

§18. VÍ DỤ VỀ CÁCH VIẾT VÀ SỬ DỤNG CHƯƠNG TRÌNH CON

Các ngôn ngữ lập trình đều có các quy tắc viết và sử dụng chương trình con. Trong mục này, ta xét cách viết và sử dụng chương trình con trong Pascal.

1. Cách viết và sử dụng thủ tục

Xét ví dụ vẽ hình chữ nhật có dạng sau:

```
* * * * *
*           *
* * * * *
```

Ta có thể vẽ hình chữ nhật trên với ba câu lệnh:

```
writeln('* * * * *');
writeln('*           *');
writeln('* * * * *');
```

Như vậy, trong một chương trình, mỗi khi cần vẽ một hình chữ nhật như trên, ta cần phải đưa vào ba câu lệnh này.

Trong chương trình sau, ta đưa ba câu lệnh trên vào một thủ tục có tên là *Ve_Hcn* (vẽ hình chữ nhật). Mỗi khi cần vẽ một hình chữ nhật ta đưa vào một *câu lệnh gọi* thủ tục đó. Chương trình *VD_thutucl* gọi thủ tục *Ve_Hcn* ba lần để vẽ ba hình chữ nhật.

```
program VD_thutucl;
procedure Ve_Hcn; {Bat dau thu tuc}
begin
  writeln('* * * * *');
  writeln('*           *');
  writeln('* * * * *');
end; {Ket thuc thu tuc}
begin
  Ve_Hcn; {goi thu tuc Ve_Hcn}
```

```
writeln;writeln;      {de cach 2 dong}
Ve_Hcn;               {goi thu tuc Ve_Hcn}
writeln;writeln;
Ve_Hcn;               {goi thu tuc Ve_Hcn}
end.
```

a) Cấu trúc của thủ tục

Thủ tục có cấu trúc như sau:

```
procedure <tên thủ tục>[(<danh sách tham số>)];
[<phần khai báo>]
begin
    [<dãy các lệnh>]
end;
```

Phần đầu thủ tục gồm tên dành riêng **procedure**, tiếp theo là tên thủ tục. Danh sách tham số có thể có hoặc không có.

Phần khai báo dùng để xác định các hằng, kiểu, biến và cũng có thể xác định các chương trình con khác được sử dụng trong thủ tục. Thủ tục *Ve_Hcn* ở trên không khai báo hằng, biến hay chương trình con nào.

Dãy câu lệnh được viết giữa cặp tên dành riêng **begin** và **end** tạo thành thân của thủ tục.

Chú ý: Sau tên dành riêng **end** kết thúc chương trình chính là dấu chấm (.) còn sau **end** kết thúc thủ tục là dấu chấm phẩy (;).

Các thủ tục, nếu có, phải được khai báo và mô tả trong phần khai báo của chương trình chính, ngay sau phần khai báo các biến.

Khi cần thực hiện, ta phải viết lệnh gọi thủ tục tương tự như các thủ tục chuẩn.

b) Ví dụ về thủ tục

Thủ tục *Ve_Hcn* trong ví dụ trên chỉ vẽ được hình chữ nhật với kích thước cố định là 7×3 . Giả sử chương trình cần vẽ nhiều hình chữ nhật với kích thước khác nhau. Để thủ tục *Ve_Hcn* có thể thực hiện được điều đó, cần có hai *tham số* cho dữ liệu vào là chiều dài và chiều rộng. Khi đó phần đầu của thủ tục được viết như sau:

```
procedure Ve_Hcn(chdai, chrong:integer);
```

Khai báo này có nghĩa thủ tục *Ve_Hcn* sẽ được thực hiện để vẽ hình chữ nhật có kích thước tùy theo giá trị của các tham số *chdai*, *chrong* và giá trị của các tham số *chdai* và *chrong* là nguyên (*integer*).

Chương trình sau đây mô tả đầy đủ thủ tục *Ve_Hcn* với các tham số *chdai*, *chrong* và sử dụng thủ tục này để vẽ các hình chữ nhật có kích thước khác nhau.

```

program VD_thutuc2;
uses crt;
var a, b, i: integer;
procedure Ve_Hcn(chdai, chrong: integer);
    var i, j: integer;
    begin
        {ve canh tren cua hinh chu nhat}
        for i:= 1 to chdai do write('*');
        writeln;
        for j:=1 to chrong-2 do {ve 2 canh ben}
            begin
                write('*');
                for i:=1 to chdai-2 do write(' ');
                writeln('*');
            end;
        {ve canh duoi}
        for i:=1 to chdai do write('*');
        writeln;
    end;
begin {bat dau chuong trinh chinh}
    clrscr;
    {ve hinh chu nhat kích thước 25×10}
    Ve_Hcn(25,10);
    writeln; writeln; {de cach 2 dong}
    {ve hinh chu nhat kích thước 5×10}
    Ve_Hcn(5,10);
    readln;
    clrscr;
    {Ve 4 hình chu nhật: hình đầu tiên có kích thước 4×2,
    mọi hình sau có kích thước gấp đôi hình trước }
    a:=4; b:=2;
    for i:=1 to 4 do

```

```

    begin
        Ve_Hcn(a,b);
        readln; clrscr;
        a:=a*2; b:= b*2;
    end;
readln;
end.

```

Trong lệnh gọi thủ tục, các tham số hình thức được thay bằng các tham số thực sự tương ứng là các *giá trị cụ thể* được gọi là các **tham số giá trị** (gọi tắt là *tham trị*).

Các tham số *chdai*, *chrong* của thủ tục *Ve_Hcn* là tham trị. Trong lệnh gọi thủ tục *Ve_Hcn(5, 3)* (vẽ hình chữ nhật kích thước 5×3) tham số *chdai* được thay bởi số nguyên 5, tham số *chrong* được thay bởi số nguyên 3.

Còn trong lời gọi thủ tục *Ve_Hcn(a, b)* vẽ hình chữ nhật kích thước $a \times b$, tham số *chdai* được thay bởi giá trị hiện thời của biến *a*, tham số *chrong* được thay bởi giá trị hiện thời của biến *b*.

Trong lệnh gọi thủ tục, các tham số hình thức được thay bằng các tham số thực sự tương ứng là tên *các biến* chứa dữ liệu ra được gọi là các **tham số biến** (gọi tắt là *tham biến*).

Để phân biệt tham biến và tham trị, Pascal sử dụng từ khoá **var** để khai báo những tham số biến.

Ví dụ, thủ tục *Hoan_doi* trong chương trình sau đổi giá trị của hai biến. Vì cả hai biến đều chứa dữ liệu ra nên cần sử dụng từ khoá **var** để khai báo cho cả hai biến.

```

program VD_thambien1;
uses crt;
var a,b: integer;
procedure Hoan_doi(var x,y: integer);
    var TG: integer;
    begin
        TG:= x;
        x:= y;
        y:= TG;
    end;
begin
    clrscr;
    a:=5; b:=10;

```

```
writeln(a:6,b:6);
Hoan_doi(a,b);
writeln(a:6,b:6);
end.
```

Khai báo sau xác định x và y là hai tham biến kiểu nguyên:

```
var x,y: integer;
```

Trong lệnh gọi *Hoan_doi(a, b)* các tham biến x và y được thay thế bởi các biến nguyên tương ứng a và b .

Giá trị của các biến này bị thay đổi khi thực hiện các lệnh:

```
TG:=a; a:=b; b:=TG;
```

Do vậy, sau khi thực hiện thủ tục *Hoan_doi*, biến a sẽ nhận giá trị của biến b và biến b nhận giá trị của biến a . Nếu giá trị của a là 5 còn giá trị của b là 10 thì trên màn hình có hai dòng:

```
5 10
10 5
```

Chương trình sau cho thấy sự khác nhau khi chương trình con dùng tham số giá trị và tham số biến (có và không khai báo với từ khoá *var*):

```
program VD_thambien2;
uses crt;
var a, b: integer;
procedure Hoan_doi(x: integer; var y: integer);
var TG: integer;
begin
    TG:=x;
    x:=y;
    y:=TG;
end;
begin
    clrscr;
    a:=5; b:=10;
    writeln(a:6, b:6);
    Hoan_doi(a,b);
    writeln(a:6, b:6);
end.
```

Khi chương trình thực hiện, sẽ nhận được kết quả:

```
5 10
5 5
```

2. Cách viết và sử dụng hàm

Điểm khác nhau cơ bản giữa thủ tục và hàm là việc thực hiện hàm luôn trả về giá trị kết quả thuộc kiểu xác định và giá trị đó được gán cho tên hàm.

Hàm có cấu trúc tương tự như thủ tục, tuy nhiên có khác nhau phần đầu. Khai báo phần đầu một hàm như sau:

```
function <tên hàm> [ (<danh sách tham số>): <kiểu dữ liệu>;
```

trong đó, *kiểu dữ liệu* là kiểu dữ liệu của giá trị mà hàm trả về và chỉ có thể là các kiểu *integer, real, char, boolean, string*.

Cũng giống như thủ tục, nếu hàm không có tham số hình thức thì không cần *danh sách tham số*.

Khác với thủ tục, trong thân hàm cần có lệnh gán giá trị cho tên hàm:

```
<tên hàm> := <biểu thức>;
```

Ví dụ 1

Xét chương trình thực hiện việc rút gọn một phân số, trong đó có sử dụng hàm tính ước chung lớn nhất (UCLN) của hai số nguyên.

```
program Rutgon_Phanso;
uses crt;
var TuSo, MauSo, a: integer;
function UCLN(x, y: integer): integer; {bat dau ham UCLN}
  var sodu: integer;
  begin
    while y<>0 do
      begin
        sodu:= x mod y;
        x:= y;
        y:= sodu;
      end;
    UCLN:=x;
  end; {het ham UCLN}
begin
  clrscr;
  write('Nhap tu so, mau so vao! '); readln(TuSo, MauSo);
  a:=UCLN(TuSo, MauSo);
```

```

    if a>1 then
        begin
            TuSo:= TuSo div a;
            MauSo:= MauSo div a;
        end;
    writeln(TuSo:5, MauSo:5);
end.

```

Ghi chú: Trong chương trình này, các biến *TuSo*, *MauSo* và *a* là các biến toàn cục, còn biến *sodu* là biến cục bộ.

Sử dụng hàm

Việc sử dụng hàm *hoàn toàn tương tự* với việc sử dụng các hàm chuẩn, khi viết lệnh gọi gồm tên hàm và tham số thực sự tương ứng với các tham số hình thức.

Lệnh gọi hàm có thể tham gia vào biểu thức như một toán hạng và thậm chí là tham số của lời gọi hàm, thủ tục khác, ví dụ:

```
A:= 6*UcIn(TuSo,MauSo)+1;
```

Ví dụ 2

Chương trình sau cho biết giá trị nhỏ nhất trong ba số nhập từ bàn phím, trong đó có sử dụng hàm tìm số nhỏ nhất trong hai số.

```

program Minbaso;
var a, b, c: real;
{Ham tim so nho nhat trong hai so a va b}
function Min(a,b:real):real;
begin
    if a<b then Min:=a
        else Min:=b;
end;
begin
    write('Nhap vao ba so: ');
    readln(a,b,c);
    writeln('So nho nhat trong ba so la: ', Min(Min(a,b),c));
    readln
end.

```