

CÂU LỆNH ĐIỀU KIỆN

1. Hoạt động phụ thuộc vào điều kiện

Trong cuộc sống hằng ngày, chúng ta thực hiện phần lớn các hoạt động một cách *tuần tự* theo thói quen hoặc theo kế hoạch đã được xác định từ trước. Ví dụ:

- Mỗi sáng em thức dậy, tập thể dục buổi sáng, làm vệ sinh cá nhân, ăn sáng và đến trường,...

- Long thường đi đá bóng cùng các bạn vào sáng chủ nhật hằng tuần.

Tuy nhiên các hoạt động của con người thường bị tác động bởi sự thay đổi của các hoàn cảnh cụ thể. Nhiều hoạt động sẽ bị thay đổi, bị điều chỉnh cho phù hợp.

- “Nếu” em bị ốm, em sẽ không tập thể dục buổi sáng.

- “Nếu” trời không mưa vào ngày chủ nhật, Long đi đá bóng; ngược lại Long sẽ ở nhà giúp mẹ dọn dẹp nhà cửa.

Trong cuộc sống hằng ngày, từ “*nếu*” trong các câu trên được dùng để chỉ một “*điều kiện*”. Các điều kiện đó là: “Em bị ốm” hoặc “Trời mưa”. Hoạt động tiếp theo của em hoặc của bạn Long sẽ phụ thuộc vào các điều kiện đó có xảy ra hay không.

Có thể liệt kê được rất nhiều tình huống tương tự khác, chẳng hạn khi đi trên đường phố nếu gặp đèn đỏ, ta phải dừng lại; nếu khách đến nhà, em pha trà mời khách,...

Tóm lại, có những hoạt động chỉ được thực hiện khi một điều kiện cụ thể được xảy ra. Điều kiện thường là một sự kiện được mô tả sau từ “*nếu*”.


2. Tính đúng hoặc sai của các điều kiện

Mỗi điều kiện nói trên được mô tả dưới dạng một phát biểu. Hoạt động tiếp theo phụ thuộc vào kết quả kiểm tra phát biểu đó đúng hay sai. Vậy kết quả kiểm tra có thể là gì?

Điều kiện	Kiểm tra	Kết quả	Hoạt động tiếp theo
Trời mưa?	Long nhìn ra ngoài trời và thấy trời mưa.	Đúng	Long ở nhà (không đi đá bóng).
Em bị ốm?	Buổi sáng thức dậy, em thấy mình hoàn toàn khoẻ mạnh.	Sai	Em tập thể dục buổi sáng như thường lệ.

Khi kết quả kiểm tra là *đúng*, ta nói điều kiện được *thoả mãn*, còn khi kết quả kiểm tra là *sai*, ta nói điều kiện *không thoả mãn*.

Ngoài những điều kiện gắn với các sự kiện đời thường như trên, trong Tin học chúng ta có thể gặp nhiều dạng điều kiện khác, ví dụ:

- Nếu nháy nút  ở góc trên, bên phải cửa sổ, (thì) cửa sổ sẽ được đóng lại.
- Nếu $X > 5$, (thì hãy) in giá trị của X ra màn hình.
- Nếu (ta) nhấn phím **Pause/Break**, (thì) chương trình (sẽ bị) ngừng.

3. Điều kiện và phép so sánh

Để so sánh hai giá trị số hoặc hai biểu thức có giá trị số, chúng ta đã sử dụng các kí hiệu toán học như: $=$, \neq , $<$, \leq , $>$ và \geq . Chúng ta biết rằng các phép so sánh có kết quả *đúng* hoặc *sai*.

Các phép so sánh có vai trò rất quan trọng trong việc mô tả thuật toán và lập trình. Chúng thường được sử dụng để biểu diễn các điều kiện. Phép so sánh cho kết quả đúng có nghĩa điều kiện được thoả mãn; ngược lại, điều kiện không được thoả mãn.

Ví dụ 1. Ta muốn chương trình in ra màn hình giá trị lớn hơn trong số hai giá trị của các biến a và b . Khi đó giá trị của biến a hoặc b được in ra phụ thuộc vào phép so sánh $a > b$ là đúng hay sai:

“*Nếu* $a > b$, in giá trị của biến a ra màn hình;
ngược lại, in giá trị của biến b ra màn hình.”

Trong trường hợp này điều kiện được biểu diễn bằng phép so sánh $a > b$.

Tương tự, khi giải phương trình bậc nhất dạng tổng quát $bx + c = 0$, để tính nghiệm của phương trình chúng ta cần kiểm tra các điều kiện được cho bằng các phép so sánh $b = 0$ và $c \neq 0$.

4. Cấu trúc rẽ nhánh

Ta đã biết rằng, khi thực hiện một chương trình, máy tính sẽ *thực hiện tuần tự* các câu lệnh, từ câu lệnh đầu tiên đến câu lệnh cuối cùng. Trong nhiều trường hợp, chúng ta muốn máy tính thực hiện một câu lệnh nào đó, nếu một điều kiện cụ thể được thoả mãn; ngược lại, nếu điều kiện không được thoả mãn thì bỏ qua câu lệnh hoặc thực hiện một câu lệnh khác.

Ví dụ 2. Một hiệu sách thực hiện đợt khuyến mãi lớn với nội dung sau: Nếu mua sách với tổng số tiền ít nhất là 100 nghìn đồng, khách hàng sẽ được giảm 30% tổng số tiền phải thanh toán. Hãy mô tả hoạt động tính tiền cho khách.

Ta có thể mô tả hoạt động tính tiền cho khách hàng bằng các bước dưới đây:

Với mỗi khách hàng, thực hiện :

Bước 1. Tính tổng số tiền T khách hàng đã mua sách.

Bước 2. Nếu $T \geq 100000$, số tiền phải thanh toán là $70\% \times T$.

Bước 3. In hoá đơn.

Cách thể hiện hoạt động phụ thuộc vào điều kiện như trên được gọi là *cấu trúc rẽ nhánh dạng thiếu*.

Ví dụ 3. Trong ví dụ 2, chúng ta chỉ biết rằng nếu tổng số tiền không nhỏ hơn 100 nghìn đồng, khách hàng sẽ được giảm 30% tổng số tiền phải thanh toán. Giả sử thêm vào đó, cửa hàng giảm 10% cho những khách hàng chỉ mua với tổng số tiền không đến 100 nghìn đồng.

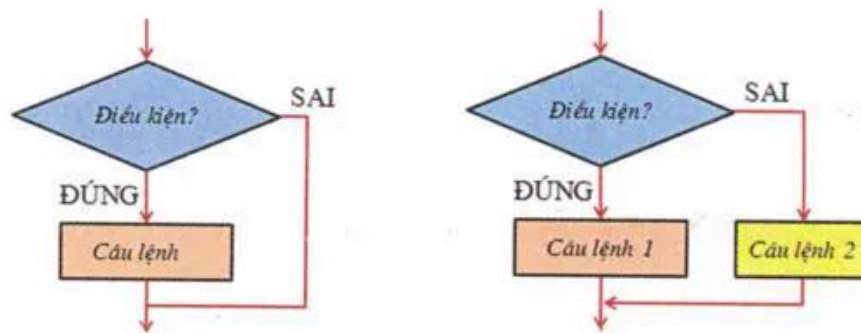
Khi đó, cần phải tính lại tiền cho khách trong cả hai trường hợp, tổng số tiền không nhỏ hơn 100 nghìn đồng và tổng số tiền nhỏ hơn 100 nghìn đồng. Thuật toán có thể được sửa lại như sau:

Bước 1. Tính tổng số tiền T khách hàng đã mua sách.

Bước 2. Nếu $T \geq 100000$, số tiền phải thanh toán là $70\% \times T$; Ngược lại, số tiền phải thanh toán là $90\% \times T$.

Bước 3. In hoá đơn.

Cách thể hiện hoạt động phụ thuộc vào điều kiện như trong ví dụ 2 được gọi là *cấu trúc rẽ nhánh dạng thiếu* (h. 32a), còn trong ví dụ 3 được gọi là *cấu trúc rẽ nhánh dạng đủ* (h. 32b).



a) Cấu trúc rẽ nhánh dạng thiếu

b) Cấu trúc rẽ nhánh dạng đủ

Hình 32

Mọi ngôn ngữ lập trình đều có các câu lệnh để thực hiện các cấu trúc rẽ nhánh dạng thiếu và đủ. Cấu trúc rẽ nhánh giúp cho việc lập trình được linh hoạt hơn.

5. Câu lệnh điều kiện

Trong các ngôn ngữ lập trình, các cấu trúc rẽ nhánh được thể hiện bằng *câu lệnh điều kiện*.

Trong Pascal, câu lệnh điều kiện dạng thiếu được viết với các từ khoá **if** và **then** như sau:

```
if <điều kiện> then <câu lệnh>;
```

Khi gặp *câu lệnh điều kiện* này, chương trình sẽ kiểm tra *điều kiện*. Nếu *điều kiện* được thoả mãn, chương trình sẽ thực hiện *câu lệnh* sau từ khoá **then**. Ngược lại, *câu lệnh* đó bị bỏ qua.

Ví dụ 4. Giả sử cần in số a ra màn hình nếu $a > b$:

Nếu $a > b$ thì in ra màn hình giá trị của a .

Thể hiện bằng câu lệnh điều kiện dạng thiếu trong Pascal:

```
if  $a > b$  then write(a);
```

Ví dụ 5. Nhiều chương trình yêu cầu người dùng nhập một số hợp lệ, chẳng hạn không lớn hơn 5, từ bàn phím. Chương trình đọc số, kiểm tra tính hợp lệ và thông báo nếu không hợp lệ. Khi đó các hoạt động của chương trình có thể biểu diễn bằng thuật toán sau đây:

Bước 1. Nhập số a ;

Bước 2. Nếu $a > 5$ thì thông báo lỗi;

Các câu lệnh điều kiện dạng thiếu của Pascal dưới đây sẽ thể hiện thuật toán trên:


```
readln(a);
```

```
if a>5 then write('So da nhap khong hop le.');
```

Ví dụ 6. Cần viết chương trình tính kết quả của a chia cho b , với a và b là hai số bất kì. Phép tính chỉ thực hiện được khi $b \neq 0$. Chương trình cần kiểm tra giá trị của b , nếu $b \neq 0$ thì thực hiện phép chia; nếu $b = 0$ sẽ thông báo lỗi.

Nếu $b \neq 0$ thì tính kết quả

ngược lại thì thông báo lỗi

Dưới đây là câu lệnh Pascal thể hiện cấu trúc rẽ nhánh dạng đủ nói trên:

```
if b<>0 then x:= a/b
```

```
else write('Mau so bang 0, khong chia duoc');
```

Câu lệnh điều kiện **if...then...else...** mô tả trong ví dụ này là câu lệnh điều kiện dạng đầy đủ. Câu lệnh điều kiện dạng đầy đủ của Pascal có cú pháp:

```
if <điều kiện > then <câu lệnh 1> else <câu lệnh 2>;
```

Với câu lệnh điều kiện này, chương trình sẽ kiểm tra **điều kiện**. Nếu **điều kiện** được thoả mãn, chương trình sẽ thực hiện **câu lệnh 1** sau từ khoá **then**. Trong trường hợp ngược lại, **câu lệnh 2** sẽ được thực hiện.

GHI NHỚ

1. Cấu trúc rẽ nhánh được sử dụng để chỉ thị cho máy tính thực hiện các hoạt động khác nhau tùy theo một điều kiện cụ thể có được thoả mãn hay không. Cấu trúc rẽ nhánh có hai dạng: Dạng thiếu và dạng đầy đủ.
2. Trong lập trình, điều kiện trong cấu trúc rẽ nhánh thường được biểu diễn bằng các phép so sánh.
3. Mọi ngôn ngữ lập trình đều có câu lệnh điều kiện để thể hiện các cấu trúc rẽ nhánh.

Câu hỏi và bài tập

1. Em hãy nêu một vài ví dụ về các hoạt động hằng ngày phụ thuộc vào điều kiện.
2. Hãy cho biết các điều kiện hoặc biểu thức sau đây cho kết quả đúng hay sai:
 - a) 123 là số chia hết cho 3.
 - b) Nếu ba cạnh a , b và c của một tam giác thoả mãn $c^2 = a^2 + b^2$ thì tam giác đó có một góc vuông.

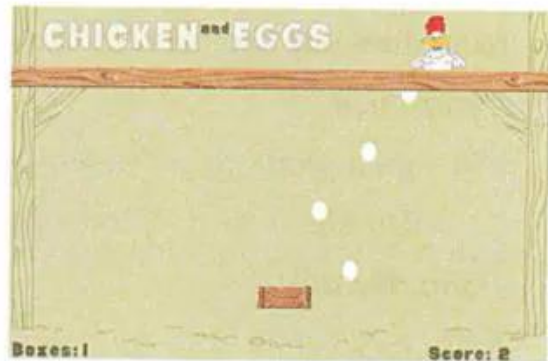
c) $15^2 > 200$.

d) $x^2 < 1$.

3. Hai người bạn cùng chơi trò đoán số. Một người nghĩ trong đầu một số tự nhiên nhỏ hơn 10. Người kia đoán xem bạn đã nghĩ số gì. Nếu đoán đúng, người đoán sẽ được cộng thêm 1 điểm, nếu sai sẽ không được cộng điểm. Luân phiên nhau nghĩ và đoán. Sau 10 lần, ai được nhiều điểm hơn, người đó sẽ thắng.

Hãy phát biểu quy tắc thực hiện một nước đi ở trò chơi. Hoạt động nào sẽ được thực hiện, nếu điều kiện của quy tắc đó thoả mãn? Hoạt động nào sẽ được thực hiện, nếu điều kiện của quy tắc đó không thoả mãn?

4. Một trò chơi máy tính rất hứng thú đối với các em nhỏ là hứng trứng. Một quả trứng rơi từ một vị trí ngẫu nhiên trên cao. Người chơi dùng các phím mũi tên \rightarrow hoặc \leftarrow để điều khiển một chiếc khay di chuyển theo chiều ngang để hứng quả trứng. Mỗi lần người chơi nhấn phím mũi tên (\rightarrow hoặc \leftarrow) thì chiếc khay sẽ dịch chuyển (sang phải hoặc sang trái) một đơn vị khoảng cách. Nếu người chơi không nhấn phím nào hoặc nhấn phím khác hai phím nói trên thì chiếc khay sẽ đứng yên.



Điều kiện để điều khiển chiếc khay trong trò chơi là gì? Hoạt động nào sẽ được thực hiện, nếu điều kiện đó thoả mãn? Hoạt động nào sẽ được thực hiện, nếu điều kiện đó không thoả mãn?

5. Các câu lệnh Pascal sau đây được viết đúng hay sai?

a) `if x := 7 then a = b;`

b) `if x > 5; then a := b;`

c) `if x > 5 then a := b; m := n;`

d) `if x > 5 then a := b; else m := n;`

6. Sau mỗi câu lệnh sau đây

a) `if (45 mod 3) = 0 then X := X + 1;`

b) `if X > 10 then X := X + 1;`

giá trị của biến X sẽ là bao nhiêu, nếu trước đó giá trị của X bằng 5?