

ỦY BAN NHÂN DÂN TỈNH HÀ NAM
TRƯỜNG CAO ĐẲNG NGHỀ HÀ NAM

GIÁO TRÌNH

MÔN HỌC: LẬP TRÌNH CĂN BẢN

NGHỀ: QUẢN TRỊ MẠNG MÁY TÍNH

TRÌNH ĐỘ: CAO ĐẲNG

*(Ban hành kèm theo Quyết định số: 285/QĐ-CDNHN
Ngày 21 tháng 07 năm 2017 của Hiệu trưởng trường Cao đẳng nghề Hà Nam)*

Hà Nam, năm 2017

TUYÊN BỐ BẢN QUYỀN

Tài liệu này thuộc loại sách giáo trình nên các nguồn thông tin có thể được phép dùng nguyên bản hoặc trích dùng cho các mục đích về đào tạo và tham khảo.

Mọi mục đích khác mang tính lệch lạc hoặc sử dụng với mục đích kinh doanh thiếu lành mạnh sẽ bị nghiêm cấm.

Dựa theo giáo trình này, có thể sử dụng để giảng dạy cho các trình độ hoặc ngành/ nghề khác của nhà trường.

TaiLieu.vn

LỜI GIỚI THIỆU

Giáo trình Lập trình căn bản được biên soạn nhằm đáp ứng yêu cầu học tập của học sinh, sinh viên bước đầu làm quen với công việc lập trình, đồng thời giúp cho sinh viên có một tài liệu học tập, rèn luyện tốt khả năng lập trình, tạo nền tảng vững chắc cho các môn học tiếp theo .

Giáo trình không chỉ phù hợp cho người mới bắt đầu mà còn phù hợp cho những người cần tham khảo. Nội dung của giáo trình được chia thành 6 chương:

Chương 1: Làm quen ngôn ngữ lập trình (Bài mở đầu)

Chương 2: Các thành phần trong ngôn ngữ lập trình

Chương 3: Các cấu trúc điều khiển

Chương 4: Hàm và thủ tục

Chương 5: Dữ liệu kiểu tập hợp, mảng và bản ghi

Chương 6: Dữ liệu kiểu chuỗi

Khi biên soạn, tôi đã tham khảo các giáo trình và tài liệu giảng dạy môn học này của một số trường Cao đẳng, Đại học để giáo trình vừa đạt yêu cầu về nội dung vừa thích hợp với đối tượng là học sinh, sinh viên của các trường thuộc hệ thống Cơ sở giáo dục nghề nghiệp.

Tôi hy vọng sớm nhận được những ý kiến đóng góp, phê bình của bạn đọc về nội dung, chất lượng và hình thức trình bày để giáo trình này ngày một hoàn thiện hơn.

Hà Nam, ngày tháng năm 2017

Tác giả biên soạn: Phạm Tất Thành

MỤC LỤC

LỜI GIỚI THIỆU.....	3
Chương 1: Làm quen với ngôn ngữ lập trình	6
Chương 2: Các thành phần cơ bản.....	16
Chương 3: Các cấu trúc điều khiển	25
Chương 4: Hàm và thủ tục.....	44
Chương 5: Dữ liệu kiểu tập hợp, mảng và bản ghi.....	54
Chương 6: Dữ liệu kiểu chuỗi	71

Tailieu.vn

GIÁO TRÌNH MÔN HỌC

Tên môn học: Lập trình căn bản;

Mã số môn học: MH 10;

I. Vị trí, tính chất, ý nghĩa và vai trò của môn học

- Vị trí: môn học được bố trí sau khi sinh viên học xong các môn học chung, các môn học tin, tin học văn phòng.
- Tính chất: là môn học lý thuyết chuyên ngành.

II. Mục tiêu môn học:

- Về kiến thức:
 - + Trình bày được khái niệm về lập trình;
 - + Mô tả được ngôn ngữ lập trình: cú pháp, công dụng của các câu lệnh;
 - + Phân tích được chương trình: xác định nhiệm vụ chương trình;
 - + Thực hiện được các thao tác trong môi trường phát triển phần mềm: biên tập chương trình, sử dụng các công cụ, điều khiển, thực đơn lệnh trợ giúp, gỡ rối, bẫy lỗi, v.v.;
- Về kỹ năng:
 - + Viết chương trình và thực hiện chương trình trong máy tính.
 - + Xác định môi trường hoạt động của hệ thống (các điều khiển, công cụ, các thành phần, tập hợp dữ liệu, nhập dữ liệu, in kết quả ...)
 - + Chú thích cho từng đoạn xử lý của chương trình.
- Về năng lực tự chủ và trách nhiệm:
 - + Có khả năng tự làm các bài tập cơ bản về lập trình hoặc làm theo nhóm, biết áp dụng những kiến thức đã học trong những bài toán thực tế.
 - + Đánh giá chất lượng bài tập thực hành sau khi hoàn thành và kết quả thực hiện của các thành viên trong nhóm.

III. Nội dung môn học:

Chương 1: Làm quen với ngôn ngữ lập trình

Giới thiệu:

Trong bài này sẽ giới thiệu về các khái niệm về lập trình, lịch sử phát triển và thực hiện các thao tác an toàn với máy tính.

Mục tiêu:

- Trình bày được các khái niệm về lập trình;
- Trình bày được lịch sử phát triển, ứng dụng của ngôn ngữ lập trình;
- Làm quen môi trường phát triển phần mềm;
- Sử dụng được hệ thống trợ giúp từ help file.
- Thực hiện các thao tác an toàn với máy tính.

Nội dung:

1.1. KHÁI NIỆM CƠ BẢN VỀ LẬP TRÌNH

Lập trình là sử dụng cấu trúc dữ liệu và các câu lệnh của ngôn ngữ lập trình cụ thể để mô tả dữ liệu và diễn đạt các thao tác của thuật toán.

Những ngôn ngữ lập trình (programming language) đầu tiên trên máy tính điện tử là ngôn ngữ máy (machine language), tổ hợp của các con số hệ nhị phân, hay các **bit** (**binary digit**) 0 và 1. Ngôn ngữ máy phụ thuộc vào hoàn toàn kiến trúc phần cứng của máy tính và các quy ước khắt khe của nhà chế tạo. Để giải các bài toán, những người lập trình phải sử dụng một tập hợp các lệnh điều khiển rất sơ cấp mà mỗi lệnh là tổ hợp các bit nhị phân nên gặp rất nhiều khó khăn, mệt nhọc, rất dễ gặp phải sai sót, nhưng rất khó sửa lỗi.

Từ những năm 1950, để giảm nhẹ việc lập trình, người ta đưa vào kỹ thuật *chương trình con* (sub-program hay sub-routine) và xây dựng các *thư viện chương trình* (library) để khi cần thì gọi đến hoặc dùng lại các đoạn chương trình đã viết.

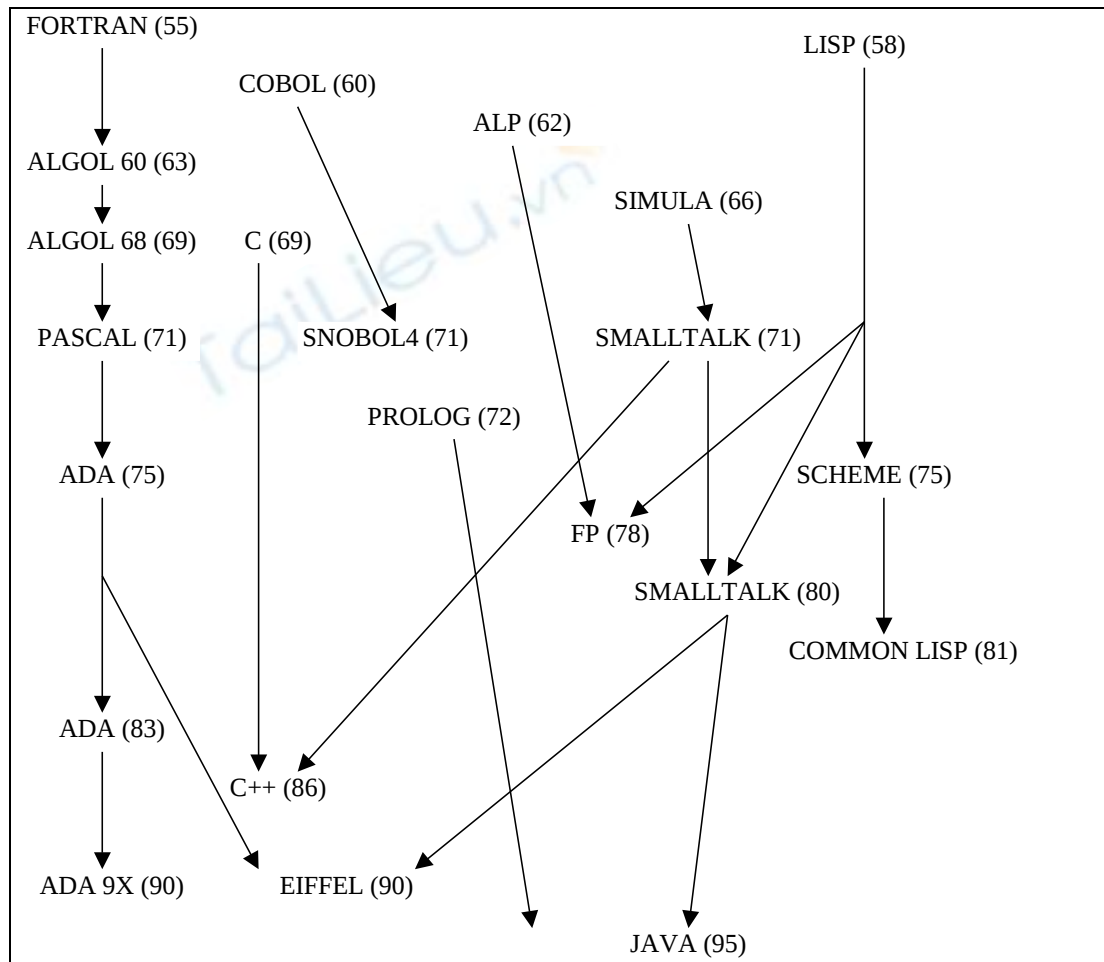
Như thế, chúng ta nhận thấy ở vào giai đoạn sơ khai ban đầu của máy tính điện tử, việc sử dụng máy tính là rất khó khăn, vì ngôn ngữ lập trình là phương tiện giao tiếp lại quá phức tạp đối với người sử dụng. Người sử dụng máy tính vào giai đoạn này chỉ là các chuyên gia về tin học. Như thế, ứng dụng của máy tính điện tử vẫn còn rất hạn chế.

1.2. LỊCH SỬ PHÁT TRIỂN VÀ ỨNG DỤNG CỦA NGÔN NGỮ LẬP TRÌNH PASCAL

Vào đầu những năm 1970 do nhu cầu học tập của sinh viên, giáo sư Niklaus Wirth - Trường Đại Học Kỹ Thuật Zurich - Thụy Sĩ đã sáng tác một ngôn ngữ lập trình cấp cao cho công tác giảng dạy sinh viên. Ngôn ngữ được đặt tên là

PASCAL đề tưởng nhớ đến nhà toán học người Pháp Blaise Pascal. Pascal là một ngôn ngữ lập trình có cấu trúc thể hiện trên 3 phương diện.

- Về mặt dữ liệu: Ngoài các kiểu dữ liệu đơn giản còn có các kiểu dữ liệu có cấu trúc. Ta có thể xây dựng các kiểu dữ liệu phức tạp từ các kiểu dữ liệu đã có.
- Về mặt câu lệnh: Từ các câu lệnh đơn giản và lệnh có cấu trúc ta có thể xây dựng các câu lệnh hợp thành.
- Về mặt chương trình: Một chương trình có thể chia làm nhiều chương trình con



Cho đến nay có hàng trăm ngôn ngữ lập trình được đề xuất nhưng trên thực tế chỉ có một số ít ngôn ngữ được sử dụng rộng rãi. Ngoài cách phân loại theo *bậc* như đã nói ở trên, người ta còn phân loại ngôn ngữ lập trình theo phương thức (paradgm), theo mức độ quan trọng, theo thể hệ, ...

Cách phân loại theo mức hay bậc là dựa trên mức độ trừu tượng so với các yếu tố phần cứng, chẳng hạn như *lệnh* (instruction) và *cấp phát bộ nhớ* (memory allocation) dưới đây:

Mức	Lệnh	Sử dụng bộ nhớ	Ví dụ
Thấp	Lệnh máy đơn	Truy cập và cấp phát trực	Hợp ngữ

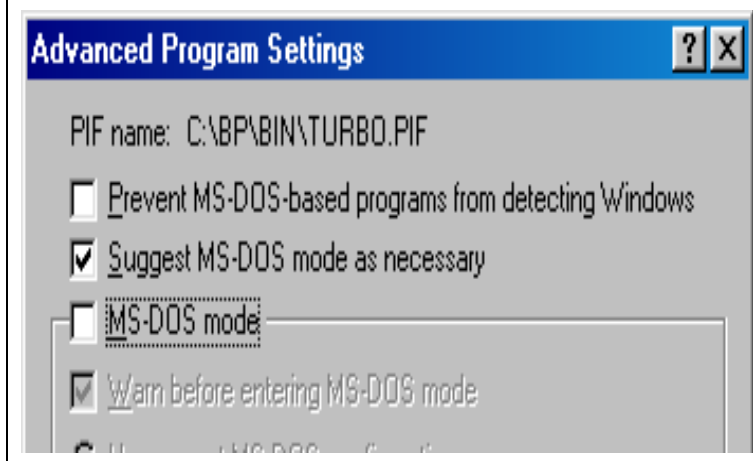
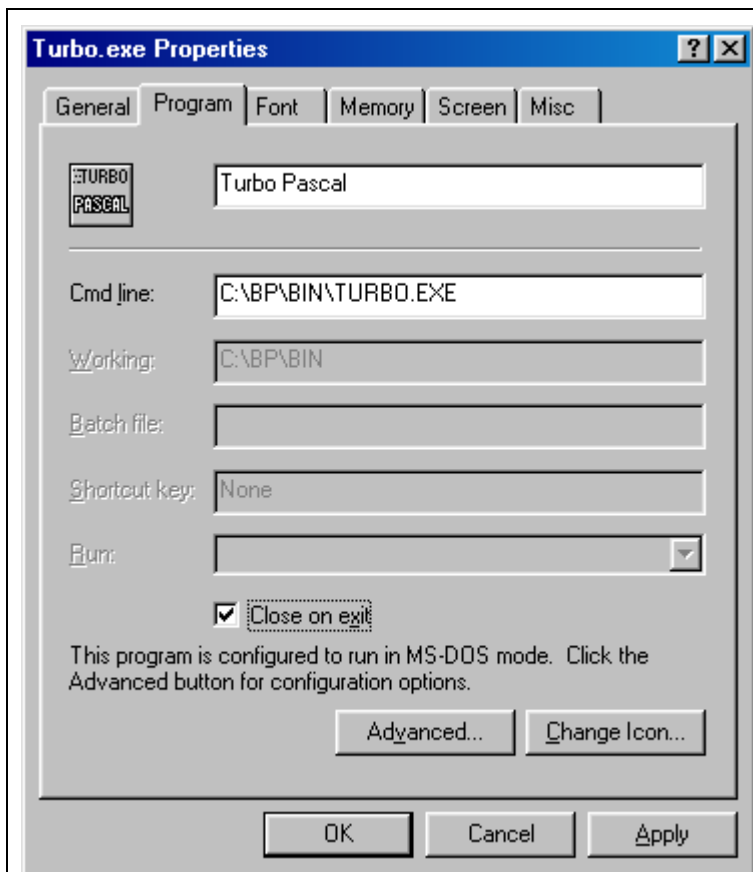
	giản	tiếp	
Cao	Biểu thức và điều kiện tương minh	Truy cập và cấp phát nhờ các phép gán	C, Pascal, Ada
Rất cao	Máy trừu tượng	Truy cập ẩn và tự động cấp phát	Prolog, Miranda

1.3. LÀM QUEN VỚI MÔI TRƯỜNG PHÁT TRIỂN PHẦN MỀM

1.3.1. KHỞI ĐỘNG CHƯƠNG TRÌNH PASCAL

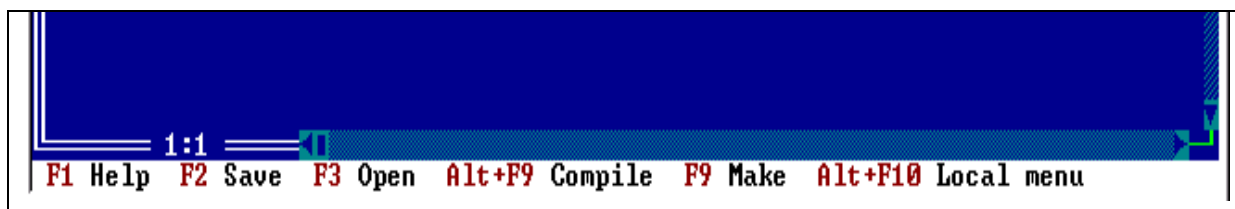
Nếu máy tính chúng ta đã cài đặt Turbo Pascal trên đĩa, ta có thể khởi động chúng như sau (Nếu máy tính chưa có, chúng ta phải cài đặt Turbo Pascal sau đó mới thực thi được)

- Từ MS-DOS: Đảm bảo rằng thư mục hiện hành đúng vị trí cài đặt (hoặc dùng lệnh PATH) Turbo Pascal. Ta đánh vào **TURBO** rồi Enter.
- Từ Windows: Ta nên giả lập MS-DOS Mode cho tập tin **TURBO.EXE** hoặc Shortcut của nó, nếu không mỗi khi ta thực thi TURBO PASCAL chương trình sẽ thoát khỏi Windows, trở về MS-DOS. Sau khi thoát Turbo Pascal ta phải đánh lệnh EXIT để khởi động lại Windows. Cách giả lập như sau:
 - Nhấp chuột phải lên tập tin TURBO.EXE hoặc Shortcut của nó, chọn Properties.
 - Chọn thẻ Program và đánh check như hình sau.



Chọn OK trên các hộp thoại, sau đó khởi động Turbo Pascal, màn hình soạn thảo sau khi khởi động TURBO PASCAL như dưới đây xuất hiện.





1.3.2. CÁC THAO TÁC SỬ DỤNG TRÊN TURBO PASCAL

Khi ta muốn **tạo mới hoặc mở** một tập tin đã có trên đĩa ta dùng phím **F3**. Sau đó đưa vào tên và vị trí của tập tin. Nếu tập tin đã tồn tại thì Turbo Pascal mở nội dung lên cho ta xem, nếu tên tập tin chưa có thì Turbo Pascal tạo một tập tin mới (với tên mà ta đã chỉ định).

Khi muốn **lưu lại** tập tin ta dùng phím **F2**. Trước khi thoát khỏi chương trình, ta nên lưu tập tin lại, nếu chưa lưu chương trình sẽ hỏi ta có lưu tập tin lại hay không. Nếu ta chọn **Yes** (ấn phím Y) thì chương trình sẽ **lưu lại**, chọn **No** (ấn phím N) chương trình sẽ **không lưu**.

Một số phím thông dụng của TURBO PASCAL 7.0

Biểu tượng	Tên phím	Diễn giải
↵	Enter	Đưa con trỏ xuống dòng.
↑	Up	Đưa con trỏ lên 1 dòng.
↓	Down	Đưa con trỏ xuống 1 dòng.
←	Left	Đưa con trỏ qua trái một ký tự.
→	Right	Đưa con trỏ qua phải một ký tự.
Home	Home	Đưa con trỏ về đầu dòng.
End	End	Đưa con trỏ về cuối dòng.
Pg Up	Page Up	Lên một trang màn hình.
Pg Down	Page Down	Xuống một trang màn hình.
Del	Delete	Xoá ký tự tại vị trí con trỏ.
←Back	BackSpace	Xoá ký tự trước con trỏ.
Insert	Insert	Thay đổi chế độ viết xen hay viết chồng.
F1	F1	Gọi chương trình giúp đỡ.
F2	F2	Lưu tập tin lại.
F3	F3	Tạo mới hoặc mở tập tin.
F4	F4	Thực thi chương trình đến dòng chứa con trỏ.

F5	F5	Phóng lớn cửa sổ.
F6	F6	Chuyển đổi các cửa sổ.
F7	F7	Chạy từng dòng lệnh (hàm xem như một lệnh).
F8	F8	Chạy từng dòng lệnh đơn.
F9	F9	Kiểm tra lỗi chương trình.
Tổ hợp	Alt + F9	Biên dịch chương trình.
Tổ hợp	Ctrl + F9	Chạy chương trình.
Tổ hợp	Ctrl + N	Thêm 1 dòng trước con trỏ.
Tổ hợp	Ctrl + Y	Xoá một dòng tại con trỏ.
Tổ hợp	Ctrl + K + B	Đánh dấu đầu khối.
Tổ hợp	Ctrl + K + K	Đánh dấu cuối khối.
Tổ hợp	Ctrl + K + C	Sao chép khối.
Tổ hợp	Ctrl + K + V	Di chuyển khối.
Tổ hợp	Ctrl + K + Y	Xoá khối.
Trong Borland Pascal các thao tác khối đơn giản và dễ hơn như sau: + Đánh dấu khối: SHIFT + (phím mũi tên) + Copy khối vào clipboard: CTRL+ Ins (phím Insert) + Dán khối (đã copy vào clipboard) vào vị trí mới: SHIFT+ Ins		
Tổ hợp	Ctrl + K + W	Ghi khối lên đĩa thành một tập tin (nội dung của tập tin là khối đã chọn).
Tổ hợp	Ctrl + K + R	Xen nội dung một tập tin (từ đĩa) vào sau vị trí con trỏ.
Tổ hợp	Ctrl + K + H	Tắt/Mở đánh dấu khối.
Tổ hợp	Ctrl + F4	Kiểm tra giá trị biến khi chạy chương trình.
Tổ hợp	Alt + X	Thoát khỏi chương trình.

1.4. CHƯƠNG TRÌNH MẪU

1.4.1. Cấu trúc cơ bản:

Chương trình Pascal đơn giản nhất phải có hai từ khóa Begin và End như sau:

Begin

End.

Chương trình trên tuy không làm gì khi chạy (ấn Ctrl - F9) nhưng là một chương trình hợp lệ do hội đủ điều kiện cần thiết là có hai từ khóa Begin và End.

Trong chương trình có thể có nhiều khối lệnh, tức có thể có nhiều cặp từ khóa Begin và End.

1.4.2. Phương pháp khai báo và tổ chức cấu trúc một chương trình Pascal:

Việc đặt các phân khai báo và soạn thảo chương trình theo thứ tự như sau:

Program ProgName;

Uses UnitName1, UnitName2, UnitNameN;

Label LabelName1, LabelName2, LabelNameN;

Const Const1 = n, Const2 = m, ConstN = k;

Type Type1 = AnyType;

Var Var1, Var2, VarN : Type;

Begin

{các lệnh của chương trình}

End.

Ví dụ: một cách khai báo tên chương trình:

Program TimUSCLN;

Begin

...

End.

- Uses: Từ khoá này dùng để khai báo việc sử dụng Unit (thư viện) cho chương trình. Thư viện là tập hợp các hàm, thủ tục do ngôn ngữ Pascal cung cấp kèm theo hoặc cũng có thể do người lập trình tạo ra để sử dụng. Ta khai báo thư viện thông qua tên của thư viện, và trong chương trình đó ta sẽ có thể sử dụng các thủ tục hoặc các hàm có trong thư viện đó. Các thư viện chuẩn của ngôn ngữ Pascal gồm: CRT, DOS, GRAPH, GRAPH3, OVERLAY, PRINTER, SYSTEM và TURBO3. Trong đó, thư viện SYSTEM mặc định được chuyển vào chương trình mà ta không cần phải khai báo. Ví dụ một cách khai báo thư viện:

...

Uses CRT, GRAPH;

...

- Label: Dùng để khai báo các nhãn cho chương trình. Nhãn là các tên dùng để đánh dấu trong chương trình để lệnh GOTO nhảy đến đúng vị trí đó. Việc sử dụng lệnh GOTO được đề cập ở bài sau. Ví dụ một cách khai báo nhãn:

...

Label TH1, N2;

...

- Const: Từ khóa này dùng để khai báo các hằng số sử dụng trong chương trình, khi báo hằng số là việc cố định một vài giá trị nào đó trong chương trình thông qua tên hằng, ví dụ cách khai báo hằng:

...

Const k = 5, Max = 500, Ten = 'Nam';

...

- Type: từ khóa dùng để khai báo các kiểu hàng dữ liệu sử dụng cho chương trình. Với từ khóa này, ta có thể tự tạo riêng cho mình những kiểu dữ liệu riêng dựa trên các kiểu dữ liệu chuẩn để tiện sử dụng trong việc lập trình. Các khái niệm về dữ liệu chuẩn và phương pháp tạo kiểu dữ liệu tự tạo sẽ được giới thiệu ở các phần sau. Ví dụ một cách để khai báo một kiểu dữ liệu tự tạo:

...

Type Day = Array [1..7] of String[8];

...

- Var: Từ khóa dùng để khai báo các biến số được sử dụng trong chương trình. Biến số là các giá trị có thể thay đổi được trong suốt quá trình chạy của chương trình. Khái niệm về biến số rất quan trọng trong việc lập trình (khái niệm này được trình bày kỹ ở bài sau). Một ví dụ về cách khai báo biến:

Var HoDem, Ten : String;

N : Integer;

...

Ghi chú:

- Thứ tự các khai báo trên là điều bắt buộc, ta phải nằm thứ tự này cho dù một số khái niệm ta chưa được biết.

- Trong chương trình Pascal, để tạo lời chú thích, ta sử dụng cặp dấu {...} hoặc (*...*) lồng các câu chú thích vào bên trong nó.

- Trên một dòng có thể viết một hoặc nhiều câu lệnh.

1.4.3. Các ví dụ đơn giản làm quen với ngôn ngữ Pascal:

Ví dụ 1:

Program GioiThieu;

Begin

Writeln (' Truong Cao dang nghe Ha Nam);

Write (' 88 Le Hoan – Quang Trung – Phu Ly – Ha Nam ');

End.

Để xem chương trình trên, ta chạy bằng Ctrl - F9 và xem lại bằng Alt - F5.

Ví dụ 2:

```

Program DonXinPhep;
Uses CRT;
Begin
  ClrScr;
  Writeln ( ' ***** ' );
  Writeln ( ' * Cong hoa Xa hoi Chu nghia Viet Nam * ' );
  Writeln ( ' *      Doc Lap - Tu Do - Hanh Phuc      * ' );
  Writeln ( ' *      DON XIN PHEP NGHI HOC      * ' );
  Writeln ( ' ***** ' );
  Writeln ( '... ' );
  Readln;
End.

```

Ví dụ 3:

```

Program TinhTong;
Uses CRT;
Begin
  ClrScr;
  Write ( ' 30 + 40 + 15 = ', 30 + 40 + 15 );
  Readln;
End.

```

Kết quả: Máy thực hiện phép tính và hiển thị $30 + 40 + 15 = 85$

1.5. BÀI TẬP

1. Khởi động Turbo Pascal.
2. Nhập vào đoạn chương trình sau:

```

Uses Crt;
Begin

  Writeln('*****
*****');

  Writeln('*  TRUONG CAO DANG NGHE DA NANG  *');
  Writeln('*              Xin chao ban !              *');

  Writeln('*****
*****');

  Readln;

End.

```

3. Dịch và chạy chương trình trên.

4. Lưu chương trình vào đĩa với tên BAI1.PAS.
5. Thoát khỏi Pascal.
6. Khởi động lại Turbo Pascal.
7. Mở file BAI1.PAS.
8. Chèn thêm vào dòng: **CLRSCR;** vào sau dòng **BEGIN**
9. Dịch và chạy thử chương trình.
10. Lưu chương trình vào đĩa.
11. Thoát khỏi Pascal.

Tailieu.vn

Chương 2: Các thành phần cơ bản

Mục tiêu:

- Trình bày và sử dụng được hệ thống kí hiệu và từ khóa.
- Mô tả được các kiểu dữ liệu.
- Trình bày được và vận dụng được các loại biến, hằng biểu thức cho từng chương trình cụ thể;
- So sánh được các lệnh, khối lệnh;
- Thực hiện được việc chạy chương trình.
- Thực hiện các thao tác an toàn với máy tính.

Nội dung :

1.6. Hệ thống từ khóa và kí hiệu được dùng trong ngôn ngữ lập trình

Các từ khoá là các từ dùng để khai báo, đặt tên cho đối tượng trong Pascal, khi ta đặt tên cho đối tượng nào đó, không được đặt trùng tên với các từ khoá.

Bảng từ khoá trong ngôn ngữ Pascal gồm:

and, array, asm, begin, case, const, constructor, destructor, div, do, downto, else, end, file, for, function, goto, if, implementation, in, inline, interface, label, mod, nil, not, object, of, or, packed, procedure, program, record, repeat, set, shl, shr, string, then, to, type, unit, until, uses, var, while, with, xor.

Turbo Pascal không phân biệt ký tự thường hoặc hoa. Ví dụ, các cách viết sau có ý nghĩa như nhau: Begin, BEGIN, begin, beGIN, bEGIN,...

1.7. Các kiểu dữ liệu cơ bản.

1.7.1. Các kiểu dữ liệu dạng số

1.7.1.1. Kiểu số nguyên: (integer)

Một giá trị kiểu số nguyên là một phần tử của tập số nguyên mà có thể biểu diễn trên máy, nghĩa là nó là một tập nhỏ của các số nguyên chứ không phải là tất cả mọi số nguyên. Kiểu số nguyên được định nghĩa với các từ khóa sau:

TỪ KHÓA	PHẠM VI
INTEGER	-32768..32767
BYTE	0..255
WORD	0..65535
SHORTINT	-128..127
LONGINT	-2147483648..2147483647

1.7.1.2. Kiểu số thực: (Real)

Tương tự như kiểu số nguyên, kiểu số thực là tập hợp các số thực có thể biểu diễn được trong máy và được máy định nghĩa sẵn với từ khóa Real.

TỪ KHÓA	PHẠM VI
REAL	$2.9 \times 10^{-39} \dots 1.7 \times 10^{38}$
SINGLE	$1.5E-39 \dots 1.7E+38$
DOUBLE	$5.0E-324 \dots 1.7E+308$
EXTENDED	$1.9E-4951 \dots 1.1E+4932$

Lưu ý:

+ *Dạng viết thập phân bình thường như:*

3.14 3.0 -24.567 -0.0089

+ *Dạng viết có phần mũ như:*

Gồm 2 phần, phần định trị và phần mũ viết sau chữ E để biểu diễn cơ số

10.

$$5678.12345 = 5.67812345 \times 10^3$$

Sẽ được viết lại cho máy tính là:

5.67812345E+03

$\uparrow$ $\uparrow$
 Phần định trị Phần mũ

1.7.2. Kiểu char, logic, string

1.7.2.1. Kiểu Char:

Dùng để biểu diễn các giá trị là các ký tự thuộc bảng chữ cái: 'A', 'b', 'x',... các con số: 0..9 hoặc các ký tự đặc biệt: '!', '@', '#', '\$', '%', '&', '*',...

Để biểu diễn thông tin, ta cần phải sắp xếp các ký tự theo một chuẩn nào đó và mỗi cách sắp xếp đó gọi là bảng mã, thông dụng nhất là bảng mã ASCII (American Standard Code for Information Interchange). Bảng mã ASCII có 256 ký tự được đánh mã số từ 0..255, mỗi ký tự có một mã số nhất định, ví dụ: ký tự 'A' có mã số là 65, 'a' có mã số là 97 trong bảng mã ASCII, v.v.

Để hiển thị bảng mã ASCII, chạy chương trình sau:

```

Program ASCII_Table;
Uses CRT;
Var I : Integer;
Begin
  ClrScr;
  For I := 0 to 255 do
    Write( I, ' = ', CHR( I ), '  ');
  Readln;
End.

```

1.7.2.2. Kiểu Logic

Kiểu logic là kiểu biểu diễn hai trạng thái là đúng (True) hoặc sai (False). Từ khóa để khai báo cho kiểu logic là BOOLEAN.

Ví dụ:

```
Var Co : Boolean;
```

Co := True;

1.7.2.3. Kiểu String (chuỗi ký tự):

String là kiểu dữ liệu chứa các giá trị là nhóm các ký tự hoặc chỉ một ký tự, kể cả chuỗi rỗng. Độ dài tối đa của một biến kiểu String là 255, tức là nó có thể chứa tối đa một dãy gồm 255 ký tự.

Cú pháp khai báo: (1) Var Biến_1, Biến_2, Biến_n: String;

Hoặc (2) Var Biến_1, Biến_2, Biến_n: String [30];

Cách khai báo (1) sẽ cho phép biến HoTen nhận tối đa 255 ký tự. Cách (2) cho phép biến HoTen nhận tối đa 30 ký tự.

1.7.3. Hằng, biến, hàm, các phép toán và biểu thức

1.7.3.1. Hằng:

Khái niệm : là đại lượng có giá trị không đổi trong suốt chương trình. Có các loại hằng số (nguyên và thực), hằng ký tự, hằng Boolean.

Trong chương trình, hằng được khai báo bằng cách đặt tên các hằng vào phần khai báo hằng sau từ khóa Const như sau:

```
CONST  
TÊN_HÀNG= GIÁ_TRỊ_CỦA_HÀNG;
```

Ví dụ:

Const

A = 5;

B = True;

C = 'S';

(Ta nên sử dụng tên hằng dưới dạng gọi nhớ nhằm rõ ràng khi cần sửa)

- Có hai phương pháp sử dụng hằng :

+ Gán trực tiếp giá trị hằng. Ví dụ: DT := R * R * 3.14; ChuVi := D * 3.14;

+ Đặt cho hằng một tên gọi và trong quá trình soạn chương trình ta dùng tên gọi thay cho việc dùng trực tiếp giá trị đó. Ví dụ: ChuVi := D * Pi; trong đó, Pi là một hằng số chuẩn của Pascal (tức là ta có thể dùng mà không cần khai báo và gán giá trị).

1.7.3.2. Biến (Variable)

- Là đại lượng mà giá trị của nó có thể thay đổi trong quá trình thực hiện chương

trình. Biến được khai báo bằng từ khóa VAR.

- Biến là tên của một vùng bộ nhớ lưu trữ dữ liệu.

- Biến được truy xuất trong chương trình thông qua tên biến.

- Biến là một cấu trúc ghi nhớ dữ liệu vì vậy phải được quy định theo một kiểu dữ liệu nào đó, ví dụ kiểu Integer, Byte, Char,...

```
VAR  
TÊN_BIẾN : KIỂU_DỮ_LIỆU_CỦA_BIẾN;
```

Ví dụ:

```
Var  
A      : Real;  
B, C   : Integer;  
TEN     : String [20];  
X       : Boolean;  
Chon    : Char;
```

1.7.3.3. Biểu thức

Biểu thức là một công thức tính toán bao gồm các phép toán, hằng, biến, hàm, các dấu ngoặc.

Ví dụ: $5 + A * \text{SQRT}(B) / \text{SIN}(X)$

Khi tính giá trị của biểu thức, luôn tuân theo thứ tự ưu tiên như sau:

Dấu ngoặc (,)	Biểu thức trong dấu ngoặc được ưu tiên trước
Not, - (Dấu trừ)	Các phép toán một toán hạng
*, /, DIV, MOD, AND	Các phép tính loại nhân cùng mức ưu tiên
+, -, OR, XOR	Các phép tính loại cộng mức ưu tiên
=, <>, <=, >=, >, <, IN	Các phép tính quan hệ cùng mức ưu tiên

Và luôn tuân theo qui tắc sau:

- ♦ Các phép toán nào có thứ tự ưu tiên cao hơn sẽ được tính trước
- ♦ Nếu các phép toán có cùng ưu tiên sẽ được tính từ trái sang phải
- ♦ Phần trong ngoặc sẽ được tính trước
- ♦ Kiểu của biểu thức là kiểu của kết quả sau cùng

1.7.4. Các lệnh, khối lệnh

Sau phần khai báo dữ liệu là phần lệnh của chương trình. Phần này xác định các công việc mà chương trình phải thực hiện xử lý các dữ liệu đã được khai báo. Câu lệnh được chia thành hai loại:

- Câu lệnh đơn giản:
 - + Lệnh gán ($:=$)
 - + Lệnh Nhập - Xuất (READ, READLN, WRITE, WRITELN).
 - + Gọi thủ tục.

- + Lệnh nhảy (GOTO).
- Câu lệnh có cấu trúc:
 - + Lệnh ghép (BEGIN... END)
 - + Lệnh lựa chọn (IF... ELSE, CASE... OF)
 - + Lệnh lặp (FOR, REPEAT... UNTIL, WHILE... DO)
 - + Lệnh WITH.

1.7.4.1. Khối lệnh

Là một nhóm câu lệnh đơn giản được đặt giữa 2 chữ Begin và End sẽ tạo nên câu lệnh hợp thành hay lệnh ghép với mẫu viết như sau:

```
BEGIN
  CÂU LỆNH 1;
  CÂU LỆNH 2;
  ...
  CÂU LỆNH N;
END;
```

1.7.4.2. Lệnh gán

Mục đích của lệnh này nhằm gán cho biến một giá trị sao cho phù hợp với kiểu khai báo của biến trước đó

Cách gán như sau:

```
TÊN_BIẾN := BIỂU THỨC;
```

Ví dụ:

```
c: = 'A';
i: = (20 + 5) * 2 Mod 4;
x: = 0.5;
```

1.7.4.3. Lệnh nhập

CÁCH VIẾT	Ý NGHĨA
READ(X1,X2,...,XN);	Nhập dữ liệu từ bàn phím vào các biến X1,X2,...Xn.
READLN(X1,X2,...,XN);	Nhập dữ liệu từ bàn phím vào các biến X1,X2,...Xn nhưng khi nhập xong, con trỏ xuống dòng.
READLN;	Dừng chương trình, đợi phím Enter để tiếp tục.
ASSIGN(F,TÊN_FILE);	Mở File F trên đĩa có tên là Tên_File
RESET(F);	Chuẩn bị đọc.
READ(F,X1,X2,...,XN);	Đọc các giá trị có ghi tên File F ra các biến X1,X2,...,Xn.
CH:=READ KEY;	Đọc một ký tự từ bàn phím vào biến ký tự CH
KEYPRESSED	Một hàm có giá trị là TRUE nếu có một phím được bấm và FALSE nếu ngược lại.
Chú ý: Biến F cần khai báo trước ở phần trên bởi lệnh:	