

BÀI 4. MỘT SỐ KIỂU DỮ LIỆU CHUẨN

- Dữ liệu là thông tin đã mã hóa trong máy tính. Dữ liệu sau khi tập hợp lại và xử lý sẽ cho ta thông tin.
 - Mỗi ngôn ngữ lập trình thường cung cấp một số *kiểu dữ liệu chuẩn* cho biết phạm vi giá trị có thể lưu trữ, dung lượng bộ nhớ cần thiết để lưu trữ và các phép toán tác động lên dữ liệu.
 - Mỗi *kiểu dữ liệu* được đặc trưng bởi tên kiểu, miền giá trị, kích thước trong bộ nhớ, các phép toán, các hàm và thủ tục sử dụng chúng.
 - Một số *kiểu dữ liệu chuẩn* thường dùng cho các biến đơn trong Pascal, đó là:
1. **Kiểu nguyên:** các *kiểu nguyên* được lưu trữ và kết quả tính toán là số nguyên nhưng có hạn chế về miền giá trị. Tập số nguyên là vô hạn và có thứ tự, đếm được nhưng trong máy tính thì *kiểu nguyên* là hữu hạn, có thứ tự.

Kiểu	Bộ nhớ lưu trữ giá trị	Phạm vi giá trị	
<u>Byte</u>	<u>1 byte</u>	từ 0 đến 255	$(0 \rightarrow 2^8 - 1)$
<u>Integer</u>	<u>2 byte</u>	từ -32766 đến 32767	$(-2^{15} \rightarrow 2^{15} - 1)$
<u>Word</u>	<u>2 byte</u>	từ 0 đến 65535	$(0 \rightarrow 2^{16} - 1)$
<u>Longint</u>	<u>4 byte</u>	từ -2147483648 đến 2147483647	$(-2^{31} \rightarrow 2^{31} - 1)$

2. **Kiểu thực:** các *kiểu thực* được lưu trữ và kết quả tính toán chỉ là gần đúng với sai số không đáng kể, nhưng **miền giá trị** được mở rộng hơn so với *kiểu nguyên*. Số thực trong máy tính rời rạc và hữu hạn. Phép toán chứa các toán hạng gồm cả *kiểu nguyên* và *kiểu thực* sẽ cho kết quả là ***kiểu thực***.

Kiểu	Bộ nhớ lưu trữ giá trị	Phạm vi giá trị
<u>Real</u>	<u>6 byte</u>	0 hoặc có giá trị tuyệt đối nằm trong phạm vi từ $2,9 \times 10^{-39}$ đến $1,7 \times 10^{38}$
<u>Extended</u>	<u>10 byte</u>	0 hoặc có giá trị tuyệt đối nằm trong phạm vi từ 10^{-4932} đến 10^{4932}

3. **Kiểu kí tự:** *kiểu kí tự* có tập hợp giá trị là các kí tự trong bảng mã **ASCII**, được dùng khi thông tin là các kí tự, xâu (string). *Kiểu kí tự* là kiểu có thứ tự, đếm được. Kí tự đặc biệt dùng để thể hiện sự ngăn cách giữa hai từ viết liên tiếp trong các văn bản là **dấu cách**.

Kiểu	Bộ nhớ lưu trữ giá trị	Phạm vi giá trị
<u>char</u>	<u>1 byte</u>	<u>256</u> kí tự trong bộ mã ASCII

4. **Kiểu logic:** *kiểu logic* trong Pascal chỉ có hai giá trị là **TRUE** hoặc **FALSE**, được dùng khi kiểm tra một *điều kiện* hoặc tìm giá trị của một biểu thức **logic**. *Kiểu logic* cũng là kiểu thứ tự đếm được.

Kiểu	Bộ nhớ lưu trữ giá trị	Phạm vi giá trị
<u>Boolean</u>	<u>1 byte</u>	<u>true</u> hoặc <u>false</u>

BÀI 5. KHAI BÁO BIẾN

- Trong Pascal, mọi biến trong chương trình đều cần phải được khai báo **tên** và **kiểu dữ liệu** của nó. Khai báo biến để cấp phát bộ nhớ cho biến và mỗi biến chỉ được khai báo **một lần** trong chương trình.
- Trong Pascal, khai báo biến bắt đầu bằng từ khóa **var** có dạng:

var <danh sách biến> : <kiểu dữ liệu>;

trong đó:

- Danh sách biến** là một hoặc nhiều tên biến, các tên biến được viết cách nhau bởi **dấu phẩy**.
- Kiểu dữ liệu** thường là một trong các **kiểu dữ liệu chuẩn** hoặc **kiểu dữ liệu** do người lập trình định nghĩa.

Sau từ khóa **var** có thể khai báo **nhiều** danh sách biến khác nhau, tức là cấu trúc:

<danh sách biến> : <kiểu dữ liệu>;

có thể xuất hiện **nhiều lần**.

Ví dụ 1.

```
var a, b, c, d, x1, x2: real;  
    m, n: integer;
```

- Khai báo **biến** thường đặt sau khai báo **hằng**.
- Sau khai báo sẽ có một vùng nhớ dành cho biến này với kích thước **đúng** bằng kích thước **kiểu dữ liệu** của nó để lưu trữ giá trị của biến. Khai báo biến cũng nhằm đưa tên biến vào danh sách các đối tượng cần quản lý của chương trình.
- Kiểu của biến giúp chương trình dịch biết cách tổ chức lưu trữ, truy cập giá trị của biến và áp dụng các thao tác thích hợp trên biến đó.
- Một số chú ý khi khai báo biến:**
 - Cần đặt tên biến sao cho gợi nhớ đến **ý nghĩa** của biến đó. Điều này rất có lợi cho việc đọc, hiểu và sửa đổi chương trình khi cần thiết.
 - Không nên đặt tên biến **quá ngắn** hay **quá dài**, dẫn đến dễ bị mắc lỗi khi viết nhiều lần tên biến.
 - Khi khai báo biến cần đặc biệt lưu ý đến **phạm vi giá trị** của nó.