

## Ghi bài 8 Tuần 6

### BÀI 4: SỬ DỤNG BIẾN VÀ HẲNG TRONG CHƯƠNG TRÌNH

#### 1. Biến là công cụ trong lập trình:

- *Biến: là đại lượng được đặt tên* 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.

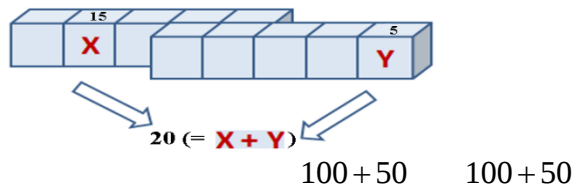
\* Ví dụ 1 :

In kết quả phép cộng  $15+5$  lên màn hình viết lệnh :

`writeln(15+5);`

In lên màn hình giá trị của biến  $x$  + giá trị của biến  $y$  viết lệnh :

`writeln(X+Y);`



\* Ví dụ 2 : Tính và in giá trị của các biểu thức  $\frac{100 + 50}{3}$  và  $\frac{100 + 50}{5}$  ra màn hình.

Cách làm :

`X = 100 + 50`

`Y = X/3`

`Z = X/5`

#### 2. Khai báo biến

- Việc khai báo biến gồm :
  - + Khai báo *tên biến*;
  - + Khai báo *kiểu dữ liệu* của biến.

**Dạng tổng quát :**

**Var** danh sách tên biến : kiểu của biến ;

\* Ví dụ :

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

Trong đó :

-`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**).

### **3. Sử dụng biến trong chương trình:**

Các thao tác được thực hiện trên các biến:

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

\* Câu lệnh gán giá trị cho biến có dạng:

Tên biến ← Biểu thức cần gán giá trị cho biến.

Trong đó: ← biểu thị cho phép gán.

\* Trong NNLT Pascal kí hiệu phép gán là dấu ‘:=’.

Ví dụ: *x:=5;*

Giá trị của biến còn có thể gán nhờ câu lệnh nhập liệu *Read* hoặc *Readln*.

Ví dụ: Nhập giá trị cho biến *m, n* bằng lệnh:

*Read(m,n);* hoặc *Readln(m,n);*

### **4. Hằng:**

\* Là đại lượng được đặt tên có giá trị không đổi trong suốt chương trình.

\* Khai báo hằng:

**Const tên hằng=giá trị;**

Ví dụ:

**Const** *pi* = 3.14;

*Bankinh* = 2;

- **Const:** Từ khóa của ngôn ngữ lập trình dùng để khai báo hằng

- Hằng **pi** được gán giá trị tương ứng là 3.14.

- Hằng **bankinh** được gán giá trị tương ứng là 2.

\* Chú ý: Ta không dùng câu lệnh để thay đổi giá trị của hằng (như đối với biến) ở bất kỳ vị trí nào trong chương trình.

## Phiếu học tập 8 Tuần 6

### Bài 4: Sử dụng biến để tính toán

#### I. Biến được khai báo và sử dụng trong chương trình với mục đích gì ?

.....

.....

.....

.....

.....

#### II. Khai báo biến:

Cú pháp

..... danh sách tên biến : .....

Vd: .....

Quy tắc đặt tên:

- Không .....
- Không .....
- Không .....

#### III. Phép gán trong pascal:

\_ Pascal dùng dấu ..... để thực hiện phép gán

|        |          |
|--------|----------|
| Toán   | $X = 5;$ |
| Pascal |          |

#### IV. Hằng số:

\_ Là 1 ..... đặc biệt dùng để lưu ..... không thay đổi.

Cú pháp ..... tên biến ..... giá trị;

VD: ..... PI ..... 3.14;

## Ghi bài 8 tuần 7

### TH3: KHAI BÁO VÀ SỬ DỤNG BIẾN

1. Cú pháp khai báo biến và hằng trong Pascal:

**Cú pháp khai báo biến:**

**Var** <danh sách biến>: <kiểu dữ liệu>;

Các biến trong danh sách phải cách nhau bằng dấu phẩy, các biến cùng kiểu khai báo cùng dòng

**Cú pháp khai báo hằng:**

**Const** tên hằng = giá trị của hằng ;

2. Cú pháp lệnh gán trong Pascal:

<tên biến>:= <biểu thức>;

Kí hiệu := để gán giá trị cho biến

3. Lệnh **read**(<danh sách biến>) hay **readln**(<danh sách biến>), trong đó danh sách biến là tên các biến đã khai báo, được sử dụng để nhập dữ liệu từ bàn phím. Sau khi nhập dữ liệu cần nhấn phím **Enter** để xác nhận. Nếu giá trị nhập vào vượt quá phạm vi của biến, nói chung kết quả tính toán sẽ sai.

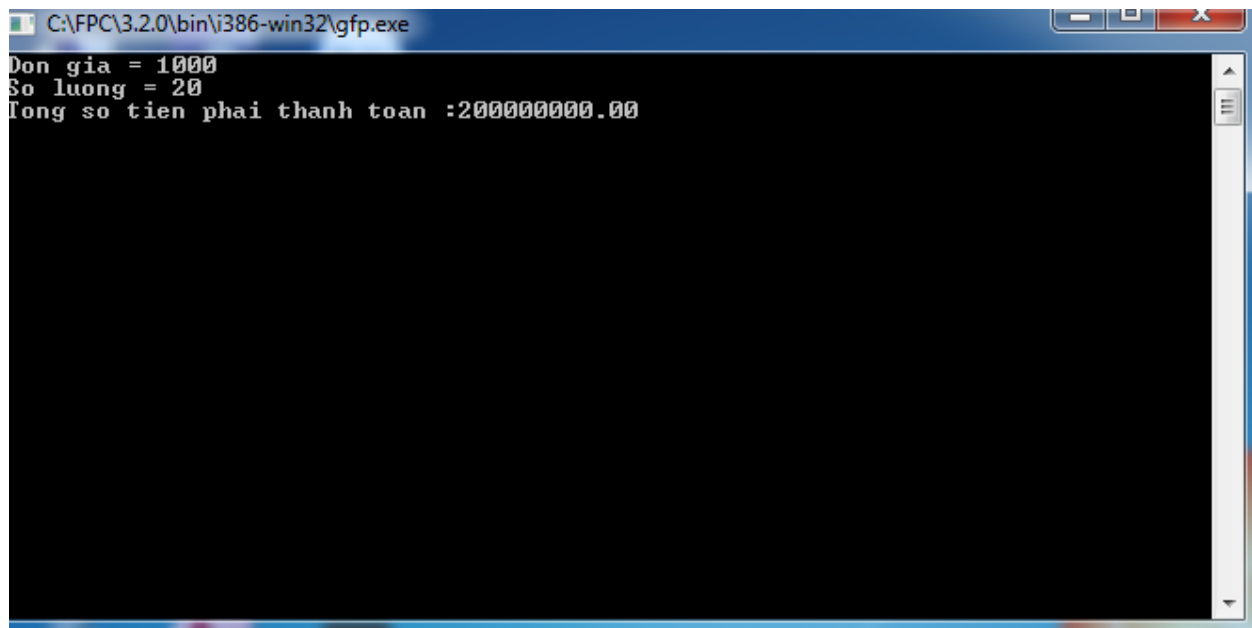
4. Nội dung chú thích nằm trong cặp dấu { và } bị bỏ qua khi dịch chương trình. Các chú thích đọc dùng để làm cho chương trình dễ đọc, dễ hiểu. Ngoài ra có thể sử dụng cặp các dấu (\* và \*) để tạo chú thích.

## Phiếu học tập 8 Tuần 7

Bài 1: trang 34

```
program Tinh_tien;  
  
uses    crt;  
  
var    soluong: integer;  
        dongia, thanh tien: real;  
        thongbao : string;  
        const phi = 10000;  
  
begin  
    clrscr;  
  
    thongbao := 'Tong so tien phai thanh toan :';  
    write('Don gia = ');  
    readln(dongia);  
  
    write('So luong = ');  
    readln(soluong);  
  
    thanh tien := soluong * dongia * phi;  
  
    writeln(thongbao, thanh tien:10:2);  
  
    readln;  
end.
```

Chạy:



```
C:\FPC\3.2.0\bin\i386-win32\gfp.exe
Don gia = 1000
So luong = 20
Tong so tien phai thanh toan :200000000.00
```

Tương tự: cho trường hợp

- đơn giá = 3500 số lượng = 200
- đơn giá = 18500 số lượng = 123