

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

### TIẾT 1

#### A. Mục đích, yêu cầu

Với những ví dụ cụ thể về cách viết và sử dụng chương trình con trong ngôn ngữ Pascal, học sinh cần:

- Thấy được rằng: nói chung thủ tục có cấu trúc tương tự như một chương trình.
- Hiểu mối liên quan giữa chương trình và thủ tục: Thủ tục giải quyết một bài toán con và trong chương trình sẽ có lời gọi thủ tục khi cần giải quyết bài toán con này. Một chương trình có thể sử dụng nhiều thủ tục, có thể sử dụng một thủ tục nhiều lần.
- Phân biệt được tham số giá trị và tham số biến trong khai báo tham số hình thức của một thủ tục: Tham số biến được dùng để ghi lại dữ liệu kết quả của việc thực hiện thủ tục, còn tham số giá trị chỉ để đưa dữ liệu vào cho thủ tục khi bắt đầu thực hiện thủ tục.
- Ở lời gọi thực hiện một thủ tục: Một hằng hay một tên biến hay một biểu thức có thể xuất hiện ở vị trí một tham số thực sự tương ứng với tham số giá trị, trong khi đó ở vị trí một tham số thực sự tương ứng với tham số biến chỉ có thể là một tên biến.

Bước đầu học sinh nắm được những kỹ năng sau:

- Nhận biết được các thành phần trong đầu của thủ tục.
- Nhận biết được hai loại tham số hình thức trong phần đầu của thủ tục.
- Nhận biết được lời gọi thủ tục ở chương trình chính cùng các tham số thực sự.

Các kỹ năng viết và sử dụng chương trình con sẽ được rèn luyện ở các tiết thực hành.

#### B. Những điểm cần lưu ý và gợi ý tổ chức dạy học

##### 1. Khó khăn

Nội dung kiến thức về tham số (ở mục 1.b) là một nội dung khó đối với học sinh, thời lượng một tiết học sẽ không đủ để giáo viên vừa viết chương trình của các ví dụ lên bảng vừa giảng giải và tổ chức hoạt động trong lớp học.

## 2. Thuận lợi

- Dễ làm cho học sinh thấy được lợi ích của việc sử dụng chương trình con.
- Các ví dụ đưa vào sách giáo khoa là đơn giản và kế thừa nhau.
- Có nhiều thuận lợi khi muốn tạo điều kiện cho học sinh tích cực và chủ động trong việc tiếp thu bài học.

## 3. Một số điểm gợi ý tổ chức dạy học

Ví dụ vẽ hình chữ nhật đưa ra ở đầu (mục 1 §18) chỉ nhằm mục đích dẫn dắt học sinh đến tình huống dùng thủ tục có thể tránh được việc lặp đi lặp lại đoạn lệnh (gồm ba lệnh *writeln(...)*) vẽ một hình chữ nhật. Qua ví dụ đơn giản đó, học sinh cũng bước đầu hiểu được thủ tục (hay chương trình con) là một đoạn lệnh chịu trách nhiệm giải quyết một bài toán con.

Nên khai thác chương trình ở ví dụ đó để đưa ra những nhận xét, giới thiệu và minh họa cho học sinh về cấu trúc của một thủ tục cũng như cách gọi thủ tục của chương trình chính.

Bắt đầu giới thiệu cho học sinh với một trường hợp đơn giản, thủ tục trong chương trình *VD\_thutuc1* là thủ tục không có tham số. Sau đó có thể liên hệ lại ví dụ ở bài trước (§17), chưa yêu cầu học sinh hiểu sâu sắc khái niệm tham số hình thức, chỉ nhằm nhắc lại rằng có những thủ tục thực hiện phụ thuộc vào những tham số nào đó. Do vậy, ở đầu của những thủ tục này cần khai báo danh sách tham số. Qua các ví dụ tiếp theo các em sẽ được học về sử dụng tham số.

Khi nói rằng: thủ tục trong Pascal có cấu trúc hoàn toàn giống như một chương trình, trừ dòng đầu tiên và dòng cuối:

```
procedure <tên thủ tục> [ (<danh sách tham số> ) ];  
    [<phân khai báo hằng, biến,...>]  
begin  
    { ... dãy các lệnh (thân của thủ tục) ... }  
end;
```

giáo viên có thể yêu cầu học sinh cho biết [<phân khai báo hằng, biến,...>] thường gồm những khai báo gì, lưu ý học sinh trong thủ tục *Ve\_Hcn* không khai báo gì vì không cần dùng kiểu, hằng hay biến nào cả.

Có thể yêu cầu học sinh nhận xét vị trí của thủ tục trong một chương trình thông qua ví dụ để giáo viên khẳng định lại: các thủ tục (nếu có) phải được khai báo trong phần khai báo của chương trình, sau phần khai báo các biến.

Nhấn mạnh cho học sinh cách sử dụng thủ tục trong chương trình chính: Khi cần, chương trình chính gọi thủ tục qua tên của nó, như gọi các thủ tục chuẩn. Có thể lấy ví dụ một vài thủ tục chuẩn đã từng được dùng trong những chương trình trước đây.

Giáo viên có thể nêu vấn đề để dẫn dắt và gợi động cơ hướng đích cho phần tiếp theo (về tham số). Trong ví dụ đã được xét, mỗi khi thực hiện thủ tục *Ve\_Hcn* đều vẽ hình chữ nhật có kích thước cố định  $7 \times 3$ . Khi cần vẽ ba hình chữ nhật cùng có kích thước như vậy, chương trình chỉ cần gọi thủ tục *Ve\_Hcn* ba lần. Nếu cần vẽ ba hình chữ nhật nhưng với kích thước khác nhau (giáo viên có thể cho ba kích thước cụ thể) thì không thể sử dụng thủ tục *Ve\_Hcn* như trước nữa. Ta mong muốn có thể viết được thủ tục vẽ hình chữ nhật sao cho mỗi khi gọi nó thực hiện, chương trình chính quy định được kích thước hình chữ nhật mà thủ tục này phải vẽ ở lần thực hiện đó.

- *Tham số giá trị*

Khi phân tích chương trình *VD\_thutuc2*, cần lưu ý:

- Thủ tục *Ve\_Hcn* ở đây đã diễn đạt một thuật toán vẽ một hình chữ nhật có kích thước chiều dài và chiều rộng tùy theo giá trị của các tham số *chdai* và *chrong*. Những tham số này được gọi là tham số hình thức.
- Mỗi lần được gọi thực hiện, thủ tục *Ve\_Hcn* sẽ vẽ một hình chữ nhật với giá trị của tham số *chdai* và *chrong* đã được xác định trong lời gọi đó của chương trình. Những tham số, trong ví dụ này là các hằng nguyên, xuất hiện trong lời gọi thủ tục có vai trò xác định những giá trị cụ thể cho lần thực hiện vẽ hình chữ nhật này và được gọi là tham số thực sự.

Qua ví dụ, giáo viên cũng nên nhận xét hoặc gợi ý học sinh nhận xét:

- Nếu một thủ tục có tham số thì các tham số (hình thức) phải được khai báo sau tên thủ tục, trong cặp dấu ngoặc tròn ( và ). Khai báo một tham số có nghĩa là chỉ ra nó thuộc loại tham số gì và thuộc kiểu dữ liệu gì.
- Câu lệnh gọi thủ tục có dạng:

*<tên thủ tục>[(các tham số thực sự)];*

tức là chỉ cần viết tên của thủ tục và thay thế các tham số hình thức bởi các tham số thực sự. Việc thay thế như vậy phải đảm bảo rằng các tham số thực sự phải tương ứng với các tham số hình thức về trật tự và kiểu dữ liệu.

Trong ví dụ, cả hai lần gọi thủ tục *Ve\_Hcn* các tham số thực sự đều là hằng nguyên. Tuy nhiên, tổng quát hơn, *chdai* và *chrong* có kiểu nguyên thì khi gọi thủ tục chúng có thể được thay thế bởi biểu thức có kiểu nguyên. Giả sử *x* là biến nguyên thì các lời gọi sau đều là hợp lệ:

```
Ve_Hcn(25, 10);  
Ve_Hcn(5, x);  
Ve_Hcn(3*x+2, 2*x);
```

Trong lời gọi thứ nhất *chdai* được thay bởi giá trị hằng 25, *chrong* được thay bởi giá trị hằng 10. Trong lời gọi thứ hai *chdai* được thay bởi giá trị hằng 5, *chrong* được thay bởi giá trị của biến nguyên  $x$ . Trong lời gọi thứ ba *chdai* được thay bởi giá trị của biểu thức  $3*x+2$ , *chrong* được thay bởi giá trị biểu thức  $2*x$ .

- *Tham số biến*

Cần giúp học sinh phân biệt được cách dùng tham số giá trị với tham số biến.

Khi thực hiện thủ tục, các tham số giá trị như là các biến riêng của thủ tục được nhận giá trị khởi đầu là giá trị của các tham số thực sự (tương ứng) trong lời gọi thủ tục. Trong khi đó, mọi tác động (mọi thay đổi) xảy ra đối với các tham số biến sẽ xảy ra đối với các tham số thực sự thay thế chúng, tham số thực sự tương ứng với tham số biến phải là biến (không thể là hằng giá trị hay biểu thức).

Chương trình *VD\_thambien1* nhằm mục đích:

- Giới thiệu với học sinh một thủ tục khai báo hai tham số biến  $x, y$  thuộc kiểu *integer*.
- Qua kết quả chạy chương trình cho học sinh thấy được: Trước và sau khi gọi thực hiện thủ tục các tham số biến đã thay đổi, giá trị của tham số biến sau lời gọi thủ tục là giá trị sau cùng khi kết thúc thực hiện thủ tục.

Chương trình *VD\_thambien2* nhằm mục đích làm nổi bật sự khác nhau giữa tham số giá trị và tham số biến:

- Khác nhau trong khai báo danh sách tham số hình thức: Các tham số có từ khoá *var* đứng trước là tham số biến, còn không có từ khoá *var* đứng trước là tham số giá trị.
- Khác nhau ở kết quả sau khi thực hiện thủ tục.

Để có nhiều thời gian giải thích và tổ chức hoạt động nhận thức cho học sinh, giáo viên nên chuẩn bị sao cho không mất thời gian chép chương trình trong ví dụ lên bảng. Nên tạo một vài tình huống để điều chỉnh và củng cố nhận thức của học sinh chẳng hạn:

- Trong chương trình *VD\_thambien2*, mỗi lời gọi thủ tục *Hoan\_doi* sau có hợp lệ không?

- + Hoan\_doi(10,A); (Trả lời: Có)
- + Hoan\_doi(B,10); (Trả lời: Không)
- + Hoan\_doi(B,A+3); (Trả lời: Không)

- Trong chương trình *VD\_thambien2* nếu đầu thủ tục được đổi lại là:

```
procedure Hoan_doi(var x: integer; y: integer);
```

thì nhận được kết quả (trên màn hình) như thế nào sau khi thực hiện chương trình?

- Quay lại chương trình *VD\_thutuc2*, thực hiện một vài thay đổi như sau: (giáo viên nên chuẩn bị sẵn chương trình sửa đổi này trên giấy khổ lớn treo trước lớp).

+ Khai báo chương trình có thêm hai biến *D, R* thuộc kiểu *integer*.

+ Đầu thủ tục khai báo *chdai* là tham số biến.

+ Thêm hai câu lệnh sau vào cuối thủ tục:

```
chdai:= chdai*5;
chrong:= chrong*5;
```

(đây là hai câu lệnh cuối trước **end**; kết thúc thủ tục).

+ Thân chương trình được viết lại như sau:

```
begin
  clrscr;
  D:=25;R:=10;
  Ve_Hcn(D,R);
  writeln; writeln;
  D:=5;R:=10;
  Ve_Hcn(D,R);
  readln;
  clrscr;
  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.
```

Chương trình sau khi thay đổi sẽ là:

```
program VD_thutuc2_sua;
uses crt;
```

```

var a, b, i, D, R: integer;
procedure Ve_Hcn(var chdai: integer; 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;
    chdai:= chdai*5;
    chrong:= chrong*5;

  end;
begin {bat dau chuong trinh chinh}
  clrscr;
  D:=25;R:=10;
  Ve_Hcn(D,R);
  writeln; writeln;
  D:=5;R:=10;
  Ve_Hcn(D,R);
  readln;
  clrscr;
  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.

```

Yêu cầu học sinh cho biết kết quả thực hiện chương trình trên (sau khi thay đổi). Do biến *chdai* trở thành tham biến nên bốn hình chữ nhật vẽ sau cùng sẽ không còn tuân theo quy luật mỗi hình vẽ sau có chiều dài gấp đôi chiều dài của hình trước.

Một số điểm cần lưu ý:

- Nếu có nhiều tham số biến cùng thuộc một kiểu dữ liệu thì có thể dùng một từ khoá *var* chung cho phần khai báo những tham số biến này trong

danh sách tham số hình thức. Cụ thể là bắt đầu bằng từ khoá *var*, tiếp theo là tên các tham số biến cùng kiểu, các tên cách nhau bởi dấu phẩy, sau đó đến dấu hai chấm và tên kiểu dữ liệu. Nói tóm lại, các tham số cùng kiểu và cùng loại ngăn cách nhau bởi dấu phẩy còn các nhóm tham số khác kiểu hoặc khác loại được ngăn cách nhau bởi dấu chấm phẩy.

#### Ví dụ

- + Trong chương trình *VD\_thambien1*, khai báo tham số hình thức ở đầu thủ tục là:

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

cho thấy *x* và *y* đều là tham số biến của thủ tục này, cũng có thể viết:

```
(var x:integer;var y:integer);
```

tuy nhiên cách viết trên ngắn gọn hơn.

- + Muốn khai báo danh sách tham số hình thức gồm hai tham số biến *x*, *y* và hai tham số giá trị *P*, *Q* đều thuộc kiểu *integer*, chúng ta viết:

```
(var x, y: integer; P,Q:integer);
```

hoặc

```
(P,Q: integer; var x,y:integer);
```

hoặc muốn khai báo thủ tục *P* có ba tham số biến *X*, *Y*, *Q* và một tham số giá trị *Z*; *X* và *Y* thuộc kiểu *integer* còn *Q* và *Z* thuộc kiểu *real*, ta có thể khai báo như sau:

```
procedure P(var X,Y:integer;var Q:real;Z:real);
```

Khi khai báo dữ liệu cho các tham số chỉ được dùng tên kiểu. Tên kiểu là tên chuẩn hoặc tên do người lập trình đặt.

#### Ví dụ

Khai báo sau là hợp lệ:

```
procedure Alpha(N : integer ; var s : string);
```

nhưng khai báo sau đây là không hợp lệ:

```
procedure Delta(M:array[1..10] of byte;var X:string[25]);
```

Muốn khai báo các tham số *M* có kiểu mảng và *X* có kiểu xâu như trên, phải xác định trước hai kiểu dữ liệu như vậy ở phần khai báo *type* của chương trình

```
type Mang = array[1..10] of byte;  
Xau = string[25];
```

rồi sau đó mới được dùng

```
procedure Delta (M: Mang; var X: Xau);
```

So với nội dung tiết 2 (của §18 này), nội dung tiết 1 nhiều hơn, do vậy một số phân gợi ý tổ chức dạy học ở cuối tiết 1, giáo viên có thể chuyển sang thực hiện ở đầu tiết 2 với mục đích củng cố, khắc sâu các kiến thức của tiết 1.

## TIẾT 2

### A. Mục đích, yêu cầu

Học sinh cần nắm được sự giống nhau và khác nhau giữa thủ tục và hàm và có kỹ năng nhận biết được các thành phần trong đầu hàm, nhận biết được câu lệnh sử dụng hàm ở chương trình chính cùng các tham số thực sự.

### B. Những điểm cần lưu ý và gợi ý tổ chức dạy học

#### 1. Thuận lợi

Phần thủ tục và các tham số đã được trình bày kỹ nên khi trình bày hàm chỉ cần nêu những điểm khác nhau giữa hàm và thủ tục.

#### 2. Một số gợi ý tổ chức dạy học

Cần làm nổi bật được sự giống và khác nhau giữa thủ tục và hàm:

- Có nhiều điểm giống nhau:
  - Thủ tục và hàm đều là chương trình con, nói chung có cấu trúc giống một chương trình.
  - Thủ tục và hàm đều có thể chứa các tham số (tham số giá trị và tham số biến), cùng tuân theo các quy định về khai báo và sử dụng các loại tham số này.
- Hàm chỉ khác thủ tục ở điểm căn bản là việc thực hiện hàm luôn trả về một giá trị thuộc kiểu xác định qua tên hàm. Do vậy:
  - Đầu hàm bắt đầu với từ khoá **function**, sau tên hàm và phân khai báo danh sách tham số (nếu có) phải chỉ ra giá trị kết quả của hàm thuộc kiểu dữ liệu nào.
  - Trong thân hàm thường có câu lệnh gán giá trị cho tên hàm.



Mục đích của chương trình *Rutgon\_Phanso* là minh họa cho những điểm giống và khác nhau giữa thủ tục và hàm.

Chương trình *Minbaso* ở ví dụ 2 có sử dụng hàm *Min*, tìm số nhỏ hơn trong hai số *A* và *B*. Tuy nhiên việc sử dụng hàm *Min* ở đây được nâng lên một mức phức tạp hơn. Để thực hiện câu lệnh:

```
writeln('Số nhỏ nhất trong ba số là: ', Min(Min(A,B),C));
```

hàm *Min* được gọi thực hiện với hai tham số thực sự: Tham số thực sự thứ nhất là kết quả thực hiện hàm *Min(A, B)* còn tham số thứ hai là giá trị của biến *C*. Để có được tham số thực sự thứ nhất đó, hàm *Min(A, B)* đã được thực hiện, kết quả trả về là số nhỏ hơn trong hai giá trị của biến *A* và biến *B*. Như vậy, việc thực hiện hàm *Min* với hai tham số thực sự là *Min(A, B)* và *C* sẽ cho kết quả là số nhỏ nhất trong ba giá trị của các biến *A, B, C*. Một cách tổng quát, một hàm có thể thay vào vị trí của một tham số giá trị trong lời gọi thực hiện một chương trình con (thủ tục hay hàm) nếu hàm đó có cùng kiểu với tham số giá trị mà nó thay thế. Điều này hợp lệ vì hàm được phép thay vào vị trí một giá trị (hay một toán hạng của biểu thức).

Cần lưu ý:

- Việc sử dụng hàm tương tự như sử dụng các hàm chuẩn.
- Kết quả trả về qua tên hàm chỉ có thể thuộc các kiểu dữ liệu chuẩn (integer, real, boolean, char, string).
- Vì kết quả trả về đã gán cho tên hàm nên thông thường các tham số dùng trong hàm là tham số giá trị.

Trong Pascal có chương trình con dưới dạng thủ tục và hàm. Thông thường khi cần thực hiện một công việc nào đó người ta dùng thủ tục, còn khi cần tính một giá trị nào đó người ta dùng hàm. Tuy nhiên, điều này là không bắt buộc. Bất cứ công việc nào được thực hiện bởi thủ tục cũng đều có thể thực hiện được bằng hàm và bất cứ giá trị nào được tính bằng hàm cũng đều có thể tính được nhờ thủ tục. Hiện nay người ta có xu hướng dùng tất cả các chương trình con dưới dạng hàm ngay cả khi không quan tâm đến giá trị trả về. Trong những trường hợp như vậy, người ta thường gán cho hàm một giá trị logic nhằm ghi nhận trạng thái thành công hay thất bại của công việc mà hàm đã tiến hành.

Cuối tiết này nên hệ thống lại để củng cố cho học sinh các khái niệm: chương trình con, thủ tục, hàm, tham số hình thức, tham số thực sự, biến toàn cục và biến cục bộ. Có thể làm điều đó đồng thời với việc sử dụng vào một chương trình nào đó (trong số các chương trình học sinh đã được học) để minh họa một cách nhanh chóng.