

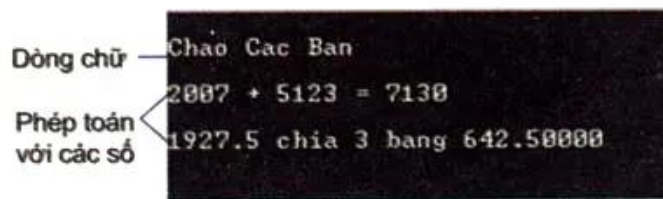
## BÀI 3

# CHƯƠNG TRÌNH MÁY TÍNH VÀ DỮ LIỆU

## 1. Dữ liệu và kiểu dữ liệu

Máy tính là công cụ xử lý thông tin, còn chương trình chỉ dẫn cho máy tính cách thức xử lý thông tin để có kết quả mong muốn. Thông tin rất đa dạng nên dữ liệu trong máy tính cũng rất khác nhau về bản chất. Để dễ dàng quản lý và tăng hiệu quả xử lý, các ngôn ngữ lập trình thường phân chia dữ liệu thành các *kiểu* khác nhau: chữ, số nguyên, số thập phân,...

**Ví dụ 1.** Hình 18 dưới đây minh họa kết quả thực hiện của một chương trình: in ra màn hình với các kiểu dữ liệu quen thuộc là chữ và số.



Hình 18

Các kiểu dữ liệu thường được xử lý theo các cách khác nhau. Chẳng hạn, ta có thể thực hiện các phép toán số học với các số, nhưng với các câu chữ thì các phép toán đó không có nghĩa.

Các ngôn ngữ lập trình định nghĩa sẵn một số kiểu dữ liệu cơ bản. Kiểu dữ liệu xác định các giá trị có thể của dữ liệu và các phép toán có thể thực hiện trên các giá trị đó. Dưới đây là một số kiểu dữ liệu thường dùng nhất:

- **Số nguyên**, ví dụ số học sinh của một lớp, số sách trong thư viện,...
- **Số thực**, ví dụ chiều cao của bạn Bình, điểm trung bình môn Toán,...
- **Xâu kí tự** (hay *xâu*) là dãy các “chữ cái” lấy từ bảng chữ cái của ngôn ngữ lập trình, ví dụ: “Chao cac ban”, “Lop 8E”, “2/9/1945”...

Trong các ngôn ngữ lập trình, dữ liệu kiểu số nguyên còn được phân chia tiếp thành các kiểu nhỏ hơn theo các phạm vi giá trị khác nhau, dữ liệu kiểu số thực được phân chia thành các kiểu có độ chính xác (số chữ số thập phân) khác nhau.

Ngoài các kiểu nói trên, mỗi ngôn ngữ lập trình cụ thể còn định nghĩa nhiều kiểu dữ liệu khác. Số các kiểu dữ liệu và tên kiểu dữ liệu trong mỗi ngôn ngữ lập trình có thể khác nhau.

Ví dụ 2. Bảng 1 dưới đây liệt kê một số kiểu dữ liệu cơ bản của ngôn ngữ lập trình Pascal:

Tên kiểu	Phạm vi giá trị
<code>integer</code>	Số nguyên trong khoảng $-2^{15}$ đến $2^{15}-1$ .
<code>real</code>	Số thực có giá trị tuyệt đối trong khoảng $2,9 \times 10^{-39}$ đến $1,7 \times 10^{38}$ và số 0.
<code>char</code>	Một kí tự trong bảng chữ cái.
<code>string</code>	Xâu kí tự, tối đa gồm 255 kí tự.

**Bảng 1**

Trong Pascal, để chỉ rõ cho chương trình dịch hiểu dãy chữ số là kiểu xâu, ta phải đặt dãy số đó trong cặp dấu nháy đơn. Ví dụ '5324', '863'.

## 2. Các phép toán với dữ liệu kiểu số

Trong mọi ngôn ngữ lập trình ta đều có thể thực hiện các phép toán số học cộng, trừ, nhân và chia với các số nguyên và số thực.

Chẳng hạn, bảng dưới đây là kí hiệu của các phép toán số học đó trong ngôn ngữ Pascal:

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

**Bảng 2**

Chúng ta đã quen thuộc với các phép toán cộng, trừ, nhân và chia. Tuy nhiên hãy lưu ý rằng hầu hết các ngôn ngữ lập trình đều xem kết quả chia hai số  $n$  và  $m$  (tức  $n/m$ ) là số thực, cho dù  $n$  và  $m$  là các số nguyên và  $n$  có chia hết cho  $m$  hay không.

Dưới đây là các ví dụ về phép chia, phép chia lấy phần nguyên và phép chia lấy phần dư:

$$5/2 = 2.5$$

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

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

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

$$5 \text{ mod } 2 = 1$$

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

Ta có thể kết hợp các phép tính số học nói trên để có các *biểu thức số học* phức tạp hơn. Sau đây là một số ví dụ về biểu thức số học và cách viết chúng trong ngôn ngữ lập trình Pascal:

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

$$a*b - c + d$$

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

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

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

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

Quy tắc tính các biểu thức số học:

- Các phép toán trong ngoặc được thực hiện trước tiên;
- Trong dãy các phép toán không có dấu ngoặc, các phép nhân, phép chia, phép chia lấy phần nguyên và phép chia lấy phần dư được thực hiện trước;
- Phép cộng và phép trừ được thực hiện theo thứ tự từ trái sang phải.

Chú ý rằng khi viết các biểu thức toán, để dễ phân biệt, ta có thể dùng các cặp dấu ngoặc tròn ( và ), dấu ngoặc vuông [ và ], dấu ngoặc nhọn { và } để gộp các phép toán, nhưng trong các ngôn ngữ lập trình chỉ được sử dụng *dấu ngoặc tròn* cho mục đích này.

Ví dụ, biểu thức  $\frac{[(a+b)(c-d)+6]}{3} - a$  khi viết trong Pascal sẽ có dạng:

$$((a + b) * (c - d) + 6) / 3 - a$$

### 3. Các phép so sánh

Ngoài các phép toán số học, ta còn thường *so sánh* các số. Các kí hiệu toán học quen thuộc dưới đây được sử dụng để kí hiệu các phép so sánh:

Kí hiệu	Phép so sánh	Ví dụ
=	bằng	$5 = 5$
<	nhỏ hơn	$3 < 5$
>	lớn hơn	$9 > 6$
≠	khác	$6 \neq 5$
≤	nhỏ hơn hoặc bằng	$5 \leq 6$
≥	lớn hơn hoặc bằng	$9 \geq 6$

**Bảng 3**

Kết quả của phép so sánh chỉ có thể là *đúng* hoặc *sai*. Ví dụ, phép so sánh  $9 \geq 6$  cho kết quả đúng,  $10 = 9$  cho kết quả sai hoặc  $5 < 3$  cũng cho kết quả sai,... Để so sánh giá trị của hai biểu thức, chúng ta cũng sử dụng các kí hiệu nói trên. Ví dụ:

$$5 \times 2 = 9; \quad 15 + 7 > 20 - 3; \quad 5 + x \leq 10$$

Biểu thức thứ nhất là sai ( $10 = 9$ ), còn biểu thức thứ hai ( $22 > 17$ ) là đúng. Biểu thức thứ ba ( $5 + x \leq 10$ ) đúng hoặc sai lại phụ thuộc vào giá trị cụ thể của  $x$ .

Khi viết chương trình, để so sánh dữ liệu (số, biểu thức,...) chúng ta sử dụng các kí hiệu do ngôn ngữ lập trình quy định.

Kí hiệu các phép toán và phép so sánh có thể khác nhau, tùy theo từng ngôn ngữ lập trình.

**Ví dụ 3.** Bảng 4 dưới đây cho thấy kí hiệu của các phép so sánh trong ngôn ngữ Pascal:

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

**Bảng 4**

## 4. Giao tiếp người - máy tính

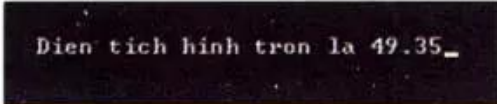
Trong khi thực hiện chương trình máy tính, con người thường có nhu cầu can thiệp vào quá trình tính toán, thực hiện việc kiểm tra, điều chỉnh, bổ sung. Ngược lại, máy tính cũng cho thông tin về kết quả tính toán, thông báo, gợi ý,... Quá trình trao đổi dữ liệu hai chiều như thế thường được gọi là giao tiếp hay tương tác giữa người và máy tính. Với các máy tính cá nhân, tương tác người-máy thường được thực hiện nhờ các thiết bị chuột, bàn phím và màn hình. Dưới đây là một số trường hợp tương tác người-máy.

### a) Thông báo kết quả tính toán

Thông báo kết quả tính toán là yêu cầu đầu tiên đối với mọi chương trình. Ví dụ, câu lệnh

```
write('Diện tích hình tròn là',X);
```

in kết quả tính diện tích hình tròn ra màn hình như hình 19 dưới đây:




```
Diện tích hình tròn là 49.35_
```

Hình 19

### b) Nhập dữ liệu

Một trong những tương tác thường gặp là chương trình yêu cầu nhập dữ liệu. Chương trình sẽ tạm ngừng để chờ người dùng “nhập dữ liệu” từ bàn phím hay bằng chuột. Hoạt động tiếp theo của chương trình sẽ tùy thuộc vào dữ liệu được nhập vào.

Ví dụ, chương trình yêu cầu nhập năm sinh từ bàn phím. Khi đó ta cần gõ một số tự nhiên ứng với năm sinh. Sau khi nhấn phím **Enter** để xác nhận, chương trình sẽ tiếp tục hoạt động.



```
Ban hay nhap nam sinh:
```

Hình 20

Hai câu lệnh Pascal dưới đây sẽ cho kết quả như hình trên:

```
write('Ban hay nhap nam sinh:');  
read(NS);
```

### c) Tạm ngừng chương trình

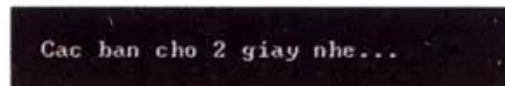
Có hai chế độ tạm ngừng của chương trình: Tạm ngừng trong một khoảng

thời gian nhất định và tạm ngừng cho đến khi người dùng nhấn phím.

**Ví dụ 4.** Giả sử trong chương trình Pascal có các câu lệnh sau:

```
Writeln('Cac ban cho 2 giay nhe...');  
Delay(2000);
```

Sau khi in ra màn hình dòng chữ “Cac ban cho 2 giay nhe...”, chương trình sẽ tạm ngừng trong 2 giây, sau đó mới thực hiện tiếp.

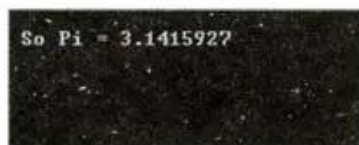


Hình 21

**Ví dụ 5.** Khi chạy đoạn chương trình Pascal có các câu lệnh

```
writeln('So Pi = ',Pi);  
readln;
```

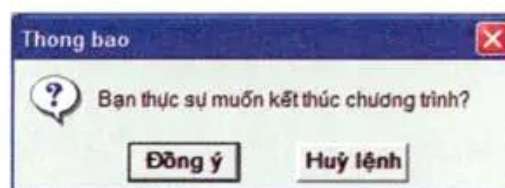
sau khi thông báo kết quả tính số  $\pi$ , chương trình sẽ tạm ngừng chờ người dùng nhấn phím **Enter**, rồi mới thực hiện tiếp.



Hình 22

#### d) Hộp thoại

Hộp thoại được sử dụng như một công cụ cho việc giao tiếp người-máy tính trong khi chạy chương trình. Ví dụ, khi người dùng thao tác để thoát khỏi chương trình đang chạy, hộp thoại dạng sau đây có thể xuất hiện:



Hình 23

Khi đó, nếu nhấp chuột vào nút **Đồng ý**, chương trình sẽ kết thúc còn nhấp nút **Huỷ lệnh**, chương trình vẫn tiếp tục như bình thường.

## GHI NHỚ

1. Các ngôn ngữ lập trình thường phân chia dữ liệu cần xử lý theo các kiểu khác nhau, với các phép toán có thể thực hiện trên từng kiểu dữ liệu đó.
2. Quá trình trao đổi dữ liệu hai chiều giữa người và máy tính khi chương trình hoạt động thường được gọi là giao tiếp hoặc tương tác người-máy.

### Câu hỏi và bài tập

1. Hãy nêu ít nhất hai kiểu dữ liệu và một phép toán có thể thực hiện được trên một kiểu dữ liệu, nhưng phép toán đó không có nghĩa trên kiểu dữ liệu kia.

2. Dãy chữ số 2010 có thể thuộc những kiểu dữ liệu nào?

3. Hãy phân biệt ý nghĩa của các câu lệnh Pascal sau đây:

```
Writeln('5+20=', '20+5'); và Writeln('5+20=', 20+5);
```

Hai lệnh sau có tương đương với nhau không? Tại sao?

```
Writeln('100'); và Writeln(100);
```

4. Viết các biểu thức toán dưới đây bằng các kí hiệu trong Pascal:

a)  $\frac{a}{b} + \frac{c}{d}$ ;

b)  $ax^2 + bx + c$ ;

c)  $\frac{1}{x} - \frac{a}{5}(b+2)$ ;

d)  $(a^2 + b)(1+c)^3$ .

5. Chuyển các biểu thức được viết trong Pascal sau đây thành các biểu thức toán:

a)  $(a + b) * (a + b) - x/y$ ;

b)  $b / (a*a + c)$ ;

c)  $a*a / ((2*b + c) * (2*b + c))$ ;

d)  $1 + 1/2 + 1/(2*3) + 1/(3*4) + 1/(4*5)$ .

6. Hãy xác định kết quả của các biểu thức sau đây:

a)  $15 - 8 \geq 3$ ;

b)  $(20 - 15)^2 \neq 25$ ;

c)  $11^2 = 121$ ;

d)  $x > 10 - 3x$ .

7. Viết các biểu thức ở bài tập 6 bằng các kí hiệu trong Pascal.