

§9. CẤU TRÚC RỄ NHÁNH

1. Rễ nhánh

Thường ngày, có rất nhiều việc chỉ được thực hiện khi một điều kiện cụ thể nào đó được thoả mãn.

Ví dụ, Châu và Ngọc thường cùng nhau chuẩn bị các bài thực hành môn Tin học.

Một lần Châu hẹn với Ngọc: "*Chiều mai nếu trời không mưa thì Châu sẽ đến nhà Ngọc*".

Một lần khác, Ngọc nói với Châu: "*Chiều mai nếu trời không mưa thì Ngọc sẽ đến nhà Châu, nếu mưa thì sẽ gọi điện cho Châu để trao đổi*".

Câu nói của Châu cho ta biết một việc làm cụ thể (Châu đến nhà Ngọc) sẽ được thực hiện nếu một điều kiện cụ thể (trời không mưa) thoả mãn. Ngoài ra không đề cập đến việc gì sẽ xảy ra nếu điều kiện đó không thoả mãn (trời mưa).

Ta nói cách diễn đạt như vậy thuộc dạng thiếu:

Nếu... thì...

Câu nói của Ngọc khẳng định một trong hai việc cụ thể (Ngọc đến nhà Châu hay Ngọc gọi điện cho Châu) chắc chắn sẽ xảy ra. Tuy nhiên, việc nào trong hai việc sẽ được thực hiện thì tùy thuộc vào điều kiện cụ thể (trời không mưa) thoả mãn hay không.

Ta nói cách diễn đạt như vậy thuộc dạng đủ:

Nếu... thì..., nếu không thì...

Từ đó có thể thấy, trong nhiều thuật toán, các thao tác tiếp theo sẽ phụ thuộc vào kết quả nhận được từ các bước trước đó.

Cấu trúc dùng để mô tả các mệnh đề có dạng như trên được gọi là cấu trúc rẽ nhánh thiếu và đủ.

Ví dụ, để giải phương trình bậc hai:

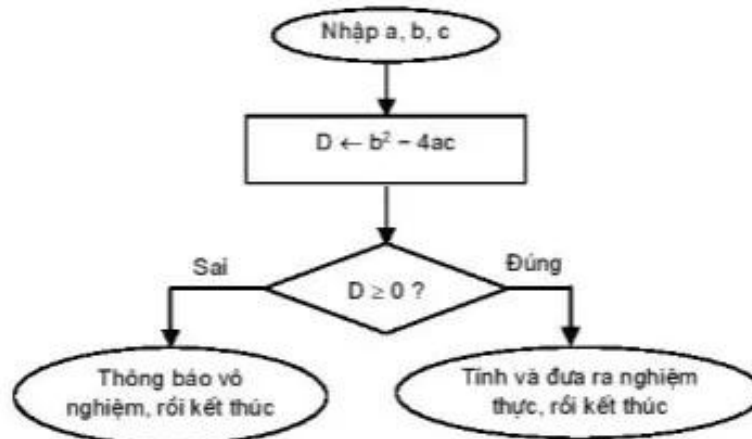
$$ax^2 + bx + c = 0, (a \neq 0)$$

trước tiên, ta tính biệt số delta $D = b^2 - 4ac$.

Nếu D không âm, ta sẽ đưa ra các nghiệm. Trong trường hợp ngược lại, ta phải thông báo là phương trình vô nghiệm.

Như vậy, sau khi tính D , tùy thuộc vào giá trị của D , một trong hai thao tác sẽ được thực hiện (h. 4).

Mọi ngôn ngữ lập trình đều có các câu lệnh để mô tả cấu trúc rẽ nhánh.



Hình 4. Sơ đồ thể hiện cấu trúc rẽ nhánh

2. Câu lệnh if-then

Để mô tả cấu trúc rẽ nhánh, Pascal dùng câu lệnh *if-then*. Tương ứng với hai dạng thiếu và đủ nói ở trên, Pascal có hai dạng câu lệnh *if-then*:

a) Dạng thiếu

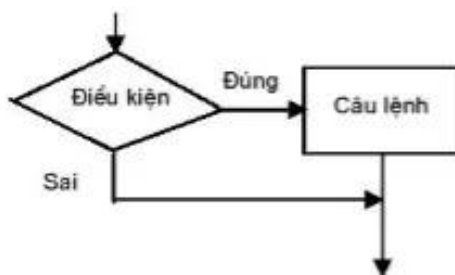
`if <điều kiện> then <câu lệnh >;`

b) Dạng đủ

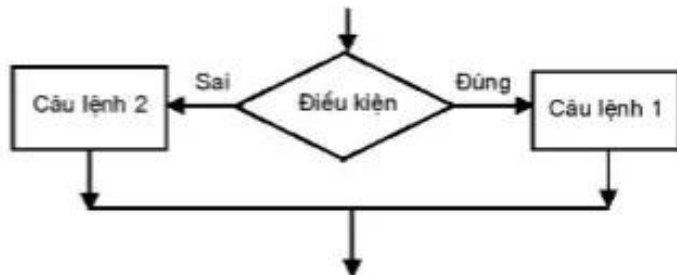
`if <điều kiện> then <câu lệnh 1> else <câu lệnh 2>;`

trong đó:

- Điều kiện là biểu thức logic.
- Câu lệnh, câu lệnh 1, câu lệnh 2 là một câu lệnh của Pascal.



Hình 5



Hình 6

Ở dạng thiếu: *điều kiện* sẽ được tính và kiểm tra. Nếu *điều kiện* đúng (có giá trị *true*) thì *câu lệnh* sẽ được thực hiện, ngược lại thì *câu lệnh* sẽ bị bỏ qua (h. 5).

Ở dạng đủ: *điều kiện* cũng được tính và kiểm tra. Nếu *điều kiện* đúng thì *câu lệnh 1* sẽ được thực hiện, ngược lại thì *câu lệnh 2* sẽ được thực hiện (h. 6).

Ví dụ 1

```
if D < 0 then writeln('Phuong trinh vo nghiem.');
```

Ví dụ 2

```
if a mod 3 = 0 then write('a chia het cho 3')
else write('a khong chia het cho 3');
```

Ví dụ 3. Để tìm số lớn nhất *max* trong hai số *a* và *b*, có thể thực hiện bằng hai cách sau:

– Dùng câu lệnh gán *max:= a* và lệnh *if-then* dạng thiếu:

```
if b > a then max:=b;
```

– Dùng một lệnh *if-then* dạng đủ:

```
if b > a then max:=b else max:=a;
```

3. Câu lệnh ghép

Theo cú pháp, sau một số từ khoá (như *then* hoặc *else*) phải là một câu lệnh. Nhưng trong nhiều trường hợp, các thao tác sau những tên dành riêng đó khá phức tạp, đòi hỏi không phải chỉ một mà là nhiều câu lệnh để mô tả. Trong các trường hợp như vậy, ngôn ngữ lập trình cho phép gộp một dãy câu lệnh thành một *câu lệnh ghép* (hay câu lệnh hợp thành). Chẳng hạn, câu lệnh ghép của Pascal có dạng:

```
begin
    <các câu lệnh>;
end;
```

Câu lệnh, câu lệnh 1, câu lệnh 2 trong các câu lệnh *if-then* ở mục trên có thể là câu lệnh ghép.

Thuật ngữ *câu lệnh* được hiểu chung cho cả câu lệnh đơn và câu lệnh ghép.

Ví dụ

```
if D < 0 then writeln('Phuong trinh vo nghiem.')
else
    begin
        x1:= (-b - sqrt(b*b - 4*a*c))/(2*a);
        x2:= -b/a-x1;
    end;
```

4. Một số ví dụ

Ví dụ 1. Tìm nghiệm thực của phương trình bậc hai:

$$ax^2 + bx + c = 0, \text{ với } a \neq 0.$$

Input: Các hệ số a, b, c nhập từ bàn phím.

Output: Đưa ra màn hình các nghiệm thực hoặc thông báo "*Phương trình vô nghiệm*".

```
program Giai_PT2;  
uses crt;  
var a,b,c: real;  
    D, x1, x2: real;  
begin  
  clrscr;  
  write(' a, b, c: ');  
  readln(a, b, c);  
  D:= b*b - 4*a*c;  
  if D < 0 then writeln('Phương trình vô nghiệm.')  
    else  
      begin  
        x1:= (-b - sqrt(D))/(2*a);  
        x2:= -b/a - x1;  
        writeln(' x1 = ', x1:8:3, ' x2 = ', x2:8:3);  
      end;  
  readln  
end.
```

Ví dụ 2

Tìm số ngày của năm N , biết rằng năm nhuận là năm chia hết cho 400 hoặc chia hết cho 4 nhưng không chia hết cho 100. Ví dụ, các năm 2000, 2004 là năm nhuận và có số ngày là 366, các năm 1900, 1945 không phải là năm nhuận và có số ngày là 365.

Input: N nhập từ bàn phím.

Output: Đưa số ngày của năm N ra màn hình.

```
program Nam_nhuan;  
uses crt;  
var N, SN: integer;  
begin  
  clrscr;  
  write('Nam: '); readln(N);  
  if (N mod 400 = 0) or ((N mod 4 = 0) and (N mod 100 <> 0))  
    then SN:= 366 else SN:= 365;  
  writeln(' So ngay cua nam ', N, ' la ', SN);  
  readln  
end.
```