

§6. PHÉP TOÁN, BIỂU THỨC, CÂU LỆNH GÁN

A. Mục đích, yêu cầu

- Biết các khái niệm: phép toán, biểu thức số học, hàm số học chuẩn, biểu thức quan hệ;
- Hiểu lệnh gán;
- Viết được lệnh gán;
- Viết được các biểu thức số học và lôgic với các phép toán thông dụng.

B. Những điểm cần lưu ý và gợi ý tổ chức dạy học

1. Về các phép toán

Cần giải thích một số phép toán mới với học sinh như *div*, *mod*, *not*, *and*, *or*, các phép toán khác giới thiệu nhanh.

Có thể phân loại thành phép toán số học, phép toán quan hệ và phép toán lôgic.

Bằng một vài ví dụ đơn giản, giáo viên minh họa một vài trường hợp vận dụng các mức độ ưu tiên thực hiện các phép toán trong một biểu thức. Ví dụ, khi giải phương trình bậc hai, viết nghiệm của phương trình có hai nghiệm phân biệt, một nghiệm là $(-b+\sqrt{(\Delta)})/2*a$ sẽ không cho ta kết quả chờ đợi, ta cần viết $(-b+\sqrt{(\Delta)})/(2*a)$.

2. Về các hàm số học chuẩn

Nên chuẩn bị sẵn bảng các hàm chuẩn trên giấy khổ lớn.

3. Về các loại biểu thức

Không nên thiên về trình bày định nghĩa các loại biểu thức, chỉ nên minh họa mỗi loại bằng một hoặc hai ví dụ để hiểu cách viết biểu thức trong một ngôn ngữ lập trình tương ứng cách viết trong toán học như thế nào. Việc hình thành các kỹ năng tạo biểu thức đúng trên cơ sở sử dụng đúng các mức độ ưu tiên của các phép toán sẽ hình thành dần qua các tiết thực hành, chữa bài tập của chương và trong quá trình học các chương sau.

4. Về câu lệnh gán

- a) Câu lệnh gán có chức năng gán giá trị cho một biến, nghĩa là thay giá trị cũ trong ô nhớ (tương ứng với biến) bởi giá trị mới. Giá trị mới là giá trị của một biểu thức. Biểu thức này đã có giá trị xác định thuộc phạm vi của biến.

Kiểu giá trị của biểu thức phải phù hợp với kiểu của biến. Một biến chỉ được coi là xác định giá trị khi đã nhận được giá trị từ ngoài (đọc từ bàn phím hoặc từ tệp,...) hoặc trực tiếp qua lệnh gán trong chương trình. Tuỳ theo ngôn ngữ lập trình, kí hiệu lệnh gán có thể khác nhau (Pascal sử dụng dấu hai chấm và dấu bằng viết liền nhau (`:=`), C/C++ chỉ dùng một dấu bằng).

- b)** Trong Pascal, giáo viên nhấn mạnh bên trái lệnh gán là tên biến, bên phải là biểu thức đã xác định. Giáo viên có thể nêu một vài ví dụ minh họa về lệnh gán không hợp lệ như vẽ trái không là tên biến hoặc vẽ phải chưa xác định được giá trị,...
- c)** Một số điểm chú ý khi sử dụng lệnh gán:
 - Phải viết đúng kí hiệu lệnh gán. Ví dụ, trong Pascal kí tự hai chấm phải viết liền kí tự dấu bằng (`:=`).
 - Biểu thức bên phải cần được xác định giá trị trước khi gán, nghĩa là mọi biến trong biểu thức đã được xác định giá trị và các phép toán trong biểu thức có thể thực hiện được trong miền giá trị của biến.
 - Kiểu của giá trị biểu thức bên phải dấu gán phải phù hợp với kiểu dữ liệu của biến.

C. Kiến thức bổ sung

- Trong Free Pascal, lệnh gán còn được kết hợp với các toán tử kiểu C (chọn Options→Compiler→C-like operators) như sau:

Kí hiệu	Ý nghĩa	Biểu diễn trong Pascal thường	Biểu diễn kết hợp với toán tử kiểu C
<code>+=</code>	Cộng thêm	<code>S := S + i;</code>	<code>S += i;</code>
<code>-=</code>	Trừ đi	<code>S := S - i;</code>	<code>S -= i;</code>
<code>*=</code>	Nhân thêm	<code>S := S * i;</code>	<code>S *= i;</code>
<code>/=</code>	Chia cho	<code>S := S / i;</code>	<code>S /= i;</code>

Cách viết này không chỉ làm cho chương trình ngắn gọn hơn mà còn tạo điều kiện cho chương trình dịch dễ tối ưu mã lệnh.

- Đối với đối tượng học sinh khá, giỏi, trong giờ ngoại khoá, giáo viên có thể giới thiệu thêm về một vài phép toán khác thường dùng và giới thiệu về độ ưu tiên của các phép toán khác (theo thứ tự các phép toán `*`, `/`, `div`, `mod` thực hiện trước và các phép toán `+`, `-` thực hiện sau). Chẳng hạn trong Pascal, có thể giới thiệu:
 - Phép dịch chuyển bit `SHL`, `SHR` trong kiểu số nguyên.

Ví dụ

$x \text{ SHL } N$ là $x * 2^N$

và $x \text{ SHR } N$ là $x / 2^N$.

- Toán tử IN trong phép toán quan hệ.

Ví dụ

Biểu thức $3 \text{ IN } [1..7]$ cho giá trị *true* vì số 3 thuộc tập gồm các số nguyên từ 1 đến 7.

- Độ ưu tiên của các phép toán trong Pascal cho ở bảng 3 sau đây:

Bảng 3

Toán tử	Độ ưu tiên
NOT	1 (cao nhất)
* / DIV MOD AND SHL SHR	2
+ - OR XOR	3
= <> < > <= >= IN	4 (thấp nhất)

Quy luật áp dụng độ ưu tiên là:

- Một toán hạng nằm giữa hai phép toán có độ ưu tiên khác nhau thì nó được thực hiện bởi phép toán có độ ưu tiên cao hơn.
- Một toán hạng nằm giữa hai phép toán có độ ưu tiên bằng nhau thì nó được thực hiện bởi phép toán bên trái nó.
- Biểu thức trong ngoặc được tính trước và sau đó sẽ được coi như một toán hạng của phép toán tiếp theo (nếu còn).