

## §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 là *bảng chữ cái*, *cú pháp* và *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 quy định trong bảng chữ cái.

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

- 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 cách (mã ASCII 32)										-	

Bảng chữ cái của các ngôn ngữ lập trình nói chung không khác nhau nhiều. Ví dụ, bảng chữ cái của ngôn ngữ lập trình C++ chỉ khác Pascal là có sử dụng thêm các kí tự như dấu nháy kép ("), dấu số ngược (\), dấu chấm than (!).

*b) Cú pháp* là bộ quy tắc để viết chương trình. 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 và tổ hợp nào là không 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.

### Ví dụ

Phần lớn các ngôn ngữ lập trình đều sử dụng dấu cộng (+) để chỉ phép cộng. Xét các biểu thức:

$$A + B \quad (1)$$

$$I + J \quad (2)$$

Giả thiết  $A, B$  là các đại lượng nhận giá trị thực và  $I, J$  là các đại lượng nhận giá trị nguyên. Khi đó dấu "+" trong biểu thức (1) được hiểu là cộng hai số thực, dấu "+" trong biểu thức (2) được hiểu là cộng hai số nguyên. Như vậy, ngữ nghĩa dấu "+" trong hai ngữ cảnh khác nhau là khác nhau.

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.

Các lỗi cú pháp được chương trình dịch phát hiện và thông báo cho người lập trình biết. Chỉ có các chương trình không còn lỗi cú pháp mới dịch được sang ngôn ngữ máy.

Các lỗi ngữ nghĩa khó phát hiện hơn. Phần lớn các lỗi ngữ nghĩa chỉ được phát hiện khi thực hiện chương trình trên dữ liệu cụ thể.

## 2. Một số khái niệm

### a) Tên

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

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*. Trong chương trình dịch Free Pascal, tên có thể có độ dài tới 255 kí tự.

Ví dụ, trong ngôn ngữ Pascal:

- Các tên đúng: A

R21

P21\_c

\_45

- Các tên sai:

A BC (chứa dấu cách)

6Pq (bắt đầu bằng chữ số)

X#Y (chứa kí tự "#" không hợp lệ)

Ngôn ngữ Pascal không phân biệt chữ hoa, chữ thường trong tên. Một số ngôn ngữ lập trình khác (ví dụ như C++) phân biệt chữ hoa, chữ thường. Ví dụ, *AB* và *Ab* là một tên trong Pascal, nhưng lại là hai tên khác nhau trong C++.

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

- Tên dành riêng;
- Tên chuẩn;
- Tên do người lập trình đặt.

### Tên dành riêng

Một số tên được ngôn ngữ lập trình quy định dùng với *ý nghĩa riêng xác định*, người lập trình không được sử dụng với ý nghĩa khác. Những tên này được gọi là *tên dành riêng* (còn được gọi là *từ khoá*).

**Ví dụ.** Một số tên dành riêng:

Trong Pascal: `program, uses, const, type, var, begin, end.`

Trong C++: `main, include, if, while, void.`

### Tên chuẩn

Một số tên được ngôn ngữ lập trình dùng với *ý nghĩa nhất định nào đó*. Những tên này được gọi là *tên chuẩn*. Tuy nhiên, người lập trình có thể khai báo và dùng chúng với ý nghĩa và mục đích khác.

Ý nghĩa của các tên chuẩn được quy định trong các *thư viện* của ngôn ngữ lập trình.

**Ví dụ.** Một số tên chuẩn

– Trong Pascal:

<code>abs</code>	<code>integer</code>	<code>real</code>
<code>sqrt</code>	<code>longint</code>	<code>extended</code>
<code>sqr</code>	<code>byte</code>	<code>break</code>

– Trong C++:

<code>cin</code>	<code>cout</code>	<code>getchar</code>
------------------	-------------------	----------------------

### Tên do người lập trình đặt

Tên do người lập trình đặt được dùng với ý nghĩa riêng, xác định bằng cách khai báo trước khi sử dụng. Các tên này không được trùng với tên dành riêng.

### **Ví dụ**

Tên do người lập trình đặt:

```
AI
DELTA
CT_Vidu
```

### **b) Hằng và biến**

#### **Hằng**

*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).
- Hằng logic là giá trị *đúng* hoặc *sai* tương ứng với *true* hoặc *false*.
- Hằng xâu là dãy kí tự trong bộ mã ASCII. Khi viết, dãy kí tự này được đặt trong cặp dấu nháy (Pascal dùng dấu nháy đơn, còn C++ dùng dấu nháy kép).

#### **Ví dụ**

– Hằng số học:    2                    0                    -5                    +18  
                         1.5                    -22.36                    +3.14159                    0.5  
                         -2.236E01                    1.0E-6

– Hằng logic:

+ Trong Pascal:    TRUE                    FALSE

– Hằng xâu:

+ Trong Pascal:    'Information'                    'Lop 11A'

+ Trong C++:        "Information"                    "Lop 11A"

**Chú ý:** Hằng dấu nháy đơn trong Pascal được viết là `'`. Để có xâu tiếng Anh *I'm a student*, trong Pascal cần viết là `'I'm a student'`.

#### **Biến**

*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.*

Tùy theo cách lưu trữ và xử lí, Pascal phân biệt nhiều loại biến. Các biến dùng trong chương trình đều phải khai báo. Việc khai báo biến sẽ được trình bày ở các phần sau.

### c) Chú thích

Có thể đặt các đoạn chú thích trong chương trình nguồn. Các chú thích này giúp cho người đọc chương trình nhận biết ý nghĩa của chương trình đó dễ hơn. Chú thích 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 \*). Một trong những cách tạo chú thích trong C++ là đặt chúng giữa cặp dấu /\* và \*/.

## TÓM TẮT

- 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: 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 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.

## CÂU HỎI VÀ BÀI TẬP

1. Tại sao người ta phải xây dựng các ngôn ngữ lập trình bậc cao?
2. Chương trình dịch là gì? Tại sao cần phải có chương trình dịch?
3. Biên dịch và thông dịch khác nhau như thế nào?
4. Hãy cho biết các điểm khác nhau giữa tên dành riêng và tên chuẩn.
5. Hãy tự viết ra ba tên đúng theo quy tắc của Pascal.
6. Hãy cho biết những biểu diễn nào dưới đây không phải là biểu diễn hằng trong Pascal và chỉ rõ lỗi trong từng trường hợp:
  - a) 150.0                      b) -22                      c) 6,23                      d) '43'
  - e) A20                      f) 1.06E-15                      g) 4+6                      h) 'C
  - i) 'TRUE'