

§6. PHÉP TOÁN, BIỂU THỨC, CÂU LỆNH GÁN

Để mô tả các thao tác trong thuật toán, mỗi ngôn ngữ lập trình đều xác định và sử dụng một số khái niệm cơ bản: phép toán, biểu thức, gán giá trị cho biến.

Dưới đây sẽ xét các khái niệm đó trong Pascal.

1. Phép toán

Tương tự trong toán học, trong các ngôn ngữ lập trình đều có những phép toán số học như cộng, trừ, nhân, chia trên các đại lượng thực, các phép toán chia nguyên và lấy phần dư, các phép toán quan hệ,...

Bảng dưới đây là kí hiệu các phép toán đó trong toán và trong Pascal:

| Phép toán | Trong toán học | Trong Pascal |
|------------------------------------|--|---------------------|
| Các phép toán số học với số nguyên | + (cộng), - (trừ), × (nhân), div (chia nguyên), mod (lấy phần dư) | +, -, *, div, mod |
| Các phép toán số học với số thực | + (cộng), - (trừ), × (nhân), : (chia) | +, -, *, / |
| Các phép toán quan hệ | < (nhỏ hơn), ≤ (nhỏ hơn hoặc bằng), > (lớn hơn), ≥ (lớn hơn hoặc bằng), = (bằng), ≠ (khác) | <, <=, >, >=, =, <> |
| Các phép toán logic | ¬ (phủ định), ∨ (hoặc), ∧ (và) | not, or, and |

Chú ý: – Kết quả của các phép toán quan hệ cho giá trị logic.

– Một trong những ứng dụng của phép toán logic là để tạo ra các biểu thức phức tạp từ các quan hệ đơn giản.

2. Biểu thức số học

Trong lập trình, biểu thức số học là một biến kiểu số hoặc một hằng số hoặc các biến kiểu số và các hằng số liên kết với nhau bởi một số hữu hạn phép toán số học, các dấu ngoặc tròn (và) tạo thành một biểu thức có dạng tương tự như cách viết trong toán học với những quy tắc sau:

- Chỉ dùng cặp ngoặc tròn để xác định trình tự thực hiện phép toán trong trường hợp cần thiết;
- Viết lần lượt từ trái qua phải;
- Không được bỏ qua dấu nhân (*) trong tích.

Các phép toán được thực hiện theo thứ tự:

- Thực hiện các phép toán trong ngoặc trước;
- Trong dãy các phép toán không chứa ngoặc thì thực hiện từ trái sang phải, theo thứ tự các phép toán nhân (*), chia (/), chia nguyên (div), lấy phần dư (mod) thực hiện trước và các phép toán cộng (+), trừ (-) thực hiện sau.

Ví dụ

| Biểu thức trong toán học | Biểu thức trong Pascal |
|--|-------------------------------|
| $5a+6b$ | $5*a + 6*b$ |
| $\frac{xy}{z}$ | $x*y/z$ |
| $Ax^2 + Bx + C$ | $A*x*x + B*x + C$ |
| $\frac{x+y}{x-\frac{1}{2}} - \frac{x-z}{xy}$ | $(x+y)/(x-1/2) - (x-z)/(x*y)$ |

- Chú ý:**
- Nếu biểu thức chứa một hằng hay biến kiểu thực thì ta có biểu thức số học thực, giá trị của biểu thức cũng thuộc kiểu thực.
 - Trong một số trường hợp nên dùng biến trung gian để có thể tránh được việc tính một biểu thức nhiều lần.

3. Hàm số học chuẩn

Để lập trình được dễ dàng, thuận tiện hơn, các ngôn ngữ lập trình đều có thư viện chứa một số chương trình tính giá trị những hàm toán học thường dùng. Các chương trình như vậy được gọi là các *hàm số học chuẩn*. Mỗi hàm chuẩn có tên chuẩn riêng. Đối số của hàm là một hay nhiều biểu thức số học và được đặt trong cặp ngoặc tròn (và) sau tên hàm. Bản thân hàm chuẩn cũng được coi là một biểu thức số học và nó có thể tham gia vào biểu thức số học như một toán hạng (giống như biến và hằng). Kết quả của hàm có thể là nguyên hoặc thực hay phụ thuộc vào kiểu của đối số.

Bảng dưới đây cho biết một số hàm chuẩn thường dùng.

| Hàm | Biểu diễn toán học | Biểu diễn trong Pascal | Kiểu đối số | Kiểu kết quả |
|-------------------|--------------------|------------------------|------------------|----------------------|
| Bình phương | x^2 | sqr(x) | Thực hoặc nguyên | Theo kiểu của đối số |
| Căn bậc hai | \sqrt{x} | sqr(x) | Thực hoặc nguyên | Thực |
| Giá trị tuyệt đối | $ x $ | abs(x) | Thực hoặc nguyên | Theo kiểu của đối số |
| Lôgarit tự nhiên | $\ln x$ | ln(x) | Thực | Thực |
| Lũy thừa của số e | e^x | exp(x) | Thực | Thực |
| Sin | $\sin x$ | sin(x) | Thực | Thực |
| Cos | $\cos x$ | cos(x) | Thực | Thực |

Ví dụ

Biểu thức toán học $\frac{-b + \sqrt{b^2 - 4ac}}{2a}$ trong Pascal có thể viết dưới dạng:

$$(-b + \text{sqr}(b*b - 4*a*c)) / (2*a)$$

hoặc

$$(-b + \text{sqr}(\text{sqr}(b) - 4*a*c)) / 2/a$$

Ngoài những hàm số học chuẩn trên, còn có các hàm chuẩn khác được giới thiệu trong những phần sau.

4. Biểu thức quan hệ

Hai biểu thức cùng kiểu liên kết với nhau bởi phép toán quan hệ cho ta một biểu thức quan hệ.

Biểu thức quan hệ có dạng:

$$\langle \text{biểu thức 1} \rangle \langle \text{phép toán quan hệ} \rangle \langle \text{biểu thức 2} \rangle$$

trong đó, *biểu thức 1* và *biểu thức 2* cùng là xâu hoặc cùng là biểu thức số học.

Ví dụ

$$x < 5$$

$$i+1 \geq 2*j$$

Biểu thức quan hệ được thực hiện theo trình tự:

- Tính giá trị các biểu thức.
- Thực hiện phép toán quan hệ.

Kết quả của biểu thức quan hệ là giá trị logic: *true* (đúng) hoặc *false* (sai).

Trong ví dụ trên, nếu x có giá trị 3, thì biểu thức $x < 5$ có giá trị *true*. Nếu i có giá trị 2 và j có giá trị 3 thì biểu thức $i + 1 \geq 2*j$ sẽ cho giá trị *false*.

Ví dụ

Điều kiện để điểm M có tọa độ (x, y) thuộc hình tròn tâm $I(a, b)$, bán kính R là:

$$\text{sqrt}((x-a)*(x-a) + (y-b)*(y-b)) \leq R$$

hoặc

$$\text{sqrt}(x-a) + \text{sqrt}(y-b) \leq R*R$$

5. Biểu thức logic

Biểu thức logic đơn giản là biến logic hoặc hằng logic.

Biểu thức logic là các biểu thức logic đơn giản, các biểu thức quan hệ liên kết với nhau bởi phép toán logic. Giá trị biểu thức logic là *true* hoặc *false* (xem phụ lục A). Các biểu thức quan hệ thường được đặt trong cặp ngoặc (và).

Phép toán **not** được viết trước biểu thức cần phủ định, ví dụ:

not $(x < 1)$ thể hiện phát biểu "x không nhỏ hơn 1" và điều này tương đương với biểu thức quan hệ $x \geq 1$.

Các phép toán **and** và **or** dùng để kết hợp nhiều biểu thức logic hoặc quan hệ thành một biểu thức, thường được dùng để diễn tả các điều kiện phức tạp.

Ví dụ 1

Để thể hiện điều kiện $5 \leq x \leq 11$, trong Pascal cần phải tách thành phát biểu dưới dạng " $5 \leq x$ và $x \leq 11$ ":

`(5 <= x) and (x <= 11)`

Ví dụ 2

Giả thiết M và N là hai biến nguyên. Điều kiện xác định M và N đồng thời chia hết cho 3 hay đồng thời không chia hết cho 3 được thể hiện trong Pascal như sau:

`((M mod 3 = 0) and (N mod 3 = 0)) or ((M mod 3 <> 0) and (N mod 3 <> 0))`

6. Câu lệnh gán

Lệnh gán là một trong những lệnh cơ bản nhất của các ngôn ngữ lập trình.

Trong Pascal câu lệnh gán có dạng:

`<tên biến> := <biểu thức>;`

Trong trường hợp đơn giản, *tên biến* là tên của biến đơn. Kiểu của giá trị biểu thức phải phù hợp với kiểu của biến.

Chức năng của lệnh gán là đặt cho biến có tên ở vế trái dấu "=" giá trị mới bằng giá trị của biểu thức ở vế phải.

Ví dụ

```
x1:= (-b - sqrt(b*b - 4*a*c))/(2*a);  
x2:= -b/a - x1;  
z:= z - 1;  
i:= i + 1;
```

Trong ví dụ trên, ý nghĩa của lệnh gán thứ ba là giảm giá trị của biến z một đơn vị. Ý nghĩa của lệnh gán thứ tư là tăng giá trị của biến i lên một đơn vị.