

ĐẠI HỌC HUẾ
TRUNG TÂM ĐÀO TẠO TỪ XA

LÊ MẠNH THẠNH (Chủ biên) – VÕ VĂN TUẤN DŨNG

GIÁO TRÌNH
**NGÔN NGỮ LẬP TRÌNH
PASCAL**

(Sách dùng cho hệ đào tạo từ xa)

HUẾ – 2007

MỤC LỤC

CHƯƠNG 1	6
GIỚI THIỆU CHUNG	6
1.1. MỞ ĐẦU	6
1.2. CÁC KHÁI NIỆM CƠ BẢN CỦA TURBO PASCAL	9
1.3. CÁC BƯỚC ĐỂ CHẠY MỘT CHƯƠNG TRÌNH TURBO PASCAL.....	11
CHƯƠNG 2	15
CÁC KIỂU DỮ LIỆU ĐƠN GIẢN.....	15
2.1. KIỂU DỮ LIỆU TRONG NGÔN NGỮ LẬP TRÌNH.....	15
2.2. CÁC KIỂU DỮ LIỆU ĐƠN GIẢN	15
CHƯƠNG 3	21
HÀNG, BIẾN, KIỂU, BIỂU THỨC, CÂU LỆNH, LỆNH GÁN.....	21
3.1. ĐỊNH NGHĨA HÀNG	21
3.2. BIẾN VÀ KHAI BÁO BIẾN	22
3.3. ĐỊNH NGHĨA KIỂU DỮ LIỆU	22
3.4. BIỂU THỨC.....	23
3.5. CÂU LỆNH (STATEMENT)	24
3.6 LỆNH GÁN (ASSIGNMENT STATEMENT)	24

CHƯƠNG 4	27
XUẤT NHẬP DỮ LIỆU	27
4.1. XUẤT DỮ LIỆU	27
4.2. NHẬP DỮ LIỆU TỪ BÀN PHÍM	31
CHƯƠNG 5	36
LỆNH CÓ CẤU TRÚC	36
5.1. CÂU LỆNH GHEP (COMPOUND STATEMENT)	36
5.2. CÂU LỆNH LỰA CHỌN	36
5.3. CÂU LỆNH LẬP	41
CHƯƠNG 6	52
Kiểu TẬP HỢP VÀ KIỂU MẢNG	52
6.1. KIỂU TẬP HỢP (SET TYPE)	52
6.2. KIỂU MẢNG (ARRAY TYPE)	54
CHƯƠNG 7	59
CHƯƠNG TRÌNH CON	59
7.1. KHÁI NIỆM VỀ CHƯƠNG TRÌNH CON	59
7.2. MỘT SỐ HÀM VÀ THỦ TỤC CỦA TURBO PASCAL	62
CHƯƠNG 8	68
Kiểu CHUỖI KÍ TỰ	68

8.1. KHAI BÁO KIỂU CHUỖI.....	68
8.2. CÁC PHÉP TOÁN TRÊN CHUỖI KÍ TỰ	69
8.3. CÁC THỦ TỤC TRÊN CHUỖI KÍ TỰ	70
8.4. CÁC HÀM TRÊN CHUỖI.....	71
CHƯƠNG 9.....	75
KIỂU BẢN GHI VÀ KIỂU TẬP TIN.....	75
9.1. KIỂU BẢN GHI (RECORD TYPE).....	75
9.2. KIỂU TẬP TIN (FILE TYPE).....	83
CHƯƠNG 10.....	100
KIỂU CON TRỎ VÀ BIẾN ĐỘNG	100
10.1. BIẾN CON TRỎ.....	100
10.2. BIẾN ĐỘNG (DYNAMIC VARIABLE).....	101
10.3. DANH SÁCH LIÊN KẾT (LINKED LIST)	104

LỜI MỞ ĐẦU

Ngày nay ở nước ta, các thành quả của Công nghệ Thông tin đã và đang được đưa vào ứng dụng trong mọi cơ quan xí nghiệp và do đó nhiệm vụ đào tạo nhân lực Công nghệ Thông tin đã trở thành cấp thiết đối với ngành Giáo dục và Đào tạo. Chúng ta phải đồng thời tiến hành 3 mục tiêu: mở mang dân trí, đào tạo nhân lực và phát triển nhân tài. Vấn đề đặt ra là cần phải xây dựng chương trình thích hợp theo từng mục tiêu, cho từng loại hình, từng đối tượng. Trong nhiều môn học thuộc chuyên ngành Công nghệ Thông tin, Ngôn ngữ lập trình Pascal là môn học được dùng cho nhiều đối tượng và nhiều cấp học với lí do: đây là một ngôn ngữ giải thuật có tính chặt chẽ vừa có thể sử dụng như là một ngôn ngữ giải thuật toán, vừa là ngôn ngữ phổ biến, thích hợp với nhiều ứng dụng khoa học, kĩ thuật, quản lí.

Giáo trình này được soạn thảo dựa trên cơ sở Turbo Pascal phiên bản 5.0 và chủ yếu dành cho các sinh viên Đại học Từ xa. Tuy nhiên, nó cũng có thể dùng làm tài liệu tham khảo và giảng dạy cho sinh viên của các trường Đại học, Trung học chuyên nghiệp, kể cả các trường Phổ thông trung học và các Trung tâm đào tạo khác.

Tác giả rất mong sự góp ý của các độc giả về nội dung, hình thức và sai sót trong giáo trình. Xin chân thành cảm ơn.

Các tác giả

Chương 1

GIỚI THIỆU CHUNG

1.1. Mở đầu

1.1.1. Cấu tạo chính của máy tính điện tử

Các máy tính điện tử nói chung có các bộ phận sau đây:

- Bộ điều khiển trung tâm CPU (Central Processing Unit) là bộ phận thực hiện các phép tính số học và logic bằng việc phân tích chức năng của các lệnh.
- Bộ nhớ trong (Main Memory) là nơi chứa các chương trình và số liệu.
- Thiết bị vào (Input Device) là bộ phận đưa chương trình và dữ liệu vào máy.
- Thiết bị ra (Output Device) là bộ phận đưa kết quả từ bộ nhớ ra ngoài.
- Bộ nhớ phụ (Auxiliary Memory) được dùng để lưu trữ chương trình và số liệu như bộ nhớ chính song dung lượng nhớ lớn hơn và truy cập chậm hơn.

Đối với máy PC (Personal Computer), các bộ phận chính là:

- Bộ xử lý trung tâm CPU bao gồm 3 khối : Khối số học và logic ALU (Arithmetic Logic Unit) có chức năng thực hiện các phép toán số học, so sánh, dịch chuyển. Khối điều khiển CU (Control Unit) có chức năng phân tích lệnh và điều khiển hoạt động của CPU. Các thanh ghi (Register) lưu trữ thông tin phục vụ trực tiếp cho tính toán và điều khiển.

Bộ nhớ trong bao gồm 2 phần chính: ROM (Read Only Memory) là bộ nhớ có cài sẵn chương trình phục vụ cho việc vào ra – gọi là BIOS (Basic Input – Output System). RAM là bộ nhớ cho phép truy cập tức thời, Chương trình và số liệu được lưu trữ tạm thời trong RAM trong quá trình thực hiện, khi tắt máy chương trình và số liệu trong RAM cũng biến mất.

- Bộ nhớ phụ của máy PC là các ổ đĩa (Disk Driver). Có 3 loại chính: đĩa cứng HDD (Hard Disk Driver – thông thường chỉ có một (hoặc hai) đĩa với dung lượng lớn và thường được cố định trong máy, được kí hiệu là C (hoặc C, D nếu có hai ổ đĩa); đĩa mềm FDD (Floppy Disk Driver) là loại ổ đĩa dùng cho đĩa mềm có dung lượng nhỏ và có thể thay đổi thường xuyên; ổ đĩa chỉ đọc CD – ROM (Compact Disk ROM) được cài đặt máy để đọc các đĩa CD.
- Thiết bị vào đối với máy PC rất đa dạng, thông dụng nhất là bàn phím. Ngoài ra còn có các loại khác như con chuột (Mouse) để đưa vào các tín hiệu điều khiển, máy quét (scanner) để nhập vào máy các hình ảnh.
- Thiết bị ra chủ yếu là màn hình (monitor), máy in (printer).

1.1.2. Thuật toán và chương trình

a. Ngôn ngữ lập trình

Ngôn ngữ cơ sở để lập chương trình cho máy là ngôn ngữ máy, đó là một bộ lệnh, mỗi lệnh được biểu diễn trong máy bằng một dãy các chữ số nhị phân chỉ cho máy các công việc cần thực hiện. Các lệnh được thể hiện ra bên ngoài bằng các số hệ cơ số 8 hoặc cơ số 16.

Để tiện cho người sử dụng, người ta xây dựng các ngôn ngữ trung gian giao tiếp giữa người và máy như Assembler, Basic, Angol, Fortran, Pascal, C, ...

Để máy hiểu được các ngôn ngữ trên, cần phải có các chương trình dịch để dịch các ngôn ngữ lập trình cấp cao ra ngôn ngữ máy. Chương trình dịch có hai loại: Biên dịch (Compiler) và thông dịch (Interpreter).

b. Thuật toán

Thuật toán là sự phân tích bài toán thành một số hữu hạn bước giải mà theo thứ tự đó ta có thể lập trình cho máy.

↪ **Ví dụ 1.** Để xây dựng chương trình giải phương trình bậc 2: $ax^2 + bx + c = 0$ ($a, b, c \neq 0$) ta phân tích như sau:

Bước 1. Đọc các hệ số a, b, c .

Bước 2. Tính $\Delta = b^2 - 4ac$.

Bước 3. Nếu $\Delta < 0$ thì trả lời : “*phương trình không có nghiệm thực*” chuyển sang Bước 5, nếu ngược lại chuyển sang Bước 4.

Bước 4. Nếu $\Delta = 0$ thì trả lời: “*phương trình có nghiệm kép $X_1 = X_2 = -b/2a$* ”, nếu ngược lại thì trả lời: “*Phương trình có hai nghiệm thực là $X_1 = (-b + \text{SQRT}(\Delta))/2a$, $X_2 = (-b - \text{SQRT}(\Delta))/2a$* ”, chuyển sang Bước 5.

Bước 5. Dừng máy.

c. Chương trình

Chương trình là một dãy hữu hạn các lệnh hoặc các chỉ thị của một ngôn ngữ lập trình nào đó được viết theo một trật tự nhất định nhằm giải quyết một lớp bài toán đã cho.

Ví dụ 2. Chương trình viết bằng ngôn ngữ Pascal sau đây giải phương trình bậc 2 với thuật toán đã chỉ ra ở trên.

Program GPTB2;

Var a, b, c, x1, x2, delta: Real;

Begin

 Readln(a, b, c); delta:=b^2-4*a*c;

 If delta < 0 then Writeln(“Phuong trinh khong co nghiem thuc”)

 else

```

If delta = 0 then
    Writeln("Phuong trinh co nghiem kep x1 = x2 =", -b/2a)
else
    x1 := (-b+sqrt(delta))/2 a;
    x2 := (-b-sqrt(delta))/2 a;
    Writeln("Phuong trinh co 2 nghiem x1=", x1, "x2=", x2)
End

```

End.

Chương trình được viết sẵn để giải quyết một loạt bài toán gọi là chương trình mẫu mà mỗi lần sử dụng nó chỉ việc đưa vào các giá trị thay cho các tham số. Chẳng hạn, trong ngôn ngữ lập trình Pascal hàm mẫu SQRT(x) là chương trình tính căn bậc 2 của x, hàm EXP(x) là chương trình con tính e^x , LN(x) tính $\ln(x)$,...

1.1.3. Giới thiệu về Pascal và Turbo Pascal

Pascal là ngôn ngữ lập trình cấp cao do Niklaus Wirth, giáo sư điện toán trường Đại học Kỹ thuật Zurich (Thụy Sĩ) đề xuất vào năm 1970. Lúc đầu mục đích của Wirth thiết kế Pascal là để giảng dạy lập trình. Nhưng trong quá trình phát triển, Pascal đã phát huy được ưu điểm của mình và tỏ ra hơn hẳn nhiều ngôn ngữ cấp cao khác, Pascal đã trở thành một ngôn ngữ mạnh được ứng dụng trong nhiều lĩnh vực khác nhau. Các tổ chức và công ty chuyên về máy tính dựa trên Pascal chuẩn đã phát triển thêm và tạo ra các chương trình dịch ngôn ngữ Pascal với nhiều phần thêm bớt khác nhau. Chẳng hạn như:

- Turbo Pascal của hãng Borland
- Quick Pascal của hãng Microsoft.
- UCSD Pascal (University of California at San Diego), của trường Đại học California.
- Ansi Pascal (American National Standard Institute), của Viện các tiêu chuẩn Quốc gia Mỹ.

Trong nhiều chương trình dịch Pascal, Turbo Pascal đã tỏ ra có nhiều ưu điểm và hiện nay được xem như là một trong những ngôn ngữ lập trình cấp cao phổ biến trên thế giới được sử dụng rộng rãi trong lĩnh vực giảng dạy và lập trình chuyên nghiệp.

Chỉ trong vòng mấy năm, Turbo Pascal được cải tiến qua nhiều phiên bản: 1.0, 2.0, 3.0, 4.0, 5.0, 5.5, (1989), 6.0 (1990), 7.0 (1992).

Hiện nay Turbo Pascal 7.0 có rất nhiều bổ sung, cải tiến so với các phiên bản trước, song mới bắt đầu tiếp cận với Turbo Pascal, bạn nên sử dụng Turbo Pascal 5.5 vì nó tương đối đơn giản, dễ sử dụng và không đòi hỏi máy có cấu hình mạnh.

1.1.4. Các tập tin chính của Turbo Pascal

Để chạy được chương trình Pascal thông thường cần sử dụng các tập tin chủ yếu sau:

Turbo.exe : Soạn thảo, dịch và liên kết chương trình.

Turbo.tpl : Tập tin thư viện, lưu trữ các đơn vị chuẩn để chạy với Turbo.exe (TPL : Turbo Pascal Library).

Với 2 tập tin trên chúng ta có thể bắt đầu viết được chương trình. Sau đó nếu muốn sử dụng đồ họa thì thêm các tập tin sau:

Graph.tpu: Đơn vị chương trình chứa các chương trình con đồ họa.

Các tập tin có phần mở rộng CHR (Sans.chr, Trip.chr,...) chứa các font chữ trong chế độ đồ họa.

Các tập tin có phần mở rộng BGI (Ega.bgi, Vga.bgi,...) để điều khiển các loại màn hình tương ứng khi dùng đồ họa.

1.2. Các khái niệm cơ bản của Turbo Pascal

1.2.1. Bộ kí tự của Turbo Pascal

Ngôn ngữ lập trình Pascal dựa vào bộ kí tự sau:

- Các chữ cái: 26 chữ hoa (A, B, C, D, ..., Z) và 26 chữ thường (a, b, c, d, ..., z).
- Các chữ số thập phân: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9.
- Các dấu toán học thông dụng: + - * / = < > () .
- Dấu gạch nối _ (khác với dấu trừ (-)).
- Các kí hiệu đặc biệt: . , ; ! ? : ‘ “ { } [] % @ & # \$ ^.

1.2.2. Từ khoá (Key word)

Trong Turbo Pascal có một từ cho sẵn, có ngữ nghĩa đã được xác định và tuân theo một cấu trúc ngữ pháp nhất định gọi là từ khoá. Sau đây là một số từ khoá quen thuộc, thường gặp trong Turbo Pascal: BEGIN, END, PROGRAM, CASE, OF, FOR, TO, DOWNTON, DO, WHILE, REPEAT, UNTIL, WITH, EXIT, GOTO, AND, OR, NOT, IN, DIV, MOD.

☞ **Chú ý.** Turbo Pascal không phân biệt kí tự in thường (lower case char) hay in hoa (upper case char). Chẳng hạn, các cách viết sau có ý nghĩa hoàn toàn như nhau: BEGIN, begin, Begin, beGIN...

1.2.3. Tên chuẩn

Tên chuẩn là tên đã được định nghĩa sẵn trong Turbo Pascal, nhưng người lập trình có thể định nghĩa lại nếu muốn. Sau đây là một số tên chuẩn:

- Tên hằng chuẩn: False, True, MaxInt, ...
- Tên kiểu chuẩn: Boolean, Char, Integer, Word, Read, Byte, Text ...
- Tên hàm chuẩn: Abs, Arctan, Chr, Cos, Sin, Exp, Ln, Sqr, Sqrt ...
- Tên thủ tục chuẩn: Read, Readln, Write, Writeln, ...

1.2.4. Tên (*identifier*)

Tên là một dãy các chữ cái, chữ số và dấu nối (_) bắt đầu không phải là chữ số và được dùng để đặt cho các đại lượng trong chương trình như tên hằng, tên kiểu dữ liệu, tên biến, tên mảng, tên hàm, tên chương trình ...

Số kí tự của tên gọi bị hạn chế trong Pascal chuẩn là 8 còn trong Turbo Pascal số kí tự của tên tối đa là 127. Tên không được đặt trùng với từ khoá.

Các ví dụ đúng về tên: PT_Bac_2, Delta, a_1.

Các ví dụ sai về tên: PT Bac 2 (có kí tự trống), 3ABC (kí tự đầu tiên là chữ số), g(x) (sử dụng dấu ()), Label (Trùng với từ khoá).

☞ **Chú ý.** Khi viết chương trình ta nên đặt các tên sao cho chúng nói lên được ý nghĩa của đối tượng mà chúng biểu thị để giúp cho chương trình được sáng sủa và dễ hiểu.

1.2.5. Cấu trúc tổng quát của một chương trình Turbo Pascal

Một chương trình Turbo Pascal đầy đủ gồm ba phần sau:

Phần 1. Phần tiêu đề của chương trình. Phần này được bắt đầu bằng từ khoá PROGRAM, tiếp theo là tên chương trình do người lập trình đặt ra và kết thúc bởi dấu chấm phẩy (;)

Phần 2. Phần khai báo bao gồm các khai báo theo thứ tự sau

- Khai báo nhãn: Label ... ;
- Khai báo sử dụng Unit: Uses ... ;
- Khai báo các hằng: Const ... ;
- Khai báo kiểu dữ liệu: Type ... ;
- Khai báo các biến: Var ... ;
- Khai báo các chương trình con thủ tục hay hàm:
 Procedure ... ;
 Function ... ;

Phần 3. Thân chương trình, bao gồm một dãy các lệnh thực hiện (có thể là lệnh đơn hoặc ghép) được viết cách nhau bởi dấu chấm phẩy, và bắt đầu bởi Begin và kết thúc bởi End.

BEGIN

..... { Các lệnh của chương trình }

END.

└─> Dấu chấm báo kết thúc chương trình.

Chú ý.

- Phần 1 chiếm 1 dòng và có thể có hay không.

- Phần 2 tùy theo chương trình cụ thể mà các khai báo có thể có hay không và có thể lặp lại một số lần.
- Phần 3 bắt buộc phải có đối với mọi chương trình.
- Lời giải thích: Turbo Pascal cho phép người lập trình có thể đưa vào văn bản chương trình các lời giải thích, ghi chú để làm cho chương trình dễ đọc, dễ hiểu hơn nhưng không ảnh hưởng đến sự thực hiện của chương trình. Các lời giải thích được đặt giữa hai dấu { } hoặc giữa 2 cụm dấu (* *) và có thể viết trên 1 dòng hay nhiều dòng.
- Dấu chấm phẩy (;) dùng để ngăn cách các câu lệnh của Turbo Pascal (không phải là dấu kết thúc lệnh). Tuy nhiên, nếu dấu chấm phẩy đứng ngay trước từ khoá END có thể bỏ.

1.3. Các bước để chạy một chương trình Turbo Pascal

1.3.1. Các bước thực hiện

- **Bước 1.** Khởi động Turbo Pascal bằng lệnh TURBO (⇩). Khi đó màn hình làm việc của Turbo Pascal xuất hiện. Dòng trên cùng màn hình là bảng chọn chính.

<u>F</u> ile	<u>E</u> dit	<u>R</u> un	<u>C</u> ompile	<u>O</u> pion	<u>D</u> ebug	<u>B</u> reak/Watch
--------------	--------------	-------------	-----------------	---------------	---------------	---------------------

- **Bước 2.** Đặt tên tập tin chương trình.

Chọn mục **File** trong bảng chọn chính, khi đó sẽ xuất hiện bảng chọn dọc. Trong bảng này chọn **Load**, tiếp theo màn hình xuất hiện dòng:

```
Load file name (*.PAS) :
_
```

Đưa vào tên tập tin chương trình cần soạn thảo rồi ấn Enter, có 3 khả năng xảy ra:

Load	F3
Pick	Alt - F3
New	
Save	
Write to	
Directory	
Change Dir	
Dos shell	
Quit	Alt - X

- Tập tin đã có trên đĩa : Tập tin chương trình này được nạp vào RAM và văn bản chương trình hiện lên màn hình.
- Tập tin chưa có trên đĩa : màn hình trống hiện ra để soạn thảo chương trình mới.
- Trong tên tập tin có dùng kí tự * hay ? : Tên các tập tin tương tự tên đã cho (bằng cách dùng các phím mũi tên và gõ Enter).

☞ **Chú ý.**

- Để chọn mục **File** có thể ấn Alt - F hoặc đưa điểm sáng đến tên **File** trên bảng chọn chính rồi gõ Enter.
- Để chọn **Load** trên bảng chọn dọc ta có thể đưa điểm sáng về từ **Load** rồi gõ Enter.

- **Bước 3.** Soạn thảo văn bản chương trình.

Một số thao tác cần thiết khi soạn thảo chương trình:

** Các phím dịch chuyển con trỏ*

- 4 mũi tên : chuyển sang trái, chuyển sang phải một kí tự, chuyển lên chuyển xuống một dòng theo hướng mũi tên.
- Enter thực hiện việc chuyển sang dòng mới.
- Home và End chuyển con trỏ về đầu dòng và cuối dòng.
- Ins : chuyển chế độ ghi chèn và ghi đè.
- Backspace: xoá kí tự trước con trỏ.
- Del: xoá kí tự sau con trỏ.
- Ctr + Y: xoá một dòng.

** Các thao tác khối :*

- Chọn khối: Dùng mũi tên đưa con trỏ về vị trí đầu khối, giữ phím shift và dùng các mũi tên để chuyển đến vị trí cuối khối, thả phím shift. Khối được chọn có màu sáng hơn vùng không được chọn.
- Xoá khối đã chọn : ấn phím Delete.
- Xoá khối đã chọn và đưa nội dung khối vào vùng đệm: ấn Shift + Del hoặc chọn thực đơn **Edit/Cut**.
- Sao chép nội dung của khối đã chọn vào vùng đệm: ấn Ctrl + Ins.
- Sao chép nội dung vùng đệm vào vùng bắt đầu từ vị trí con trỏ : ấn Shift + Ins.

• **Bước 4.** Lưu văn bản.

Khi văn bản đã đánh xong cần lưu vào trong đĩa ta làm như sau:

- Ấn Alt + F để mở thực đơn **File** hoặc ấn phím F3.
- Dùng mũi tên đưa dòng sáng về **Save as** và ấn Enter (hoặc ấn phím A) để mở hộp thoại Save As.
- Đặt tên tập tin với đầy đủ đường dẫn hoặc dùng phím Tab chuyển điểm nháy đến hộp Dir/Drive để chọn ổ đĩa, thư mục (nếu cần).
- Ghi tên tập tin vào hộp File name.
- Ấn phím Enter.

Nếu tập tin văn bản đã được đặt tên thì chỉ cần ấn Alt + F và chọn **Save** hoặc phím S.

Muốn ghi văn bản chương trình vào tập tin đang mở có thể ấn phím F2.

• **Bước 5.** Biên dịch và thực hiện chương trình : gõ phím Ctrl – F9.

Khi biên dịch, máy sẽ kiểm tra ngữ pháp của từng lệnh, gặp lệnh sai hay nhập sai máy dừng lại, trên màn hình có thông báo nguyên nhân sai của lỗi, tiến hành sửa lỗi và gõ lại Ctrl – F9 để chạy tiếp. Lặp lại quá trình này đến khi sửa hết lỗi.

Sau khi biên dịch và thực hiện chương trình xong mà không thấy kết quả chương trình trên màn hình thì gõ Alt – F5 để xem kết quả, xem xong gõ phím bất kì để trở lại màn hình ban đầu.

Khi chương trình đã chạy thông suốt, muốn dịch chương trình ra tập tin trên đĩa có mở rộng EXE để sau này có thể chạy độc lập trong môi trường DOS thì thực hiện bước 6 và bước 7.

- **Bước 6.** Gõ Alt – C (chọn mục **Compile**), trên màn hình xuất hiện bảng chọn đọc Compile, trong bảng chọn này chọn **Destination** là **Disk** để kết quả dịch chương trình được ghi ra đĩa, rồi gõ Alt – F9 để dịch chương trình. Khi xuất hiện thông báo “Success: Press any key” thì dịch xong. Nếu có lỗi máy cũng dừng lại và thông báo nguyên nhân lỗi trên màn hình.
- **Bước 7.** Thoát tạm thời khỏi Turbo Pascal trở về DOS để chạy chương trình đã dịch thành công: Gõ Alt – F để mở menu **File**. Trong menu **File** chọn **Dos Shell**. Sau đó từ DOS cho chạy tập tin chương trình có đuôi EXE đã dịch. Chạy xong chương trình, trở lại Turbo Pascal bằng lệnh Exit.
- **Bước 8.** Thoát khỏi hẳn Turbo Pascal: Gõ Alt – X hoặc chọn **Quit** trên bảng chọn đọc **File**.

1.3.2. Ví dụ một chương trình hoàn chỉnh

Sau đây là một chương trình hoàn chỉnh thực hiện việc tính chu vi, diện tích hình tròn với bán kính r nhập vào từ bàn phím.

Program HINH_TRON;

Uses Ctr;

Var r, P, S: Real;

Begin

ClrScr; { xoá màn hình }

Write('Nhập bán kính r= ');Readln(r);

Writeln('Diện tích S= ', pi*r*r:2:4);

Writeln('Chu vi P= ',2*pi*r:2:4);

Readln;

End.

Kết quả thực hiện chương trình:

Nhap bán kính r= 3.5 <enter>

Diện tích S= 38.4845

Chu vi P= 21.9911

Câu hỏi – Bài tập chương 1

1. Trình bày các phần tử cơ bản của ngôn ngữ Turbo Pascal.
2. Từ khoá là gì? Tại sao nên học thuộc các từ khoá của Turbo Pascal?
3. Phân biệt sự khác nhau giữa từ khoá và tên chuẩn.
4. Tên là gì?
5. Các tên sau đây tên nào không đúng trong Turbo Pascal? Tại sao? Alpha1, Nghiem – nguyên, 15A, \$I.
6. Trình bày cấu trúc đầy đủ của một chương trình viết bằng Turbo Pascal. Trong cấu trúc đó phần nào bắt buộc phải có, phần nào không bắt buộc phải có?
7. Sắp xếp các từ sau đây thành 3 cột theo các nội dung : Tên chuẩn, Từ khoá, Tên thông thường :
Label, pascal, alpha, logic, boolean, real, with, exit, Lamda, Turbo, compiler, record.
8. Xây dựng thuật toán giải hệ phương trình bậc nhất hai ẩn số sau :

$$\begin{cases} ax + by = c \\ dx + hy = g \end{cases}$$

trong đó x, y là các ẩn số còn a, b, c, d, h, g là hệ số cần được nhập vào từ bàn phím.

9. Xây dựng thuật toán tìm ước số chung lớn nhất của hai số tự nhiên a và b.
10. Cho hai dãy số thực a_1, a_2, \dots, a_n và b_1, b_2, \dots, b_n . Hãy xây dựng thuật toán đọc các số liệu vào và tính biểu thức :

$$a_1b_1 + a_2b_2 + \dots + a_nb_n$$

Chương 2

CÁC KIỂU DỮ LIỆU ĐƠN GIẢN

2.1. Kiểu dữ liệu trong ngôn ngữ lập trình

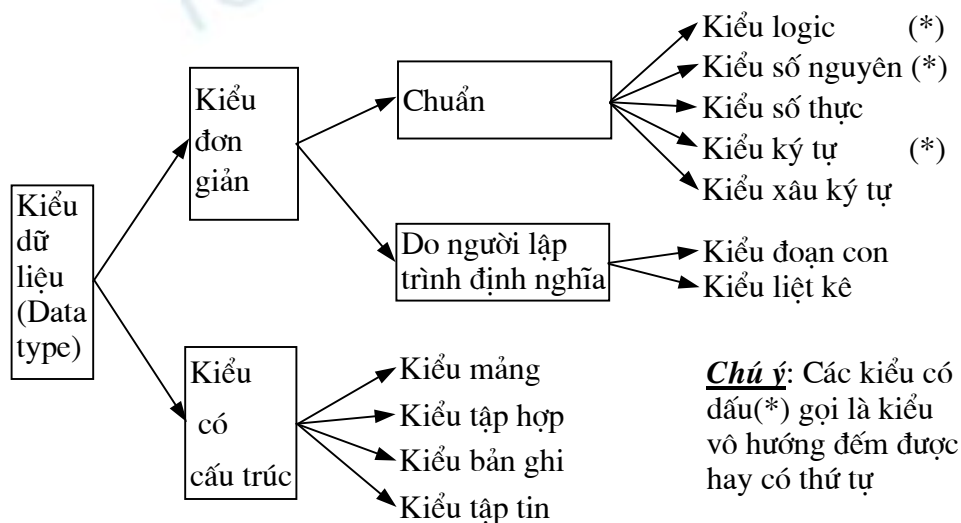
Một kiểu dữ liệu (Data type) là một sự quy định về cấu trúc, miền trị của dữ liệu và tập các phép toán tác động lên miền giá trị đó.

Một ngôn ngữ lập trình chỉ chấp nhận và xử lý những dữ liệu tuân theo sự quy định về kiểu của ngôn ngữ lập trình đó. Trong một ngôn ngữ lập trình, một dữ liệu bao giờ cũng thuộc về một kiểu dữ liệu nhất định.

Dữ liệu trong Turbo Pascal được chia thành 2 loại chính:

- Kiểu dữ liệu đơn giản (simple type).
- Kiểu dữ liệu có cấu trúc (structure type).

Tổng quan phân loại các kiểu dữ liệu trong Turbo pascal



2.2. Các kiểu dữ liệu đơn giản

2.2.1. Kiểu Logic (Boolean type)

Dữ liệu kiểu logic chỉ có 2 giá trị True (đúng) và False (sai). Một giá trị Logic được lưu trữ trong 1 byte. Quy ước : False < True

Các phép toán tác động được lên các đại lượng kiểu logic là : AND, OR, XOR, NOT. Kết quả của các phép toán này được thể hiện qua bảng sau

X	Y	X AND Y	X OR Y	X XOR Y	NOT X
FALSE	FALSE	FALSE	FALSE	FALSE	TRUE
FALSE	TRUE	FALSE	TRUE	TRUE	
TRUE	FALSE	FALSE	TRUE	TRUE	FALSE
TRUE	TRUE	TRUE	TRUE	FALSE	

(X và Y là giá trị kiểu Boolean)

Chẳng hạn, kết quả của các tính toán sau sẽ cho giá trị logic :

$\Delta > 0$; $A <> 0$; $(\Delta > 0 \text{ AND } (A <> 0))$.

2.2.2. Kiểu số nguyên (Integer type)

a. Kiểu số nguyên

Có 5 kiểu số nguyên mà từ khoá, phạm vi số và số byte cần thiết được xác định bởi bảng sau:

Từ khoá	Phạm vi	Yêu cầu bộ nhớ
Shortint	-128 đến 127	1 byte
Integer	-32768 đến 32767	2 byte
Longint	-2147483648 đến 2147483647	4 byte
Byte	0 đến 255	1 byte
Word	0 đến 65535	1 byte

b. Các phép tính số học đối với số nguyên

Đối với các đại lượng kiểu số nguyên chúng ta có thể sử dụng các phép tính sau :

+ (cộng), - (trừ), * (nhân),

/ (phép chia lấy kết quả là số thực read),

DIV (phép chia lấy số nguyên)

MOD (phép chia lấy số dư của hai số nguyên).

☞ Chú ý.

1. Khi viết các số nguyên phải tuân theo các quy định:

- Không có khoảng trống giữa các số.
- Dấu + hay - (nếu có) phải đặt ngay trước chữ số đầu tiên.
- Không được sử dụng dấu chấm thập phân.

2. Khi thực hiện các phép tính trên số nguyên, cần hết sức thận trọng xem phép toán đó có cho kết quả vượt quá phạm vi biểu diễn của số nguyên không.

↪ **Ví dụ 1.**

Khai báo a, b là dữ liệu Integer. Khi đó 2 câu lệnh sau:

a:=80;

b:=(a*1245) div 200

cho kết quả sai vì $a*1245=99600$ vượt quá phạm vi của Integer.

2.2.3. Kiểu số thực (Real type)

a. Kiểu số thực

Kiểu số thực được định nghĩa bằng một trong các từ khoá REAL, SINGLE, DOUBLE tùy theo yêu cầu về độ chính xác cần đến. Cụ thể chúng ta có bảng (xem trang sau) :

Từ khoá	Phạm vi	Yêu cầu bộ nhớ
Real	2.9×10^{99} đến 1.7×10^{38}	6 byte
Single	1.5E-45..3.4E+38	4 byte
Double	5.0E-324..1.7E+308	8 byte
Extended	1.9E-4951..1.1E+4932	10 byte

b. Cách viết một số thực

Các giá trị thực được biểu diễn theo 2 cách :

Cách 1. Số thực dấu phẩy tính được biểu diễn như một số thực thông thường trong Toán học chỉ khác dấu chấm thay cho dấu phẩy trong cách viết của tiếng Việt Nam.

Chẳng hạn: 25.3456789; 0.0000007; -2.12; +4.0.

Cách 2. Số trong biểu diễn dưới dạng dấu phẩy động được tách thành 2 phần: phần định trị và phần mũ.

- Phần định trị là một số nguyên hay số thực viết theo kí pháp tự nhiên.
- Phần mũ là một số nguyên.
- 2 phần cách nhau bởi chữ E hay e.

↪ Ví dụ 2.

123.456E-4	biểu diễn số	0.0123456
0.12E+5	biểu diễn số	12000.0
2E8	biểu diễn số	200000000.0
-52.4e2	biểu diễn số	5240.0

Phần định trị ↑ ↑Phần mũ

c. Các phép toán số học đối với số thực

+ (cộng), - (trừ), * (nhân), / (chia)

d. Các hàm số học dùng cho số nguyên và số thực

Tên hàm	Kiểu của đối số	Kiểu của kết quả	Ý nghĩa
Sqr Abs	Nguyên hay Thực	giống đối số	Sqr(x) = x^2 Abs(x) = $ x $: trị tuyệt đối của x
Sqrt, Sin, Cos, Arctan Ln, Exp	Nguyên hay Thực	Thực	Sqrt(x) = \sqrt{x} ; Sin(x), Cos(x) Arctan(x), Ln(x) : tính giá trị sin(x), cos(x), arctan(x), ln(x) Exp(x) = e^x
Trunc Round	Nguyên hay thực	Nguyên	Trunc(x) : cắt bỏ phần lẻ thập phân của x nếu có Round(x) : cho số nguyên gần x nhất
Succ Pred	Nguyên	Nguyên	Succ(n) = $n + 1$ Pred(n) = $n - 1$
Odd	Nguyên	Kí tự	Odd(n) cho kết quả true nếu n lẻ, False nếu ngược lại.

2.2.4. Kiểu kí tự (Character type)

a. Kiểu dữ liệu

Các kiểu dữ liệu có dạng chữ viết (kí tự) được khai báo bởi từ khoá Char. Một giá trị kiểu Char chiếm 1 byte và biểu diễn được một kí tự thông qua bảng mã ASCII.

Mã của 1 kí tự chính là số thứ tự của nó trong bảng mã. Có tất cả 256 giá trị có thể được biểu diễn trong một byte (đánh số từ 0 đến 255) nên có thể mã hoá 256 giá trị kiểu Char.

Trong bảng mã ASCII, các kí tự từ 0 đến 31 là các kí tự điều khiển không in ra được và dùng để điều khiển quá trình vào ra trên các thiết bị ngoại vi. Chẳng hạn, khi máy nhận được kí tự có mã là 7 (Bell) nó sẽ phát ra một tiếng chuông.

Để biểu diễn một kí tự có thể dùng 1 trong 3 cách :

- Đặt kí tự trong 2 dấu nháy đơn.
- Dùng hàm Char (sẽ trình bày bài sau).
- Dùng #n (trong đó n là mã của kí tự cần biểu diễn).

Chẳng hạn, để biểu diễn kí tự A ta có thể viết: 'A', hoặc Char(65), hoặc #65.

Sau đây là một phần của bảng mã ASCII

Mã (số thứ tự)	Kí tự
7	Bell
32	Space
43	+
48	0
49	1
65	A
66	B
97	a
98	b
134	Σ

b. Phép toán

Chỉ có phép toán so sánh theo giá trị mã kí tự. Chẳng hạn, 'A' < 'a' (vì mã ASCII của A là 65, còn của a là 97).

c. Các hàm chuẩn liên quan đến kiểu kí tự

- Hàm **UPCASE**(ch) cho ra kí tự hoa tương ứng với kí tự ch. Chẳng hạn, Upcase('a')='A'; Upcase('b')='B'
- Hàm **ORD**(x) cho số thứ tự (mã ASCII) của kí tự x trong bảng mã. Chẳng hạn, ord('A')=65, ord('1')=49.
- Hàm **PRED**(x) cho kí tự đứng trước kí tự x trong bảng mã ASCII. Chẳng hạn, Pred('b')='a'.
- Hàm **SUCC**(x) cho kí tự đứng sau kí tự x trong bảng mã ASCII. Chẳng hạn, Succ('c')='d'.

2.2.5. Kiểu chuỗi kí tự (String type)

Một giá trị kiểu String là một dãy kí tự bất kì. Số kí tự của dãy không quá 255. Chuỗi không có kí tự nào cả gọi là chuỗi rỗng. Hằng chuỗi kí tự được đặt trong cặp nháy đơn.

↪ **Ví dụ 3.** Sau đây là các hằng chuỗi kí tự:

‘Họ và tên’ là một chuỗi gồm 9 kí tự

“ ” là một chuỗi rỗng.

Dữ liệu kiểu String sẽ được trình bày chi tiết hơn ở các chương sau.

Câu hỏi – Bài tập Chương 2

1. Kiểu dữ liệu là gì?
2. Trình bày các kiểu dữ liệu đơn giản của Turbo Pascal.
3. Trong Turbo Pascal có thể biểu diễn được bao nhiêu kí tự khác nhau?
4. Khi cần tính các số có giá trị từ 0 đến 245 ta nên dùng dữ liệu kiểu gì?
5. Các giá trị sau thuộc kiểu dữ liệu đơn giản chuẩn nào?
 - a) 15.0
 - b) 1996
 - c) ‘I’
 - d) Sqrt(4)
 - e) Int(12.5)
 - f) 4 >= 5
 - g) -1e10
 - h) 15/6
6. Nếu sử dụng 3 byte để lưu trữ một số nguyên không âm thì giá trị của số nằm trong khoảng nào?
7. Bạn hãy giải thích tại sao để lưu trữ số thực trong Pascal lại có đến 4 kiểu dữ liệu : Real, Single, Double, Extended?
8. Trong trường hợp nào thì cần sử dụng dạng số thực dấu phẩy động?
9. Trong các phép toán dưới đây phép nào có thể thực hiện được trong Turbo Pascal và là kết quả của phép toán thuộc kiểu dữ liệu nào? Trong trường hợp không thực hiện được hãy giải thích tại sao?
 - a) $14/10 + \text{ord}('a')$
 - b) $\sin(0.5) + 12$
 - c) $\text{abs}(a) + 25.7$
 - d) $\text{odd}(12) + \text{ord}(12)$