

Bài 6. CÂU LỆNH ĐIỀU KIỆN

Thời lượng: 2 tiết

1. Mục đích, yêu cầu

- Biết sự cần thiết của cấu trúc rẽ nhánh trong lập trình;
- Biết cấu trúc rẽ nhánh được sử dụng để chỉ dẫn cho máy tính thực hiện các thao tác phụ thuộc vào điều kiện;
- Biết cấu trúc rẽ nhánh có hai dạng: Dạng thiếu và dạng đủ;
- Biết mọi ngôn ngữ lập trình đều có câu lệnh để thể hiện cấu trúc rẽ nhánh;
- Hiểu cú pháp, hoạt động của các câu lệnh điều kiện dạng thiếu và dạng đủ trong Pascal;
- Bước đầu viết được câu lệnh điều kiện trong Pascal.

2. Những điểm cần lưu ý và gợi ý dạy học

a) Trước khi bắt đầu bài học mới, GV có thể kiểm tra bài cũ và yêu cầu HS nhận xét về thứ tự thực hiện câu lệnh trong các chương trình đã học. GV cần nhấn mạnh cho HS biết rằng: các lệnh trong chương trình được thực hiện theo thứ tự từ trên xuống dưới. Thực hiện các lệnh tuần tự từ đầu đến cuối là thứ tự thực hiện ngầm định (và là cấu trúc điều khiển) của mọi ngôn ngữ lập trình.

b) Bắt đầu bài học mới, GV có thể xuất phát từ những hoạt động phụ thuộc vào điều kiện trong đời sống để dẫn dắt HS đến sự cần thiết cần có cấu trúc rẽ nhánh trong ngôn ngữ lập trình. Nhiều bài toán mà máy tính giúp con người giải quyết là những vấn đề của đời sống thực tiễn. Trong thực tiễn, một công việc có thể được thực hiện nếu như một điều kiện nào đó được thoả mãn. Vì vậy, mọi ngôn ngữ lập trình đều phải có cấu trúc điều khiển cho phép giải quyết các tình huống này. Nghĩa là, trong chương trình một số câu lệnh có thể được thực hiện hoặc không được thực hiện phụ thuộc vào một điều kiện cụ thể nào đó.

c) Cũng từ những ví dụ về hoạt động phụ thuộc vào điều kiện, GV có thể khái quát lên hai cấu trúc rẽ nhánh dạng thiếu và dạng đủ bằng ngôn ngữ tự nhiên như sau:

Nếu... thì...

Nếu... thì... ngược lại thì...

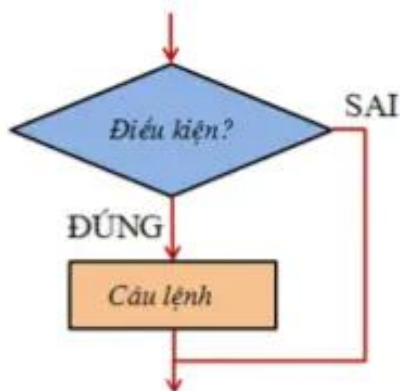
Tương ứng với hai cấu trúc rẽ nhánh dạng thiếu và dạng đủ, trong Pascal có hai câu lệnh điều kiện dạng thiếu và dạng đủ như sau:

Dạng thiếu: `if <điều kiện> then <câu lệnh>;`

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

Sau đó giới thiệu cho HS về sơ đồ hoạt động của câu lệnh điều kiện dạng thiếu và dạng đủ.

Ở dạng thiếu: Nếu *điều kiện* thoả mãn thì *câu lệnh* được thực hiện, ngược lại thì bỏ qua *câu lệnh*.



`if <điều kiện> then <câu lệnh>;`

Ở dạng đủ: Nếu *điều kiện* thoả mãn thì *câu lệnh 1* được thực hiện, ngược lại thì thực hiện *câu lệnh 2*.



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

Trong đó, *câu lệnh*, *câu lệnh 1*, *câu lệnh 2* là câu lệnh của Pascal.

Điều kiện thường là phép so sánh (lưu ý, do HS THCS chưa học về biểu thức quan hệ nên ở đây dùng cụm từ *phép so sánh* để nói về *điều kiện* trong câu lệnh điều kiện). Phép so sánh cho kết quả là đúng tương đương *điều kiện* được thoả mãn, ngược lại phép so sánh cho kết quả sai tương đương với *điều kiện* không thoả mãn.

d) Trong SGK dành mục 2 để nói về điều kiện và phép so sánh. Trọng tâm của mục này là cần cho HS biết trong lập trình điều kiện thường được thể hiện bằng phép so sánh (biểu thức so sánh) và khái niệm điều kiện *được thoả mãn* (hay *không được thoả mãn*) trong đời sống tương đương với khái niệm phép so sánh cho kết quả là *đúng* (hay *sai*) trong ngôn ngữ lập trình.

Các phép so sánh, kí hiệu các phép so sánh trong Pascal đã được giới thiệu ở bài học trước. Tuy nhiên, HS chưa được luyện tập, do vậy GV cần lưu ý hướng dẫn HS luyện tập về phép so sánh cùng với việc tìm hiểu, tập viết câu lệnh điều kiện trong bài học này.

e) Đến đây GV có thể sử dụng các ví dụ ở mục 4 (SGK) để HS luyện tập nhằm hiểu rõ về hoạt động của câu lệnh điều kiện dạng thiếu, dạng đủ, biết ý nghĩa *câu lệnh*, *câu lệnh 1*, *câu lệnh 2* và *điều kiện*. Cần dành thời gian thích đáng cho HS luyện tập qua các ví dụ minh hoạ (khoảng 1 tiết học). Các ví dụ ở đây cần đơn giản, dễ hiểu để HS có thể dễ dàng nắm bắt được hoạt động của câu lệnh điều kiện, ý nghĩa của biểu thức điều kiện, câu lệnh. GV có thể chủ

động chọn các ví dụ khác, nhưng tránh ra những ví dụ quá phức tạp về điều kiện, nhiều phép so sánh.

f) Cuối bài học GV cần khái quát hoá để HS biết cấu trúc rẽ nhánh, hoạt động của cấu trúc rẽ nhánh là giống nhau ở mọi ngôn ngữ lập trình nhưng mỗi ngôn ngữ lập trình lại có những câu lệnh riêng để thể hiện cấu trúc rẽ nhánh.

Lưu ý: Hoàn toàn có thể theo các trình tự giới thiệu trong SGK để tiến hành dạy học bài học này. Tuy nhiên, tiến trình dạy học theo cách giới thiệu ở trên là một phương án khả thi. Phương án này đi từ câu lệnh cụ thể của Pascal, sau đó khái quát thành những kiến thức, nguyên tắc chung cho mọi ngôn ngữ lập trình. Căn cứ vào điều kiện thực tế, GV chủ động lựa chọn cách tiến hành phù hợp.

Việc dịch câu lệnh *if... then...* và *if... then... else...* ra tiếng Việt tương ứng là *nếu... thì...* và *nếu... thì... ngược lại thì...* có thể là cần thiết cho HS để nhớ ý nghĩa của câu lệnh, nhất là với HS chưa được học tiếng Anh.

3. Hướng dẫn trả lời câu hỏi và bài tập

Bài 1. Có thể nêu vài ví dụ về các hoạt động hằng ngày phụ thuộc vào điều kiện. Dưới đây là một số ví dụ:

- Nếu đạt điểm tổng kết cả năm cao hơn 8.0, em sẽ đạt danh hiệu "*HS giỏi*".
- Nếu bị ốm, em (cần phải) đến phòng khám để bác sĩ khám bệnh.
- Nếu không được tưới đủ nước đúng thời kì phát triển, lúa sẽ không cho thu hoạch cao.

Bài 2. Đáp án: a) Đúng; b) Sai; c) Đúng; d) Sai, nếu $x \geq 1$ hoặc $x \leq -1$.

Bài 3. Giả sử *Điểm₁* là số điểm của người thứ nhất và *Điểm₂* là số điểm của người thứ hai, ngoài ra người thứ nhất nghĩ trong đầu một số tự nhiên $n < 10$.

Điều kiện ở trò chơi là người thứ hai đoán đúng số n . Khi đó *Điểm₂* được cộng thêm 1; ngược lại, *Điểm₂* được giữ nguyên. Tương tự, nếu người thứ hai nghĩ số tự nhiên m và điều kiện thứ hai là người thứ nhất đoán đúng số m đó. Khi đó *Điểm₁* được cộng thêm 1; ngược lại, *Điểm₁* được giữ nguyên.

Điều kiện ở trò chơi là sau 10 lần, nếu $Điểm_1 > Điểm_2$ thì người thứ nhất được tuyên bố thắng cuộc; ngược lại, người thứ hai thắng. Trường hợp $Điểm_1 = Điểm_2$ thì không có người thắng và người thua.

Bài 4. Điều kiện để điều khiển chiếc khay trong trò chơi là người chơi nhấn phím mũi tên \rightarrow hoặc \leftarrow . Nếu người chơi nhấn phím \rightarrow , biểu tượng chiếc khay sẽ di chuyển sang phải một đơn vị khoảng cách; nếu phím \leftarrow được nhấn, biểu tượng chiếc khay sẽ di chuyển sang trái. Nếu một phím khác ngoài hai phím mũi tên trên được nhấn, chiếc khay vẫn giữ nguyên vị trí.

Bài 5. a) Sai (thừa ":" chấm ở lệnh $x:=1$, thiếu ":" ở lệnh $a=b$);

b) Sai (thừa dấu chấm phẩy thứ nhất);

c) Đúng, nếu phép gán $m:=n$ không phụ thuộc điều kiện $x>5$; ngược lại, sai và cần đưa hai câu lệnh $a:=b$; $m:=n$; vào giữa cặp từ khoá *begin* và *end*;

d) Sai (thừa dấu chấm phẩy thứ nhất trước *else*).

Bài 6. a) Vì 45 chia hết cho 3, điều kiện được thoả mãn nên giá trị của X được tăng lên 1, tức là bằng 6;

b) Điều kiện không được thoả mãn nên câu lệnh không được thực hiện, tức là X giữ nguyên giá trị 5.

Bài 7. Các bước thực hiện thuật toán:

1) Khai báo biến nhớ n .

2) Nhập số n từ bàn phím.

3) Kiểm tra nếu n chia hết cho 2 thì thông báo n là số chẵn; ngược lại thông báo n là số lẻ.

Chương trình:

```
Var n: integer;  
Begin  
Write('Hay nhap so n: '); Readln(n);  
if (n mod 2) = 0 then writeln(n, ' la so chan')  
  else witeln(n, ' la so le');  
End.
```