

## II - HƯỚNG DẪN CHI TIẾT

### §1. KHÁI NIỆM LẬP TRÌNH VÀ NGÔN NGỮ LẬP TRÌNH

#### A. Mục đích, yêu cầu

- Hiểu khả năng của ngôn ngữ lập trình bậc cao, phân biệt được với ngôn ngữ máy và hợp ngữ.
- Hiểu ý nghĩa và nhiệm vụ của chương trình dịch. Phân biệt được biên dịch và thông dịch.

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

##### *1. Về ngôn ngữ lập trình*

Trước hết, giáo viên có thể giúp học sinh liên hệ với kiến thức đã học ở lớp 10 bằng những câu hỏi như:

- Có những loại ngôn ngữ lập trình nào?
- Phân biệt ngôn ngữ bậc cao với các ngôn ngữ lập trình khác ở những nội dung nào?

- Kể tên một số ngôn ngữ lập trình mà học sinh biết.
- Tại sao người ta phải xây dựng các ngôn ngữ lập trình bậc cao? (câu hỏi 1, phần câu hỏi và bài tập cuối chương I).

Giáo viên cần nêu được hai nội dung chính phân biệt ngôn ngữ lập trình bậc cao với các ngôn ngữ khác là chương trình viết bằng ngôn ngữ lập trình bậc cao nói chung không phụ thuộc vào loại máy và phải được dịch sang ngôn ngữ máy mới thực hiện được.

## 2. Về khái niệm lập trình

Khái niệm lập trình chưa được nêu đầy đủ ở lớp 10 nên giáo viên cần phân tích rõ nội dung này như sách giáo khoa đã nêu và nhấn mạnh ý nghĩa của lập trình là để tạo ra các chương trình giải được các bài toán trên máy tính. Trong lập trình, khía cạnh tổ chức dữ liệu cần được quan tâm đúng mức. Trong từng trường hợp, người lập trình cần cân nhắc chọn cấu trúc dữ liệu cho thích hợp.

Giáo viên có thể giải thích thêm về câu lệnh:

- Câu lệnh diễn tả các thao tác trong các bước của thuật toán.
- Câu lệnh đơn thực hiện bước có một thao tác.
- Câu lệnh cấu trúc thực hiện bước gồm dãy các thao tác.

## 3. Về chương trình dịch

Trước hết cần nêu rõ vai trò của chương trình dịch. Thông qua ví dụ nêu trong sách giáo khoa (hoặc ví dụ khác), giáo viên hướng dẫn học sinh phân biệt được biên dịch và thông dịch (những điểm giống và khác nhau). Với chương trình dịch, có thể coi *chương trình nguồn* (chương trình viết trên ngôn ngữ bậc cao) là dữ liệu vào và *chương trình đích* (chương trình đã được dịch thành chương trình máy tính hiểu được - thông thường là chương trình trên ngôn ngữ máy) là kết quả ra.



Có thể giới thiệu tên một số trình thông dịch (như Python, Perl, PHP,...), giúp tiết kiệm thời gian phát triển ứng dụng (vì không cần biên dịch và liên kết). Trong chế độ tương tác, trình thông dịch này có thể cho phép nhập vào một

biểu thức tại từng thời điểm rồi nhấn phím **Enter** sẽ cho ngay kết quả, điều này có lợi cho quá trình nghiên cứu tính năng của ngôn ngữ hoặc chạy thử khi phát triển phần mềm.

### C. Kiến thức bổ sung

- Khái niệm lập trình gắn chặt với các khái niệm thuật toán và cấu trúc dữ liệu. Lập trình thực chất là dùng các cấu trúc dữ liệu và câu lệnh của một ngôn ngữ lập trình để mô tả cấu trúc dữ liệu thích hợp và cài đặt thuật toán thành chương trình nhằm giải bài toán trên máy tính. Thuật toán và cấu trúc dữ liệu luôn đi liền với nhau. Vì vậy sách giáo khoa đã nêu: "*Lập trình là sử dụng cấu trúc dữ liệu và các câu lệnh của ngôn ngữ lập trình cụ thể để mô tả dữ liệu và diễn đạt các thao tác của thuật toán*". Tuy nhiên, khái niệm lập trình như vậy chỉ được hiểu trong phạm vi tin học phổ thông. Thực ra, lập trình còn bao hàm một khía cạnh khác nữa đó là giai đoạn tìm tòi, phát minh thuật toán hoặc tìm cách tổ chức cấu trúc dữ liệu mới.
- Chương trình dịch thực hiện hai giai đoạn: phân tích và tổng hợp. Giai đoạn phân tích nhằm phân tích chương trình nguồn về từ vựng và cú pháp. Giai đoạn tổng hợp nhằm tạo ra chương trình đích gồm ba bước:
  - Sinh mã trung gian (chuyển chương trình nguồn về chương trình trung gian);
  - Tối ưu mã (chỉnh sửa, tối ưu chương trình trung gian);
  - Sinh mã (tạo chương trình đích từ chương trình trung gian đã tối ưu).
- Cụm từ "bộ dịch" trong nhiều trường hợp được coi là môi trường lập trình bao gồm cả ngôn ngữ lập trình, chương trình dịch và môi trường tích hợp phát triển IDE (Integrated Development Environment, gồm bộ tiền xử lí, bộ chương trình soạn thảo, bộ kết hợp và gỡ rối, bộ quản lí tệp, liên kết tệp, thực hiện chương trình,... tạo ra nhiều dịch vụ được hỗ trợ trên môi trường hệ điều hành). Ví dụ chương trình gỡ rối (debug) có khả năng phát hiện lỗi rồi thông báo về vị trí lỗi, loại lỗi và các lỗi khác liên quan tạo các điều kiện thuận lợi cho người lập trình sửa lỗi đó. Chẳng hạn, môi trường lập trình Dev-C++ được coi là một môi trường tích hợp phát triển đầy đủ IDE cho ngôn ngữ lập trình C/C++, dùng Mingw của GCC (GNU Compiler Collection), chương trình gỡ rối GDB, các thư viện tĩnh và thư viện động DLL,... Môi trường lập trình Free-C có môi trường tích hợp phát triển Borland C++ Builder 6.0 và chương trình dịch Mingw32 2.95,...
- Một ngôn ngữ lập trình có thể có nhiều bộ dịch khác nhau. Ví dụ, với C/C++ có thể kể ra các bộ dịch như C1, C2, C3, C4.5++, pch, whereis, MMComBsc, zlib, Free C, Dev-C++, Boolean C, Turbo C, VC++ 6, VC++7.0, Digital Mas,... Với

Pascal, có thể kể ra các bộ dịch như Turbo Pascal 7.0, Free Pascal 2.0.2,... Các bộ dịch khác nhau ở tốc độ dịch, khả năng hỗ trợ các dịch vụ trên các môi trường khác nhau.

- Trong quá trình phát triển phần mềm của hệ thống tin học, đến nay đã có hàng nghìn ngôn ngữ lập trình xuất hiện. Việc phân chia các ngôn ngữ lập trình thành ba lớp được căn cứ vào *mức độ mô tả các thao tác* với mức thấp là tập hữu hạn các lệnh máy đến mức cao là ngày càng gần với ngôn ngữ tự nhiên. Ba lớp đó là ngôn ngữ máy, hợp ngữ, ngôn ngữ bậc cao. Trong ngôn ngữ bậc cao có thể kể ra một số ngôn ngữ như ALGOL, APL, BASIC, C, C++, C#, COBOL, Delphi, Fortran, Java, JavaScript, Lisp, Logo, Pascal, Perl, PHP, PL/I, Prolog, Python, Ruby, Visual Basic, Visual Foxpro,... (xem bài đọc thêm 1. "*Bạn biết gì về các ngôn ngữ lập trình?*" trong sách giáo khoa Tin học 11 hoặc tra cứu trên Internet).