



Bài thực hành 3

KHAI BÁO VÀ SỬ DỤNG BIẾN

Thời lượng: 2 tiết

1. Mục đích, yêu cầu

- Thực hiện được khai báo đúng cú pháp, lựa chọn được kiểu dữ liệu phù hợp cho biến;
- Kết hợp được giữa lệnh *write*, *writeln* với *read*, *readln* để thực hiện việc nhập dữ liệu cho biến từ bàn phím;
- Hiểu về các kiểu dữ liệu chuẩn: kiểu số nguyên, kiểu số thực;
- Sử dụng được lệnh gán giá trị cho biến;
- Hiểu cách khai báo và sử dụng hằng;
- Hiểu và thực hiện được việc trao đổi giá trị của hai biến.

2. Những điểm cần lưu ý và gợi ý dạy học

a) Chương trình trong câu a, bài 1 là chương trình giả định số tiền phải trả bao gồm số tiền mua hàng (bảng đơn giá nhân với số lượng) và số tiền cước phí vận chuyển (cố định là 10000). Tổng số tiền phải trả bằng số tiền mua hàng cộng với cước phí.

Với bài này HS tập khai báo biến trong Pascal, cần cho HS tìm hiểu cú pháp khai báo biến, đặt tên đúng theo quy định của Pascal, chọn đúng kiểu dữ liệu của biến.

Rèn luyện soạn thảo, dịch, hiệu chỉnh, chạy và kiểm chứng kết quả cũng là một mục tiêu của bài này.

Cần hướng dẫn để HS tìm hiểu chức năng của lệnh *readln(tên biến)* để nhập giá trị của biến. Sự kết hợp của *write* và *readln* trong việc nhập giá trị biến từ bàn phím. Việc sử dụng biến trong biểu thức:

$$\text{thanhtien} := \text{soluong} * \text{dongia} + \text{phi}.$$

Các chú thích đặt trong cặp dấu ngoặc {} hoặc (* *) được dùng để giải thích câu lệnh, ý đồ của người viết chương trình. Gặp cặp dấu ngoặc này Pascal bỏ qua, không dịch những nội dung bên trong. Việc viết chú thích trong chương trình đôi khi rất cần thiết để giúp người khác có thể nhanh chóng hiểu được chương trình, thậm chí là để chính người đã viết ra chương trình dễ dàng hơn khi xem lại hoặc chỉnh sửa chương trình của mình.

Có thể gợi ý HS nhập số lượng là một số thực, ví dụ 6.5 chẳng hạn và giải thích hiện tượng xảy ra. Nguyên nhân là do kiểu dữ liệu nhập vào là số thực không phù hợp với kiểu của biến *Soluong* đã khai báo trong chương trình là số nguyên. Có thể gợi ý để HS thay đổi kiểu của biến số lượng đã khai báo để có thể nhập số lượng là một số thực.

Khi nhập bộ số liệu (1, 35000), kết quả không còn đúng nữa, nguyên nhân của hiện tượng này là do tràn số. Biến *Soluong* có kiểu là *integer* nên chỉ cho phép chứa các giá trị trong khoảng từ -32768 đến 32767, giá trị 35000 nằm ngoài phạm vi giá trị trên cho nên đã gây ra lỗi, kết quả đưa ra không chính xác. Có thể gợi ý cho HS chỉnh sửa khai báo kiểu dữ liệu để khắc phục hạn chế này.

b) Một trong những nội dung quan trọng của bài này là giúp HS luyện tập việc nhận biết và khai báo kiểu dữ liệu hợp lý cho các biến. Khai báo kiểu dữ liệu hợp lý một mặt sẽ giúp cho việc sử dụng bộ nhớ một cách tối ưu (ví dụ, với đại lượng chỉ nhận giá trị số tự nhiên không vượt quá 255 thì không cần thiết phải khai báo biến kiểu *integer*, mà chỉ cần kiểu *byte*), mặt khác giúp tránh lỗi tràn dữ liệu và dẫn đến kết quả sai. Trong các bài thực hành sau, GV nên lưu ý HS đến điểm này.

c) Với bài 2, trọng tâm của bài này là cho HS luyện tập với lệnh gán và thực hiện việc trao đổi giá trị của hai biến *x, y*. Đây là một công việc hay gặp trong lập trình và qua ví dụ này HS có thể hiểu rõ hơn về biến, cách sử dụng biến. Bài này cũng giới thiệu cách viết câu lệnh nhập nhiều dữ liệu từ bàn phím bằng một câu lệnh *readln* hoặc *read*.

Để thực hành bài này, có thể tiến hành như sau:

Cho HS gõ chương trình trong SGK, tiến hành dịch, chỉnh sửa và cho chạy chương trình.

Do không có thông báo cho người dùng về yêu cầu nhập giá trị tương ứng của các biến x , y nên HS có thể gặp khó khăn không biết nhập thế nào. GV cần hướng dẫn HS cú pháp của câu lệnh và cách nhập hai số nguyên (cách nhau bởi dấu cách) rồi nhấn phím Enter và quan sát kết quả.

Nên gợi ý cho HS cải tiến chương trình trên để hướng dẫn người dùng nhập giá trị cho x , y từ bàn phím. In ra màn hình giá trị của x , y vừa được người dùng nhập vào và in ra màn hình giá trị x , y sau khi đã trao đổi giá trị. Có thể tham khảo chương trình *TinhTien.pas* để thực hiện việc này.

Về việc trao đổi giá trị giữa biến x và biến y , có thể lấy ví dụ minh họa như việc muốn trao đổi nước giữa hai cốc. Giả sử có hai cốc nước, một cốc nước chứa nước màu đỏ, một cốc nước chứa nước màu xanh. Làm thế nào để trao đổi nước giữa hai cốc nước này? Đương nhiên là phải dùng cốc thứ ba làm trung gian để thực hiện điều này, cụ thể: Giả sử cốc X chứa nước màu đỏ, cốc Y chứa nước màu xanh và cốc Z là cốc trung gian, không chứa gì cả. Cách trao đổi nước chứa trong cốc X và cốc Y như sau:

Đổ nước màu đỏ trong cốc X sang cốc Z;

Đổ nước màu xanh trong cốc Y sang cốc X;

Đổ nước màu đỏ trong cốc Z sang cốc Y.

Sau khi thực hiện như trên nước trong hai cốc đã được trao sang nhau.

Việc trao đổi giá trị của biến cũng tương tự, trong chương trình đã phải sử dụng biến z làm biến trung gian để lưu giữ giá trị ban đầu của biến x . Cụ thể:

```
z:=x; {Lưu giá trị x cho biến z}
```

```
x:=y; {Giá trị của biến x được thay bằng giá trị của biến y}
```

```
y:=z; {Giá trị của biến y được thay bằng giá trị của biến z,  
      giá trị của biến z lúc này chính bằng giá trị của biến x  
      ban đầu}
```

* Có một khác biệt cần được lưu ý ở đây. Khi đổ nước ở cốc Y sang nước ở cốc X thì cốc X có nước còn cốc Y hết nước. Khác với khi gán $X:=Y$ thì giá trị biến X bằng giá trị biến Y, nhưng giá trị biến Y không mất. Nếu HS thắc mắc thì GV có thể giải thích điều này, nếu HS không thắc mắc GV không nên giải thích để tránh làm phức tạp vấn đề.

Chương trình sau khi chỉnh sửa có thể như sau:

```

Program hoan_doi;
var x,y,z:integer;
begin
  write('Nhap gia tri bien x = '); readln(x);
  write('Nhap gia tri bien y = '); readln(y);
  Writeln('Truoc trao doi, gia tri cua bien x: ', x);
  Writeln('Truoc trao doi, gia tri cua bien y: ', y);
  {Bat dau thuc hien trao doi}
  z:=x;
  x:=y;
  y:=z;
  {Ket thuc trao doi}
  Writeln('Sau trao doi, gia tri cua bien x = ', x);
  Write('Sau trao doi, gia tri cua bien y = ', y);
  readln
end.

```

Lưu ý, để HS dễ tiếp thu, trong chương trình trên đã lựa chọn cách viết nhiều lệnh đơn giản, mặc dù có thể ghép một số lệnh thành một lệnh để chương trình ngắn gọn hơn. Ví dụ, có thể thay hai lệnh:

```

Writeln('Truoc trao doi, gia tri cua bien x = ', x);
Writeln('Truoc trao doi, gia tri cua bien y = ', y);

```

bằng một lệnh như sau:

```

Writeln('Truoc trao doi, gia tri cua bien x=', x, ' y=', y);

```