn Ất: **Kỹ thuật lập trình C/C++***
 2. *Stanley B. Lippman, Josée Lajoie: **C++ Primer**. 3rd Edition. Addison-Wesley 1998.*
 3. *Bjarne Stroustrup: **The C++ Programming Language**. 3rd Edition. Addison-Wesley 1997.*
 4. *David Musser,...: **C++ Programming with Standard Template Library**. 2nd Edition, Addison-Wesley 1998.*
 5. *Bruce Eckel: **Thinking in C++**. www.bruceeckel.com, 2003.*

1.2 Giới thiệu về kỹ thuật lập trình

- *Kỹ thuật lập trình là gì?*
 - Là kỹ thuật thực thi một giải pháp phần mềm
 - Dựa trên nền tảng một phương pháp luận
 - Dùng một hoặc nhiều ngôn ngữ lập trình phù hợp với đặc thù của ứng dụng
- *Kỹ thuật lập trình = Kỹ thuật mã hóa?*

Kỹ thuật lập trình = Tư tưởng thiết kế + Kỹ thuật mã hóa
= Cấu trúc dữ liệu + Giải thuật + Ngôn ngữ lập trình

Thế nào là lập trình

- *Viết chương trình tính giai thừa của số 100?*
- *Viết một chương trình in ra 100 số nguyên tố đầu tiên?*

Không phải lập trình

- *Viết một hàm tính giai thừa?*
- *Viết một chương trình in ra N số nguyên tố đầu tiên?*

Lập trình

Thế nào là lập trình tốt?

- *Đúng yêu cầu của khách hàng*
- *Tin cậy*
 - *Chương trình chạy đúng*
 - *Chạy ít lỗi (số lượng lỗi ít, cường độ lỗi thấp)*
 - *Mức độ lỗi nhẹ*
- *Hiệu suất*
 - *Chương trình nhỏ gọn, sử dụng ít bộ nhớ*
 - *Tốc độ nhanh, sử dụng ít thời gian CPU*
- *Hiệu quả:*
 - *Thời gian lập trình ngắn,*
 - *Khả năng bảo trì dễ dàng*
 - *Giá trị sử dụng lại lớn*
 - *Sử dụng đơn giản, thân thiện*
 - *Nhiều chức năng tiện ích*

Làm thế nào để lập trình tốt?

- *Học cách tư duy và phương pháp lập trình*
 - *Tư duy toán học, tư duy logic, tư duy có cấu trúc, tư duy hướng đối tượng, tư duy tổng quát*
 - *Tìm hiểu về cấu trúc dữ liệu và giải thuật*
- *Hiểu sâu về máy tính*
 - *Tương tác giữa CPU, chương trình và bộ nhớ*
 - *Cơ chế quản lý bộ nhớ*
- *Nắm vững ngôn ngữ lập trình*
 - *Biết rõ các khả năng và hạn chế của ngôn ngữ*
 - *Kỹ năng lập trình (đọc thông, viết thạo)*
- *Tự rèn luyện trên máy tính*
 - *Hiểu sâu được các điểm nêu trên*
 - *Rèn luyện kỹ năng lập trình*
 - *Thúc đẩy sáng tạo*

Các nguyên tắc cơ bản

◆ *Triều tượng hóa*

- *Chắt lọc ra những yếu tố quan trọng, bỏ qua những chi tiết kém quan trọng*

◆ *Đóng gói*

- *Che giấu và bảo vệ các dữ liệu quan trọng qua một giao diện có kiểm soát*

◆ *Module hóa*

- *Chia nhỏ đối tượng vấn đề thành nhiều module nhỏ để dễ can thiệp và giải quyết*

◆ *Phân cấp*

- *Phân hạng hoặc sắp xếp trật tự đối tượng theo các quan hệ trên dưới*

1.3 Một số kỹ thuật lập trình cơ bản

- *Lập trình tuần tự*
- *Lập trình có cấu trúc*
- *Lập trình module*
- *Lập trình hướng đối tượng*
- *Lập trình tổng quát*
- ...

Lập trình tuần tự (Sequential Programming)

- *Phương pháp cổ điển nhất.*
- *Chương trình gồm một chuỗi các lệnh kế tiếp nhau.*
- *Mức trừu tượng thấp*
- *Điều khiển chương trình bằng các lệnh rẽ nhánh, lệnh nhảy, lệnh gọi chương trình con.*
- *Ví dụ ngôn ngữ đặc thù:*
 - *Ngôn ngữ máy*
 - *ASSEMBLY*
 - *BASIC*

Lập trình tuần tự: Ví dụ tính giai thừa

- 1: MOV AX, n
- 2: DEC n
- 3: CMP n, 1
- 4: JMPI
- 5: MUL AX, n
- 6: JMP 2
- 7: MOV n, AX
- 8: RET

Lập trình có cấu trúc (structured programming)

- *Cấu trúc hóa dữ liệu (xây dựng kiểu dữ liệu) và cấu trúc hóa chương trình để tránh các lệnh nhảy.*
- *Phân tích và thiết kế theo trật tự từ trên xuống và thực hiện từ dưới lên.*
- *Trong lập trình có cấu trúc chỉ sử dụng các cấu trúc điều khiển rẽ nhánh (if then else...), vòng lặp (while...) và thoát ra (exit).*
- *Ví dụ các ngôn ngữ đặc thù:*
 - *PASCAL, FORTRAN, C,...*

Lập trình có cấu trúc: Ví dụ tính giai thừa (PASCAL)

```
function gthua(n: interger) : interger
var gt: interger;
begin
    gt := n;
    while (n > 1) do
        begin
            dec(n) ;
            gt := gt * n;
        end
        gthua := gt;
    end
end;
```

Lập trình module (modular programming)

- *Lập trình module là một dạng cải tiến của lập trình có cấu trúc. Chương trình được cấu trúc nghiêm ngặt hơn, dùng đơn vị cấu trúc là module.*
- *Module:*
 - *Một đơn vị cấu trúc độc lập, được chuẩn hóa dùng để tạo lập một hệ thống.*
 - *Mỗi module bao gồm phần giao diện (mở) và phần thực hiện (che giấu)*
 - *Các module giao tiếp với nhau thông qua các giao diện được đặc tả rất chính xác.*
- *Ví dụ ngôn ngữ tiêu biểu:*
 - *Modula-2, xây dựng trên cơ sở PASCAL, do Niclaus Wirth thiết kế năm 1977.*

Lập trình hướng đối tượng (object-oriented programming)

- *Xây dựng chương trình ứng dụng dựa trên các cấu trúc dữ liệu trừu tượng (lớp), các thể nghiệm của các cấu trúc đó (đối tượng) và quan hệ giữa chúng (quan hệ lớp, quan hệ đối tượng).*
- *Ba nguyên lý cơ bản:*
 - *Đóng gói dữ liệu (data encapsulation)*
 - *Dẫn xuất / thừa kế (subtyping / inheritance)*
 - *Đa hình đa xạ (polymorphism)*
- *Ví dụ ngôn ngữ hỗ trợ tiêu biểu:*
 - *C++, C#*
 - *Java,*
 - *...*

Ví dụ minh họa: Quản lý sinh viên (C++)

```
class Date {
    int Day, Month, Year;
public:
    void setDate(int, int, int);
    ...
};

class Student {
    string name;
    Date    dob;
    int     code;
public:
    Student(string n, Date d, int c);
    ...
};

class StudentList {
    Student* list;
public:
    void addStudent(Student*);
    ...
};
```

Lập trình tổng quát (generic programming)

- Một tư duy lập trình mở, trên quan điểm tổng quát hóa tất cả những gì có thể nhằm đưa ra một **khôn mẫu giải pháp** cho nhiều bài toán lập trình cụ thể.
- Ưu điểm:
 - Giảm tối đa lượng mã nguồn
 - Tăng nhiều lần giá trị sử dụng lại của phần mềm
 - Có thể kết hợp tùy ý với các phương pháp luận khác
 - Tính khả chuyển cao
- Các hình thức tổng quát hóa:
 - Kiểu dữ liệu
 - Phép toán cơ bản
 - Cấu trúc dữ liệu
 - Quản lý bộ nhớ,...

1.4 Các bước phát triển chương trình

- *Tạo mã nguồn chương trình*
- *Biên dịch chương trình → mã đích*
- *Liên kết chương trình với các thư viện → chương trình chạy được*
- *Nạp và chạy chương trình*

Môi trường/công cụ phát triển

- *IDE (Integrated Development Environment)*
 - *Hỗ trợ toàn bộ các bước phát triển chương trình*
 - *Ví dụ: MS Visual C++, Borland C++ (Builder), Visual studio C++*
- *Các công cụ tiêu biểu*
 - *Trình soạn thảo (Editor)*
 - *Trình biên dịch (Compiler)*
 - *Trình liên kết (Linker)*
 - *Trình nạp (Loader)*
 - *Trình gỡ rối (Debugger)*
 - *Trình quản lý dự án (Project Manager)*

1.5 Lịch sử ngôn ngữ C/C++

Lịch sử ngôn ngữ C

- *Tiến hóa từ hai ngôn ngữ lập trình phi kiểu là BCPL và B.*
- *Được Dennis Ritchie (Bell Laboratories, AT&T) phát triển bằng cách kiểu hóa dữ liệu và bổ sung các yếu tố khác*
- *Ngôn ngữ phát triển hệ điều hành UNIX*
- *Có tính khả chuyển, không phụ thuộc phần cứng*
- *1989: ANSI chuẩn hóa (ANSI-C)*
- *1990: Công bố chuẩn ANSI và ISO*
 - *ANSI/ISO 9899: 1990*

Lịch sử ngôn ngữ C++

- *Được mở rộng, tiến hóa từ C*
- *Bjarne Stroustrup (Bell Laboratories)*
 - *Đầu những năm 1980: “C with classes”*
 - *1984: Tên C++*
 - *1987: “The C++ Programming Language” 1st Edition*
 - *1997: “The C++ Programming Language” 3rd Edition*
 - *Chuẩn hóa quốc tế: ANSI/ISO 1996*
- *Bổ sung các đặc tính hỗ trợ:*
 - *Lập trình hướng đối tượng*
 - *Lập trình tổng quát*
 - *Lập trình toán học,...*
 - *LAPA*