

TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM
TRƯỜNG TRUNG CẤP KINH TẾ - KỸ THUẬT BÌNH THUẬN

GIÁO TRÌNH
MÔN HỌC: LẬP TRÌNH CƠ BẢN
NGÀNH: CÔNG NGHỆ THÔNG TIN
TRÌNH ĐỘ: TRUNG CẤP

*(Ban hành kèm theo Quyết định số:/QĐ-TC .ngày....tháng...năm 202...
của Trường Trung cấp Kinh tế - Kỹ thuật Bình Thuận)*

Bình Thuận, năm 2023
(Lưu hành nội bộ)

TUYÊN BỐ BẢN QUYỀN

Tài liệu này thuộc loại sách giáo trình nên các nguồn thông tin có thể được phép dùng nguyên bản hoặc trích dùng cho các mục đích về đào tạo và tham khảo.

Mọi mục đích khác mang tính lệch lạc hoặc sử dụng với mục đích kinh doanh thiếu lành mạnh sẽ bị nghiêm cấm.

TaiLieu.vn

LỜI GIỚI THIỆU

Thông tin là nguồn tài nguyên quý giá của một tổ chức. Các phần mềm máy tính là những công cụ hiệu quả để xử lý thông tin và hệ quản trị cơ sở dữ liệu là công cụ phổ biến cho phép lưu trữ và rút trích thông tin một cách hiệu quả.

Hệ quản trị cơ sở dữ liệu được dùng phổ biến nhất hiện nay và được hỗ trợ bởi nhiều nhà cung cấp phần mềm. Tính hiệu quả của các ứng dụng phụ thuộc vào chất lượng của việc tổ chức dữ liệu. Những cải tiến trong kỹ thuật và xử lý cơ sở dữ liệu đưa đến các cơ hội sử dụng thông tin một cách linh hoạt và hiệu quả khi dữ liệu được tổ chức và lưu trữ trong các cấu trúc quan hệ. Hệ quản trị cơ sở dữ liệu là một thành công trong lĩnh vực thương mại.

Hệ Quản trị cơ sở dữ liệu Access có hai khả năng chính cho phép phân biệt nó với các kiểu hệ thống lập trình khác:

- Khả năng quản lý dữ liệu tồn tại lâu dài: đặc điểm này chỉ ra rằng có một cơ sở dữ liệu tồn tại trong một thời gian dài, nội dung của cơ sở dữ liệu này là các dữ liệu mà hệ quản trị cơ sở dữ liệu truy nhập và quản lý.

- Khả năng truy nhập các khối lượng dữ liệu lớn một cách hiệu quả.

Sự bùng nổ của các loại hình dịch vụ cùng sự phát triển của hệ thống ứng dụng thông tin nghiệp vụ đòi hỏi cần phải có những máy chủ với khả năng xử lý mạnh, những hệ quản trị cơ sở dữ liệu có thể đáp ứng được những yêu cầu khắt khe về hiệu năng, tính an toàn và tính sẵn sàng dữ liệu. Đó chính là lý do hệ quản trị cơ sở dữ liệu ra đời và được sử dụng rất phổ biến trong các hệ thống thông tin của các tổ chức, doanh nghiệp lớn.

Trên thị trường phần mềm hiện nay ở Việt Nam đã xuất hiện khá nhiều phần mềm hệ quản trị cơ sở dữ liệu như: Microsoft Access, Foxpro, DB2, SQL Server, Oracle,...

Nếu bạn chưa có cơ hội làm quen với một hệ quản trị cơ sở dữ liệu nào, hãy nắm bắt cơ hội để bắt đầu làm quen với Hệ Quản trị cơ sở dữ liệu Microsoft Access 2010.

Bình Thuận, ngày tháng năm 202

Tham gia biên soạn

1.
2.
3.
4.
5.

MỤC LỤC

GIÁO TRÌNH MÔN HỌC

Tên môn học: Lập trình cơ bản

Mã môn học: MH12

Thời gian thực hiện môn học: 75 giờ (Lý thuyết: 15 giờ; Thực hành: 57 giờ; Kiểm tra: 3 giờ)

I. Vị trí, tính chất của môn học

Vị trí: Lập trình cơ bản là môn học cơ sở của chương trình đào tạo nghề Công nghệ thông tin, môn học này được bố trí sau các môn học chung.

Tính chất: Là môn học lý thuyết chuyên về thiết kế cài đặt các thuật toán trong lĩnh vực lập trình.

II. Mục tiêu môn học

- Kiến thức:
 - + Trình bày những khái niệm cơ bản về ngôn ngữ lập trình C: câu lệnh, từ khóa, cú pháp, cách khai báo và sử dụng biến, hàm và các cấu trúc lệnh trong ngôn ngữ C;
 - + Trình bày được ý nghĩa, cách khai báo, cách truy xuất với một số cấu trúc dữ liệu;
 - + Phân tích được một số thuật toán để xử lý một số yêu cầu đơn giản.
- Kỹ năng:
 - + Cài đặt được một số chương trình ứng dụng đơn giản bằng ngôn ngữ lập trình C;
 - + Đọc hiểu và chỉnh sửa các chương trình C có sẵn.
- Năng lực tự chủ và trách nhiệm:
 - + Nghiêm túc và tích cực trong việc học lý thuyết và làm bài tập, chủ động tìm kiếm các bài tập ứng dụng liên quan.

III. Nội dung môn học:

1. Chương trình khung

Stt	Mã MH/MD	Tên môn học, mô đun	Số tín chỉ	Thời gian đào tạo (giờ)	Học kỳ									
									Tổng số	Trong g đó	1	2	3	4
					Lý thuyết	Thực hành	Kiểm tra							
I. Các môn học chung					15	316	116	184	16	151	165	0	0	
1	MH01	Giáo dục chính trị	2	30	15	13	2		30					
2	MH02	Pháp luật	1	15	9	5	1	15						
3	MH03	Giáo dục thể chất	1	30	4	24	2	30						
4	MH04	Giáo dục quốc phòng và an ninh	2	45	21	21	3		45					
5	MH05	Tin học	2	45	15	29	1	45						
6	MH06	Tiếng Anh	4	90	30	56	4		90					
7	MH07	Giáo dục SKSS, SKTD và phòng chống HIV/AIDS	1	16	7	9		16						
8	MH08	Kỹ năng mềm	2	45	15	27	3	45						
II. Các môn học, mô đun đào tạo bắt buộc					57	1.380	317	1.010	53	285	225	435	435	
II.1. Các môn học, mô đun cơ sở					17	375	122	234	19	225	150	0	0	
9	MH09	An toàn	2	30	27	1	2	30						

2. Chương trình chi tiết môn học

Stt	Tên các chương, mục	Thời gian (giờ)			
		Tổng số	Lý thuyết	Thực hành	Kiểm tra
1	Chương 1: Tổng quan về ngôn ngữ C++	4	1	3	
2	Chương 2: Các thành phần cơ bản của C++	4	1	3	
3	Chương 3: Cấu trúc rẽ nhánh	4	1	3	
4	Chương 4: Cấu trúc lựa chọn	4	1	3	
5	Chương 5: Cấu trúc lặp có số bước lặp xác định	4	0	4	
6	Chương 6: Cấu trúc lặp có số bước lặp không xác định	4	0	4	
7	Chương 7: Cấu trúc các lệnh đặc biệt	4	1	2	1
8	Chương 8: Xây dựng chương trình con bằng hàm	4	1	3	
9	Chương 9: Xây dựng hàm sử dụng tham biến và	4	0	4	

	tham trị				
10	Chương 10: Xây dựng hàm sử dụng biến cục bộ, toàn cục	4	0	4	
11	Chương 11: Lập trình sử dụng mảng một chiều	4	1	3	
12	Chương 12: Lập trình sử dụng mảng hai chiều	4	1	2	1
13	Chương 13: Hướng dẫn sử dụng biến con trỏ	4	1	3	
14	Chương 14: Hướng dẫn khai báo và cú pháp của các hàm về chuỗi	4	1	3	
15	Chương 15: Giải các bài toán sử dụng hàm xử lý chuỗi	4	1	2	1
16	Chương 16: Đệ qui	4	1	3	
17	Ôn tập	11	3	8	
Cộng			75	15	57
					3

IV. Điều kiện thực hiện môn học

1. Phòng học chuyên môn hóa/nhà xưởng: Phòng thực hành đạt chuẩn
2. Trang thiết bị máy móc: Máy tính cài đặt các công cụ lập trình (C, C++), máy chiếu.
3. Học liệu, dụng cụ, nguyên vật liệu: máy chiếu đa phương tiện, giáo trình, tài liệu, bài giảng môn Lập trình căn bản.

V. Nội dung và phương pháp đánh giá:

Nội dung:

– Kiến thức:

- + Mối liên hệ giữa cấu trúc dữ liệu và giải thuật;
- + Kiến thức về phân tích thiết kế và đánh giá độ phức tạp thuật toán;
- + Một số thuật toán cơ bản;
- + Các kiểu dữ liệu cơ sở;
- + Các cấu trúc dữ liệu trừu tượng: cách cài đặt và các thao tác;
- + Cấu trúc dữ liệu cây nhị phân: cài đặt và các thao tác trên cây nhị phân;
- + Sắp xếp và tìm kiếm: Tầm quan trọng và các thuật toán.

– Kỹ năng:

- + Sử dụng một số thuật toán cơ bản;
- + Cài đặt danh sách liên kết, ngăn xếp, hàng đợi và các phép toán liên quan;
- + Cài đặt cây nhị phân và các thao tác liên quan;
- + Cài đặt các thuật toán sắp xếp và tìm kiếm.
- Năng lực tự chủ và trách nhiệm: Nghiêm túc và tích cực trong việc học lý thuyết và làm bài tập, chủ động tìm kiếm các bài tập ứng dụng liên quan.

a.i.2. Phương pháp:

– Tự luận;

– Trắc nghiệm;

– Thực hành.

VI. Hướng dẫn thực hiện môn học

1. Phạm vi áp dụng chương trình:

Chương trình môn học được sử dụng để giảng dạy nghề Công nghệ thông tin (ứng dụng phần mềm) và làm tài liệu tham khảo cho các nghề thuộc các ngành nghề kỹ thuật.

2. Hướng dẫn một số điểm chính về phương pháp giảng dạy môn học:

Giáo viên trước khi giảng dạy cần phải căn cứ vào nội dung của từng bài học chuẩn bị đầy đủ các điều kiện thực hiện bài học để đảm bảo chất lượng giảng dạy.

3. Những trọng tâm chương trình cần chú ý

- Một số thuật toán cơ bản
- Các cấu trúc dữ liệu trừu tượng: cách cài đặt và các thao tác;
- Cấu trúc dữ liệu cây nhị phân: cài đặt và các thao tác trên cây nhị phân;
- Sắp xếp và tìm kiếm: Tầm quan trọng và các thuật toán.

4. Tài liệu cần tham khảo:

- Quách Tuấn Ngọc (1998), Ngôn ngữ lập trình C, NXB Giáo Dục.
 - Hoàng Kiếm (2001), Giải một bài toán trên máy tính như thế nào?, tập 1, NXB Giáo Dục. 3.
 - Brian W.Kernigan & Dennis M.Ritchie (2000), *The C Programming Language*. 4.
 - H.M. Deitel and P.J. Deitel (1998), C++ How to program, 2nd Edition, Prentice Hall.
 - Lê Mạnh Thanh, Hoàng Hữu Hạnh, Trương Công Tuấn, Ngôn ngữ lập trình C, 1998, NXB Giáo dục, Việt Nam.

Tailieu.vn

Bài 1

TỔNG QUAN VỀ NGÔN NGỮ LẬP TRÌNH

Mục tiêu:

Trình bày được các khái niệm về lập trình và thuật toán.

Mô tả được các bước xây dựng thuật toán, quá trình xây dựng một phần mềm đúng trình tự.

Rèn luyện kỹ năng làm việc và tư duy logic.

Nội dung chính:

1. Khái niệm về lập trình, thuật toán

1.1. Ngôn ngữ lập trình là gì?

Ngôn ngữ lập trình là một hệ thống được ký hiệu hóa để miêu tả những tính toán (qua máy tính) trong một dạng mà cả con người và máy đều có thể đọc và hiểu được.

Theo định nghĩa ở trên thì một ngôn ngữ lập trình phải thỏa mãn được hai điều kiện cơ bản sau:

- Dễ hiểu và dễ sử dụng đối với người lập trình, để có thể dùng để giải quyết nhiều bài toán khác nhau.

- Miêu tả một cách đầy đủ và rõ ràng các tiến trình, để chạy được trên các hệ máy tính khác nhau.

Mỗi ngôn ngữ lập trình cũng chính là một chương trình, nhưng nó có thể được dùng để tạo nên các chương trình khác. Văn bản được viết bằng ngôn ngữ lập trình để tạo nên chương trình được gọi là **mã nguồn**.

Ngôn ngữ lập trình là một tập con của ngôn ngữ máy tính, được thiết kế và chuẩn hóa để truyền các chỉ thị cho các máy có bộ xử lý (CPU), nói riêng là máy tính. Ngôn ngữ lập trình được dùng để lập trình máy tính, tạo ra các chương trình máy nhằm mục đích điều khiển máy tính hoặc mô tả các thuật toán để người khác đọc hiểu.

Theo tiến trình lịch sử phát triển, ngôn ngữ lập trình có thể được chia ra làm ba loại chủ yếu như sau:

Ngôn ngữ máy (machine language)

Ngôn ngữ máy (machine language) là các chỉ thị dưới dạng nhị phân, can thiệp trực tiếp vào trong các mạch điện tử. Chương trình được viết bằng ngôn ngữ máy thì có thể được thực hiện ngay không cần qua bước trung gian nào. Tuy nhiên chương trình viết bằng ngôn ngữ máy dễ sai sót, cồng kềnh và khó đọc, khó hiểu vì toàn những con số 0 và 1.

Hợp ngữ (assembly language)

Hợp ngữ là một bước tiến vượt bậc đưa ngôn ngữ lập trình thoát ra khỏi ngôn ngữ máy khó hiểu. Ngôn ngữ này xuất hiện vào những năm 1950, nó được thiết kế để máy tính trở nên thân thiện hơn với người sử dụng. Hợp ngữ đưa ra khái niệm biến (variable), nhờ đó mà ta có thể gán một ký hiệu cho một vị trí nào đó trong bộ nhớ mà không phải viết lại địa chỉ này dưới dạng nhị phân mỗi lần sử dụng. Hợp ngữ cũng chứa vài “phép toán giả”, tức là ta có thể biểu diễn mã phép toán dưới dạng phát biểu (hay còn gọi là câu lệnh) thay vì dưới dạng nhị phân. Các câu lệnh bao gồm hai phần: phần mã lệnh (viết tựa tiếng Anh) chỉ phép toán cần thực hiện và phần tên biến chỉ địa chỉ chứa toán hạng của phép toán đó.

Để máy thực hiện được một chương trình viết bằng hợp ngữ thì chương trình đó phải được dịch sang ngôn ngữ máy. Công cụ thực hiện việc dịch đó được gọi là Assembler

Ngôn ngữ cấp cao (High level language)

Ngôn ngữ cấp cao (High level language): là ngôn ngữ được tạo ra và phát triển nhằm phản ánh cách thức người lập trình nghĩ và làm. Ngôn ngữ cấp cao rất gần với ngôn ngữ con người (Anh ngữ) nhưng chính xác như ngôn ngữ toán học. Nhờ ngôn ngữ cấp cao mà lĩnh vực lập trình trở nên phổ biến, rất nhiều người có thể viết được chương trình, và nhờ thế mà các phần mềm phát triển như vũ bão, phục vụ nhiều lĩnh vực của cuộc sống. Cùng với sự phát triển của các thế hệ máy tính, ngôn ngữ lập trình cấp cao cũng được phát triển rất đa dạng và phong phú, việc lập trình cho máy tính vì thế mà cũng có nhiều khuynh hướng khác nhau: lập trình cấu trúc, lập trình hướng đối tượng, lập trình logic, lập trình hàm... Một chương trình viết bằng ngôn ngữ cấp cao được gọi là chương trình nguồn (source programs). Để máy tính “hiểu” và thực hiện được các lệnh trong chương trình nguồn thì phải có một chương trình dịch để dịch chương trình nguồn (viết bằng ngôn ngữ cấp cao) thành chương trình đích.

Trong khuôn khổ tài liệu này, thuật ngữ ngôn ngữ lập trình dùng để chỉ ngôn ngữ lập trình cấp cao.

Việc nghiên cứu tốt ngôn ngữ lập trình sẽ đạt được các lợi ích như sau:

Cho phép lựa chọn một ngôn ngữ lập trình phù hợp với dự án thực tế

Hiện nay có rất nhiều dự án công nghệ thông tin ứng dụng vào nhiều lĩnh vực khác nhau của cuộc sống. Do tính chất của từng dự án mà phần mềm có thể được cài đặt bằng các ngôn ngữ lập trình khác nhau. Với một vốn kiến thức rộng về ngôn ngữ lập trình, những người làm dự án có thể lựa chọn nhanh chóng một ngôn ngữ lập trình phù hợp với đề án thực tế. Chẳng hạn có thể lựa chọn ngôn ngữ lập trình Java cho các dự án lập trình truyền thông, hay hướng lập trình logic cho các dự án về trí tuệ nhân tạo.

Sử dụng một cách có hiệu quả các công cụ của ngôn ngữ

Các ngôn ngữ nói chung đều cung cấp những công cụ đặc biệt để tạo ra các tiện ích cho lập trình viên, nhưng khi sử dụng chúng không đúng đắn có thể sẽ gây ra những sai lầm lớn. Một ví dụ điển hình là phép đệ quy- một công cụ lập trình đặc biệt có hiệu lực trong nhiều ngôn ngữ. Khi sử dụng đệ quy một cách đúng đắn thì có thể cài đặt một giải thuật đẹp đẽ và có hiệu quả. Nhưng trong trường hợp khác nó có thể gây ra một sự lãng phí thời gian chạy máy rất lớn cho một giải thuật đơn giản. Điều này có thể tránh được nếu như lập trình viên có một sự hiểu biết sâu sắc về ngôn ngữ lập trình và các cài đặt bên trong nó.

Làm tăng vốn kinh nghiệm khi xây dựng các chương trình

Nếu người lập trình đã có sự nghiên cứu một cách rộng rãi nhiều ngôn ngữ mà một trong chúng có cài đặt sẵn những công cụ nào đó thì anh ta có thể tự thiết lập những công cụ tương tự khi phải viết chương trình bởi một ngôn ngữ mà trong đó các công cụ như thế chưa được cài đặt.

Tạo sự dễ dàng học một ngôn ngữ mới

Mặc dù có nhiều ngôn ngữ lập trình khác nhau nhưng chúng đều có những nguyên tắc chung của ngôn ngữ lập trình. Rất nhiều ngôn ngữ có chung cú pháp (sai khác nhau chút ít về cách viết), có chung các kiểu dữ liệu (sai khác nhau chút ít về tên gọi). Việc nắm vững các nguyên lý cơ bản của ngôn ngữ lập trình sẽ là một điều kiện thuận lợi lớn để tiếp cận một cách nhanh chóng với một ngôn ngữ lập trình cụ thể mới. Thực tế cho thấy rằng với những người nắm vững ngôn ngữ lập trình, khi gặp một ngôn ngữ lập trình cụ thể mới, họ có thể vừa nghiên cứu ngôn ngữ mới này vừa áp dụng để lập trình giải quyết một bài toán theo yêu cầu.

Tạo tiền đề để thiết kế một ngôn ngữ mới

Việc thiết kế ngôn ngữ mới là một đòi hỏi của khoa học phát triển ngôn ngữ lập trình. Nếu chúng ta không nghiên cứu về ngôn ngữ lập trình thì không thể nào có kiến thức để xây dựng một ngôn ngữ mới.

Những yếu tố sau tạo nên một ngôn ngữ tốt, nó cũng là những tiêu chuẩn để người lập trình đánh giá ngôn ngữ này tốt hơn ngôn ngữ kia khi lựa chọn một ngôn ngữ để sử dụng. Ngoài ra khi thiết kế một ngôn ngữ lập trình mới, ta cũng phải quan tâm đến các tiêu chuẩn này để có được một ngôn ngữ tốt.

Tính dễ đọc

Tính dễ đọc của một ngôn ngữ lập trình là sự dễ dàng đọc hiểu một chương trình được viết bằng ngôn ngữ đó. Tính dễ đọc được đặc trưng bởi các thuộc tính sau:

Sự giản dị.

Một ngôn ngữ được gọi là có tính giản dị nếu ngôn ngữ đó có ít các thành phần cơ sở, tức là ít các yếu tố được định nghĩa trước. Các ngôn ngữ mà chúng ta có thể đạt được một phép toán bằng nhiều cách khác nhau thì không phải là một ngôn ngữ giản dị. Chẳng hạn trong ngôn ngữ C để tăng thêm một đơn vị cho biến count ta có thể sử dụng nhiều cách như $\text{count} = \text{count} + 1$, $\text{count} += 1$, $\text{count}++$ hoặc $++\text{count}$. Các phép toán chồng (overload) cũng làm cho ngôn ngữ trở nên phức tạp. Chẳng hạn toán tử + có thể hiểu là cộng hai số nguyên, cộng hai số thực, hợp hai tập hợp hay ghép nối hai chuỗi ký tự...

Cấu trúc điều khiển

Các lệnh có cấu trúc cho phép viết các chương trình sáng sủa, dễ đọc, dễ hiểu. Chúng ta có thể nhận thấy điều này trong các ngôn ngữ thuộc thập niên 1960 như BASIC, FORTRAN trong đó do thiếu các cấu trúc điều khiển nên chương trình phải sử dụng nhiều lệnh GOTO, rất khó theo dõi để hiểu chương trình. Ta hãy so sánh hai đoạn chương trình in ra màn hình 10 số tự nhiên đầu tiên được viết bằng ngôn ngữ BASIC (không có lệnh cấu trúc FOR) và ngôn ngữ Pascal.

Kiểu dữ liệu và cấu trúc dữ liệu

Xem xét kiểu dữ liệu và cấu trúc dữ liệu của một ngôn ngữ cũng góp phần đánh giá một ngôn ngữ có dễ đọc hay không. Chẳng hạn trong các ngôn ngữ không có kiểu dữ liệu logic thì phải sử dụng kiểu số để thay thế và do đó mà chương trình trở nên khó đọc. Ví dụ ta hay sử dụng biến found trong các chương trình tìm kiếm một phần tử x trong một mảng a gồm n phần tử. Nếu ngôn ngữ sử dụng có kiểu logic thì ta có thể gán cho found giá trị TRUE hoặc FALSE để biểu diễn trạng thái tìm thấy phần tử cần tìm hay không, ngược lại đối với các ngôn ngữ không có kiểu logic thì ta phải dùng kiểu số và gán cho found giá trị 1 hoặc 0.

Cú pháp

Cú pháp của ngôn ngữ có ảnh hưởng lớn đến sự dễ đọc hiểu của chương trình. Chúng ta xét một số thí dụ sau để thấy rõ vấn đề này.

- Một số ngôn ngữ quy định độ dài tối đa của danh biểu quá ngắn, chẳng hạn trong FORTRAN 77 độ dài tối đa của danh biểu là 6, do đó tên biến nhiều khi phải viết tắt nên khó đọc hiểu.

- Việc sử dụng từ khóa cũng góp phần làm cho ngôn ngữ trở nên dễ đọc. Chẳng hạn trong ngôn ngữ Pascal chỉ sử dụng một từ khóa end để kết thúc một khối, kết thúc một lệnh case hay kết thúc một lệnh hợp thành do đó chương trình trở nên khó đọc, trong khi Ada dùng các từ khóa end if để kết thúc lệnh if, end loop để kết thúc lệnh vòng lặp thì chương trình dễ đọc hơn.

Tính dễ viết

Tính dễ viết của một ngôn ngữ là khả năng sử dụng ngôn ngữ đó để viết một chương trình cho một vấn đề nào đó một cách dễ dàng hay không. Thông thường các ngôn ngữ dễ đọc thì đều dễ viết. Tính dễ viết phải được xem xét trong ngữ cảnh của vấn đề mà ngôn ngữ được sử dụng để giải quyết. Theo đó không thể so sánh tính dễ viết của hai ngôn ngữ cho cùng một bài toán mà một trong hai được thiết kế để dành riêng giải quyết bài toán đó. Ví dụ để giải quyết bài toán quản trị dữ liệu, chúng ta không thể so sánh Pascal với một hệ quản trị cơ sở dữ liệu như Foxpro, Access hay Oracle.

Sau đây là một số yếu tố quan trọng nhất ảnh hưởng tới tính dễ viết của ngôn ngữ.

Sự giản dị

Nếu một ngôn ngữ có quá nhiều cấu trúc thì một số người lập trình sẽ không quen sử dụng hết tất cả chúng. Tốt nhất là có một số nhỏ các cấu trúc ban đầu và một quy tắc để kết hợp chúng thành các cấu trúc phức tạp hơn.

Hỗ trợ cho trừu tượng

Một cách ngắn gọn, trừu tượng (abstraction) là khả năng để định nghĩa và sử dụng các cấu trúc hoặc các phép toán phức tạp theo cách thức mà nó cho phép bỏ qua các chi tiết. Một ví dụ về trừu tượng là chương trình con, từ chương trình gọi, chúng ta gọi chương trình con để thực hiện một tác vụ nào đó mà không cần biết các cài đặt chi tiết bên trong chương trình con đó. Thực chất trừu tượng hóa chính là làm cho chương trình sáng sủa hơn.

Khả năng diễn đạt

Là những công cụ của ngôn ngữ mà người lập trình có thể sử dụng để diễn đạt giải thuật một cách dễ dàng. Nói cách khác, một ngôn ngữ có khả năng diễn đạt là ngôn ngữ cung cấp cho người lập trình những công cụ sao cho người lập trình có thể nghĩ sao thì viết chương trình như vậy. Chẳng hạn lệnh lặp FOR trong Pascal để sử dụng cho cấu trúc lặp với số lần lặp xác định hơn là lệnh WHILE.

Độ tin cậy

Độ tin cậy của một ngôn ngữ lập trình là khả năng của ngôn ngữ hỗ trợ người lập trình tạo ra các chương trình đúng đắn. Độ tin cậy được thể hiện bởi các đặc trưng sau:

Kiểm tra kiểu

Là kiểm tra lỗi về kiểu của chương trình trong giai đoạn dịch hoặc trong khi thực hiện. Kiểm tra kiểu là một yếu tố quan trọng đảm bảo độ tin cậy của ngôn ngữ. Kiểm tra kiểu sẽ báo cho người lập trình biết các lỗi về kiểu và yêu cầu họ có các sửa chữa cần thiết để có một chương trình đúng.

Xử lý ngoại lệ (Exception Handling)

Là một công cụ cho phép chương trình phát hiện các lỗi trong thời gian thực hiện, tạo khả năng để sửa chữa chúng và sau đó tiếp tục thực hiện mà không phải dừng chương trình.

Sự lẩn tên (Aliasing)

Khi có hai hay nhiều tên cùng liên kết tới một ô nhớ ta gọi là sự lẩn tên. Chẳng hạn các biến con trỏ trong ngôn ngữ Pascal cùng trỏ đến một ô nhớ. Sự lẩn tên có thể làm giảm độ tin cậy do người lập trình không kiểm soát được giá trị được lưu trữ trong ô nhớ.

Chi phí

Chi phí của một ngôn ngữ cũng thường được quan tâm như là một tiêu chuẩn để đánh giá ngôn ngữ. Chi phí ở đây phải được hiểu là cả tiền bạc và thời gian. Chi phí này bao gồm:

- Chi phí đào tạo lập trình viên sử dụng ngôn ngữ. Chi phí này phụ thuộc vào sự giản dị của ngôn ngữ.
- Chi phí cài đặt chương trình. Chi phí này phụ thuộc vào tính dễ viết của ngôn ngữ.
- Chi phí dịch chương trình.
- Chi phí thực hiện chương trình.

- Chi phí bảo trì chương trình.
- Chi phí mua trình biên dịch

1.2. Thuật toán

Thuật ngữ thuật toán (Algorithm) là từ viết tắt của tên một nhà toán học ở thế kỷ IX: Abu Ja'fa Mohammed ibn Musa al-Khowarizmi.

Thuật toán là một dãy hữu hạn các thao tác được bố trí theo một trình tự xác định, được đề ra trước, nhằm giải quyết một bài toán nhất định.

Thao tác hay còn gọi là tác vụ, phép toán (Operation) hay lệnh (Command), chỉ thị (Instruction)... là một hành động cần được thực hiện bởi cơ chế thực hiện thuật toán.

Mỗi thao tác biến đổi bài toán từ một trạng thái trước (hay trạng thái nhập) sang trạng thái sau (hay trạng thái xuất). Thực tế mỗi thao tác thường sử dụng một số đối tượng trong trạng thái nhập (các đối tượng nhập) và sản sinh ra các đối tượng mới trong trạng thái xuất (các đối tượng xuất). Quan hệ giữa 2 trạng thái xuất và nhập cho thấy tác động của thao tác. Dãy các thao tác của thuật toán nối tiếp nhau nhằm biến đổi bài toán từ trạng thái ban đầu đến trạng thái kết quả.

* Các đặc trưng của thuật toán

Tính xác định: Các thao tác, các đối tượng, phương tiện trong thuật toán phải có ý nghĩa rõ ràng, không được gây nhầm lẫn. Nói cách khác, hai cơ chế hoạt động khác nhau cùng thực hiện một thuật toán, sử dụng các đối tượng, phương tiện nhập phải cho cùng một kết quả.

Tính dừng: Đòi hỏi thuật toán phải dừng và cho kết quả sau một số hữu hạn các bước.

Tính đúng của thuật toán: Thuật toán đúng là thuật toán cho kết quả thỏa mãn đặc tả thuật toán với mọi trường hợp của các đối tượng, phương tiện nhập.

Tính phổ dụng: Thuật toán để giải một lớp bài toán gồm nhiều bài cụ thể, lớp đó được xác định bởi đặc tả. Dĩ nhiên là có lớp bài toán chỉ gồm 1 bài. Thuật toán khi đó sẽ không cần sử dụng đối tượng, phương tiện nhập nào cả.

* Phương pháp biểu diễn

Thuật toán có thể diễn đạt dưới nhiều hình thức, chẳng hạn dưới dạng lưu đồ, dạng ngôn ngữ tự nhiên, dạng mã giả hoặc một ngôn ngữ lập trình nào khác.

Dạng ngôn ngữ tự nhiên: Thuật toán có thể trình bày dưới dạng ngôn ngữ tự nhiên theo trình tự các bước thực hiện trong thuật toán.

Dạng ngôn ngữ lập trình: Dùng cấu trúc lệnh, dữ liệu của một ngôn ngữ lập trình nào đó để mô tả.

Dạng mã giả: Thuật toán trình bày trong dạng văn bản bằng ngôn ngữ tự nhiên tuy dễ hiểu nhưng khó cài đặt. Dùng một ngôn ngữ lập trình nào đó để diễn tả thì phức tạp, khó hiểu. Thông thường thuật toán cũng được trao đổi dưới dạng văn bản - tuy không ràng buộc nhiều vào cú pháp xác định như các ngôn ngữ lập trình, nhưng cũng tuân theo một số quy ước ban đầu - ta gọi là dạng mã giả. Tùy theo việc định hướng cài đặt thuật toán theo ngôn ngữ lập trình nào ta diễn đạt thuật toán gần với ngôn ngữ ấy.

Dạng lưu đồ: Trong các phương pháp biểu diễn, chúng ta sẽ chủ yếu nghiên cứu phương pháp biểu diễn theo dạng này. Dạng lưu đồ dùng các hình vẽ (có quy ước) để diễn đạt

thuật toán. Lưu đồ cho hình ảnh trực quan và tổng thể của thuật toán, cho nên thường được sử dụng nhiều nhất.

Các ký hiệu sử dụng trong phương pháp biểu diễn thuật toán bằng lưu đồ:

STT	Ký hiệu	Giải thích
1		Bắt đầu và kết thúc chương trình
2		Điểm nối, đường đi (luồng xử lý)
3		Điều khiển lựa chọn
4		Thao tác nhập, xuất.
5	—	Thao tác xử lý hoặc tính toán.
6		Trả về giá trị (return)
7		<u>Điểm nối liên kết tiếp theo (sử dụng khi lưu đồ vượt quá trang)</u>

2. Quá trình xây dựng một phần mềm

Để thấy rõ vai trò của ngôn ngữ lập trình trong công nghệ phần mềm chúng ta hãy xét các giai đoạn chủ yếu để xây dựng một phần mềm. Các giai đoạn đó bao gồm:

- **Xác định:** Mục tiêu của giai đoạn xác định là để hiểu rõ các yêu cầu của khách hàng. Kết quả của giai đoạn này là mô hình thể giới thực được phản ánh thông qua một tài liệu đặc tả yêu cầu.

- **Phân tích:** Mục tiêu của giai đoạn này là xác định chính xác hệ thống sẽ làm những gì theo quan điểm của người sử dụng. Kết quả của giai đoạn phân tích là một tài liệu đặc tả chức năng mô tả hệ thống sẽ có những chức năng gì.

- **Thiết kế:** Mục tiêu của giai đoạn thiết kế là xác định chính xác hệ thống sẽ làm việc như thế nào. Kết quả của giai đoạn này là một tài liệu đặc tả thiết kế. Đây là một tài liệu kỹ thuật mà những người thực hiện sẽ căn cứ vào đó mà tạo ra phần mềm.

- **Cài đặt:** Là việc thực hiện cách giải quyết vấn đề đã được đề xuất bởi người thiết kế bằng một ngôn ngữ lập trình. Kết quả của giai đoạn này là một hệ chương trình máy tính.

- **Tích hợp và kiểm thử hệ thống:** Do các chuyên viên tin học thực hiện nhằm ghép nối các bộ phận của hệ thống và kiểm tra xem hệ thống có được thực hiện đúng theo thiết kế không.

- **Chấp nhận:** Do các chuyên viên tin học cùng với khách hàng tiến hành nhằm xác nhận hệ thống chương trình bảo đảm các yêu cầu của người sử dụng.

- **Vận hành khai thác:** Hệ thống được triển khai để sử dụng.

Ở trên chỉ trình bày một mô hình làm phần mềm, gọi là mô hình thác nước (water fall), ngoài ra còn có nhiều mô hình khác. Tuy nhiên trong tất cả các mô hình ấy đều phải có giai đoạn **cài đặt**. Trong đó ngôn ngữ lập trình đóng vai trò là một **công cụ** giúp con người thực hiện bước cài đặt này. Công cụ đó ngày càng được cải tiến hoàn thiện và có thể nói mọi tiến bộ trong tin học đều thể hiện ra trong ngôn ngữ lập trình. ngôn ngữ lập trình vừa là công cụ giúp các nhà tin học giải quyết các vấn đề thực tế nhưng đồng thời cũng là nơi mà những nghiên cứu mới nhất của tin học được đưa vào. Lĩnh vực này vừa mang tính truyền thống vừa mang tính hiện đại.

Câu hỏi cuối bài:

1. Ngôn ngữ lập trình là gì? Lợi ích của việc nghiên cứu ngôn ngữ lập trình?
 2. Trình bày các tiêu chuẩn đánh giá một ngôn ngữ lập trình tốt?
 3. Thuật toán là gì? Các cách biểu diễn thuật toán?
- Trình bày các giai đoạn xây dựng một phần mềm theo mô hình thác

TaiLieu.vn

Bài 2

NGÔN NGỮ LẬP TRÌNH C

Mục tiêu:

Trình bày được các khái niệm cơ bản như: tên, từ khóa, biến và các kiểu dữ liệu

Giải thích được cấu trúc của một chương trình C và mô tả được môi trường lập trình turbo C

Vận dụng được các loại biến, hằng, biểu thức cho từng chương trình cụ thể.

Thực hiện được việc chạy chương trình.

Rèn luyện kỹ năng làm việc và tư duy logic.

Nội dung chính:

1. Các khái niệm cơ bản

1.1. Từ khóa

Bộ chữ viết trong C

Ngôn ngữ C được xây dựng trên bộ ký tự sau:

- 26 chữ cái hoa: A B C .. Z
- 26 chữ cái thường: a b c .. z
- 10 chữ số: 0 1 2 .. 9
- Các ký hiệu toán học: + - * / = ()
- Ký tự gạch nối: _
- Các ký tự khác: . , : ; [] { } ! \ & % # \$...
- Dấu cách (space) dùng để tách các từ. Ví dụ chữ SINH VIEN có 9 ký tự, còn SINHVIEN chỉ có 8 ký tự.

Chú ý:

Khi viết chương trình, ta không được sử dụng bất kỳ ký tự nào khác ngoài các ký tự trên.

Từ khóa

Từ khóa là những từ được sử dụng để khai báo các kiểu dữ liệu, để viết các toán tử và các câu lệnh. Một số từ khóa của TURBO C được liệt kê như sau:

asm	break	case	cdecl
char	const	continue	default
do	double	else	enum
extern	Far	float	for
goto	Huge	if	int
interrupt	Long	near	pascal
register	return	short	signed
sizeof	static	struct	switch
typedef	union	unsigned	void
volatile	while		

Chú ý:

- Không được dùng các từ khoá để đặt tên cho các hằng, biến, mảng, hàm ...
- Từ khoá phải được viết bằng chữ thường, ví dụ: viết từ khoá khai báo kiểu nguyên là int chứ không phải là INT.

1.2. Tên

Khái niệm tên rất quan trọng trong quá trình lập trình, nó không những thể hiện rõ ý nghĩa trong chương trình mà còn dùng để xác định các đại lượng khác nhau khi thực hiện chương trình.

Tên thường được đặt cho hằng, biến, mảng, con trỏ, nhãn... Chiều dài tối đa của tên là 32 ký tự.

Tên biến hợp lệ là một chuỗi ký tự liên tục gồm: Ký tự chữ, số và dấu gạch dưới. Ký tự đầu của tên phải là chữ hoặc dấu gạch dưới. Khi đặt tên không được đặt trùng với các từ khóa.

Ví dụ 1:

Các tên đúng: delta, a_1, Num_ODD, Case

Các tên sai:

3a_1 (ký tự đầu là số)

num-odd (sử dụng dấu gạch ngang)

int (đặt tên trùng với từ khóa)

del ta (có khoảng trắng)

f(x) (có dấu ngoặc tròn)

Lưu ý: Trong C, tên phân biệt chữ hoa, chữ thường

Ví dụ 2: number khác Number

case khác Case

(case là từ khóa, do đó bạn đặt tên là Case vẫn đúng)

1.3. Kiểu dữ liệu

1.3.1. Kiểu số nguyên

Trong C cho phép sử dụng số nguyên kiểu int, số nguyên dài kiểu long và số nguyên không dấu kiểu unsigned. Kích cỡ và phạm vi biểu diễn của chúng được chỉ ra trong bảng dưới đây:

Kiểu	Phạm vi biểu diễn	Kích thước
int	-32768 đến 32767	2 byte
unsigned int	0 đến 65535	2 byte
long	-2147483648 đến 2147483647	4 byte
unsigned long	0 đến 4294967295	4 byte

1.3.2. Kiểu số thực

Trong C cho phép sử dụng số thực gồm 3 kiểu: float, double và long double. Kích cỡ và phạm vi biểu diễn của chúng được chỉ ra trong bảng dưới đây:

Kiểu	Phạm vi biểu diễn	Số chữ số có nghĩa	Kích thước
Float	3.4E-38 đến 3.4E+38	7 đến 8	4 byte
Double	1.7E-308 đến 1.7E+308	15 đến 16	8 byte
Long double	3.4E-4932 đến 1.1E4932	17 đến 18	10 byte

Giải thích:

Máy tính có thể lưu trữ được các số kiểu float có giá trị tuyệt đối từ 3.4E-38 đến 3.4E+38. Các số có giá trị tuyệt đối nhỏ hơn 3.4E-38 được xem bằng 0. Phạm vi biểu diễn của số double được hiểu theo nghĩa tương tự.

1.4. Các phép toán

1.4.1. Các phép toán số học

Phép toán	Ý nghĩa	Ví dụ
+	Phép cộng	a+b
-	Phép trừ	a-b
*	Phép nhân	a*b
/	Phép chia	a/b (Chia số nguyên sẽ chệch phần thập phân)
%	Phép lấy phần dư	a%b (Cho phần dư của phép chia a cho b)

Có phép toán một ngôi - ví dụ -(a+b) sẽ đảo giá trị của phép cộng (a+b). Ví dụ:

$$11/3=3$$

$$11\%3=2$$

$$-(2+6)=-8$$

Các phép toán + và - có cùng thứ tự ưu tiên, có thứ tự ưu tiên nhỏ hơn các phép *, /, % và cả ba phép này lại có thứ tự ưu tiên nhỏ hơn phép trừ một ngôi.

Các phép toán số học được thực hiện từ trái sang phải. Số ưu tiên và khả năng kết hợp của phép toán được chỉ ra trong một mục sau này.

1.4.2. Các phép toán quan hệ và logic

Phép toán quan hệ và logic cho ta giá trị đúng (1) hoặc giá trị sai (0). Nói cách khác, khi các điều kiện nêu ra là đúng thì ta nhận được giá trị 1, trái lại ta nhận giá trị 0.

Các phép toán quan hệ:

Phép toán	Ý nghĩa	Ví dụ
>	So sánh lớn hơn	$a > b$ $4 > 5$ có giá trị 0
>=	So sánh lớn hơn hoặc bằng	$a \geq b$ $6 \geq 2$ có giá trị 1
<	So sánh nhỏ hơn	$a < b$ $6 < 7$ có giá trị 1
<=	So sánh nhỏ hơn hoặc bằng	$a \leq b$ $8 \leq 5$ có giá trị 0
==	So sánh bằng nhau	$a == b$ $6 == 6$ có giá trị 1
!=	So sánh khác nhau	$a != b$ $9 != 9$ có giá trị 0

Bốn phép toán đầu có cùng số ưu tiên, hai phép sau có cùng số thứ tự ưu tiên nhưng thấp hơn số thứ tự của bốn phép đầu.

Các phép toán quan hệ có số thứ tự ưu tiên thấp hơn so với các phép toán số học, cho nên biểu thức: $i < n - 1$ được hiểu là $i < (n - 1)$.

Các phép toán logic

Trong C sử dụng ba phép toán logic:

- Phép phủ định một ngôi (!)