

NGÔN NGỮ LẬP TRÌNH C++

Ngô Quốc Dũng

Email : quocdung.ngo@gmail.com

Facebook nhóm C++ : <https://www.facebook.com/groups/CPP4PTIT13>

Tài liệu tham khảo : <http://www.tutorialspoint.com/cplusplus/index.htm>



Học Viện Bưu Chính Viễn Thông

NGÔN NGỮ LẬP TRÌNH C++



Tổng quan môn học

Part 1: Giới thiệu tổng quan về các phương pháp lập trình với C++

(trang 15 - 127)

Part 2: Lập trình hướng đối tượng với C++

(trang 128 - 197)

Part 3: Lập trình nâng cao với C++

Cách tính điểm :

Điểm chuyên cần (10%)

Điểm kiểm tra trên lớp (20%)

Điểm đồ án (20% - 5sv/nhóm)

Điểm thi (50%)

Chương 1: Giới thiệu tổng quan về các phương pháp lập trình

- Các phương pháp lập trình
- Phương pháp lập trình hướng cấu trúc
- Các đặc trưng cơ bản của phương pháp hướng đối tượng
- Giới thiệu ngôn ngữ C++
- Bài tập làm quen với C++

Lịch sử và các phương pháp lập trình

phần mềm máy tính là gì?

Một phần mềm máy tính thực hiện một nhiệm vụ cụ thể, và có thể tương tác với người sử dụng và phần cứng của máy tính.



Hướng dẫn - Quy
trình



Yêu cầu công việc



Người lao động



Sản phẩm



phần mềm máy
tính



Yêu cầu công việc



máy tính



Sản phẩm

Lịch sử và các phương pháp lập trình

Các phương pháp lập trình

- **Lập trình tuyến tính : mang tính tuần tự**

vd : Assembly, basic

- **Lập trình hướng cấu trúc : cấu trúc dữ liệu + giải thuật**

vd : Pascal, Foxpro, C

- **Lập trình hướng đối tượng : Dữ liệu + Hành vi của dữ liệu**

vd : Java, C++, C#

Lịch sử và các phương pháp lập trình

Các phương pháp lập trình - Lập trình cấu trúc



Yêu cầu công việc:

Chương trình hello world với C++

```
// Chương trình hello World C++  
#include <iostream> // thư viện cho phép hiện thông tin trên màn hình máy tính  
using namespace std;
```

```
int main() // hàm chính của chương trình  
{  
    cout << "Hello World!"; // lệnh hiện thị lên màn hình  
    return 0; // chỉ định chương trình kết thúc thành công  
} // kết thúc chương trình
```

Lịch sử và các phương pháp lập trình

Các phương pháp lập trình - Lập trình cấu trúc

Một số kiểu dữ liệu đơn giản

Type	Typical Bit Width	Typical Range
char	1 byte	-127 to 127 or 0 to 255
unsigned char	1 byte	0 to 255
signed char	1 byte	-127 to 127
int	4 bytes	-2147483648 to 2147483647
unsigned int	4 bytes	0 to 4294967295
signed int	4 bytes	-2147483648 to 2147483647
short int	2 bytes	-32768 to 32767
unsigned short int	Range	0 to 65,535
signed short int	Range	-32768 to 32767
long int	4 bytes	-2,147,483,647 to 2,147,483,647
signed long int	4 bytes	same as long int
unsigned long int	4 bytes	0 to 4,294,967,295
float	4 bytes	+/- 3.4e +/- 38 (~7 digits)
double	8 bytes	+/- 1.7e +/- 308 (~15 digits)
long double	8 bytes	+/- 1.7e +/- 308 (~15 digits)
wchar_t	2 or 4 bytes	1 wide character

Lịch sử và các phương pháp lập trình

Các phương pháp lập trình - Lập trình cấu trúc

Yêu cầu công việc:

Tính tổng 2 số tự nhiên :

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

int main ()
{
    int a,b;
    cout << "nhap so dau tien : ";
    cin >> a;
    cout << "nhap so thu hai : ";
    cin >> b;
    cout << "tong hai so la : "<< a+b;
    return 0;
}
```

nhap so dau tien : 3.4
nhap so thu hai : tong hai so la : 3

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

int main ()
{
    float a,b;
    cout << "nhap so dau tien : ";
    cin >> a;
    cout << "nhap so thu hai : ";
    cin >> b;
    cout << "tong hai so la : "<< a+b;
    return 0;
}
```

nhap so dau tien : 3.4
nhap so thu hai : 4.5
tong hai so la : 7.9

Lịch sử và các phương pháp lập trình

Các phương pháp lập trình - Lập trình cấu trúc

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

int main()
{
    int a;
    cout << "nhap gia tri mot so bat ki : ";
    cin >> a;

    if (a > 10){
        cout << "so ban vua nhap lon hon 10 "<< endl;
    }
    else{
        cout << "so ban vua nhap nho hon hoac bang 10 " << endl;
    }
    return 0;
}
```

nhap gia tri mot so bat ki : 9
so ban vua nhap nho hon 10

Lịch sử và các phương pháp lập trình

Các phương pháp lập trình - Lập trình cấu trúc

Yêu cầu công việc

Nhập 3 số tự nhiên bất kì.

Tính tổng 2 số lớn nhất

Hiệu (dương) của 2 số bé nhất



Lịch sử và các phương pháp lập trình

Các phương pháp lập trình - Lập trình cấu trúc

```
for(lệnh ban đầu ; điều kiện ; lệnh tiếp theo){  
    Lệnh trong vòng lặp  
}
```

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

```
int main()  
{  
    int i;  
  
    for(i=0 ; i<10 ; i++)  
        cout << "Toi di hoc day du va dung gio" << endl;  
    return 0;  
}
```

Lịch sử và các phương pháp lập trình

Các phương pháp lập trình - Lập trình cấu trúc

Tổng mười số nguyên đầu tiên

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

int main()
{
    int i;
    int a = 0;
    for(i=0 ; i<10 ; i++){
        a = a+i;
    }
    cout << "Tổng 10 số nguyên đầu tiên là : " << a << endl;
    return 0;
}
```

Tổng 10 số nguyên đầu tiên là : 45

Lịch sử và các phương pháp lập trình

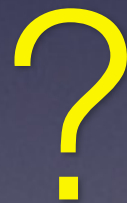
Các phương pháp lập trình - Lập trình cấu trúc

Yêu cầu công việc

Nhập một số nguyên bằng bàn phím : n

Tính giai thừa $n!$

Tính tổng 4 số cuối cùng của dãy số.



Lịch sử và các phương pháp lập trình

Các phương pháp lập trình - Lập trình cấu trúc

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

int main()
{
    // 10 số bất kì
    int elements[10] = { 3, 14, 15, 9, 26, 5, 35, 8, 7, 10 };

    int i, j;

    for (i=0, j=9; i <= j; i++, j--)
        cout << " Thứ tự "
            << " [" << i << "]= " << So[i]
            << " [" << j << "]= " << So[j]
            << endl;

    return 0;
}
```

Lịch sử và các phương pháp lập trình

Các phương pháp lập trình - Lập trình cấu trúc

Yêu cầu công việc:

Lập một chương trình tính so sánh giữa 2 số tự nhiên (a,b).

Tính diện tích một hình vuông có cạnh = a.

Tính tiền điện tiêu thụ với số điện tiêu thụ là a.

STT	Mức sử dụng của một hộ trong một tháng	Giá bán điện (đồng/kWh)
1	Bậc 1: Cho kWh từ 0-50	1.388
2	Bậc 2: Cho kWh từ 51-100	1.433
3	Bậc 3: Cho kWh từ 101-200	1.660
4	Bậc 4: Cho kWh từ 201-300	2.082
5	Bậc 5: Cho kWh từ 301-400	2.324
6	Bậc 6: Cho kWh từ 401 trở lên	2.399

Theo Quyết định số 4887/QĐ-BCT ngày 30/5/2014

Lịch sử và các phương pháp lập trình

Các phương pháp lập trình - Lập trình cấu trúc

Yêu cầu công việc:

Lập một chương trình tính điểm trung bình sinh viên :

- Nhập điểm lý thuyết và điểm thực hành
- Cách tính điểm trung bình = $(\text{điểm lý thuyết} + \text{điểm thực hành})/2$.
- Sinh viên qua nếu điểm trung bình trên 5
- Hiện tên sinh viên, lớp theo học và kết quả : Qua hay không

Lịch sử và các phương pháp lập trình

Các phương pháp lập trình - Lập trình cấu trúc

Chương trình tính điểm trung bình

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

int main() {
    string ho_ten, ten_lop;
    float diemLyThuyet, diemThucHanh, diemTrungBinh;
    cout << "Nhập họ tên : ";
    cin >> ho_ten;
    cout << "Nhập tên lớp : « ;
    cin >> ten_lop;
    cout << "Nhập điểm lý thuyết : ";
    cin >> diemLyThuyet;
    cout << "Nhập điểm thực hành : ";
    cin >> diemThucHanh;

    diemTrungBinh = (diemLyThuyet+diemThucHanh)/2;

    cout << "\nHọ tên: " << ho_ten;
    cout << "\nđiểm lý thuyết: " << diemLyThuyet;
    cout << "\nđiểm thực hành: " << diemThucHanh;
    cout << "\nđiểm trung bình: " << diemTrungBinh;

    if(diemTrungBinh>=5)
        cout << " => " << ho_ten << "lớp " << ten_lop << " Được Qua";
    else
        cout << " => " << ho_ten << "lớp " << ten_lop << " Bị Trượt";
    return 0;
}
```

Lịch sử và các phương pháp lập trình

Các phương pháp lập trình - Lập trình cấu trúc

Chương trình tính điểm trung bình P1 (5 điểm)

Chương trình tính điểm trung bình và xét kết quả P2 (6 điểm)

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

int main() {
    string ho_ten, ten_lop;
    float diemLyThuyet, diemThucHanh, diemTrungBinh;
    cout << "Nhập họ ten : ";
    cin >> ho_ten;
    cout << "Nhập ten lop : « ;
    cin >> ten_lop;
    cout << "Nhập điểm lý thuyết : ";
    cin >> diemLyThuyet;
    cout << "Nhập điểm thực hành : ";
    cin >> diemThucHanh;

    diemTrungBinh = (diemLyThuyet+diemThucHanh)/2;

    cout << "\nHo ten: " << ho_ten;
    cout << "\nđiểm lý thuyết: " << diemLyThuyet;
    cout << "\nđiểm thực hành: " << diemThucHanh;
    cout << "\nđiểm trung bình: " << diemTrungBinh;

    if(diemTrungBinh>=5)
        cout << " => " << ho_ten << " Được Qua";
    else
        cout << " => " << ho_ten << " Bị Trượt";

    return 0;
}
```

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

int main() {
    string ho_ten, ten_lop;
    float diemLyThuyet, diemThucHanh, diemTrungBinh;
    cout << "Nhập họ ten : ";
    cin >> ho_ten;
    cout << "Nhập ten lop : « ;
    cin >> ten_lop;
    cout << "Nhập điểm lý thuyết : ";
    cin >> diemLyThuyet;
    cout << "Nhập điểm thực hành : ";
    cin >> diemThucHanh;

    diemTrungBinh = (diemLyThuyet+diemThucHanh)/2;

    cout << "\nHo ten: " << ho_ten;
    cout << "\nđiểm lý thuyết: " << diemLyThuyet;
    cout << "\nđiểm thực hành: " << diemThucHanh;
    cout << "\nđiểm trung bình: " << diemTrungBinh;

    if(diemTrungBinh>=6)
        cout << " => " << ho_ten << " Được Qua";
    else
        cout << " => " << ho_ten << " Bị Trượt";

    return 0;
}
```

Lịch sử và các phương pháp lập trình

Các phương pháp lập trình - Lập trình cấu trúc

Làm sao để phát triển một chương trình tổng quan ?

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

```
int main() {
```

```
    string ho_ten, ten_lop;
    float diemLyThuyet, diemThucHanh, diemTrungBinh;
    cout << "Nhập họ tên : ";
    cin >> ho_ten;

    cout << "Nhập tên lớp : « ;
    cin >> ten_lop;
    cout << "Nhập điểm lý thuyết : ";
    cin >> diemLyThuyet;
    cout << "Nhập điểm thực hành : ";
    cin >> diemThucHanh;
```

```
    diemTrungBinh = (diemLyThuyet+diemThucHanh)/2;
```

```
    cout << "\nHọ tên: " << ho_ten;
    cout << "\nđiểm lý thuyết: " << diemLyThuyet;
    cout << "\nđiểm thực hành: " << diemThucHanh;
    cout << "\nđiểm trung bình: " << diemTrungBinh;
```

```
    if(diemTrungBinh>=5)
```

```
        cout << " => " << ho_ten << " Được Qua";
```

```
    else
```

```
        cout << " => " << ho_ten << " Bị Trượt";
```

```
    return 0;
```

```
}
```

Lấy thông tin điểm
và sinh viên

Xét điểm qua hay không

Lịch sử và các phương pháp lập trình

Các phương pháp lập trình - Lập trình cấu trúc

Hàm (*Function, Subroutine, Procedure, Method, Proccess*): một đoạn chương trình thực hiện một nhiệm vụ nào đó phục vụ mục tiêu chủ quan của người lập trình, được xây dựng một lần, sử dụng nhiều lần ở mọi lúc, mọi nơi, mọi thời điểm trong chương trình.

Hàm còn có nghĩa là đơn vị chương trình (unit program): lập trình có cấu trúc là phương pháp lập trình trên các hàm hoặc thủ tục. Một chương trình theo kiểu của lập trình cấu trúc là dãy các lời gọi hàm hoặc thủ tục.

Hàm còn có nghĩa là chức năng (Function): một ứng dụng lớn thường được chia thành các chức năng. Mỗi chức năng thực hiện một ứng dụng nhỏ hơn.

Hàm còn có nghĩa là quá trình xử lý (proccess): khi nó thực hiện một nhiệm vụ hoặc thuật toán đơn lẻ.

Khi giải quyết một bài toán hoặc xây dựng ứng dụng thì điều quan trọng nhất là làm thế nào ta phân tích được lời giải bài toán hoặc ứng dụng để có được các hàm.

Cú pháp xây dựng hàm:

```
[Kiểu hàm]   Tên-hàm ( danh sách đối của hàm ) {  
    <Thân-hàm>; //dãy các chỉ thị thực hiện để đạt được mục tiêu  
    return (giá trị); //giá trị trả về của hàm  
}
```

Lịch sử và các phương pháp lập trình

Các phương pháp lập trình - Lập trình cấu trúc

Ví dụ. Cấu trúc *chương trình*. Viết chương trình tính tổng, hiệu hai số a và b.

//Phần 1. Khai báo việc sử dụng chương trình

```
#include <iostream> //khai báo sử dụng các hàm trong iostream
```

```
using namespace std; //khai báo sử dụng không gian tên của iostream
```

//Phần 2: mô tả các hàm sử dụng trong chương trình

```
int tong( int a, int b) { //hàm tính tổng hai số
```

```
    int c = a + b; // lấy c = a+b
```

```
    return (c); //trả lại c
```

```
}
```

```
int hieu( int a, int b) { //hàm tính hiệu hai số
```

```
    int c = a - b; // lấy c = a - b
```

```
    return (c); //trả lại c
```

```
}
```

// Chương trình chính

```
int main () { // điểm đầu thực hiện chương trình
```

```
    int a, b; //khai báo hai biến nguyên a, b;
```

```
    cout<<"Nhập a ="; cin>>a; //nhập a từ bàn phím
```

```
    cout<<"Nhập b ="; cin>>b; //nhập b từ bàn phím
```

```
    cout<<"Tổng a + b ="<<tong(a,b)<<endl; //đưa ra tổng a+b và xuống dòng
```

```
    cout<<"Hiệu a - b ="<<hieu(a,b)<<endl; //đưa ra tổng a-b và xuống dòng
```

```
    system("PAUSE"); //dừng lại xem kết quả
```

```
}
```

Lịch sử và các phương pháp lập trình

Các phương pháp lập trình - Lập trình cấu trúc

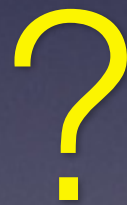
Làm sao để phát triển một chương trình tổng quan ?

Hàm lấy thông tin điểm và sinh viên

layThongTinDiem()

layTenLop()

layTenHocSinh()



Lịch sử và các phương pháp lập trình

Các phương pháp lập trình - Lập trình cấu trúc

Làm sao để phát triển một chương trình tổng quan ?

Hàm Xét Qua Hay Không

```
void xetQuaHayKhong(float diemTrungBinh, string ho_ten, int diemQua){  
    if(diemTrungBinh >= diemQua)  
        cout << " => " << ho_ten << " Duoc Qua";  
    else  
        cout << " => " << ho_ten << " Bi Truot";  
}
```

```
int main() {  
    string ho_ten, ten_lop;  
    float diemTrungBinh;  
    cout << "Nhap ho ten : ";  
    cin >> ho_ten;  
    cout << "Nhap ten lop : ";  
    cin >> ten_lop;  
    diemTrungBinh = layThongTinDiem();  
    xetQuaHayKhong(diemTrungBinh, ho_ten, 6);  
    return 0;  
}
```

Lịch sử và các phương pháp lập trình

Các phương pháp lập trình - Lập trình cấu trúc

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

```
float layThongTinDiem(){
    float diemLyThuyet, diemThucHanh, diemTrungBinh;
    cout << "Nhập điểm lý thuyết : ";
    cin >> diemLyThuyet;
    cout << "Nhập điểm thực hành : ";
    cin >> diemThucHanh;
    diemTrungBinh = (diemLyThuyet + diemThucHanh) / 2;

    return diemTrungBinh;
}
```

```
void xetQuaHayKhong(float diemTrungBinh, string ho_ten, int diemQua){
    if(diemTrungBinh >= diemQua)
        cout << " => " << ho_ten << " Được Qua";
    else
        cout << " => " << ho_ten << " Bị Trượt";
}
```

```
int main() {
    string ho_ten, ten_lop;
    float diemTrungBinh;
    cout << "Nhập họ tên : ";
    cin >> ho_ten;
    cout << "Nhập tên lớp : ";
    cin >> ten_lop;
    diemTrungBinh = layThongTinDiem();
    xetQuaHayKhong(diemTrungBinh, ho_ten, 6);
    return 0;
}
```

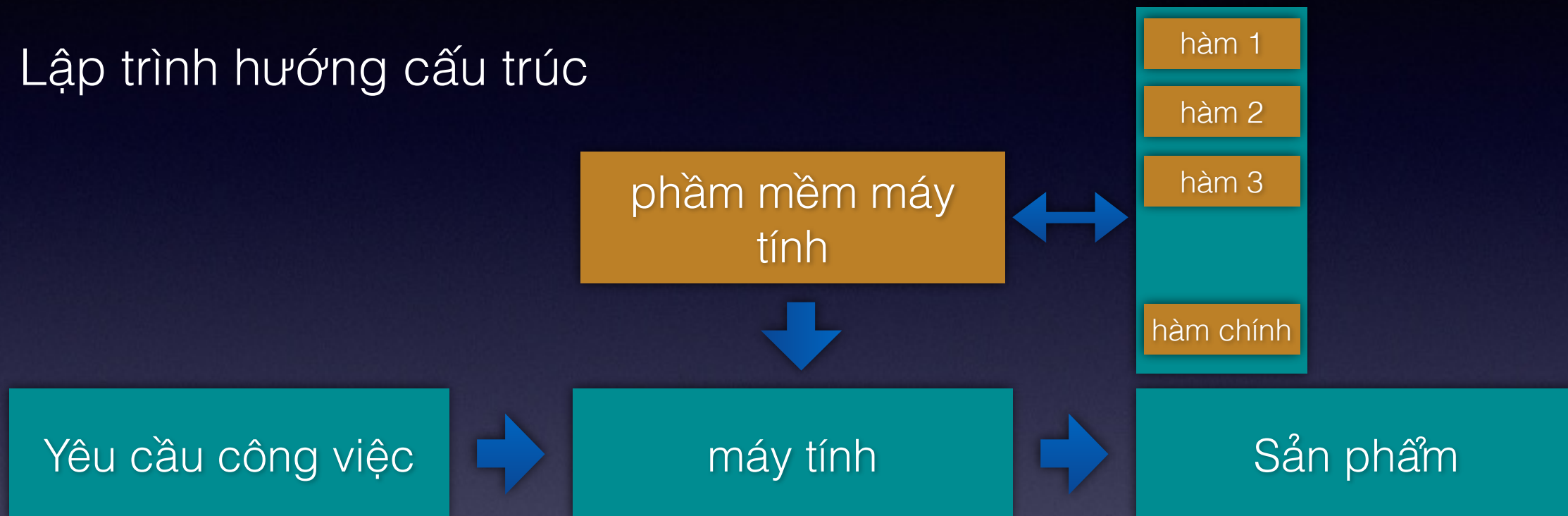
```
int main() {
    function 1()
    function 2()
    function...n()
    return 0;
}
```

Lịch sử và các phương pháp lập trình

Các phương pháp lập trình - Lập trình cấu trúc

các phương pháp hiện lập trình

Lập trình hướng cấu trúc



Trong lập trình hướng cấu trúc, chương trình chính được chia nhỏ thành các chương trình con và mỗi chương trình con thực hiện một công việc xác định. Chương trình chính sẽ gọi đến chương trình con theo một giải thuật, hoặc một cấu trúc được xác định trong chương trình chính

Lịch sử và các phương pháp lập trình

Các phương pháp lập trình - Lập trình cấu trúc

Đặc trưng cơ bản nhất của lập trình cấu trúc thể hiện ở mối quan hệ:

Chương trình = Cấu trúc dữ liệu + Giải thuật

Trong đó:

Cấu trúc dữ liệu là cách tổ chức dữ liệu, cách mô tả bài toán dưới dạng ngôn ngữ lập trình

Giải thuật là một quy trình để thực hiện một công việc xác định Trong chương trình, giải thuật có quan hệ phụ thuộc vào cấu trúc dữ liệu:

Một cấu trúc dữ liệu chỉ phù hợp với một số hạn chế các giải thuật.

Nếu thay đổi cấu trúc dữ liệu thì phải thay đổi giải thuật cho phù hợp.

Một giải thuật thường phải đi kèm với một cấu trúc dữ liệu nhất định.

Tính chất

Mỗi chương trình con có thể được gọi thực hiện nhiều lần trong một chương trình chính.

Các chương trình con có thể được gọi đến để thực hiện theo một thứ tự bất kì, tùy thuộc vào giải thuật trong chương trình chính mà không phụ thuộc vào thứ tự khai báo của các chương trình con.

Các ngôn ngữ lập trình cấu trúc cung cấp một số cấu trúc lệnh điều khiển chương trình.

Lịch sử và các phương pháp lập trình

Các phương pháp lập trình - Lập trình cấu trúc

Ưu điểm

Chương trình sáng sủa, dễ hiểu, dễ theo dõi.

Tư duy giải thuật rõ ràng.

Nhược điểm

Lập trình cấu trúc không hỗ trợ việc sử dụng lại mã nguồn: Giải thuật luôn phụ thuộc chặt chẽ vào cấu trúc dữ liệu, do đó, khi thay đổi cấu trúc dữ liệu, phải thay đổi giải thuật, nghĩa là phải viết lại chương trình.

Không phù hợp với các phần mềm lớn: tư duy cấu trúc với các giải thuật chỉ phù hợp với các bài toán nhỏ, nằm trong phạm vi một modul của chương trình. Với dự án phần mềm lớn, lập trình cấu trúc tỏ ra không hiệu quả trong việc giải quyết mối quan hệ vĩ mô giữa các modul của phần mềm.

Vấn đề

Vấn đề cơ bản của lập trình cấu trúc là bằng cách nào để phân chia chương trình chính thành các chương trình con cho phù hợp với yêu cầu, chức năng và mục đích của mỗi bài toán.

Thông thường, để phân rã bài toán trong lập trình cấu trúc, người ta sử dụng phương pháp thiết kế trên xuống (top-down).

Lịch sử và các phương pháp lập trình

Các phương pháp lập trình - Lập trình cấu trúc

Các cấu trúc lệnh trong C++ bao gồm:

Cấu trúc lệnh tuần tự (Sequential): dãy các chỉ thị hoặc lời gọi hàm thực hiện một cách tuần tự. Các lệnh có thể là lệnh đơn, lệnh có cấu trúc hoặc lệnh phức hợp. Các ví dụ được nêu ở trên đều được thể hiện bằng cấu trúc lệnh tuần tự.

Cấu trúc lệnh tuyến chọn: bao gồm cấu trúc if..else và switch(). Điểm khác biệt duy nhất giữa hai cấu trúc lệnh này là if..else lựa chọn một trong hai khả năng đúng, sai của biểu thức điều kiện để thực hiện, trái lại switch() lựa chọn khả năng đúng của biểu thức điều kiện trong nhiều khả năng để thực hiện.

Cấu trúc lệnh if..else.

```
if (biểu-thức-điều-kiện) {  
    <Câu lệnh 1>;  
}  
else {  
    <Câu lệnh 2>;  
}
```

Cấu trúc lệnh if..else khuyết.

```
if (biểu-thức-điều-kiện) {  
    <Câu lệnh 1>;  
}
```

Cấu trúc lệnh switch().

```
switch (biểu-thức) {  
    case H1: < câu lệnh 1>; break;  
    case H2: < câu lệnh 2>; break;  
    .....;  
    case Hn: < câu lệnh n>; break;  
    default: < câu lệnh n+1>; break;  
}
```


Lịch sử và các phương pháp lập trình

Các phương pháp lập trình - Lập trình cấu trúc

```
#include <iostream>
using namespace std;
int main() {
    int number;
    cout<<"Nhap mot so:"; cin>>number;
    if (number % 2) {
        cout<<"Số lẻ"<<endl;
    }
    else {
        cout<<"Số chẵn"<<endl;
    }
    system("PAUSE");
    return 0;
}
```

Lịch sử và các phương pháp lập trình

Các phương pháp lập trình - Lập trình cấu trúc

Ví dụ : Dịch tên tháng sang tiếng anh sử dụng cấu trúc switch().

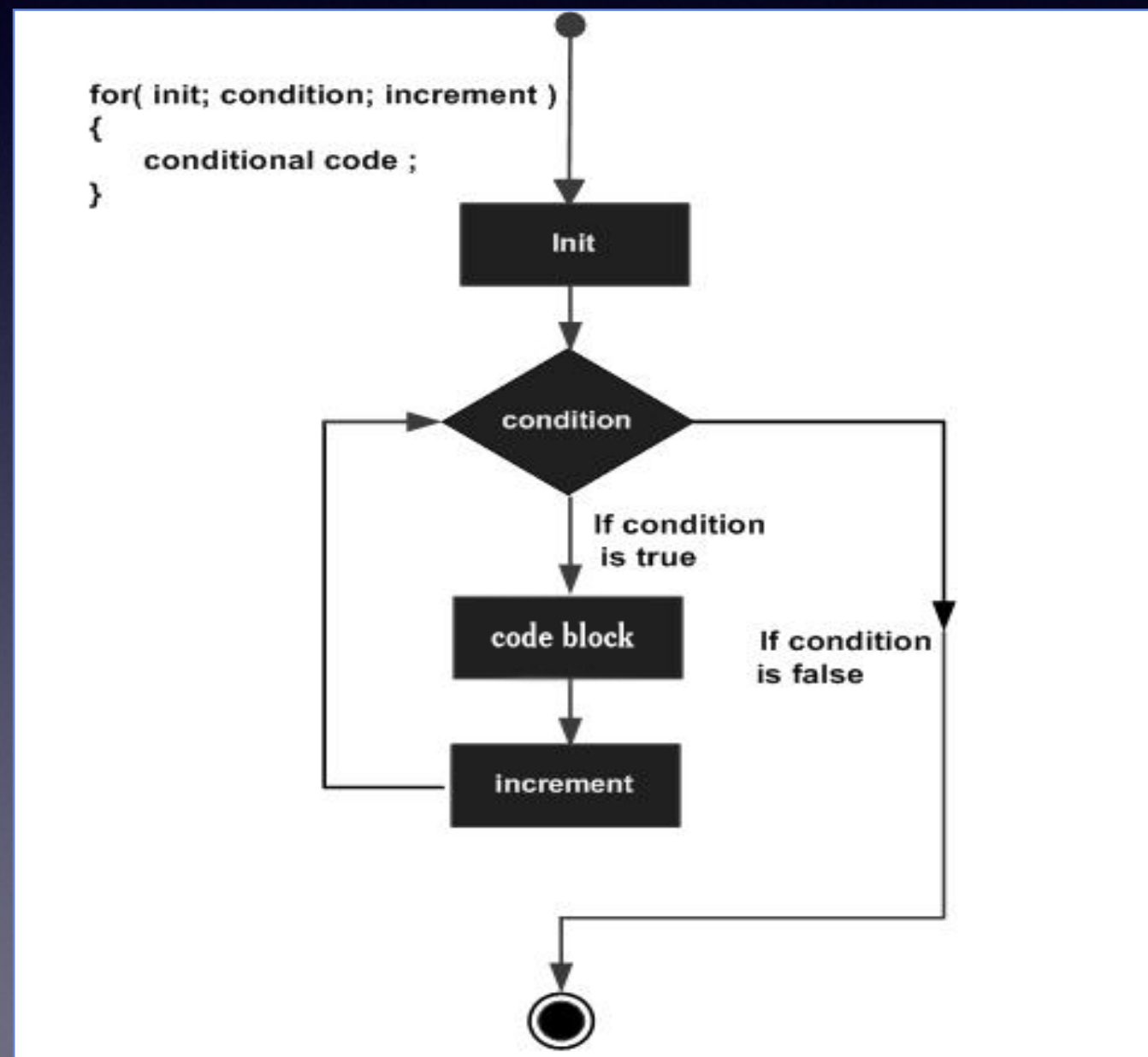
```
#include <iostream>
using namespace std;
int main( void ) {
    int thang;
    cout<<"Nhap thang:"; cin>>thang;
    switch(thang) {
        case 1: cout<<"January"<<endl; break;
        case 2: cout<<"Febuary"<<endl; break;
        case 3: cout<<"March"<<endl; break;
        case 4: cout<<"April"<<endl; break;
        case 5: cout<<"May"<<endl; break;
        case 6: cout<<"June"<<endl; break;
        case 7: cout<<"July"<<endl; break;
        case 8: cout<<"August"<<endl; break;
        case 9: cout<<"September"<<endl; break;
        case 10: cout<<"October"<<endl; break;
        case 11: cout<<"November"<<endl; break;
        case 12: cout<<"December"<<endl; break;
    }
    system("PAUSE");
    return 0;
}
```

Lịch sử và các phương pháp lập trình

Các phương pháp lập trình - Lập trình cấu trúc

Cấu trúc lệnh lặp (loop): for, while, do..while.

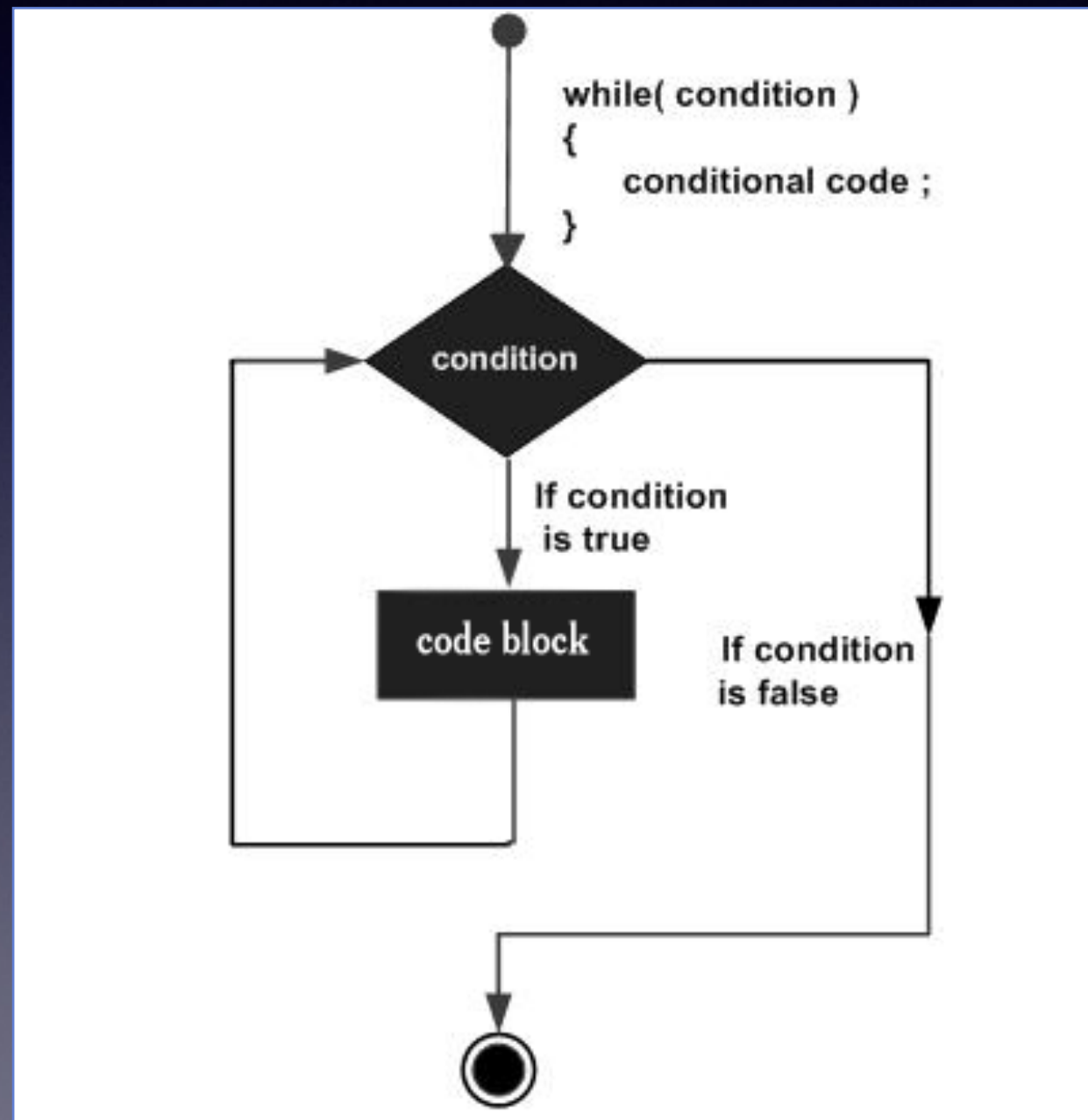
Cấu trúc lệnh lặp for:



Lịch sử và các phương pháp lập trình

Các phương pháp lập trình - Lập trình cấu trúc

Cấu trúc lệnh lặp while: trong khi biểu thức điều kiện còn đúng thì thực hiện câu lệnh.



Lịch sử và các phương pháp lập trình

Các phương pháp lập trình - Lập trình cấu trúc

Ví dụ : Tìm N và S để $(1/N) \leq \text{Epsilon}$ cho trước.

$$S = 1 + 1/2 + 1/3 + \dots + 1/N$$

```
#include <iostream>
using namespace std;
int main() {

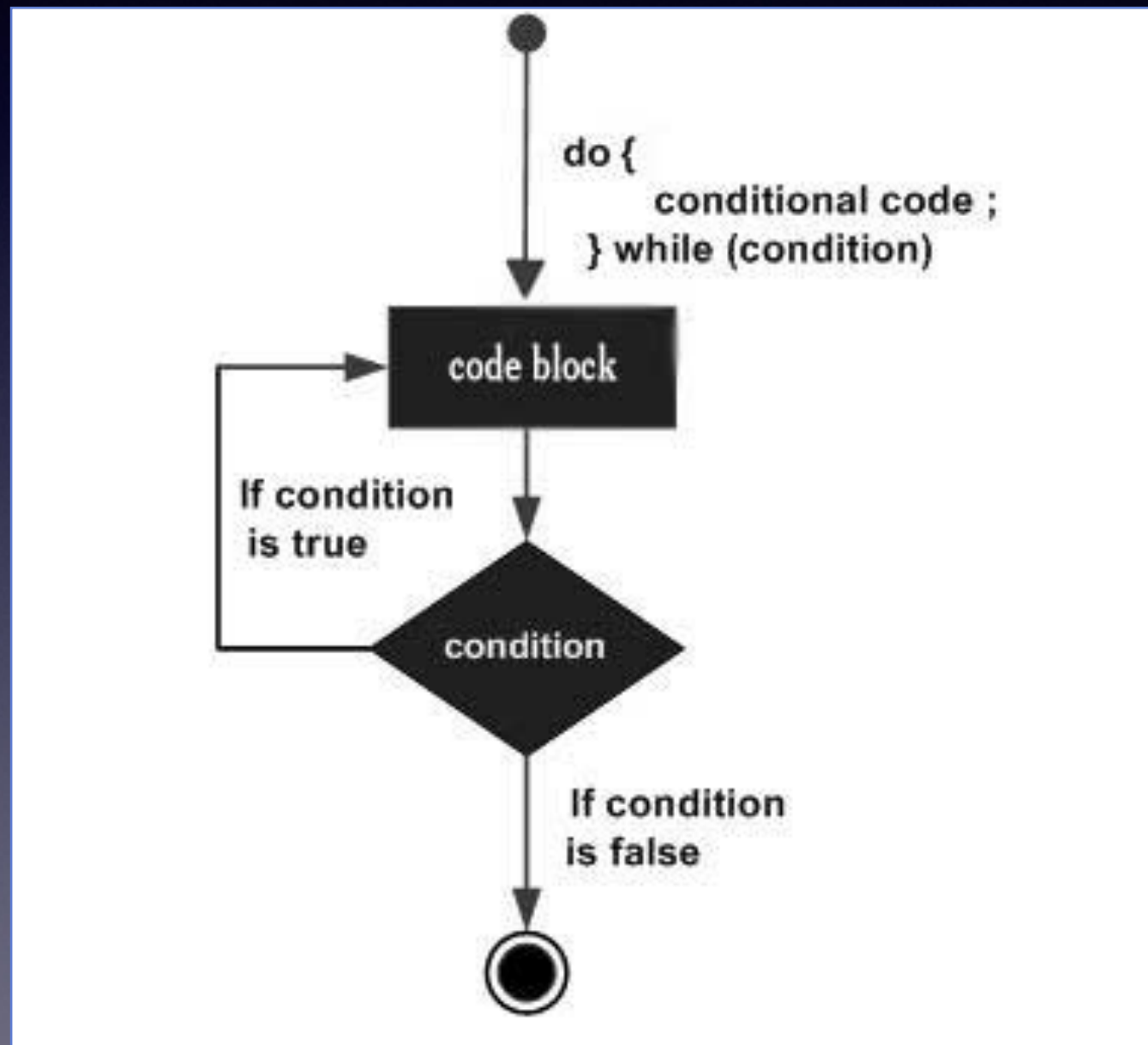
    float S = 0, Epsilon; int N=1;
    cout<<"Nhap Epsilon="; cin>>Epsilon;
    while( (Epsilon<=(float)1/ (float)N)) {
        S = S + ((float)1) / ((float) N);
        N = N + 1;
    }

    cout<<" N = "<<N <<" S ="<<S;
    system("PAUSE");
}
```

Lịch sử và các phương pháp lập trình

Các phương pháp lập trình - Lập trình cấu trúc

Cấu trúc lệnh lặp do..while: thực hiện câu lệnh trong khi biểu thức điều kiện còn đúng.



Lịch sử và các phương pháp lập trình

Các phương pháp lập trình - Lập trình cấu trúc

Ví dụ : Nhập đúng password là một số sử dụng cấu trúc lặp do..while.

```
#include <iostream>
using namespace std;
int main() {
    int number=1001, pass;
    do {
        system("cls");//xóa màn hình
        cout<<"Nhập Password:"; cin>>pass;
    } while(pass!=number);
    system("PAUSE");
    return 0;
}
```

Lịch sử và các phương pháp lập trình

Các phương pháp lập trình - Lập trình cấu trúc

Yêu cầu công việc:

Lập một chương trình tính so sánh giữa 2 số tự nhiên (a,b).

Tính diện tích một hình vuông có cạnh = a.

Tính tiền điện tiêu thụ với số điện tiêu thụ là a.

STT	Mức sử dụng của một hộ trong một tháng	Giá bán điện (đồng/kWh)
1	Bậc 1: Cho kWh từ 0-50	1.388
2	Bậc 2: Cho kWh từ 51-100	1.433
3	Bậc 3: Cho kWh từ 101-200	1.660
4	Bậc 4: Cho kWh từ 201-300	2.082
5	Bậc 5: Cho kWh từ 301-400	2.324
6	Bậc 6: Cho kWh từ 401 trở lên	2.399

Theo Quyết định số 4887/QĐ-BCT ngày 30/5/2014

Lịch sử và các phương pháp lập trình

Các phương pháp lập trình - Lập trình cấu trúc

Bài tập. Giải các bài tập dưới đây sử dụng các kiểu dữ liệu cơ bản và các cấu trúc lệnh.

Cho số tự nhiên N . Hãy tìm tổng các chữ số của N . Ví dụ $N = 12345$ thì

$$S = 1 + 2 + 3 + 4 + 5 = 15.$$

Cho số tự nhiên N . Hãy liệt kê tất cả các số nguyên tố nhỏ hơn N .

Hãy liệt kê tất cả các số nguyên tố có N chữ số sao cho tổng các chữ số của số đó đúng bằng S cho trước.

Cho số tự nhiên N , hãy phân tích N thành tích các thừa số nguyên tố.

Cho số tự nhiên a và b , hãy tìm ước số chung lớn nhất của hai số.

Một số tự nhiên được gọi là số hoàn hảo nếu tổng các ước số thực sự của nó kể cả 1 bằng chính nó. Hãy liệt kê các số hoàn hảo nhỏ hơn N .

Cặp số a, b được gọi là hữu nghị nếu tổng các ước số thực sự của a là b và tổng các ước số thực sự của b là a . Hãy liệt kê các số hoàn hảo có N chữ số.

Hãy liệt kê mã của các ký tự từ A đến Z cùng với mã nhị phân của ký tự.

Viết chương trình giải phương trình bậc 2.

Hãy liệt kê tất cả các số đối xứng có N chữ số và tổng các chữ số đúng bằng S cho trước.

Lịch sử và các phương pháp lập trình

Các phương pháp lập trình - Lập trình cấu trúc

Bài tập. Giải các bài tập dưới đây sử dụng các kiểu dữ liệu cơ bản và các cấu trúc lệnh.

11. Hãy viết chương trình liệt kê tất cả các số tự nhiên K thỏa mãn đồng thời những điều kiện dưới đây:

K là số có N chữ số;

K là số nguyên tố;

Đảo ngược các chữ số trong K là một số nguyên tố;

Tổng các chữ số trong K cũng là một số nguyên tố;

Mỗi chữ số trong K cũng là các số nguyên tố.

12. Hãy viết chương trình liệt kê tất cả các số tự nhiên K thỏa mãn đồng thời những điều kiện dưới đây:

K là số có N chữ số;

K là số đối xứng;

Tổng các chữ số của K là số chia hết cho 10;

Các chữ số của K đều khác 0.

Lịch sử và các phương pháp lập trình

Các phương pháp lập trình - Lập trình cấu trúc

13. Nhân dịp phát hành các số điện thoại 0919xxx.xxx. Công ty Vinaphone dự định phát hành N số điện thoại loại 1, M số điện thoại loại 2, K số điện thoại loại 3. Trong đó, số điện thoại loại 1, 2, 3 được định nghĩa như sau:

Loại 1: là những số điện thoại có sáu số cuối cùng của nó tạo thành một số đối xứng (thuận nghịch) có sáu chữ số.

Loại 2: là những số điện thoại loại 1 có sáu số cuối cùng của nó là các chữ số khác 0.

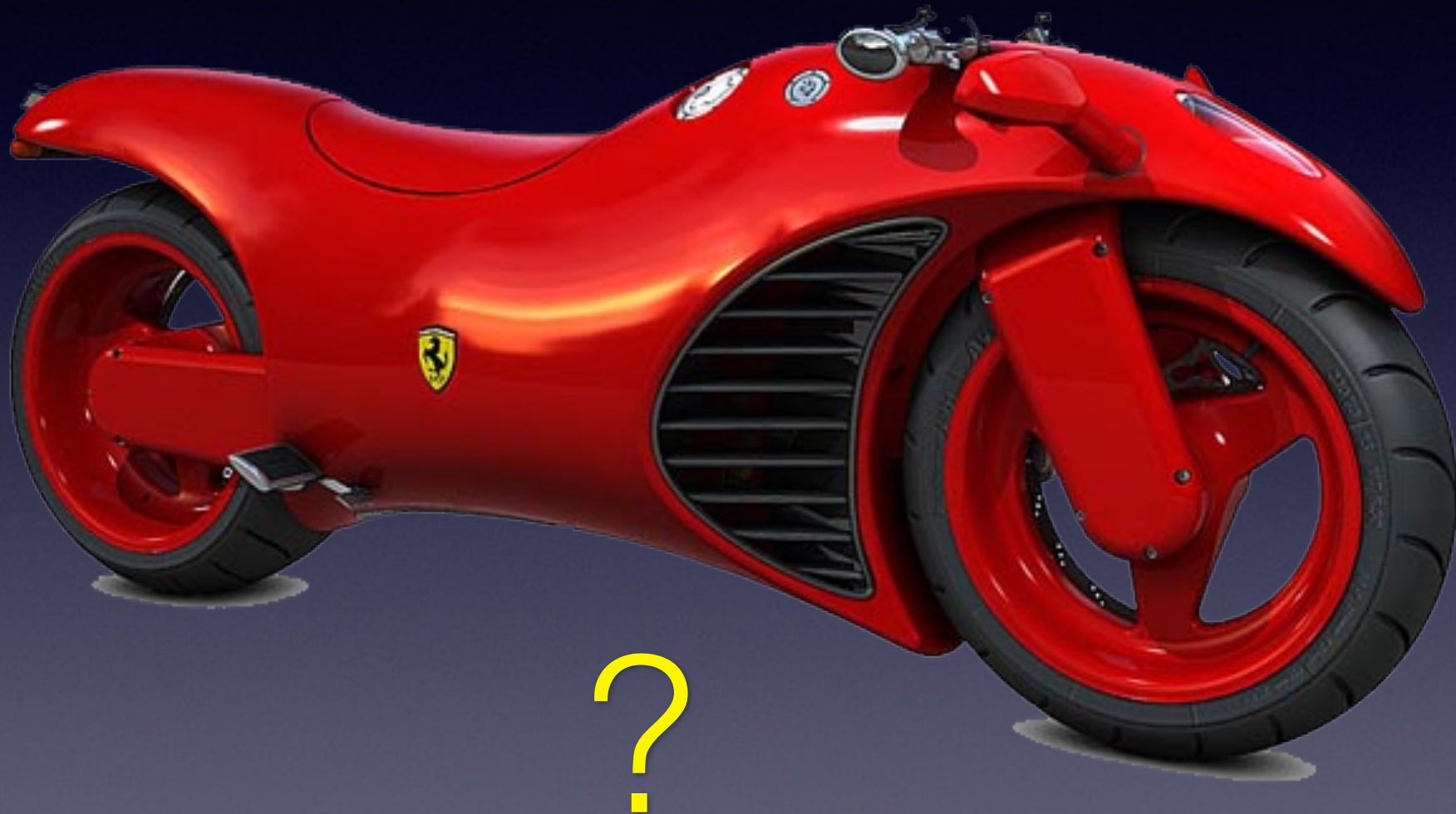
Loại 3: là những số điện thoại loại 2 có tổng của sáu chữ số cuối cùng là một số chia hết cho 10.

Bài toán được đặt ra là: cho một phương án phát hành N, M, K. Hãy cho biết công ty Vinaphone có thể thực hiện được phương án phát hành kể trên hay không? Đưa ra kết quả “YES” nếu phương án phát hành thực hiện được, đưa ra kết quả “NO” nếu phương án phát hành không thể thực hiện được.

Lịch sử và các phương pháp lập trình

Các phương pháp lập trình - Lập trình cấu trúc

Thiết kế trên xuống (Top-Down)



Thiết kế mô tô với phương pháp Top - Down