

Họ tên:

TIN HỌC LỚP 11

Lớp:

CHƯƠNG VI: CHƯƠNG TRÌNH CON: THỦ TỤC VÀ HÀM

(Tiết 1)

1. Khái niệm về chương trình con

Chương trình con là một đoạn chương trình thực hiện trọn vẹn hay một chức năng nào đó. Trong Pascal, có 2 dạng chương trình con:

- Thủ tục (PROCEDURE): Dùng để thực hiện một hay nhiều nhiệm vụ nào đó.
- Hàm (FUNCTION): Trả về một giá trị nào đó (có kiểu vô hướng, kiểu string hoặc kiểu con trỏ). Hàm có thể sử dụng trong các biểu thức. Ngoài ra, trong Pascal còn cho phép các CTC lồng vào nhau.

2. Cấu trúc chung

```

PROGRAM Tên_chương_trình;
USES      CRT;
CONST .....;
TYPE .....;
VAR .....;

PROCEDURE THUTUC[(Các tham số)];
    [Khai báo Const, Type, Var]
BEGIN
    .....
END;

FUNCTION HAM[(Các tham số)]:<Kiểu dữ liệu>;
    [Khai báo Const, Type, Var]
BEGIN
    .....
    HAM:=<Giá trị>;
END;

BEGIN {Chương trình chính}
    .....
    THUTUC[...];
    .....
    A:= HAM[...];
    .....
END.

```

↪ Các khái niệm:

Biến toàn cục (*global variable*): Còn được gọi là biến chung, là biến được khai báo ở đầu chương trình, nó được sử dụng bên trong chương trình chính và cả bên trong chương trình con. Biến toàn cục sẽ tồn tại trong suốt quá trình thực hiện chương trình.

Biến cục bộ (*local variable*): Còn được gọi là biến riêng, là biến được khai báo ở đầu chương trình con, và nó chỉ được sử dụng bên trong thân chương trình con hoặc bên trong thân chương trình con khác nằm bên trong nó (các chương trình con lồng nhau). Biến cục bộ chỉ tồn tại khi chương trình con đang hoạt động, nghĩa là biến cục bộ sẽ được cấp phát bộ nhớ khi chương trình con được gọi để thi hành, và nó sẽ được giải phóng ngay sau khi chương trình con kết thúc.

Tham số thực (*actual parameter*) là một tham số mà nó có thể là một biến toàn cục, một biểu thức hoặc một giá trị số (cũng có thể biến cục bộ khi sử dụng chương trình con lồng nhau) mà ta dùng chúng khi truyền giá trị cho các tham số hình thức tương ứng của chương trình con.

Tham số hình thức (*formal parameter*) là các biến được khai báo ngay sau **Tên chương trình con**, nó dùng để nhận giá trị của các tham số thực truyền đến. Tham số hình thức cũng là một biến cục bộ, ta có thể xem nó như là các đối số của hàm toán học.

3. Thủ tục (Procedure):

Thủ tục là một đoạn cấu trúc chương trình được chứa bên trong chương trình Pascal như là một chương trình con. Thủ tục được đặt tên và có thể chứa danh sách tham số hình thức (*formal parameters*). Các tham số này phải được đặt trong dấu ngoặc đơn (). Ta có thể truy xuất thủ tục bằng cách **gọi tên của thủ tục**. Chương trình sẽ tự động truy xuất thủ tục đúng tên đã gọi và thực hiện các lệnh chứa trong thủ tục đó.

Có 2 loại thủ tục:

- Thủ tục không tham số
- Thủ tục có tham số.

2. Hàm (Function) :

Hàm là một chương trình con cho ta 1 giá trị kiểu vô hướng. Hàm tương tự như thủ tục nhưng trả về **một** giá trị thông qua tên hàm và lời gọi hàm tham gia trong biểu thức.

Cấu trúc một hàm tự đặt gồm:

FUNCTION <Tên hàm> (<Tham số hình thức : kiểu biến>) :

<Kiểu kết quả> ;

{ các khai báo hằng, biến, kiểu cục bộ... }

BEGIN

{ ... các khai báo trong nội bộ hàm ... }

END;

Trong đó:

- **Tên hàm** là tên tự đặt cần tuân thủ theo nguyên tắc đặt tên trong Pascal.
- **Kiểu kết quả** là một kiểu vô hướng, biểu diễn kết quả giá trị của hàm.
- Một hàm có thể có 1 hay nhiều tham số hình thức, khi có nhiều tham số hình thức cùng một kiểu giá trị thì ta có thể viết chúng cách nhau bằng dấu phẩy (,). Trường hợp các tham số hình thức khác kiểu thì ta viết chúng cách nhau bằng dấu chấm phẩy (;).

- Trong hàm có thể sử dụng các hằng, kiểu, biến đã được khai báo trong chương trình chính nhưng ta có thể khai báo thêm các hằng, kiểu, biến dùng riêng trong nội bộ hàm. Chú ý là phải có một *biến trung gian có cùng kiểu kết quả của hàm* để lưu kết quả của hàm trong quá trình tính toán để cuối cùng ta có 1 lệnh gán giá trị của biến trung gian cho tên hàm.

Dùng hàm	Dùng thủ tục
<ul style="list-style-type: none">- Kết quả của bài toán trả về 1 giá trị duy nhất (kiểu vô hướng, kiểu string hoặc kiểu con trỏ).- Lời gọi CTC cần nằm trong các biểu thức tính toán.	<ul style="list-style-type: none">- Kết quả của bài toán không trả về giá trị nào hoặc trả về nhiều giá trị hoặc trả về kiểu dữ liệu có cấu trúc (Array, Record, File).- Lời gọi CTC không nằm trong các biểu thức tính toán.

Họ tên:

TIN HỌC LỚP 11

Lớp:

CHƯƠNG VI: CHƯƠNG TRÌNH CON: THỦ TỤC VÀ HÀM

(Tiết 2)

Ví dụ về chương trình con:**Bài tập 1:** Viết CTC để tính $n! = 1.2...n$.

↳ Ý tưởng:

- Nếu người dùng nhập số $n < 0$ thì yêu cầu nhập lại.
- Sử dụng chương trình con để tính giai thừa của một số.

 $n! = 1$ nếu $n = 0$; $n! = 1.2.3.4.5...n$ (Tích của n thừa số).

```

Program CTC_2;
uses crt;
var n:integer;x:longint;
Procedure Giai_Thua(var GT:longint; n:byte);
begin
  GT:=1;
  while n>0 do
  begin
    GT:=GT*n;
    n:=n-1;
  end;
end;
begin
  repeat
  clrscr;
  write('Nhap so n: ');readln(n);
  if n < 0 then begin write('Nhap so n>=0');readln;
end;
  until n>=0;
  Giai_Thua(x,n);
  writeln('Giai thua cua ',n,'la:',x);
  readln
end.

```

Bài tập 2: Viết chương trình cho phép thực hiện rút gọn phân số.

↳ Ý tưởng:

- Tìm UCLN của tử số và mẫu số.
- Chia tử và mẫu của phân số cho UCLN vừa tìm được.

```
Program Rut_gon_phan_so;
uses crt;
var tu, mau: integer;
Function UCLN(a, b: integer): integer;
var r: integer;
begin
  r := a mod b;
  while r <> 0 do
  begin
    a := b;
    b := r;
    r := a mod b;
  end;
  UCLN := b;
end;
begin
  clrscr;
  write('Nhap tu: '); readln(tu);
  write('Nhap mau: '); readln(mau);
  write('Ket qua rut gon: ', tu, '/', mau, '=', tu div
UCLN(tu, mau), '/', mau div UCLN(tu, mau));
  readln
end.
```