

## Bài 7. CÂU LỆNH LẶP

Thời lượng: 3 tiết

### 1. Mục đích, yêu cầu

- Biết nhu cầu cần có cấu trúc lặp trong ngôn ngữ lập trình;
- Biết ngôn ngữ lập trình dùng cấu trúc lặp để chỉ dẫn máy tính thực hiện lặp đi lặp lại công việc nào đó một số lần;
- Hiểu hoạt động của câu lệnh lặp với số lần biết trước *for...do* trong Pascal;
- Viết đúng được lệnh *for...do* trong một số tình huống đơn giản;
- Biết lệnh ghép trong Pascal.

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

a) Giống với cấu trúc rẽ nhánh, cần xuất phát từ những hoạt động trong đời sống thực tiễn có tính chất lặp đi lặp lại để HS hiểu về khái niệm lặp. Ví dụ, tiếng gà trống gáy, tiếng chim hót, tiếng chuông đồng hồ báo thức gọi em dậy mỗi buổi sáng; Các ngày trong tuần các em đều lặp đi lặp lại các hoạt động buổi sáng đến trường và buổi chiều trở về nhà; Hoặc trên lớp, giờ trả bài kiểm tra cô giáo lặp đi lặp lại việc gọi tên HS và ghi điểm của HS vào sổ điểm, cô giáo sẽ ngừng lại khi đã vào điểm cho tất cả HS trong lớp.

Giả sử cô giáo đề nghị em viết chương trình Pascal để in lời chào từng bạn của lớp em (hoặc của nhóm em), cụ thể chương trình cho phép từng bạn nhập tên của mình từ bàn phím và in ra lời chào tương ứng, ví dụ khi một bạn nhập tên là *Mai*, thì chương trình sẽ in ra "*Chao bạn Mai*", một bạn khác nhập tên là *Trung* thì sẽ in ra "*Chao bạn Trung*". Như vậy em sẽ cần viết một chương trình Pascal cho phép lặp đi lặp lại việc nhập tên và hiển thị ra màn hình lời chào. Làm thế nào để chương trình Pascal của em có thể thực hiện việc lặp này?

Giả sử lớp của em có 40 bạn, em hoàn toàn có thể viết 40 lần lệnh để nhập tên và lệnh hiển thị dòng chào. Các lệnh này hoàn toàn giống nhau. Tuy nhiên, một chương trình như vậy thì vừa dài, vừa nhàm chán, dễ sai sót.

Trong Pascal cung cấp một câu lệnh lặp như sau:

```
for <biến đếm>:= <giá trị đầu> to <giá trị cuối> do <câu lệnh>;
```

trong đó:

- *biến đếm* là biến đơn có kiểu nguyên;

- giá trị đầu và giá trị cuối là các biểu thức có cùng kiểu với biến đếm;
- câu lệnh có thể là câu lệnh đơn giản hay câu lệnh ghép.

Nói chung, câu lệnh lặp sẽ thực hiện *câu lệnh* sau từ khoá *do* nhiều lần, mỗi lần là một vòng lặp. Số vòng lặp là biết trước và bằng

$$\text{giá trị cuối} - \text{giá trị đầu} + 1.$$

Trong trường hợp giá trị đầu lớn hơn giá trị cuối thì *câu lệnh* không được thực hiện lần nào.

Khi thực hiện, ban đầu biến đếm sẽ nhận giá trị là *giá trị đầu*, sau mỗi vòng lặp, biến đếm được tự động tăng thêm một đơn vị cho đến khi bằng *giá trị cuối*.

#### **Lưu ý:**

- Để tránh phức tạp, gây khó hiểu với HS, mô tả hoạt động của lệnh *for...do* ở trên được ngầm định *biến đếm*, *giá trị đầu*, *giá trị cuối* là số nguyên. Về lí thuyết, biến đếm, giá trị đầu, giá trị cuối có thể thuộc một kiểu dữ liệu bất kì có thứ tự (ví dụ kiểu số nguyên, kiểu kí tự hoặc kiểu đoạn con của số và kí tự), tuy nhiên, ở đây không đề cập đến những vấn đề này với mục đích để giảm lược nội dung, giúp HS dễ tiếp thu kiến thức mà vẫn đảm bảo những kiến thức, kĩ năng cần thiết theo yêu cầu.

- Trong Pascal cấu trúc *for...do* có hai dạng tiến và lùi:

Dạng tiến:

*for* <biến đếm>:= <giá trị đầu> to <giá trị cuối> *do* <câu lệnh>;

Dạng lùi:

*for* <biến đếm>:= <giá trị cuối> downto <giá trị đầu> *do* <câu lệnh>;

Trong SGK chỉ giới thiệu dạng tiến. Về cơ bản dạng tiến gần gũi với cách suy nghĩ tự nhiên của HS THCS hơn và chỉ cần dạng tiến là đủ, không yêu cầu phải giới thiệu thêm dạng lùi.

b) Khi thực hiện câu lệnh lặp *for...do* các giá trị đầu và giá trị cuối phải được xác định trước. Chính vì thế mà ta biết trước được số lần thực hiện câu lệnh sau từ khoá *do* (số lần lặp bằng *giá trị cuối* - *giá trị đầu* + 1). Chính vì vậy, câu lệnh *for...do* còn được gọi là câu lệnh lặp với số lần biết trước.

c) GV có thể sử dụng chương trình *Lap* trong SGK (ví dụ 3), phân tích ví dụ này để cho HS hiểu rõ về hoạt động của câu lệnh lặp, hiểu về biến đếm, giá trị đầu, giá trị cuối và câu lệnh.

Có thể hướng dẫn HS lập bảng quá trình thực hiện chương trình trên như dưới đây:

Lần lặp thứ	i	Kết quả viết ra màn hình
1	1	Day la lan lap thu 1
2	2	Day la lan lap thu 2
3	3	Day la lan lap thu 3
4	4	Day la lan lap thu 4
5	5	Day la lan lap thu 5
6	6	Day la lan lap thu 6
7	7	Day la lan lap thu 7
8	8	Day la lan lap thu 8
9	9	Day la lan lap thu 9
10	10	Day la lan lap thu 10

d) Sau khi cùng với HS phân tích và thực hiện các chương trình ở ví dụ 3 và 4, GV có thể cùng HS sử dụng câu lệnh *for...do* để viết đoạn câu lệnh nhập tên và hiển thị ra màn hình lời chào cho các bạn trong lớp. Giả sử lớp có 40 bạn thì chương trình có thể được viết như sau:

```

Program Chao_hoi;
var i: integer; ten: string;
begin
  for i:= 1 to 40 do
    Begin
      write('Nhap ten cua ban: '); Readln(ten);
      writeln('Chao ban ', ten);
    end;
end.

```

**Lưu ý:** Có thể giới thiệu nhanh về câu lệnh **ghép** (lệnh nằm trong hai từ khoá **begin ... end**). Khác với chương trình *Lap*, sau từ khoá *do* chỉ có một câu lệnh cần thực hiện, ở chương trình *Chao\_hoi*, (sau từ khoá *do* có hai câu lệnh cần thực hiện). Muốn vậy, hai câu lệnh này cần phải được "gói" trong cặp từ khoá *begin...end*. Cách viết này của Pascal được gọi là câu lệnh ghép. Câu lệnh **ghép** có thể chứa nhiều câu lệnh khác của Pascal. Lưu ý HS trong cấu trúc câu lệnh ghép này sau *end* là dấu chấm phẩy (;), không phải là dấu chấm (.)

e) Cần lấy thêm một số ví dụ khác để HS biết và tập làm quen với các tình huống sử dụng câu lệnh *for...do* và lệnh ghép. Có thể yêu cầu HS đọc hiểu ví dụ có trong SGK (như chương trình *Tinh\_tong*, *Tinh\_Giai\_thua*) hoặc đưa ra bài toán đơn giản cần sử dụng đến câu lệnh *for...do* và hướng dẫn HS viết chương trình.

f) GV cần khái quát cho HS cấu trúc lặp với số lần biết trước có ở mọi ngôn ngữ lập trình, mỗi ngôn ngữ lập trình có câu lệnh riêng để mô tả cấu trúc này. Trên đây các em đã được tìm hiểu về câu lệnh lặp với số lần biết trước trong Pascal (*for...do*).

g) Căn cứ vào tình hình tiếp thu của HS, GV cần lựa chọn và giao một số bài tập cho HS luyện tập, không nhất thiết phải làm hết tất cả các bài tập cuối bài này.

### 3. Hướng dẫn trả lời câu hỏi và bài tập

**Bài 1.** Có thể nêu rất nhiều ví dụ về các hoạt động lặp. Dưới đây là một số ví dụ:

- Hằng ngày em đặt đồng hồ báo thức lúc 6 giờ để dậy sớm tập thể dục.
- Hằng ngày (hoặc hằng tuần) bác lái xe khách lái xe để chuyên chở hành khách xuất phát vào một thời gian và từ một địa điểm nhất định và đi theo một tuyến đường đã xác định trước.
- Mỗi lần được khởi động, máy tính của em sẽ thực hiện cùng các hoạt động tự kiểm tra các thành phần máy tính, sau đó khởi động hệ điều hành theo một trình tự đã được quy định trước.

**Bài 2.** Chương trình này không thực hiện hoạt động nào mặc dù có hẳn một lệnh lặp.

**Bài 3.** Thuật toán tính tổng  $A = \frac{1}{1.3} + \frac{1}{2.4} + \frac{1}{3.5} + \dots + \frac{1}{n(n+2)}$ .

Bước 1.  $A \leftarrow 0, i \leftarrow 1$ .

Bước 2.  $A \leftarrow A + \frac{1}{i(i+2)}$ .

Bước 3.  $i \leftarrow i + 1$ .

Bước 4. Nếu  $i \leq n$ , quay lại bước 2.

Bước 5. Ghi kết quả  $A$  và kết thúc thuật toán.