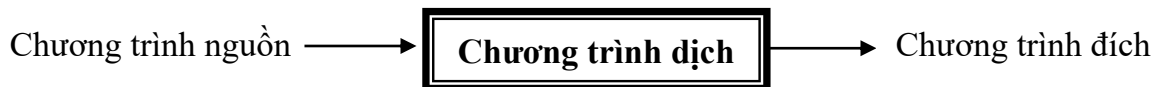


CHƯƠNG I: MỘT SỐ KHÁI NIỆM VỀ LẬP TRÌNH VÀ NGÔN NGỮ LẬP TRÌNH

BÀI 1: KHÁI NIỆM LẬP TRÌNH VÀ NGÔN NGỮ LẬP TRÌNH

Như chúng ta đã biết **ngôn ngữ lập trình** là ngôn ngữ để viết chương trình, là phương tiện dùng để diễn đạt cho máy tính những việc con người muốn máy tính thực hiện và nó được chia thành các lớp: *ngôn ngữ máy*, *hợp ngữ* và *ngôn ngữ bậc cao*.

- **Ngôn ngữ lập trình bậc cao** là ngôn ngữ lập trình gần với ngôn ngữ tự nhiên hơn, thuận tiện cho đông đảo người lập trình (không chỉ cho những người lập trình chuyên nghiệp).
- **Lập trình** là sử dụng *cấu trúc dữ liệu* và các *câu lệnh* của ngôn ngữ lập trình cụ thể để mô tả dữ liệu và diễn đạt các thao tác của thuật toán, là tạo ra các chương trình giải được các bài toán trên máy tính.
- Chương trình viết bằng ngôn ngữ lập trình bậc cao nói chung không phụ thuộc vào máy, nghĩa là một chương trình có thể thực hiện trên nhiều loại máy. Chương trình viết bằng ngôn ngữ máy có thể được nạp trực tiếp vào bộ nhớ và thực hiện ngay còn chương trình viết bằng ngôn ngữ lập trình bậc cao phải được chuyển đổi thành chương trình trên ngôn ngữ máy mới có thể thực hiện được.
- Chương trình đặc biệt có chức năng chuyển đổi chương trình được viết bằng ngôn ngữ lập trình bậc cao thành chương trình thực hiện trên máy tính cụ thể được gọi là chương trình dịch.
- **Chương trình dịch** nhận đầu vào là chương trình viết bằng *ngôn ngữ lập trình bậc cao* (chương trình nguồn) và thực hiện chuyển đổi sang *ngôn ngữ máy* (chương trình đích):



- Ngôn ngữ máy là ngôn ngữ duy nhất máy tính điện tử có thể trực tiếp hiểu và thực hiện các *câu lệnh*.
 - Chương trình dịch có 2 loại: thông dịch và biên dịch
- a) Thông dịch** (interpreter) được thực hiện bằng cách lặp lại dãy các bước sau:
- ① Kiểm tra tính đúng đắn của *câu lệnh* tiếp theo trong chương trình nguồn;
 - ② Chuyển đổi *câu lệnh* đó thành một hay nhiều *câu lệnh* tương ứng trong *ngôn ngữ máy*;
 - ③ Thực hiện các *câu lệnh* vừa chuyển đổi được.

Như vậy, quá trình dịch và thực hiện các *câu lệnh* là luân phiên. Các chương trình *thông dịch* lần lượt dịch và thực hiện từng câu lệnh. Nó thích hợp cho môi trường đối thoại giữa người và hệ thống, được ứng dụng cho các ngôn ngữ khai thác hệ quản trị cơ sở dữ liệu, ngôn ngữ đối thoại với hệ điều hành, ... Tuy nhiên, nếu một *câu lệnh* nào đó phải thực hiện bao nhiêu lần thì nó phải được dịch bấy nhiêu lần.

b) Biên dịch (compiler) được thực hiện qua hai bước:

- ① Duyệt, phát hiện lỗi, kiểm tra tính đúng đắn của các *câu lệnh* trong chương trình nguồn;
- ② Dịch toàn bộ chương trình nguồn thành một chương trình đích có thể thực hiện trên máy và có thể lưu trữ để sử dụng lại khi cần thiết.

Như vậy, trong thông dịch, không có chương trình đích để lưu trữ, còn trong biên dịch thì cả chương trình nguồn và chương trình đích đều có thể lưu trữ lại để sử dụng về sau. Nó còn được ứng dụng vào việc biên soạn, lưu trữ, tìm kiếm, cho biết các kết quả trung gian, ... Toàn bộ các dịch vụ trên tạo thành một môi trường làm việc trên một ngôn ngữ lập trình cụ thể. *Ví dụ* Turbo Pascal 7.0, Free Pascal 1.2... là các môi trường làm việc trên ngôn ngữ Pascal.

BÀI 2: CÁC THÀNH PHẦN CỦA NGÔN NGỮ LẬP TRÌNH

1. Các thành phần cơ bản

Mỗi ngôn ngữ lập trình thường có **ba** thành phần cơ bản, đó là: **bảng chữ cái, cú pháp, ngữ nghĩa.**

a) **Bảng chữ cái** là tập các kí tự được dùng để viết chương trình. Không được phép dùng bất kì kí tự nào ngoài các kí tự qui định trong bảng chữ cái.

Trong Pascal, bảng chữ cái bao gồm các kí tự:

▪ Các chữ cái thường và các chữ cái in hoa của bảng chữ cái tiếng Anh:

a b c d e f g h i j k l m n o p q r s t u v w x y z

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

▪ 10 chữ số thập phân Ả Rập: 0 1 2 3 4 5 6 7 8 9

▪ Các kí tự đặc biệt:

+	-	*	/	=	<	>	[]	.	, (dấu phẩy)
;	#	^	\$	@	&	()	{	}	: ' (dấu nháy đơn)
dấu cách (mã ASCII 32)										_ (dấu gạch dưới)

• Bảng chữ cái của các ngôn ngữ lập trình nói chung không khác nhau nhiều.

b) **Cú pháp** là bộ qui tắc để viết chương trình, mà dựa vào chúng, người lập trình và chương trình dịch biết được tổ hợp nào của các kí tự trong bảng chữ cái là hợp lệ. Nhờ đó, có thể mô tả chính xác thuật toán để máy thực hiện.

c) **Ngữ nghĩa** xác định ý nghĩa thao tác cần phải thực hiện, ứng với tổ hợp kí tự dựa vào ngữ cảnh của nó.

Tóm lại, cú pháp cho biết cách viết một chương trình hợp lệ, còn ngữ nghĩa xác định ý nghĩa của các tổ hợp kí tự trong chương trình.

2. Một số khái niệm

a) Tên

▪ Mọi đối tượng trong chương trình đều phải được đặt tên theo qui tắc của ngôn ngữ lập trình và từng chương trình dịch cụ thể.

▪ **Qui tắc đặt tên** trong Turbo Pascal: tên là một dãy liên tiếp **không quá 127 kí tự bao gồm chữ số, chữ cái hoặc dấu gạch dưới và bắt đầu bằng chữ cái hoặc dấu gạch dưới.**

Ví dụ, trong ngôn ngữ Pascal

Các tên đúng:	Các tên sai:
R21	A BC (chứa dấu cách)
NTT74_c	1Pq (bắt đầu bằng chữ số)
_42	X#Y (chứa kí tự # không hợp lệ)

▪ Nhiều ngôn ngữ lập trình, trong đó có Pascal, phân biệt **ba loại tên**, đó là:

- Tên dành riêng (*từ khóa*);
- Tên chuẩn;
- Tên do người lập trình đặt.

- **Tên dành riêng:** tên được dùng với ý nghĩa xác định, người lập trình không được sử dụng với ý nghĩa khác và chúng còn được gọi là *từ khóa*.
+ *Ví dụ*, một số tên dành riêng: **program, uses, const, type, var, begin, end.**
- **Tên chuẩn:** tên dùng với ý nghĩa nhất định nào đó, nhưng người lập trình có thể khai báo và dùng chúng với ý nghĩa và mục đích khác. Ý nghĩa của chúng được qui định trong các thư viện của ngôn ngữ lập trình.
+ *Ví dụ*, tên chuẩn: **abs, sqr, sqrt, byte, integer, longint, real, break ...**
- **Tên do người lập trình đặt:** tên được dùng với ý nghĩa riêng, xác định bằng cách khai báo trước khi sử dụng và chúng không được trùng với tên dành riêng (*từ khóa*).
+ *Ví dụ:* tên do người lập trình tự đặt: **A1, bt1, bai_thi, ...**

Lưu ý: tên do người lập trình tự đặt phải tuân thủ **qui tắc đặt tên** trong Pascal

b) Hằng và biến

- **Hằng** là đại lượng có giá trị không thay đổi trong quá trình thực hiện chương trình.
Trong các ngôn ngữ lập trình thường có các hằng số học, hằng logic, hằng xâu
 - **Hằng số học** là các số nguyên hay số thực (dấu phẩy tĩnh hoặc dấu phẩy động) có dấu hoặc không dấu. *Ví dụ:* 2, 0, -5, +18, 1.5, 1.0E-6 ...
 - **Hằng logic** là giá trị **đúng** hoặc **sai** tương ứng với true hoặc false.
Ví dụ, hằng logic trong Pascal: **true, false**
 - **Hằng xâu** là dãy kí tự trong bộ mã ASCII. Khi viết, dãy kí tự này được đặt trong cặp **dấu nháy** (ngầm hiểu là **dấu nháy đơn** do Pascal dùng dấu nháy đơn).
Ví dụ, hằng xâu trong Pascal: 'Lop 11A1' 'Information'
- **Biến** là đại lượng được đặt tên, dùng để lưu trữ giá trị và giá trị có thể được thay đổi trong quá trình thực hiện chương trình. Các biến dùng trong chương trình đều phải được khai báo.

c) Chú thích

Chúng ta có thể đặt các đoạn chú thích trong chương trình nguồn. Chúng giúp cho người đọc chương trình nhận biết ngữ nghĩa của chương trình đó dễ hơn. Nó không ảnh hưởng đến nội dung chương trình nguồn và được chương trình dịch bỏ qua.

Trong Pascal các đoạn chú thích được đặt giữa cặp dấu { và } hoặc (* và *)

TÓM TẮT CHƯƠNG I

- Cần có chương trình dịch để chuyển chương trình nguồn thành chương trình đích.

- Có hai loại chương trình dịch: *thông dịch* và *biên dịch*.
- Các thành phần của ngôn ngữ lập trình: bảng chữ cái, cú pháp và ngữ nghĩa.
- Mọi đối tượng trong chương trình đều phải được đặt tên:
 - Tên dành riêng (*từ khóa*): được dùng với ý nghĩa riêng, không được dùng với ý nghĩa khác.
 - Tên chuẩn: tên dùng với ý nghĩa nhất định, khi cần dùng với ý nghĩa khác thì phải khai báo.
 - Tên do người lập trình (tự) đặt: cần khai báo trước khi sử dụng.
- Hằng: đại lượng có giá trị **không thay đổi** trong quá trình thực hiện chương trình.
- Biến: đại lượng được **đặt tên**. Giá trị của biến có thể **thay đổi** trong quá trình thực hiện chương trình.