CHƯƠNG II : CHƯƠNG TRÌNH ĐƠN GIẢN

Bài 3: CẤU TRÚC CHƯƠNG TRÌNH

1. **Cấu trúc chung**

[<phần khai báo>]

<phần thân>

Phần khai báo có thể có hoặc không tùy theo chương trình cụ thể.

Phần thân chương trình nhất thiết phải có.

1. **Các thành phần của chương trình**
2. ***Phần khai báo***

* Khai báo tên chương trình

**program *<****tên chương trình****>;***

Ví dụ :

**program** Chuong\_trinh\_B2;

* Khai báo thư viện

**uses *<****tên thư viện****>;***

Thư viện là nơi cung cấp một số chương trình thông dụng đã lập sẵn. Để sử dụng các chương trình đó cần phải khai báo thư viện chứa nó.

Ví dụ :

**uses** CRT;

* Khai báo hằng

**const<** *tên hằng****>* = *<****giá trỊ****>;***

Hằng thường được sử dụng cho những giá trị xuất hiện nhiều lần trong chương trình.

* Khai báo biến

**var<** *danh sách biến****>* : *<****kiểu dữ Hệư****>;***

Tất cả các biến dùng trong chương trình đều phải đặt tên và phải khai báo cho chương trình dịch biết để lưu trữ và xử lí.

Biến chỉ nhận một giá trị tại mỗi thời điểm thực hiện chương trình gọi là biến đơn.

1. ***Phần thân chương trình***

Dãy lệnh trong phạm vi được xác định bởi cặp dấu hiệu bắt đầu **begin** và kết thúc **end.** tạo thành thân chương trình

**begin**

[< *dãy lệnh>]*

**end.**

1. **Ví dụ chương trình đơn giản**

Chương trình Pascal sau đưa các thông báo *"Xin chao cac ban* !" và *"Moi cac ban lamquen voi Pascal'* ra màn hình.

**program** vi\_du;

**begin**

writeln('Xin chao cac ban !);

writeln('Moi cac ban lam quen voi Pascal.');

**end**.

* Phần khai báo chỉ có khai báo tên : gồm tên dành riêng **program** và tên chương trình là **vi\_du**.
* Phần thân chương trình có hai câu lệnh **writeln** đưa hai thông báo tương ứng ra màn hình.

xxxxx

Bài 4: MỘT SỐ KIỂU DỮ LIỆU CHUẨN

1. **Kiểu nguyên**

|  |  |  |
| --- | --- | --- |
| **Kiểu** | **Phạm vi giá trị** | **Bộ nhớ** |
| byte | 0... 255 | 1 byte |
| integer | -215'.. 215 - 1 | 2 byte |
| word | 0... 216 - 1 | 2 byte |
| longint | -231... 231 - 1 | 4 byte |

1. **Kiểu thực**

|  |  |  |
| --- | --- | --- |
| **Kiểu** | **Phạm vi giá trị** | **Bộ nhớ** |
| real | 0 hoặc tuyệt đối từ 10-38 ... 1038 | 6 byte |
| extended | 0 hoặc tuyệt đối từ 10-4932 ...  104932 | 10 byte |

1. **Kiểu kí tự**

|  |  |  |
| --- | --- | --- |
| **Kiểu** | **Phạm vi giá trị** | **Bộ nhớ** |
| CHAR | 256 kí tự trong bộ mã ASCII | 1 byte |

1. **Kiểu logic**

|  |  |  |
| --- | --- | --- |
| **Kiểu** | **Phạm vi giá trị** | **Bộ nhớ** |
| BOOLEAN | true hoặc false | 1 byte |

xxxxx

Bài 5: KHAI BÁO BIẾN

Trong Pascal, khai báo biến bắt đầu bằng từ khóa **var** và có dạng :

***var<*** *danh sách biến****>* : *<****kiểu dữHệư****>;***

*<danh sách biến>* là một hoặc nhiều tên biến, các tên biến được viết cách nhau bởi dấu phẩy ***,***

*<kiểu dữ'liệư>* thường là một trong các kiểu dữ liệu chuẩn hoặc kiểu dữ liệu do người lập trình định nghĩa.

* Cần đặt tên biến sao cho gợi nhớ đến ý nghĩa của biến đó.
* Không nên đặt tên biến quá ngắn hay quá dài.
* Khi khai báo biến cần đặc biệt lưu ý đến phạm vi giá trị của nó.

xxxxx

Bài 6: PHÉP TOÁN, BIỂU THỨC, CÂU LỆNH GÁN

**1. Phép toán**

|  |  |  |
| --- | --- | --- |
| **Phép toán** | **Trong toán học** | **Trong Pascal** |
| Phép toán số học với số nguyên | + (cộng), - (trừ), x (nhân), div (chia nguyên), mod (lấy phần dư) | +, -, \*, div,  mod |
| Phép toán số học với số thực | + (cộng), - (trừ), x (nhân), : (chia) | +, -, \*, / |
| Phép toán quan hệ | < (nhỏ hơn), < (nhỏ hơn hoặc bằng),  > lớn hơn, > (lớn hơn hoặc bằng),  = (bằng), (khác) | <, <=,  >, >=,  =, <> |
| Phép toán logic | (phủ định), V (hoặc), A (và) | not, or, and |

1. **Biểu thức số học**

* Biểu thức số học là một biến kiểu số hoặc một hằng số hoặc các biến kiểu số và các hằng số liên kết với nhau bởi một số hữu hạn phép toán số học, các dấu ngoặc tròn ( và ).
* Qui tắc viết :

o Chỉ dùng cặp ngoặc tròn để xác định trình tự thực hiện phép toán khi cần thiết.

o Viết lần lượt từ trái sang phải.

o Không được bỏ qua dấu nhân (\*) trong tích.

* Thứ tự ưu tiên thực hiện của các phép toán :

o Thực hiện các phép toán trong ngoặc trước.

o Trong dãy các phép toán không chứa ngoặc thì thực hiện từ trái sang phải, theo thứ tự sau : \* / div mod + -

(phép cộng và phép trừ thực hiện sau cùng).

1. **Hàm số học chuẩn**

Các hàm số học chuẩn là các chương trình tính giá trị những hàm toán học thường, có sẵn trong thư viện của các NNLT.

Trong Pascal có một số hàm chuẩn:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Hàm** | **Biểu diễn toán học** | **Biểu diễn trong Pascal** | **Kiểu đối số** | **Kiểu kết quả** |
| Bình phương | *X2* | SQR(X) | Thực hoặc Nguyên | Theo kiểu của đối số |
| Căn bậc hai | *4X* | SQRT(X) | Thực hoặc Nguyên | Thực |
| Giá trị tuyệt đối | *X*I | ABS(X) | Thực hoặc Nguyên | Theo kiểu của đối số |
| Lôgarit tự nhiên | LnX | LN(X) | Thực | Thực |
| Luỹ thừa của số  e | *ex* | EXP(X) | Thực | Thực |
| Sin | SinX | SIN(X) | Thực | Thực |
| Cos | CosX | COS(X) | Thực | Thực |

1. **Biểu thức quan hệ**

Biểu thức quan hệ là các biểu thức cùng kiểu liên kết với nhau bởi các phép toán quan hệ.

Biểu thức quan hệ có dạng:

***<biểu thức 1><phép toán quan hệ><biểu thức2>***

Biểu thức quan hệ được thực hiện theo trình tự

o Tính giá trị các biểu thức

o Thực hiện phép toán quan hệ

o Kết quả của biểu thức là giá trị logic true (đúng) hoặc false (sai).

1. **Biểu thức logic**

*Biểu thức logic đơn giản* là biến logic hoặc hằng logic.

*Biểu thức logic* là các biểu thức logic đơn giản, các biểu thức quan hệ liên kết với nhau bởi phép toán logic. Biểu thức logic có giá trị là *true* hoặc *false.*

1. **Câu lệnh gán**

Trong Pascal lệnh gán có dạng

***<tên biến>* := *<biểu thức>* ;**

* Chức năng của lệnh gán là đặt cho biến có tên ở vế trái dấu ":=" giá trị mới bằng giá trị của biểu thức ở vế phải.
* Kiểu của biểu thức bên phải lệnh ":=" phù hợp với kiểu của biến ở bên trái lệnh ":=".

Ví dụ :

x := 5; sau lệnh gán này x có giá trị 5

y := x - 2; sau lệnh gán này y có giá trị 3

Bài 7: CÁC THỦ TỤC CHUẨN VÀO / RA ĐƠN GIẢN

Các chương trình đưa dữ liệu vào (từ bàn phím, đĩa) và đưa dữ liệu ra (hiển thị trên màn hình, in ra giấy, lưu trên đĩa) được gọi chung là các *thủ tục vào/ra đơn giản.*

1. **Nhập dữ liệu vào từ bàn phím**

Dùng các thủ tục chuẩn **read/readln** có dạng:

**read (<***danh sách biến vào****>);***

**readln (<***danh sách biến vào****>);***

Trong đó,*danh sách biến vào* là *một*ìoặc *nhều*tên biến đơn. Trong trường hợp nhiều biến thì các tên biếnđược viết cách nhau bởi *dấu phẩy.*

1. **Đưa dữ liệu ra màn hình**

Để đưa thông tin ra màn hình từ vị trí con trỏ, có 2 thủ tục chuẩn

**write/writeln**dạng:

**write (<** *danh sách kết quả ra****>);***

**writeln(<** *danh sách kết quả ra****>);***

Trong đó,*danh sách kết quả in* ra có thể là *tên bến, biểu thức, hàm* hoặc *hằng.* Trong trường hợp danh sách kết quả ra gồm nhiều thành phần thì các thành phần được viết cách nhau bởi dấu phẩy.

* Với thủ tục *write,* sau khi đưa các kết quả ra màn hình, con trỏ không chuyển xuống dòng tiếp theo.
* Với thủ tục *writeln,* sau khi đưa thông tin ra màn hình, con trỏ sẽ được chuyển xuống đầu dòng tiếp theo.

*Qui cách ra : (tham khảo thêm SGK)*

o *Đối với số thực :*

*: <độ rộng> : <số chữ số thập phân>*

o *Đối v ới các kết quả khác*

*: <độ rộng>*

xxxxx

Bài 8: SOẠN THẢO, DỊCH, THỰC HIỆN VÀ HIỆU CHỈNH CHƯƠNG  
TRÌNH

1. **Soạn thảo**

* Gõ nội dung của chương trình.
* Nhấn phím F2 để lưu.

1. **Biên dịch chương trình**

ALT + F9

1. **Chạy chương trình**

CTRL + F9

1. **Đóng cửa sổ chương trình**

ALT + F3

1. **Thoát khỏi phần mềm**

ALT + X

xxxxx