

CÔNG NGHỆ PHẦN MỀM

SOFTWARE ENGINEERING

TS. Võ Đức Hoàng

Khoa Công nghệ thông tin

Trường Đại học Bách khoa - Đại học Đà Nẵng

■ ***Giáo trình chính:***

- Andrew Troelsen, *Pro C# 5.0 and The .NET 4.5 Framework*, Apress, 2012.

■ ***Tài liệu tham khảo:***

- Simon Kendal, *Object Oriented Programming using C#*, BookBoon, 2012.
- Microsoft MSDN, *C# Programming Guide for Visual Studio 2013*.
- Joe Mayo, *LINQ Programming*, McGraw-Hill Education, 2009.
- Andrew Clymer, *Pro Asynchronous Programming with .NET*, Apress, 2013...

Lập trình và ngôn ngữ lập trình (6)

Lý do nghiên cứu các khái niệm về ngôn ngữ lập trình:

- Tăng khả năng thể hiện ý tưởng và giải quyết vấn đề.
 - Học ngôn ngữ lập trình có thể tăng năng lực và khả năng thể hiện ý tưởng ở dạng hình thức, có tính toán.
- Tăng khả năng tự động hóa tiến trình
 - Các chương trình được xây dựng chủ yếu để các tiến trình đơn giản hoặc thậm chí phức tạp được thực thi tự động.

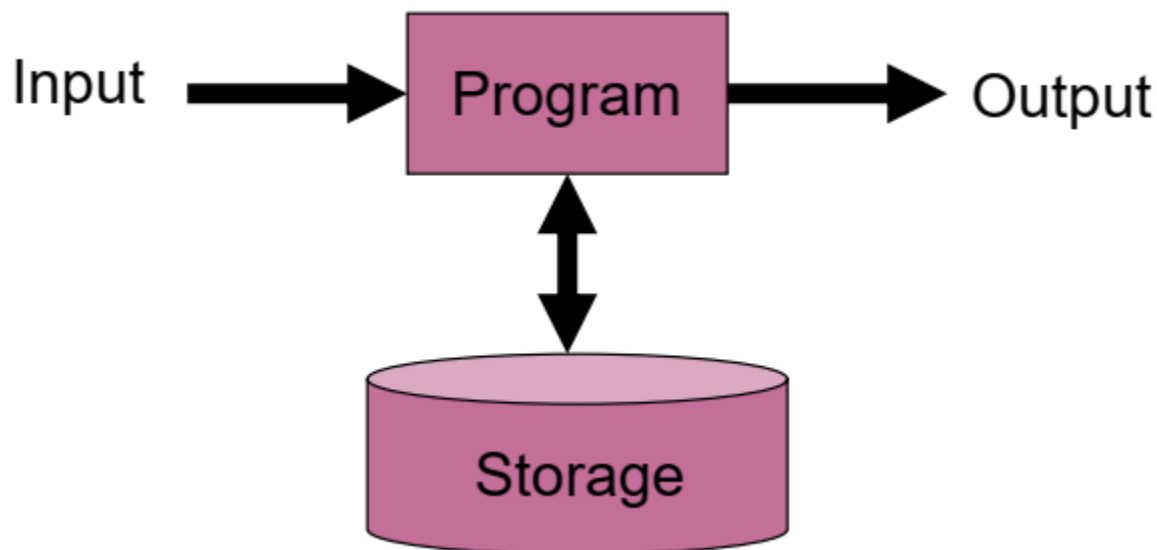
- Có nhiều phương pháp lập trình khác nhau nhiều ngôn ngữ lập trình khác nhau
- Điểm chung của các ngôn ngữ lập trình (NNLT)
 - Dễ diễn đạt
 - Dễ hiểu
 - Dễ thực thi trên máy tính

- Phát triển một chương trình máy tính.
- Yêu cầu kiến thức, kỹ năng và sự sáng tạo.
- Cả kỹ năng và kiến thức trong giải quyết vấn đề và ngôn ngữ lập trình.

- Hanly [2001], “Danh sách các lệnh hướng dẫn máy tính chuyển đổi thông tin từ dạng này sang dạng khác.”
- Thông tin lưu trữ ở vị trí bộ nhớ cụ thể.
- Nó được viết bằng ngôn ngữ lập trình.

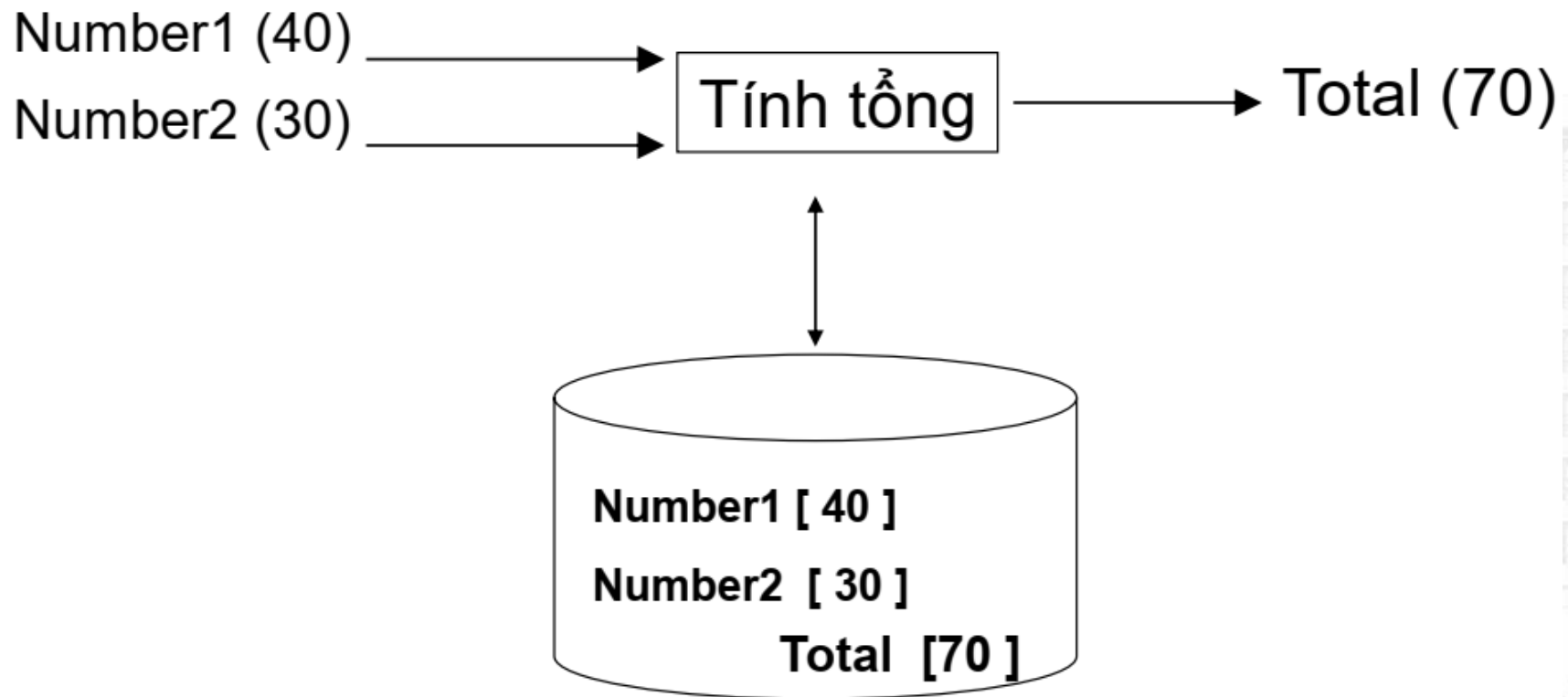
Chương trình máy tính

- Biến dữ liệu (đầu vào) thành thông tin hữu ích (đầu ra).
- Chương trình sử dụng các biến để lưu trữ dữ liệu



Chương trình máy tính

■ Ví dụ:



- Một bộ quy tắc, từ vựng và ký hiệu được sử dụng để viết chương trình máy tính.
- Thể hệ ngôn ngữ lập trình
 - Ngôn ngữ máy
 - Mã số nhị phân được hiểu bởi một CPU cụ thể.
 - Hợp ngữ
 - Mã ghi nhớ ứng với các chỉ lệnh ngôn ngữ máy
 - Ngôn ngữ cấp cao (Pascal, C, C ++, Java, ...)

■ Chương trình dịch

- Trình biên dịch

- Một chương trình dịch một chương trình trên ngôn ngữ bậc cao thành ngôn ngữ máy như một đơn vị hoàn chỉnh

- Trình thông dịch

- Một bộ chương trình thực hiện các chỉ lệnh trong khi chúng được dịch. Trình thông dịch thực hiện một chương trình từ hình thức bậc cao của nó.

■ Môi trường phát triển tích hợp (IDE)

- Một chương trình cung cấp cho người dùng môi trường để chỉnh sửa, gỡ lỗi và biên dịch mã nguồn chương trình trực tuyến.

■ Phân tích vấn đề

- Đánh giá báo cáo vấn đề sau và xác định đầu vào và đầu ra.

Lực lượng Không quân Quốc gia đã yêu cầu viết một chương trình để gán nhãn máy bay siêu thanh là “quân sự” hoặc “dân sự”. Chương trình sẽ được cung cấp tốc độ quan sát được của máy bay được tính bằng km/h và chiều dài ước tính của nó tính bằng mét. Đối với những chiếc máy bay di chuyển vượt quá 1100km/h, sẽ gán nhãn những chiếc máy bay dài hơn 52 mét ”dân sự” và máy bay ngắn hơn là “quân sự”. Đối với các máy bay di chuyển với tốc độ chậm hơn, sẽ đưa ra một thông báo “không xác định loại máy bay”.

Phát triển chương trình

- Sơ đồ Đầu vào – Xử lý – Đầu ra (Input-Process-Output - IPO)

Đầu vào	Xử lý	Đầu ra
<ul style="list-style-type: none">• Tốc độ• Chiều dài	<ul style="list-style-type: none">● Đánh giá tốc độ và chiều dài<ul style="list-style-type: none">• Tốc độ > 1100km/h AND Chiều dài > 52m• Tốc độ > 1100km/h AND Chiều dài ≤ 52m• Tốc độ ≤ 1100km/h	<ul style="list-style-type: none">● Phân loại<ul style="list-style-type: none">Có thể là 01 trong các giá trị sau:<ul style="list-style-type: none">Dân sựQuân sựKhông xác định loại máy bay

■ Thiết kế chương trình

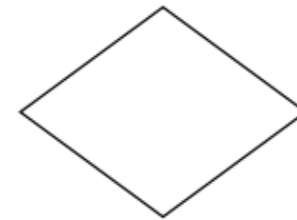
- Phát triển thuật toán
- Các công cụ sử dụng để thiết kế chương trình:
 - Sơ đồ khối - Flow Chart
 - Mã giả - Pseudo Code

- Thuật toán là hành động chính xác từng bước để thực hiện tất cả nhiệm vụ của chương trình.
- Có thể được biểu diễn bằng cách sử dụng sơ đồ khối hoặc mã giả

■ Các ký hiệu được sử dụng trong sơ đồ khối



Bắt đầu và Kết thúc



Điều kiện



Xử lý

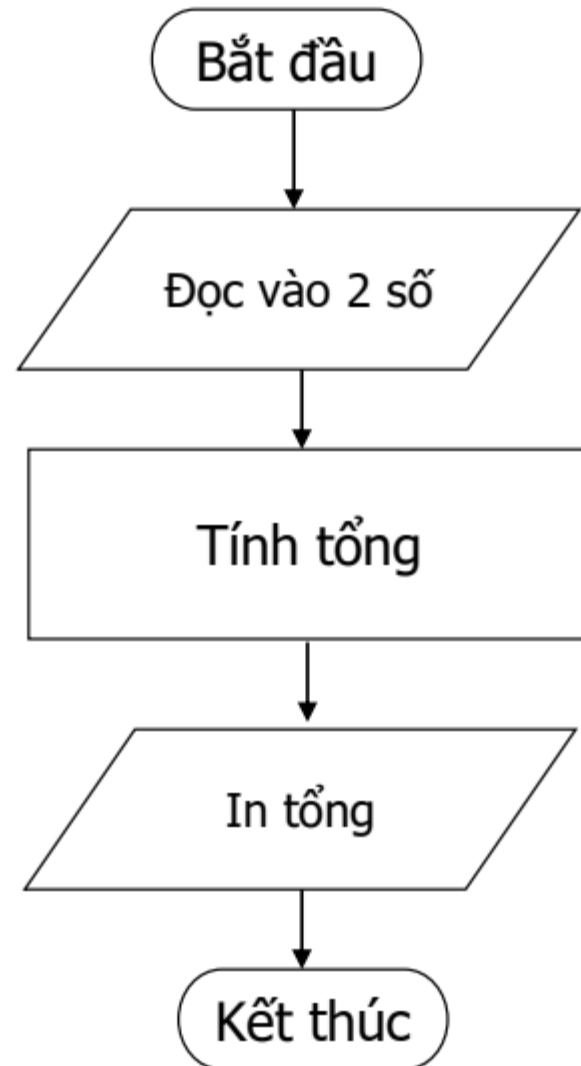


Thao tác Vào/Ra

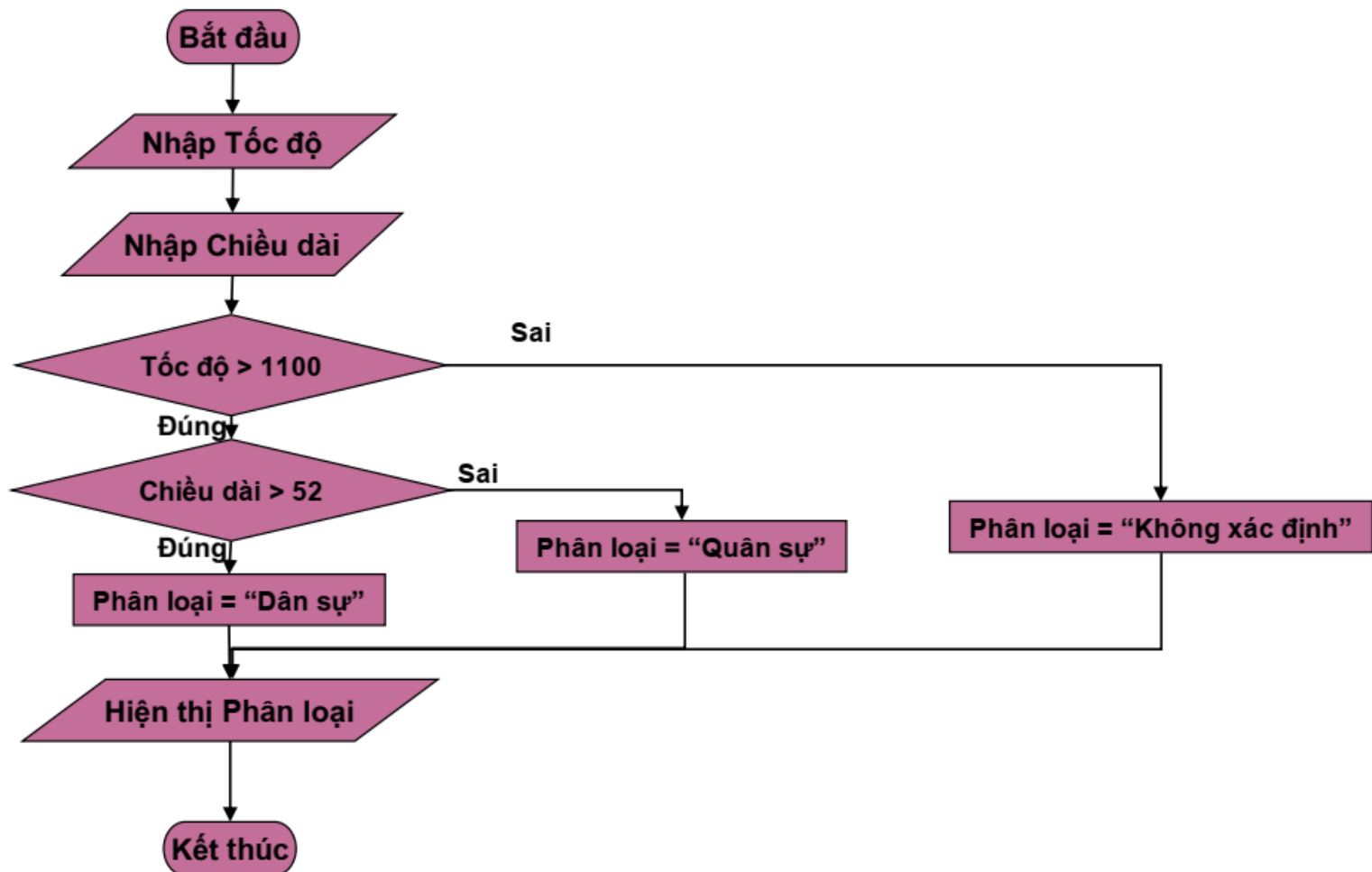


Hướng

■ Ví dụ (Flow Chart)



■ Sơ đồ khối



■ Kiểm tra thuật toán

Tốc độ	Chiều dài	Phân loại
1170	35	?
1180	56	?
900	66	?
800	34	?

■ Mã hoá

- Triển khai sơ đồ hoặc mã giả thành các quy tắc ngôn ngữ lập trình cụ thể (cú pháp)
 - Xác định yêu cầu lưu trữ
- Biên dịch
- Sửa lỗi
 - Lỗi cú pháp,
 - Lỗi logic hoặc
 - Lỗi runtime

■ Source Code

```
#include <iostream.h>

#include <string.h>

//program to classify the type of aircraft
main()
{
    int speed, length;
    char classification[30];

    cout << "Enter Observed speed : ";
    cin >> length;

    cout << "Enter estimated length : ";
    cin >> length;

    if (speed > 1100)
        if (length > 52)
            strcpy(classification, "civilian");
        else
            strcpy(classification, "Military");
    else
        strcpy(classification, "Aircraft type unknown");

    cout << "Classification is " << classification << endl;

    return 0;
} //end main()
```

■ Kiểm thử

- Chương trình phải được sửa các lỗi cú pháp
- Sử dụng một bộ dữ liệu thử nghiệm để đánh giá đầu ra.
- Chương trình phải tạo ra đầu vào hợp lệ và sinh ra đầu ra chính xác.
- Chương trình phải xử lý đầu vào không hợp lệ một cách hiệu quả.
- Chương trình có chấp nhận giá trị ngoài phạm vi không?

■ Tài liệu

- Hướng dẫn sử dụng
- Mô tả chương trình
- khả năng, giới hạn, hướng dẫn người dùng

- Ngôn ngữ định kiểu (types languages)
 - có hệ thống kiểu
 - cho phép kiểm tra sử dụng kiểu phù hợp mà không cần thực thi chương trình
 - kiểm tra tĩnh
- Ngôn ngữ định kiểu cho phép
 - phát hiện sớm một số lỗi liên quan đến kiểu
- Các ngôn ngữ định kiểu
 - C, Java, C++...

- Ưu điểm của hệ thống kiểu và kiểm tra
 - kiểu chặt chẽ
 - dễ kiểm tra
- Tuy nhiên
 - Hệ thống kiểu phải mềm dẻo trong sử dụng
 - Tính đa hình

■ Một số tình huống đa hình

- Viết hàm áp dụng cho các mảng có số phần tử khác nhau
 - kiểu mảng được kiểm tra khi biên dịch
 - số phần tử của mảng được kiểm tra khi thực thi
- Áp dụng hàm cho các kiểu dữ liệu khác nhau
- xây dựng nhiều phiên bản của hàm tương ứng với các kiểu khác nhau
- hoặc chỉ xây dựng một phiên bản của hàm, xử lý khác nhau được thực hiện khi thực thi
 - Template (C++), generic (Java)

- Xuất hiện vào những năm 70 đóng vai trò quan trọng để tạo ra phần mềm chất lượng
- Thiết kế hướng mô-đun
 - phần mềm = tập hợp các mô-đun và quan hệ giữa chúng
- Hầu hết các NNLT đều hỗ trợ mô-đun hóa

- Một mô-đun gồm hai phần
 - Phần giao diện
 - giao tiếp với bên ngoài mô-đun hay mô-đun khác
 - Phần thân
 - Nội dung của mô-đun
 - cục bộ đối với mỗi mô-đun, che dấu đối với mô-đun khác

- Các mô-đun chỉ trao đổi dữ liệu qua phần giao diện
 - không sử dụng biến toàn cục
- Nếu thay đổi phần thân thì ít ảnh hưởng (hoặc không ảnh hưởng) đến các mô-đun khác
- Trong ngôn ngữ lập trình cấu trúc
 - mô-đun = hàm
- Trong ngôn ngữ lập trình hướng đối tượng
 - mô-đun = lớp / phương thức

- Lập trình thủ tục/cấu trúc (procedural programming)
- Lập trình hướng đối tượng (object-oriented programming)
- Lập trình hàm (functional programming)
- Lập trình lô-gíc (logic programming)

Lập trình thủ tục

- Được sử dụng phổ biến
- Lập trình có cấu trúc
- phù hợp với thiết kế hướng chức năng
- NNLT thủ tục: Fortran, Ada, Pascal, C...

Lập trình hướng đối tượng



- Khái niệm cơ bản
 - đối tượng,
 - lớp
 - đóng gói
 - thừa kế
 - đa hình
- Xu hướng phát triển của các NNLT hiện đại
- NNLT hướng đối tượng: Smalltalk, C++, Java, Delphi...

- tính toán các biểu thức
- hàm tính toán dựa trên các giá trị của tham số
- thao tác trên danh sách
- áp dụng
 - lĩnh vực tính toán
 - trí tuệ nhân tạo
- NNLT hàm: LISP, Scheme...

- Thực hiện các biểu thức lô-gíc
 - khái niệm hợp giải (resolution)
 - tìm kiếm giá trị của các biến sao cho biểu thức lô-gíc có giá trị đúng
- Ứng dụng
 - xây dựng hệ chuyên gia
 - xử lý ngôn ngữ tự nhiên
- NNLT lô-gíc: Prolog

- quyết định quan trọng trong phát triển phần mềm
 - giảm chi phí
 - mã nguồn chất lượng
 - dễ bảo trì, phát triển

- dựa vào nhiều yếu tố (1)
 - Yêu cầu của khách hàng
 - Khách hàng tự bảo trì sản phẩm
 - Chương trình dịch
 - Cần có chương trình dịch có chất lượng tốt
 - Công cụ hỗ trợ
 - Dễ dàng quá trình lập trình, bảo trì
 - Kinh nghiệm của lập trình viên
 - Chọn NNLT mà lập trình làm chủ

- dựa vào nhiều yếu tố (2)
 - Yêu cầu tính khả chuyển (portability)
 - Thực hiện trên nhiều máy tính/platform khác nhau
 - Lĩnh vực ứng dụng
 - Hệ thống nhúng: C, Assembly...
 - Hệ thống quản lý: .NET, VB, C++...
 - Hệ chuyên gia: Prolog
 - Mạng: Java, .NET...
 - Website: PHP, ASP...
 - Không tồn tại ngôn ngữ đa năng cho mọi ứng dụng

Phong cách lập trình

- Cần có chương trình dễ hiểu
 - Phụ thuộc vào đặc điểm NNLT
 - Phong cách của người lập trình
- Phong cách lập trình không do lập trình viên tự đặt ra mà do tổ chức/doanh nghiệp/dự án đặt ra
 - Các luật lập trình
 - Các quy ước lập trình
- Mục đích
 - Mã nguồn dễ hiểu, dễ kiểm thử, dễ bảo trì
 - Ít lỗi

■ Một số nguyên tắc lập trình

- Đặt tên
 - Có ý nghĩa, gợi nhớ
- Trình bày
 - Rõ ràng, dễ hiểu
- Chú thích
 - Đầy đủ, dễ đọc
- Hạn chế sử dụng cấu trúc khó hiểu
 - break, continue, goto...
- ví dụ
 - quy ước lập trình C++

