

Bài 1: Lập trình cơ sở

1 Mục đích bài thực hành

Giúp học viên làm quen với công cụ lập trình Visual C++, thử nghiệm viết chương trình theo phong cách hướng hàm, qua đó nắm vững các kiến thức lập trình cơ sở như:

- 1) Phương pháp thể hiện thuật toán bằng các cấu trúc điều kiện, vòng lặp
- 2) Cách sử dụng mảng, con trỏ, bộ nhớ động
- 3) Khai báo, thực hiện và sử dụng hàm; các phương pháp truyền tham số hàm.

2 Nhiệm vụ thí nghiệm

2.1 Bài tập

1. Xây dựng một hàm thực hiện phân loại các số chẵn và lẻ trong một dãy số nguyên, kết quả được nhớ vào hai dãy số mới. Sử dụng mảng động để biểu diễn hai dãy số kết quả sao cho bộ nhớ được cấp phát động chỉ vừa đủ dùng. Kết quả trả về của hàm sẽ là số các số chẵn trong dãy ban đầu, hay cũng chính là chiều dài của một dãy kết quả.
2. Viết chương trình minh họa cách sử dụng hàm trong câu 1 với hai trường hợp:
 - dãy số ban đầu được chứa trong một mảng tĩnh và giá trị các số trong dãy được khởi tạo ngay khi khai báo.
 - dãy số ban đầu được chứa trong một mảng động và giá trị các số cần được đưa vào qua bàn phím

Hai dãy số kết quả được đưa ra màn hình.

2.2 Nhiệm vụ chuẩn bị ở nhà

- 1) Ôn lại các kiến thức cơ sở nêu trong phần 1, xem lại các ví dụ và bài tập trên lớp.
- 2) Xây dựng thuật toán và viết chương trình sẵn trên giấy (viết tay, không chấp nhận bản sao!)

2.3 Nhiệm vụ trên phòng thí nghiệm

- 1) Làm quen với Visual C++, tạo một dự án mới (New Project), đặt tên dự án (ví dụ "SortApp") và chọn "Win32 Console Application"
- 2) Viết hàm trên máy và lưu vào một 2 files riêng: Header file (ví dụ sort.h) chứa mã khai báo hàm và file nguồn (*.cpp) chứa mã thực hiện hàm. Bổ sung 2 files đó vào dự án. Biên dịch file nguồn (ví dụ sort.cpp), tìm cách giải thích và sửa các lỗi biên dịch (*compile error*).
- 3) Viết chương trình minh họa trong file chính (ví dụ sortapp.cpp). Dịch toàn bộ dự án (Build), sửa các lỗi biên dịch (*compile error*), lỗi liên kết (*link error*) nếu có và chạy thử chương trình.
- 4) Làm quen sử dụng Debugger để tìm lỗi chạy (*runtime error*), tìm cách giải thích và sửa lỗi nếu có.
- 5) Mở menu "Project Setting" và tìm hiểu ý nghĩa các thông số lựa chọn cho dự án.
- 6) Thử nghiệm thay đổi cách thể hiện vòng lặp (for... hoặc while..., sử dụng phương pháp dịch con trỏ thay cho sử dụng chỉ số).
- 7) Thử nghiệm không dùng mảng động, mà dùng mảng tĩnh để chứa dãy số đầu vào và các dãy số kết quả trong hàm. Phân tích các vấn đề gặp phải.
- 8) Ghi chép các bước thực hiện để làm báo cáo.

3 Báo cáo thực hành

Mỗi học viên phải làm một báo cáo riêng, nộp cho cán bộ hướng dẫn thí nghiệm chậm nhất là một tuần sau buổi thực hành. Báo cáo có thể viết tay hoặc in, nhưng các bản sao photocopy sẽ không được chấp nhận. Bài báo cáo phải có đủ các phần sau:

1. Toàn bộ mã chương trình đã chạy chia làm 3 files như hướng dẫn trong phần 2.3
2. Mô tả từng bước thực hiện như hướng dẫn trong phần 2.3. Trong mỗi bước, cần nêu bật các công việc đã làm, các vấn đề đã gặp phải, các lỗi chủ yếu đã gặp, lý giải lỗi và mô tả cách giải quyết.
3. Nêu các vấn đề mà anh (chị) thấy chưa thật hiểu, cần giảng viên giải thích kỹ hơn.

```

[1]     #include <iostream.h>
[2]     #include <conio.h>

[3]     int phanloai(int* a, int n, int*& chan, int*& le) {
[4]         int i = 0, sochan = 0;
[5]         while (i < n) {
[6]             if (a[i] % 2 == 0)          // so chan
[7]                 ++sochan;
[8]             ++i;
[9]         }
[10]        chan = new int[sochan];
[11]        le = new int[n-sochan];
[12]        i = 0;
[13]        int ic=0,il=0;
[14]        while (i < n) {
[15]            if (a[i] % 2)                // so le
[16]                le[il++] = a[i];
[17]            else
[18]                chan[ic++] = a[i];
[19]            ++i;
[20]        }
[21]        return sochan;
[22]    }

[23]    void print(int* a, int n) {
[24]        for (int i = 0; i < n; ++i)
[25]            cout << '\t' << a[i];
[26]    }

[27]    void main() {
[28]        // mang tinh
[29]        int n = 10;
[30]        int as[10] = {1,2,3,4,5,6,7,8,9,10};
[31]        int *c, *l;
[32]        int sochan = phanloai(as,n,c,l);
[33]        cout << "\nDay so ban dau:";
[34]        print(as,n);
[35]        cout << "\nDay so chan:";
[36]        print(c,sochan);
[37]        cout << "\nDay so le:";
[38]        print(l,n-sochan);
[39]        delete [] c; delete [] l;

[40]        // mang dong
[41]        cout << "\nCho chieu dai day so:";
[42]        cin >> n;
[43]        int *ad = new int[n];
[44]        for (int i = 0; i < n; ++i) {
[45]            cout << "\nad[" << i << "]=";
[46]            cin >> ad[i];
[47]        }
[48]        sochan = phanloai(ad,n,c,l);
[49]        cout << "\nDay so ban dau:";
[50]        print(ad,n);
[51]        cout << "\nDay so chan:";
[52]        print(c,sochan);
[53]        cout << "\nDay so le:";
[54]        print(l,n-sochan);
[55]        delete [] c; delete [] l;
[56]        cout << "\n Nhan mot phim bat ky de ket thuc";
[57]        cout.flush();
[58]        getch();
[59]    }

```

Các câu hỏi gợi ý, kiểm tra bổ sung:

1. Dòng 1 và 2 có ý nghĩa gì ?
2. Tại sao các tham biến chan và le ở dòng 3 lại có kiểu int*& ?
3. Giải thích các dòng lệnh 16 và 18
4. Tại sao ở dòng 30, ta không thể viết $a[n] = \{...\}$ thay cho $a[10]$?
5. Tại sao ở dòng 39, ta không viết delete c; delete l?
6. Dòng 57 có ý nghĩa gì? Nếu xóa dòng đó đi thì sao?

Bài 2: Lập trình hướng đối tượng: Lớp và chồng toán tử

1 Mục đích bài thực hành

Học viên biết cách xây dựng một cấu trúc dữ liệu tiêu biểu dưới dạng một lớp với các phép toán thông dụng, thử nghiệm viết chương trình theo phong cách hướng đối tượng, qua đó nắm vững các khái niệm và phương pháp thực hiện như:

- 1) Đóng gói dữ liệu
- 2) Hàm tạo, hàm hủy, hàm tạo bản sao, hàm toán tử gán, chồng toán tử
- 3) Tham số mặc định, truyền tham số và kiểu trả về
- 4) Xây dựng một thư viện lớp

2 Nhiệm vụ thí nghiệm

2.1 Bài tập

1. Xây dựng một lớp Vector để biểu diễn kiểu vector toán học có các phần tử (thành phần) kiểu double với các yêu cầu sau:
 - Có thể khai báo một vector với kích thước (số chiều) bất kỳ. Nếu không đưa kích thước thì nó sẽ được mặc định là 0. Tất cả các phần tử được khởi tạo giá trị bằng 0.
 - Một vector tự quản lý bộ nhớ động, có khả năng tự sao chép và thực hiện phép gán tương tự như với các kiểu dữ liệu cơ bản
 - Có thể xác định số chiều vector cho một vector bất kỳ
 - Có thể truy nhập từng phần tử vector thông qua toán tử `[]` với số thứ tự tính từ 0 và toán tử gọi hàm `()` với số thứ tự tính từ 1.
 - Có thể thực hiện các phép `+`, `-` cũng như `+=`, `-=` và các phép so sánh giữa hai vector và các phép `*`, `/` cũng như `*=`, `/=` giữa một vector và một số thực (double), đảo dấu `-` và đồng dấu `+`.
2. Viết chương trình minh họa cách sử dụng lớp Vector câu 1. Yêu cầu đưa số chiều, giá trị các phần tử qua bàn phím và kết quả thực hiện ra màn hình. Định nghĩa lại các toán tử `<<`, `>>` tương ứng để có thể sử dụng lại.

2.2 Nhiệm vụ chuẩn bị ở nhà

- 1) Ôn lại các kiến thức cơ sở nêu trong phần 1, xem lại các ví dụ và bài tập trên lớp.
- 2) Xây dựng thuật toán và viết chương trình sẵn trên giấy (viết tay, không chấp nhận bản sao!)

2.3 Nhiệm vụ trên phòng thí nghiệm

- 1) Tạo một dự án thư viện mới (New Project), đặt tên dự án (ví dụ "Mathlib") và chọn "Win32 Static Library". Chấp nhận các tham số mặc định.
- 2) Tạo một Header file (ví dụ vector.h) chứa phần khai báo lớp và một file nguồn (vector.cpp) chứa phần thực hiện các hàm thành viên còn lại. Bổ sung 2 files đó vào dự án.
- 3) Thực hiện lớp Vector theo các yêu cầu nêu trong phần 2.1, dịch toàn bộ dự án (Build), tìm cách giải thích và sửa các lỗi.
- 4) Bổ sung một dự án mới "Win32 Console Application" vào trong Workspace đã có để viết chương trình minh họa. Bổ sung thư viện được tạo ra ở trên (mathlib.lib) vào dự án và dịch toàn bộ dự án mới, sửa các lỗi biên dịch (*compile error*), lỗi liên kết (*link error*) nếu có và chạy thử chương trình.
- 5) Thử nghiệm thay đổi một số kiểu tham số và kiểu trả về trong hàm tạo bản sao, hàm toán tử gán và các hàm toán học. Phân tích các vấn đề gặp phải. Ghi chép các bước thực hiện để làm báo cáo.

3 Báo cáo thực hành

Mỗi học viên phải làm một báo cáo riêng, nộp cho cán bộ hướng dẫn thí nghiệm chậm nhất là một tuần sau buổi thực hành. Bài báo cáo phải có đủ các phần sau:

- 1) Toàn bộ mã chương trình đã chạy chia làm các files như hướng dẫn trong phần 2.3
- 2) Mô tả từng bước thực hiện như hướng dẫn trong phần 2.3. Trong mỗi bước, cần nêu bật các công việc đã làm, các vấn đề đã gặp phải, các lỗi chủ yếu đã gặp, lý giải lỗi và mô tả cách giải quyết.

Bài 3: Dẫn xuất lớp và cơ chế đa xạ

1 Mục đích bài thực hành

Học viên hiểu sâu về các cơ chế thừa kế và đa xạ trong dẫn xuất lớp, qua đó nắm vững các vấn đề như:

- 1) Phân tích và mô tả quan hệ tĩnh giữa các lớp đối tượng thông qua dẫn xuất lớp
- 2) Kiểm soát truy nhập thành viên ở lớp cơ sở và tác dụng đối với các lớp dẫn xuất
- 3) Phân biệt giữa hàm ảo và hàm thành viên thông thường, lớp trừu tượng và lớp thông thường
- 4) Phân biệt giữa thừa kế giao diện và thừa kế phần thực hiện và các tác dụng của chúng
- 5) Ý nghĩa của việc sử dụng con trỏ trong lập trình hướng đối tượng

2 Nhiệm vụ thí nghiệm

2.1 Bài tập

1. Xây dựng một thư viện lớp (có ít nhất 4 lớp đối tượng) để biểu diễn các khối hàm sử dụng trong mô phỏng và thực hiện các hệ thống điều khiển theo phương pháp sơ đồ khối theo bài giảng trên lớp.
2. Tự chọn một ví dụ mô phỏng và viết chương trình để minh họa cách sử dụng thư viện hàm này.

2.2 Nhiệm vụ chuẩn bị ở nhà

- 1) Ôn lại các kiến thức cơ sở nêu trong phần 1, xem lại các ví dụ trên lớp.
- 2) Vẽ biểu đồ lớp và viết mã chương trình sẵn trên giấy (viết tay, không chấp nhận bản sao!)

2.3 Nhiệm vụ trên phòng thí nghiệm

- 1) Tạo một dự án thư viện mới. Khai báo và thực hiện mỗi lớp trong một file *.h và một file *.cpp. Bổ sung các file này vào dự án. Dịch thử dự án và sửa lỗi.
- 2) Bổ sung một dự án mới vào Workspace để viết chương trình minh họa cách mô phỏng một hệ thống sử dụng các lớp đã xây dựng. Sử dụng một mảng động có các phần tử là con trỏ để chứa các đối tượng (khối hàm) trong hệ thống. Bổ sung thư viện được tạo ra ở trên vào dự án mới. Dịch toàn bộ dự án mới này, sửa các lỗi và chạy thử chương trình.
- 3) Thí nghiệm với việc thay đổi quyền truy nhập (public, protected, private) của dữ liệu cũng như một số hàm thành viên ở lớp cơ sở và các lớp dẫn xuất, dịch lại lần lượt cả hai dự án để xem tác dụng.
- 4) Thí nghiệm với việc thay đổi thuộc tính "virtual" của một số hàm thành viên ở lớp cơ sở và các lớp dẫn xuất, dịch lại lần lượt cả hai dự án để xem tác dụng.
- 5) Trong chương trình minh họa, thử không dùng các biến con trỏ vào lớp cơ sở để tạo và lưu giữ địa chỉ các đối tượng, mà dùng biến tham chiếu hoặc biến trực tiếp thông thường và tìm hiểu các vấn đề gặp phải trong việc tạo cũng như quản lý và sử dụng các đối tượng.

3 Báo cáo thực hành

Mỗi học viên phải làm một báo cáo riêng, nộp cho cán bộ hướng dẫn thí nghiệm chậm nhất là một tuần sau buổi thực hành. Báo cáo có thể viết tay hoặc in, nhưng các bản sao photocopy sẽ không được chấp nhận. Bài báo cáo phải có đủ các phần sau:

- 1) Toàn bộ mã chương trình đã chạy chia làm các files như hướng dẫn trong phần 2.3
- 2) Mô tả từng bước thực hiện như hướng dẫn trong phần 2.3. Trong mỗi bước, cần nêu bật các công việc đã làm, các vấn đề đã gặp phải, các lỗi chủ yếu đã gặp, lý giải lỗi và mô tả cách giải quyết.
- 3) Nêu các vấn đề mà anh (chị) thấy chưa thật hiểu, cần giảng viên giải thích kỹ hơn.

Bài 4: Lập trình tổng quát

1 Mục đích bài thực hành

Cho học viên luyện tập về phương pháp lập trình tổng quát, qua đó nắm vững các kỹ thuật như:

- 1) Khuôn mẫu hàm, khuôn mẫu lớp
- 2) Tổng quát hóa phép toán
- 3) Tổng quát hóa phương pháp truy nhập phần tử qua Iterator

2 Nhiệm vụ thí nghiệm

2.1 Bài tập

1. Xây dựng các thuật toán sao chép (*copy*) và tìm kiếm phần tử cực trị (*find*) tổng quát. Bổ sung một số khuôn mẫu lớp toán tử tiêu biểu (*Greater*, *Less*, ...) để sử dụng với thuật toán tìm kiếm tổng quát. Viết chương trình minh họa như sau:
 - Khai báo 2 mảng tĩnh các số thực với giá trị đầu tùy ý.
 - Áp dụng thuật toán sao chép để sao chép từ mảng 1 sang mảng 2
 - Áp dụng thuật toán tìm kiếm để tìm phần tử lớn nhất/nhỏ nhất trong mảng 1
 - Các kết quả quan trọng được in ra màn hình
2. Tổng quát hóa lớp *Vector* trong bài thí nghiệm 2 thành một khuôn mẫu lớp. Bổ sung hai hàm thành viên *begin()* và *end()* để có thể áp dụng các thuật toán tổng quát. Sử dụng thuật toán sao chép tổng quát trong câu 1 để thực hiện hàm tạo bản sao và hàm toán tử gán. Bổ sung các phép toán cộng và trừ vector là các khuôn mẫu hàm toán tử bên ngoài (có thể áp dụng phương pháp thuật toán tổng quát). Viết chương trình minh họa cách sử dụng kết hợp với các thuật toán tổng quát trong câu 1 như sau:
 - Khai báo một mảng tĩnh và một biến *Vector* kiểu số thực và có số phần tử như nhau. Các phần tử của mảng tĩnh được đặt giá trị đầu tùy ý.
 - Áp dụng thuật toán sao chép để sao chép giá trị các phần tử từ mảng sang *Vector*
 - Áp dụng thuật toán tìm kiếm để tìm phần tử lớn nhất/nhỏ nhất của *Vector*
 - Các kết quả quan trọng được đưa ra màn hình.

2.2 Nhiệm vụ chuẩn bị ở nhà

- 1) Ôn lại các kiến thức cơ sở nêu trong phần 1, xem lại các ví dụ và bài tập trên lớp.
- 2) Xây dựng thuật toán và viết chương trình sẵn trên giấy (viết tay, không chấp nhận bản sao!)

2.3 Nhiệm vụ trên phòng thí nghiệm

- 1) Thực hiện các bước như trong phần 2.1 yêu cầu. Các thuật toán tổng quát và các khuôn mẫu lớp toán tử được lưu trong một file riêng "algo.h". Khuôn mẫu lớp *Vector* được lưu riêng trong file "vector.h". Các chương trình thử nghiệm có tên lần lượt là "ct1.cpp" và "ct2.cpp".
- 2) Ghi chép các bước thực hiện để làm báo cáo.

3 Báo cáo thực hành

Mỗi học viên phải làm một báo cáo riêng, nộp cho cán bộ hướng dẫn thí nghiệm chậm nhất là một tuần sau buổi thực hành. Báo cáo có thể viết tay hoặc in, nhưng các bản sao photocopy sẽ không được chấp nhận. Bài báo cáo phải có đủ các phần sau:

- 1) Toàn bộ mã chương trình đã chạy chia làm các files như hướng dẫn trong phần 2.3
- 2) Mô tả từng bước thực hiện như hướng dẫn trong phần 2.1 và 2.3. Trong mỗi bước, cần nêu bật các công việc đã làm, các vấn đề đã gặp phải, các lỗi chủ yếu đã gặp, lý giải lỗi và mô tả cách giải quyết.