

BÀI 4

SỬ DỤNG BIẾN TRONG CHƯƠNG TRÌNH

1. Biến là công cụ trong lập trình

Hoạt động cơ bản của chương trình máy tính là xử lý dữ liệu. Trước khi được máy tính xử lý, mọi dữ liệu nhập vào đều được lưu trong bộ nhớ của máy tính. Ví dụ, nếu muốn cộng hai số a và b , trước hết hai số đó sẽ được nhập và lưu trong bộ nhớ máy tính, sau đó máy tính sẽ thực hiện phép cộng $a + b$.

Để chương trình luôn biết chính xác dữ liệu cần xử lý được lưu ở vị trí nào trong bộ nhớ, các ngôn ngữ lập trình cung cấp một công cụ lập trình rất quan trọng. Đó là *biến nhớ*, hay được gọi ngắn gọn là *biến*.

Trong lập trình, *biến* được dùng để *lưu trữ dữ liệu* và dữ liệu được biến lưu trữ có thể thay đổi trong khi thực hiện chương trình.

Dữ liệu do biến lưu trữ được gọi là *giá trị của biến*.

Chúng ta hãy xét một số ví dụ để hiểu vai trò biến nhớ trong lập trình.

Ví dụ 1. Giả sử cần in kết quả của phép cộng $15 + 5$ ra màn hình. Trong bài thực hành 2, ta đã biết có thể sử dụng câu lệnh Pascal sau đây:

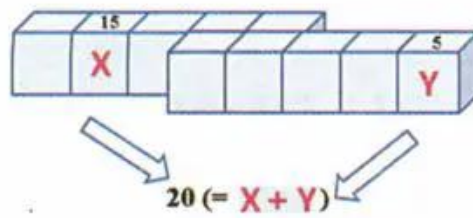
```
writeln(15+5);
```

Tình hình sẽ khác, nếu như hai số 15 và 5 được nhập trước từ bàn phím (hoặc là các kết quả tính toán trước đó). Sau khi nhận được các số 15 và 5, chương trình lưu trữ các số này ở những vị trí nào đó trong bộ nhớ. Chúng ta không thể biết giá trị của các số được nhập từ trước (hoặc các kết quả tính toán trung gian) nên không thể sử dụng lệnh in ra màn hình như trên. Vì thế ta sử dụng hai biến X và Y để lưu giá trị của các số được nhập vào, tức 15 và 5 (hoặc kết quả tính toán trung gian), sau đó có thể sử dụng lệnh

```
writeln(X+Y);
```

để in kết quả ra màn hình.

Với việc sử dụng biến như trên, chương trình sẽ tự biết lấy các số 15 và 5 từ những vị trí nào trong bộ nhớ để thực hiện phép cộng (h. 24).



Hình 24

Ví dụ này cũng cho thấy, một cách hình ảnh, có thể xem hai biến X và Y như là “tên” của các vùng nhớ chứa các giá trị tương ứng.

Ví dụ 2. Giả sử cần tính giá trị của các biểu thức $\frac{100+50}{3}$ và $\frac{100+50}{5}$ và in kết quả ra màn hình. Chúng ta có thể tính các biểu thức này một cách trực tiếp. Để ý rằng tử số trong các biểu thức là như nhau. Do đó có thể tính giá trị tử số và lưu tạm thời trong một biến trung gian X , sau đó thực hiện các phép chia. Về mặt toán học, điều này được thực hiện như sau (h. 25):

```
X = 100 + 50
Y = X/3
Z = X/5
```

Hình 25

2. Khai báo biến

Tất cả các biến dùng trong chương trình cần phải được khai báo ngay trong phần khai báo của chương trình. Việc khai báo biến gồm:

- Khai báo *tên biến*;
- Khai báo *kiểu dữ liệu* của biến.

Tên biến phải tuân theo quy tắc đặt tên của ngôn ngữ lập trình.

Ví dụ 3. Hình 26 là một ví dụ về cách khai báo biến trong Pascal:

```
var m,n : integer;
    S, dientich: real;
    thong_bao: string;
```

Hình 26

Trong ví dụ trên:

- `var` là từ khoá của ngôn ngữ lập trình dùng để khai báo biến,
- `m, n` là các biến có kiểu nguyên (`integer`),
- `S, dientich` là các biến có kiểu thực (`real`),
- `thong_bao` là biến kiểu xâu (`string`).

Tùy theo ngôn ngữ lập trình, cú pháp khai báo biến có thể khác nhau.

3. Sử dụng biến trong chương trình

Sau khi khai báo, ta có thể sử dụng các biến trong chương trình. Các thao tác có thể thực hiện với các biến là:

- *Gán* giá trị cho biến;
- *Tính toán* với giá trị của biến.

Kiểu dữ liệu của giá trị được gán cho biến thường phải trùng với kiểu của biến và khi được gán một giá trị mới, giá trị cũ của biến bị xoá đi. Ta có thể thực hiện việc gán giá trị cho biến tại bất kì thời điểm nào trong chương trình, do đó giá trị của biến có thể thay đổi.

Câu lệnh gán giá trị trong các ngôn ngữ lập trình thường có dạng:

Tên biến ← *Biểu thức cần gán giá trị cho biến*;

trong đó, dấu ← biểu thị phép gán. Ví dụ:

$x \leftarrow -c/b$ (biến x nhận giá trị bằng $-c/b$);

$x \leftarrow y$ (biến x được gán giá trị của biến y);

$i \leftarrow i + 5$ (biến i được gán giá trị hiện tại của i cộng thêm 5 đơn vị).

Việc gán giá trị cho biến còn có thể thực hiện bằng câu lệnh nhập dữ liệu.

Tùy theo ngôn ngữ lập trình, kí hiệu của câu lệnh gán cũng có thể khác nhau. Ví dụ, trong ngôn ngữ Pascal, người ta kí hiệu phép gán là dấu kép := để phân biệt với dấu bằng (=) phép so sánh.

Ví dụ 4. Bảng dưới đây mô tả lệnh gán giá trị và tính toán với các biến trong Pascal:

Lệnh trong Pascal	Ý nghĩa
<code>X:=12;</code>	Gán giá trị số 12 vào biến nhớ X.
<code>X:=Y;</code>	Gán giá trị đã lưu trong biến nhớ Y vào biến nhớ X.
<code>X:=(a+b)/2;</code>	Thực hiện phép toán tính trung bình cộng hai giá trị nằm trong hai biến nhớ a và b. Kết quả gán vào biến nhớ X.
<code>X:=X+1;</code>	Tăng giá trị của biến nhớ X lên 1 đơn vị, kết quả gán trở lại biến X.

4. Hằng

Ngoài công cụ chính để lưu trữ dữ liệu là biến, các ngôn ngữ lập trình còn có công cụ khác là **hằng**. Khác với biến, hằng là đại lượng có *giá trị không đổi* trong suốt quá trình thực hiện chương trình.

Giống như biến, muốn sử dụng hằng, ta cũng cần phải khai báo tên của hằng. Tuy nhiên hằng phải được gán giá trị ngay khi khai báo.

Dưới đây là ví dụ khai báo hằng trong Pascal:

```
const pi = 3.14;  
      bankinh = 2;
```

Hình 27

trong đó:

- **const** là từ khoá để khai báo hằng,
- Các hằng **pi**, **bankinh** được gán giá trị tương ứng là 3.14 và 2.

Với khai báo trên, để tính chu vi của hình tròn, ta có thể dùng câu lệnh sau:

```
chuvi := 2*pi*bankinh;
```

Việc sử dụng hằng rất hiệu quả nếu giá trị của hằng (bán kính) được sử dụng trong nhiều câu lệnh của chương trình. Nếu sử dụng hằng, khi cần thay đổi giá trị, ta chỉ cần chỉnh sửa một lần, tại nơi khai báo mà không phải tìm và sửa trong cả chương trình.

Cần lưu ý rằng ta không thể dùng câu lệnh để thay đổi giá trị của hằng (như đối với biến) ở bất kì vị trí nào trong chương trình. Ví dụ, đối với các hằng **pi** và **bankinh** đã khai báo ở trên, các câu lệnh gán sau đây trong chương trình là không hợp lệ:

```
pi := 3.1416;  
bankinh := bankinh + 2;
```

GHI NHỚ

1. Biến và hằng là các đại lượng được đặt tên dùng để *lưu trữ dữ liệu*. Giá trị của biến có thể thay đổi, còn giá trị của hằng được giữ nguyên trong suốt quá trình thực hiện chương trình.
2. Biến và hằng phải được khai báo trước khi sử dụng.

Câu hỏi và bài tập

1. Giả sử A được khai báo là biến với kiểu dữ liệu số thực, X là biến với kiểu dữ liệu xâu. Các phép gán sau đây có hợp lệ không?

- a) $A := 4;$ b) $X := 3242;$
c) $X := '3242';$ d) $A := 'Ha Noi'.$

2. Nêu sự khác nhau giữa biến và hằng. Cho một vài ví dụ về khai báo biến và hằng.

3. Giả sử ta đã khai báo một hằng π với giá trị 3.14. Có thể gán lại giá trị 3.1416 cho π trong phần thân chương trình được không? Tại sao?

4. Trong Pascal, khai báo nào sau đây là đúng?

- a) `var tb: real;`
b) `var 4hs: integer;`
c) `const x : real;`
d) `var R = 30;`

5. Hãy liệt kê các lỗi nếu có trong chương trình dưới đây và sửa lại cho đúng:

```
var a, b := integer;  
const c := 3;  
begin  
    a := 200  
    b := a/c;  
    write(b);  
    readln  
end.
```

6. Hãy cho biết kiểu dữ liệu của các biến cần khai báo dùng để viết chương trình để giải các bài toán dưới đây:

- a) Tính diện tích S của hình tam giác với độ dài một cạnh a và chiều cao tương ứng h (a và h là các số tự nhiên được nhập vào từ bàn phím).
b) Tính kết quả c của phép chia lấy phần nguyên và kết quả d của phép chia lấy phần dư của hai số nguyên a và b .