

§12. KIỂU XÂU

Dữ liệu trong các bài toán không chỉ thuộc kiểu số mà cả kiểu phi số - dạng kí tự. Dữ liệu kiểu xâu là dãy các kí tự.

Ví dụ. Các xâu kí tự đơn giản:

'Bach khoa' 'KI SU' '2007 la nam Dinh Hoi'

Xâu là dãy các kí tự trong bộ mã ASCII, mỗi kí tự được gọi là một phần tử của xâu. Số lượng kí tự trong một xâu được gọi là độ dài của xâu. Xâu có độ dài bằng 0 gọi là xâu rỗng.

Các ngôn ngữ lập trình đều có quy tắc, cách thức cho phép xác định:

- Tên kiểu xâu;
- Cách khai báo biến kiểu xâu;
- Số lượng kí tự của xâu;
- Các phép toán thao tác với xâu;
- Cách tham chiếu tới phần tử của xâu.

Có thể xem xâu là mảng một chiều mà mỗi phần tử là một kí tự. Các kí tự của xâu được đánh số thứ tự, thường bắt đầu từ 1.

Tương tự mảng, tham chiếu tới phần tử của xâu được xác định bởi tên biến xâu và chỉ số đặt trong cặp ngoặc [và].

Ví dụ, giả sử có biến *Hoten* lưu trữ giá trị hàng xâu '*Nguyen Le Huyen*' thì *Hoten*[6] cho ta kí tự '*n*' là kí tự thứ sáu của biến xâu *Hoten*.

Dưới đây trình bày cách khai báo dữ liệu kiểu xâu, các thao tác xử lý xâu và một số ví dụ sử dụng kiểu xâu trong Pascal.

1. Khai báo

Để khai báo dữ liệu kiểu xâu ta sử dụng tên dành riêng **string**, tiếp theo là độ dài lớn nhất của xâu (không vượt quá 255) được ghi trong cặp ngoặc [và].

Biến kiểu xâu có thể khai báo như sau:

```
var <tên biến>: string[độ dài lớn nhất của xâu];
```

Ví dụ

```
var Hoten: string[26];
```

Trong mô tả xâu có thể bỏ qua phần khai báo độ dài, chẳng hạn:

```
var Chugiai: string;
```

Khi đó độ dài lớn nhất của xâu sẽ nhận giá trị ngầm định là 255.

2. Các thao tác xử lí xâu

- a) Phép ghép xâu, kí hiệu là dấu cộng (+), được sử dụng để ghép nhiều xâu thành một. Có thể thực hiện phép ghép xâu đối với các hằng và biến xâu.

Ví dụ

Phép ghép xâu:

```
'Ha' + ' Noi' + ' - ' + 'Viet Nam'
```

cho xâu kết quả là 'Ha Nói - Việt Nam'.

- b) Các phép so sánh bằng (=), khác (<>), nhỏ hơn (<), lớn hơn (>), nhỏ hơn hoặc bằng (<=), lớn hơn hoặc bằng (>=) có thứ tự ưu tiên thực hiện thấp hơn phép ghép xâu và thực hiện việc so sánh hai xâu theo các quy tắc sau:

- Xâu A là lớn hơn xâu B nếu như kí tự đầu tiên khác nhau giữa chúng kể từ trái sang trong xâu A có mã ASCII lớn hơn.
- Nếu A và B là các xâu có độ dài khác nhau và A là đoạn đầu của B thì A là nhỏ hơn B.

Ví dụ

```
'May tinh' < 'May tinh cuatoi'
```

- Hai xâu được coi là bằng nhau nếu như chúng giống nhau hoàn toàn.

Ví dụ

```
'TIN HOC' = 'TIN HOC'
```

Để xử lí các xâu, có thể sử dụng các thủ tục và hàm chuẩn dưới đây:

- c) Thủ tục $delete(st, vt, n)$ thực hiện việc xoá n kí tự của biến xâu st bắt đầu từ vị trí vt .

Ví dụ

Giá trị st	Thao tác	Kết quả
'abcdef'	$delete(st, 5, 2);$	'abcd'
'Song Hong'	$delete(st, 1, 5);$	'Hong'

- d) Thủ tục $insert(s1, s2, vt)$ chèn xâu $s1$ vào xâu $s2$, bắt đầu ở vị trí vt .

Ví dụ

Giá trị $s1$	Giá trị $s2$	Thao tác	Kết quả
' PC '	'IBM486'	$insert(s1, s2, 4);$	'IBM PC 486'
'1'	'Hinh .2'	$insert(s1, s2, 6);$	'Hinh 1.2'

- e) Hàm $copy(S, vt, N)$ tạo xâu gồm N kí tự liên tiếp bắt đầu từ vị trí vt của xâu S .

Ví dụ

Giá trị s	Biểu thức	Kết quả
'Bai hoc thu 9'	$copy(s, 9, 5);$	'thu 9'

- f) Hàm $length(s)$ cho giá trị là độ dài xâu s .

Ví dụ

Giá trị s	Biểu thức	Kết quả
'500 ki tu'	$length(s)$	9

- g) Hàm $pos(s1, s2)$ cho vị trí xuất hiện đầu tiên của xâu $s1$ trong xâu $s2$.

Ví dụ

Giá trị $s2$	Biểu thức	Kết quả
'abcdef'	$pos('cd', s2)$	3
'abcdef'	$pos('k', s2)$	0

- h) Hàm `upcase(ch)` cho chữ cái in hoa ứng với chữ cái trong `ch`.

Ví dụ

Giá trị <code>ch</code>	Biểu thức	Kết quả
'd'	<code>upcase(ch)</code>	'D'
'E'	<code>upcase(ch)</code>	'E'

3. Một số ví dụ

Ví dụ 1

Chương trình dưới đây nhập họ tên của hai người vào hai biến xâu và đưa ra màn hình xâu dài hơn, nếu bằng nhau thì đưa ra xâu nhập sau.

```
var
  a,b:string;
begin
  write('Nhập họ tên thứ nhất: '); readln(a);
  write('Nhập họ tên thứ hai: '); readln(b);
  if length(a)>length(b) then write(a) else write(b);
  readln
end.
```

Ví dụ 2

Chương trình dưới đây nhập hai xâu từ bàn phím và kiểm tra kí tự đầu tiên của xâu thứ nhất có trùng với kí tự cuối cùng của xâu thứ hai không.

```
var x: byte;
  a,b: string;
begin
  write('Nhập xâu thứ nhất: ');
  readln(a);
  write('Nhập xâu thứ hai: ');
  readln(b);
  x:=length(b);
  {xác định độ dài xâu b để biết vị trí của kí tự cuối cùng}
  if a[1]=b[x] then write('Trung nhau')
    else write('Khác nhau');
  readln
end.
```

Ví dụ 3

Chương trình sau nhập một xâu vào từ bàn phím và đưa ra màn hình xâu đó nhưng được viết theo thứ tự ngược lại.

```

var i,k: byte;
    a: string;
begin
    write('Nhập xâu: ');
    readln(a);
    k:= length(a); {xác định độ dài xâu}
    for i:= k downto 1 do write(a[i]);
    readln
end.

```

Ví dụ 4

Chương trình sau nhập một xâu vào từ bàn phím và đưa ra màn hình xâu thu được từ nó sau khi loại bỏ các dấu cách nếu có.

```

var i,k: byte;
    a, b: string;
begin
    write('Nhập xâu: ');
    readln(a);
    k:= length(a);
    b:= ''; (* Khởi tạo xâu rỗng *)
    for i:= 1 to k do
        if a[i]<>' ' then b:=b+a[i];
    writeln('Kết quả: ',b);
    readln
end.

```

Ví dụ 5

Chương trình sau nhập vào từ bàn phím xâu kí tự s1, tạo xâu s2 gồm tất cả các chữ số có trong s1 (giữ nguyên thứ tự xuất hiện của chúng) và đưa kết quả ra màn hình.

```

program XuLiXau;
var s1, s2: string;
    i: byte;
begin
    write('Nhập vào xâu s1: ');
    readln(s1);
    s2:= ''; {Khởi tạo xâu s2 rỗng}
    for i:= 1 to length(s1) do
        if ('0'<=s1[i]) and (s1[i]<= '9') then s2:= s2+s1[i];
    writeln('Kết quả: ', s2);
    readln
end.

```