

Bài 1: MÁY TÍNH VÀ CHƯƠNG TRÌNH MÁY TÍNH

1. Viết chương trình - ra lệnh cho máy tính làm việc

Ví dụ rô - bốt nhặt rác

- Bước 1: Tiến 2 bước.
- Bước 2: Quay trái, tiến 1 bước.
- Bước 3: Nhặt rác.
- Bước 4: Quay phải, tiến 3 bước.
- Bước 5: Quay trái, tiến 2 bước.
- Bước 6: Bỏ rác vào thùng
- Một lệnh đơn giản không đủ để chỉ dẫn cho máy tính. Vì thế việc viết nhiều lệnh và tập hợp lại trong một chương trình giúp con người điều khiển máy tính một cách đơn giản và hiệu quả hơn.
- Chương trình máy tính là một dãy các lệnh mà máy tính có thể hiểu và thực hiện được.

2. Chương trình và ngôn ngữ lập trình

- Các dãy bit là cơ sở để tạo ra ngôn ngữ dành cho máy tính, được gọi là ngôn ngữ máy.
- Ngôn ngữ lập trình là ngôn ngữ dùng để viết các chương trình máy tính.
- Bước 1: Viết chương trình bằng ngôn ngữ lập trình.
- Bước 2: Dịch chương trình thành ngôn ngữ máy.

Bài 2: LÀM QUEN VỚI CHƯƠNG TRÌNH VÀ NGÔN NGỮ LẬP TRÌNH

1. Ngôn ngữ lập trình gồm những gì?

Ví dụ về chương trình.

- Ví dụ 1:

```
Program CT_Dau_tien;
```

```
Uses CRT;
```

```
Begin
```

```
Writeln('Chao cac ban');
```

```
End.
```

- Các câu lệnh được viết từ các chữ cái của bảng chữ cái đó.

+ Bảng chữ cái tiếng Anh: A --> Z.

+ Các kí hiệu phép toán: +, -, *, /.

+ Các dấu ' ', (), ...

2. Từ khoá và tên

Các từ khóa được quy định của ngôn ngữ lập trình: Program, uses, begin, end, ...

- Từ khoá của ngôn ngữ lập trình là từ dành riêng được viết bằng tiếng Anh, không được dùng các từ khóa này cho bất kì mục đích nào khác ngoài mục đích sử dụng do ngôn ngữ lập trình quy định.

- Quy tắc đặt tên:

+ Tên không được trùng với các từ khoá.

+ Tên không chứa dấu cách.

+ Tên không chứa các kí tự đặc biệt.

+ Tên không bắt đầu bằng số.

Ví dụ:

Tên hợp lệ: Stangia hoặc ban_kinh, ..

Tên không hợp lệ: Lop em hoặc 8^a, ..

3. Cấu trúc chung của chương trình

- Cấu trúc của một chương trình gồm 2 phần:

+ Phần khai báo:

. Khai báo tên chương trình.

. Khai báo thư viện.

+ Phần thân của chương trình gồm các câu lệnh mà máy tính cần thực hiện: Nằm trong cặp từ khoá BEGIN...END.

*Lưu ý: Phần thân là phần quan trọng nhất và bắt buộc phải có trong tất cả các chương trình.

4. Ví dụ về ngôn ngữ lập trình

- Khởi động Free Pascal -> Soạn thảo chương trình -> Lưu tên chương trình.Pas

- Dịch chương trình: Alt + F9

- Chạy chương trình: Ctrl + F9

- Quan sát kết quả: Alt + F5

Bài thực hành 1: LÀM QUEN VỚI FREE PASCAL

1. Bài tập 1

a. Khởi động Free Pascal bằng một trong hai cách:



- Cách 1: Nhấn đúp ↑

- Cách 2: Nháy đúp ↑ vào tên tệp **FreePascsl.exe** trong thư mục chứa tệp này. (thường là TP\BIN).

b. Quan sát màn hình Free Pascal.

c. Nhận biết các thành phần:

- Thanh bảng chọn, tên tệp đang mở, con trỏ, dòng trợ giúp phía dưới màn hình.

- Nhấn phím F10 để mở bảng chọn, sử dụng các phím mũi tên sang trái và sang phải để di chuyển qua lại giữa các bảng chọn.

d. Nhấn phím Enter để mở bảng chọn.

e. Quan sát các lệnh trong từng bảng chọn.

f. Sử dụng các phím mũi tên lên và xuống để di chuyển giữa các lệnh trong một bảng chọn.

g. Nhấn phím ALT + X để thoát khỏi Pascal.

- Cách 1: ALT + X.

- Cách 2: Chọn File → Exit.

Bài tập 2:

a. Soạn thảo, lưu, dịch và chạy một chương trình đơn giản.

Chương trình:

```
Program CT_Dau_Tien;
```

```
Uses CRT;
```

```
BEGIN
```

```
  CLRSCR;
```

```
  Writeln('Chao cac ban');
```

```
  Writeln('Toi la Free Pascal');
```

```
Readln;
```

```
END.
```

b. Lưu chương trình.

F2

c. Dịch chương trình.

Tổ hợp phím ALT + F9

d. Chạy chương trình.

Tổ hợp phím CTRL + F9

3. Bài 3:

Chỉnh sửa chương trình và nhận biết một số lỗi.

Lệnh **Writeln** thông báo ra màn hình và đưa con trỏ xuống đầu dòng tiếp theo.

Lệnh **Write** thông báo ra màn hình, nhưng không đưa con trỏ xuống đầu dòng tiếp theo

Câu lệnh **Clrscr** dùng để xóa màn hình và chỉ sử dụng được khi đã khai báo thư viện **crt**. Thư viện **crt** chứa các lệnh viết sẵn để thao tác với màn hình và bàn phím.

Bài 4: Hãy chỉnh sửa chương trình để in ra lời chào và tên của em, ví dụ

Chao cac ban

Tôi tên là Phạm Nhu Anh

Bài 3: CHƯƠNG TRÌNH MÁY TÍNH VÀ DỮ LIỆU

1. Dữ liệu và kiểu dữ liệu.

NNLT phân chia dữ liệu thành các kiểu khác nhau.

Kiểu dữ liệu xác định miền giá trị và các phép toán có thể thực hiện

Một số kiểu dữ liệu thường dùng nhất:

+Số nguyên

+Số thực

+Kí tự

+Xâu kí tự

Một số kiểu dữ liệu cơ bản của NNLT Pascal:

Kiểu	Phạm vi giá trị
byte	Số nguyên từ 0 → 255
integer	Số nguyên từ $-2^{15} \rightarrow 2^{15} - 1$. (-32768 → 32767)
real	Số thực có giá trị tuyệt đối từ $1.5 \times 10^{-39} \rightarrow 3.4 \times 10^{38}$ và số 0.
char	Một kí tự trong bảng chữ cái.
string	Xâu kí tự, tối đa gồm 255 kí tự.

Chú ý: Dữ liệu kiểu kí tự và kiểu xâu trong Pascal được đặt trong cặp dấu nháy đơn.

2. Các phép toán với dữ liệu kiểu số.

VD:

Kí hiệu	Phép toán	Kiểu dữ liệu
+	cộng	số nguyên, số thực
-	trừ	số nguyên, số thực
*	nhân	số nguyên, số thực
/	chia	số thực
div	chia lấy phần nguyên	số nguyên
mod	chia lấy phần dư	số nguyên

$$5/2 = 2.5;$$

$$-12/5 = -2.4.$$

$$5 \text{ div } 2 = 2;$$

$$-12 \text{ div } 5 = -2$$

$$5 \bmod 2 = 1;$$

$$-12 \bmod 5 = -2$$

Biểu thức số học

Ngôn ngữ toán

$$a \times b - c + d$$

$$15 + 5 \times \frac{a}{2}$$

$$\frac{x+5}{a+3} - \frac{y}{b+5} (x+2)^2$$

Ngôn ngữ pascal

$$a*b-c+d$$

$$15+5*(a/2)$$

$$(x+5)/(a+3)-y/(b+5)*(x+2)*(x+2)$$

Quy tắc tính các biểu thức số học:

- Trong cặp dấu ngoặc

*, /, div, mod

+ - thứ tự từ trái sang phải.

Chú ý: Trong Pascal chỉ sử dụng cặp dấu ngoặc tròn () để gộp các phép toán

3. Các phép so sánh.

- Kí hiệu: =, >, <, ≠, ≥, ≤

- Khi thực hiện cá phép so sánh sẽ cho kết quả là đúng hoặc sai.

- Ví dụ: $15 + 8 > 20 - 2$

Kí hiệu trong Pascal	Phép so sánh	K.hiệu trong toán học
=	Bằng	=
<>	Khác	≠
<	Bé	<
>	Lớn	>
<=	Bé hơn hoặc bằng	≤
>=	Lớn hơn hoặc bằng	≥

4. Giao tiếp người - máy tính.

a. Thông báo kết quả tính toán.

- Write.

- Writeln.

Ví dụ: Write ('Diện tích hình chu nhật là:', s);

b. Nhập dữ liệu.- Read.

- Readln.

- **Ví dụ:** Readln (a,b);

c. Thông báo kết quả tính toán.

- Delay (x);

- Ví dụ: Delay (5000);

d. Hộp thoại

Bài thực hành 2: VIẾT CHƯƠNG TRÌNH ĐỂ TÍNH TOÁN

Bài tập 1.

a. Gõ các biểu thức toán học dưới dạng biểu thức trong Pascal.

$$- 15 * 4 - 30 + 12$$

- $(10 + 5) / (3 + 1) - 18 (5 + 1)$
- $(10 + 2) * (10 + 2) / (3 + 1)$
- $(10 + 2) * (10 + 2) - 24 / (3 + 1)$

b. Chương trình:

BEGIN

```
Writeln('15*4-30+12=',15*4-30+12);
Writeln('(10+5)/(3+1)-18(5+1=',
        (10+5)/(3+)-18(5+1));
Writeln('(10+2)*(10+2)/(3+)=',
        (10+2)*(10+ 2)/(3+1));
Writeln('(10+2)*(10+2)-24/(3+)=',
        (10 + 2) * (10 + 2) - 24 / (3 + 1);
Readln;
```

END.

Bài tập 2.

Chương trình:

Uses CRT;

BEGIN

CLRSCLR;

```
Writeln('16/3=',16/3);
Writeln('16 div 3=',16 div 3);
Writeln('16 mod 3=',16 mod 3);
Writeln('16 mod 3=',16-(16 div 3)*3);
Writeln('16 div 3=',(16-(16 mod 3))/3);
Readln;
```

END.

Bài tập 3.

```
Writeln((10+5)/(3+1)-18 (5+1):4:2);
Writeln((10+2)*(10+2)/(3+):4:2);
Writeln ((10+2)*(10+2)-24/(3+):4:2);
```

Bài 4: SỬ DỤNG BIẾN VÀ HẰNG TRONG CHƯƠNG TRÌNH

1. Biến là công cụ trong lập trình.

- Mọi dữ liệu nhập vào đều được lưu trong bộ nhớ của máy tính.
- 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 gọi là giá trị của biến.

Ví dụ 1: In kết quả của tổng $15 + 5$

Cách 1: Writeln (15 + 5);

Cách 2: Writeln (x+y);

Ví dụ 2: Tính giá trị của biểu thức:

$100 + 50/3$ và $100 + 50/5$

$x = 100 + 50$

$y = x / 3$

$z = x / 5$

2. Khai báo biến.

- **Cú pháp:**

VAR <tên biến> : <Tên kiểu dữ liệu của biến>;

Các biến dùng trong chương trình cần phải khai báo ngay trong phần khai báo của chương trình.

Ví dụ: Var m, n: interger;

S, dientich: Real;

Thong_bao:string;

Trong đó:

Var là từ khóa của Pascal dùng để khai báo biến,

m, n là tên các biến có kiểu số nguyên (Integer),

S, dientich là tên các biến có kiểu số thực (Real),

Thong_bao là tên biến có kiểu xâu (string).

3. Sử dụng biến trong chương trình.

- Các thao tác thực hiện với biến:

+ Gán giá trị cho biến.

+ Tính toán với giá trị của biến.

- **Cú pháp câu lệnh gán:**

<Tên biến> := <Biểu thức>;

Ví dụ 3: Bảng mô tả phép gán giá trị và tính toán với các biến trong Pascsal

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

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

4. Hằng

- Là đại lượng có giá trị không đổi trong suốt quá trình thực hiện chương trình.

- **Cú pháp:**

CONST <tên hằng> = <Giá trị>;

Ví dụ: Const n = 45;

Const pi = 3.14;

Bankinh = 2;

Trong đó:

Const là từ khóa để khai báo hằng,

Các hằng pi, bankinh được gán giá trị tương ứng là 3.14 và 2

Bài thực hành 3: KHAI BÁO VÀ SỬ DỤNG BIẾN

Bài tập 1: Một cửa hàng cung cấp dịch vụ bán hàng thanh toán tại nhà. Khách hàng chỉ cần đăng ký số lượng mặt hàng cần mua, nhân viên cửa hàng sẽ trả hàng và nhận tiền thanh toán tại nhà khách hàng. Ngoài giá trị hàng

hoá, khách hàng còn phải trả thêm phí dịch vụ. Hãy viết chương trình Pascal để tính tiền thanh toán trong trường hợp khách hàng chỉ mua một mặt hàng duy nhất.

Chương trình:

```
Program Tinh_Tien;
```

```
Uses CRT;
```

```
  Var soluong: Integer;
```

```
      dongia, thanhtien: Real;
```

```
      thongbao: String;
```

```
  Const phi = 10000;
```

```
BEGIN
```

```
  CLRSCR;
```

```
  thong bao:= 'Tong so tien phai tra';
```

```
  Writeln('Don gia = ');
```

```
  Readln(dongia);
```

```
  Writeln('So luong = ');
```

```
  Readln(soluong);
```

```
  thanhtien:= soluong * dongia + phi;
```

```
  Writeln('thongbao, thanhtien:10:2);
```

```
  Readln;
```

```
END.
```

Bài tập 2: Viết chương trình nhập các số nguyên x và y, in giá trị của x và y ra màn hình. Sau đó hoán đổi giá trị của x và y rồi in lại ra màn hình giá trị của x và y.

Chương trình:

```
Program Hoan_doi;
```

```
Uses CRT;
```

```
  Var x,y,z: Integer;
```

```
BEGIN
```

```
  CLRSCR;
```

```
  Writeln('NHap x = ');
```

```
  Readln(x);
```

```
  Writeln('Nhap y = ');
```

```
  Readln(y);
```

```
  Writeln('Truoc trao doi, gia tri cua x la',x);
```

```
  Writeln('Truoc trao doi, gia tri cua y la',y);
```

```
  z:=x;
```

```
  x:=y;
```

```
  y:=z;
```

```
  Writeln('Sau trao doi, gia tri cua x la',x);
```

```
  Writeln('Sau trao doi, gia tri cua y la',y);
```

```
  Readln;
```

END.

ÔN TẬP

1. Cấu trúc của chương trình.

- Chương trình gồm có 2 phần:
- + Phần khai báo.
- + Phần thân.

2. Các khái niệm cơ bản và cách đặt tên.

a. Khái niệm:

- Chương trình.
- Biến.
- Hằng.

b. Cách đặt tên:

- Tên không trùng với từ khoá.
- Tên không chứa dấu cách.
- Tên không chứa các kí tự đặc biệt.
- Tên không bắt đầu bằng số.

3. Các câu lệnh cơ bản.

a. In kết quả ra màn hình.

- Write
- Writeln

b. Nhập dữ liệu.

- Read
- Readln

c. Khai báo biến.

- Cú pháp câu lệnh gán:
 <Tên biến> := <Biểu thức>;

d. Khai báo hằng.

- Cú pháp:
 CONST <tên hằng> = <Giá trị>;

4. Bài tập.

Chương trình:

```
Program Dien_tich_HT;  
  Var a,b,h,s: Real;  
BEGIN  
  Writeln('Nhap vao day lon, day be, chieu cao:');  
  Readln(a,b,h);  
  s:=((a+b)*h))/2;  
  Writeln('Dien tich hinh thang la:', s);  
  Readln;  
END.
```

1. Khai báo biến

Các biến dùng trong chương trình cần phải được khai báo ngay trong phần khai báo của chương trình.

- Khai báo biến bao gồm
- + Khai báo tên biến
- + Khai báo kiểu dữ liệu của biến

2. Hằng

Hằng là đại lượng có giá trị không đổi trong suốt quá trình thực hiện chương trình.

a. Khái niệm:

- Chương trình.
- Biến.
- Hằng.

b. Cách đặt tên:

- Tên không trùng với từ khoá.
- Tên không chứa dấu cách.
- Tên không chứa các kí tự đặc biệt.
- Tên không bắt đầu bằng số.

3. Các câu lệnh cơ bản.

a. In kết quả ra màn hình.

- Write
- Writeln

b. Nhập dữ liệu.

- Read
- Readln

c. Khai báo biến.

- Cú pháp câu lệnh gán:
 $\langle \text{Tên biến} \rangle := \langle \text{Biểu thức} \rangle;$

d. Khai báo hằng.

- Cú pháp:
 $\text{CONST } \langle \text{tên hằng} \rangle = \langle \text{Giá trị} \rangle;$

4. Bài tập.

Chương trình:

```

Program Dien_tich_HT;
  Var a,b,h,s: Real;
BEGIN
  Writeln('Nhap vao day lon, day be, chieu cao:');
  Readln(a,b,h);
  s:=((a+b)*h)/2;
  Writeln('Dien tich hinh thang la:', s);
  Readln;
END.

```

Bài 5: TỪ BÀI TOÁN ĐẾN CHƯƠNG TRÌNH

1. Xác định bài toán.

- Bài toán là một công việc hay một nhiệm vụ cần phải giải quyết.
- Để giải quyết bài toán cụ thể, cần xác định bài toán tức là xác định rõ các điều kiện cho trước và kết quả cần thu được.

Ví dụ 1:

a) Tính diện tích tam giác:

- Điều kiện: Một cạnh và đường cao tương ứng với cạnh đó.
- Kết quả: Diện tích hình tam giác.

b) Tìm đường đi tránh các điểm nghẽn giao thông.

- Điều kiện: Vị trí điểm nghẽn giao thông và các con đường có thể đi từ vị trí hiện tại.
- Kết quả: Đường đi tới vị trí cần tới nhưng không qua điểm nghẽn giao thông.

c) Bài toán nấu một món ăn.

- Điều kiện: Các thực phẩm.
- Kết quả: Một món ăn.

2. Quá trình giải bài toán trên máy tính

- Thuật toán: Là các bước (thao tác) để giải một bài toán.
- Dãy hữu hạn các thao tác cần thực hiện để giải một bài toán được gọi là *thuật toán*.
- Quá trình giải bài toán trên máy tính gồm các bước:
 - + Xác định bài toán: Từ phát biểu của bài toán, xác định thông tin đã cho (Input), thông tin cần tìm (Output).
 - + Mô tả thuật toán: Tìm cách giải bài toán và các thao tác (lệnh) cần phải thực hiện.
 - + Viết chương trình: Dựa vào thuật toán để viết chương trình bằng ngôn ngữ lập trình.

3. Thuật toán và mô tả thuật toán

Ví dụ 1: Pha trà mời khách

- Xác định INPUT và OUTPUT:
 - + INPUT: Trà, nước sôi, ấm, chén.
 - + OUTPUT: Chén trà đã pha để mời khách.
- Thuật toán:
 - + Bước 1: Tráng ấm, chén bằng nước sôi.
 - + Bước 2: Cho trà vào ấm.
 - + Bước 3: Rót nước sôi vào ấm và đợi khoảng 3 đến 4 phút.
 - + Bước 4: Rót trà ra chén để mời khách.

Ví dụ 2: Giải PT bậc nhất dạng tổng quát $bx + c = 0$.

- Xác định INPUT và OUTPUT:
 - + INPUT: Các số b, c .
 - + OUTPUT: Nghiệm của PT bậc nhất.
- Thuật toán:
 - + Bước 1: Nếu $b = 0$ chuyển tới bước 3.
 - + Bước 2: Tính nghiệm của PT $x = -c/b$ và chuyển tới bước 4.
 - + Bước 3: Nếu $c \neq 0$ thông báo PT đã cho vô nghiệm. Ngược lại nếu $c = 0$ thông báo PT có vô số nghiệm.
 - + Bước 4: Kết thúc.

Ví dụ 3: Làm món trứng tráng.

- Xác định INPUT và OUTPUT:
 - + INPUT: Trứng, dầu ăn, muối, hành. +OUTPUT: Trứng tráng.
- Thuật toán:
 - + Bước 1: Đập trứng, tách vỏ và cho trứng vào bát.
 - + Bước 2: Cho một chút muối và hành tươi thái nhỏ vào bát trứng. Dùng đũa khuấy mạnh cho đến khi đều.
 - + Bước 3: Cho một thìa dầu ăn vào chảo, đun nóng rồi đổ trứng vào. Đun tiếp trong khoảng 1 phút.
 - + Bước 4: Lật mặt trên của miếng trứng úp xuống dưới. Đun tiếp trong khoảng 1 phút.
 - + Bước 5: Lấy trứng ra đĩa.
- * Thuật toán là dãy các thao tác cần thực hiện theo một trình tự xác định để thu được kết quả cần tìm từ những điều kiện cho trước.

4. Một số ví dụ về thuật toán

Ví dụ 2: Tính diện tích hình A.

- Xác định INPUT và OUTPUT:
 - + INPUT: Số a là $1/2$ chiều rộng của hình chữ nhật và là bán kính của hình bán nguyệt, b là chiều dài của hình chữ nhật.
 - + OUTPUT: Diện tích của hình A.
- Thuật toán:
 - Bước 1: $S_1 := 2 * a * b$
 - Bước 2: $S_2 := (\pi * a * a) / 2$;
 - Bước 3: $S := S_1 + S_2$ và kết thúc

Ví dụ 3: Tính tổng 100 số tự nhiên đầu tiên.

- Xác định INPUT và OUTPUT:

+ INPUT: Dãy 100 số tự nhiên đầu tiên: 1, 2, 3 ..., 100

+OUTPUT: Giá trị của tổng $1 + 2 + \dots + 100$.

- Thuật toán:

Bước 1: $SUM := 0; i := 0;$

Bước 2: $SUM := SUM + i; i := i + 1$

Bước 3: Nếu $i \leq 100$, thì quay lại bước 2, ngược lại thông báo giá trị của SUM và kết quả thuật toán.

Ví dụ 4: Đổi giá trị của 2 biến x và y.

- Xác định INPUT và OUTPUT:

+ INPUT: Hai biến x và y có giá trị tương ứng là a và b.

+OUTPUT: Hai biến x, y có giá trị tương ứng là b và a.

- Thuật toán:

Bước 1: $z := x;$

Bước 2: $x := y;$

Bước 3: $y := z;$

4. Một số ví dụ về thuật toán

Ví dụ 5: So sánh 2 số thực.

- Xác định INPUT và OUTPUT:

+ INPUT: Hai số thực a và b.

+ OUTPUT: Kết quả so sánh.

- Thuật toán:

Bước 1: Nếu $a > b$ kết quả là a lớn hơn b và chuyển đến bước 3.

Bước 2: Nếu $a < b$ kết quả là a nhỏ hơn b. Ngược lại, kết quả là a bằng b.

+ Bước 3: Kết thúc thuật toán.

Ví dụ 6: Tìm số lớn nhất trong dãy A các số a_1, a_2, \dots, a_n cho trước.

- Xác định INPUT và OUTPUT:

+ INPUT: Dãy A các số a_1, a_2, \dots, a_n ($n \geq 1$).

+ OUTPUT: Giá trị $Max = \max\{a_1, a_2, \dots, a_n\}$

- Mô tả thuật toán:

Bước 1: $Max \leftarrow a_1; i \leftarrow 1.$

Bước 2: Nếu $a_i > MAX$, gán $MAX \leftarrow a_i$

Bước 3: $i \leftarrow i + 1.$

Bước 4: Nếu $i \leq n$, quay lại bước 2.

Bước 5: Thông báo giá trị MAX và kết thúc thuật toán.

Bài 6: CÂU LỆNH ĐIỀU KIỆN

1. Hoạt động phụ thuộc vào điều kiện..

- Những hoạt động chỉ được thực hiện khi một điều kiện cụ thể được xảy ra. Điều kiện thường là một sự kiện được mô tả sau từ "nếu".

- Kết quả kiểm tra đúng: Điều kiện được thoả mãn.

- Kết quả kiểm tra sai: Điều kiện không thoả mãn.

2. Điều kiện và phép so sánh.

Để so sánh ta thường sử dụng các kí hiệu toán học như: $<, >, =, \neq, \leq, \geq$.

- Phép so sánh được sử dụng để biểu diễn các điều kiện.

- **Ví dụ:** In ra màn hình giá trị lớn hơn trong hai số tương ứng với giá trị của 2 biến a và b.

+ Nếu $a > b$, in giá trị của a ra màn hình.

+ Ngược lại in giá trị của b

3. Cấu trúc rẽ nhánh.

- Máy tính thực hiện tuần tự các câu lệnh, từ câu lệnh đầu tiên cho đến câu lệnh cuối cùng.

Ví dụ 2: Mô tả hoạt động tính tiền cho khách.

+ Bước 1: Tính tổng số tiền T khách hàng đã mua sách.

+ Bước 2: Nếu $T \geq 100000$, số tiền phải thanh toán là $70\% \times T$.

+ Bước 3: In hóa đơn.

⇒ Cách phụ thuộc vào điều kiện như trên gọi là cấu trúc rẽ nhánh dạng thiếu.

Ví dụ 3: Mô tả hoạt động tính tiền cho khách.

+ Bước 1: Tính tổng số tiền T khách hàng đã mua sách.

+ Bước 2: Nếu $T \geq 100000$, số tiền phải thanh toán là $70\% \times T$; ngược lại số tiền phải thanh toán là $90\% \times T$.

+ Bước 3: In hóa đơn.

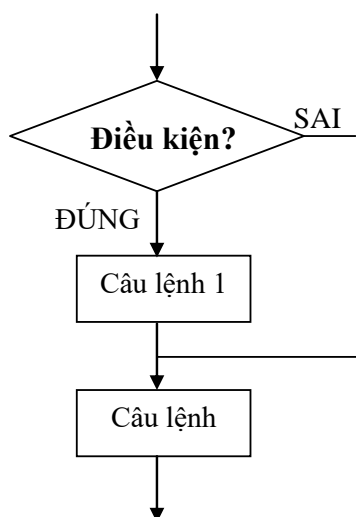
⇒ Cách phụ thuộc vào điều kiện như trên gọi là cấu trúc rẽ nhánh dạng đầy đủ.

4. Câu lệnh điều kiện.

- Dạng thiếu:

IF <Điều kiện> THEN <Câu lệnh>;

Chương trình kiểm tra điều kiện. Nếu điều kiện thỏa mãn, chương trình sẽ thực hiện câu lệnh sau từ khóa **then**. Ngược lại, câu lệnh đó bị bỏ qua.

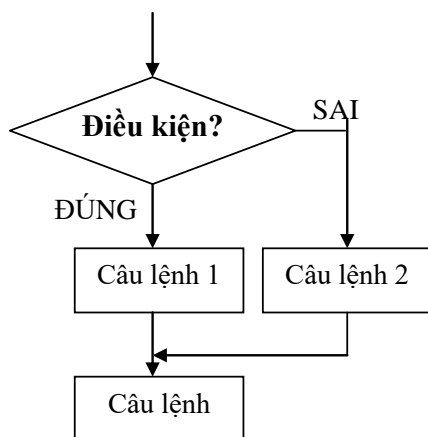


Hình 1.33a. Cấu trúc rẽ nhánh dạng thiếu

- Dạng đủ

IF <Điều kiện> THEN <Câu lệnh 1> ELSE <Câu lệnh 2>;

Chương trình sẽ kiểm tra điều kiện. Nếu điều kiện thỏa mãn, chương trình sẽ thực hiện câu lệnh 1 sau từ khóa **then**. Trong trường hợp ngược lại, câu lệnh 2 sẽ được thực hiện.



Hình 1.33b. Cấu trúc rẽ nhánh dạng đủ

Bài thực hành 4: SỬ DỤNG CÂU LỆNH ĐIỀU KIỆN

Bài tập 1: *Viết chương trình nhập hai số nguyên a và b khác nhau từ bàn phím và in hai số đó ra màn hình theo thứ tự không giảm.*

Chương trình:

Program SS_haiso;

Uses crt;

Var a,b: real;

Begin

CLRSCR;

Writeln('nhap vao hai so');

Readln(a,b);

If a < b then writeln(a, ' ', b)

else writeln(b, ' ', a);

Readln;

End.

Bài tập 2: *Viết chương trình nhập chiều cao của hai bạn Long và Trang, in ra màn hình kết quả so sánh chiều cao của hai bạn.*

Chương trình:

Program Ai_cao_hon;

Uses crt;

Var Long, Trang: real;

Begin

CLRSCR;

Write ('nhap vao chieu cao cua hai ban');

Readln(a,b);

If Long > Trang then writeln('Long cao hon')

Else If Long < Trang then writeln('Trang cao hon')

Else writeln('Hai ban cao bang nhau');

Readln;

End.

Bài tập 3: *Viết chương trình nhập ba số nguyên dương a, b, c từ bàn phím, kiểm tra và in ra màn hình kết quả kiểm tra ba số đó có thể là độ dài các cạnh của một tam giác hay không?*

Chương trình:

Program Ba_canh_tam_giac;

Uses crt;

Var a,b,c: real;

Begin

CLRSCR;

Writeln('nhap vao ba so');

Readln(a,b,c);

If (a + b > c) and (b + c > a) and (c + a > b) then writeln(a, b, c, 'la ba canh cua mot tam giac')

else writeln(a, b, c, 'khong phai la ba canh cua mot tam giac');

Readln;

End.