

LÝ THUYẾT NGÔN NGỮ LẬP TRÌNH

Mục tiêu của môn học

- Nắm được các nguyên lý (NL) của ngôn ngữ lập trình (NNLT)
- Tìm hiểu sự phát triển của NNLT
- Nắm được các kiểu dữ liệu và các cấu trúc điều khiển
- Hiểu được các phương thức (paradigm) lập trình (LT) như hướng đối tượng, hàm, logic và song song

Yêu cầu môn học

- Lập trình bằng: C, C++, C#, Java, python
- Sinh viên phải dự lớp đầy đủ
- Tham gia những lần kiểm tra tại lớp

Đánh giá môn học

- Dự lớp: 10%
- Kiểm tra: 30%
- Thi hết môn : 60%
- Question & discussion!!!

Nội dung môn học

- Giới thiệu ngôn ngữ lập trình
- Các kiểu dữ liệu
- Các cấu trúc điều khiển
- Chương trình con
- Ngôn ngữ lập trình hướng đối tượng
- Ngôn ngữ lập trình hàm
- Ngôn ngữ lập trình logic
- Ngôn ngữ lập trình song song

Tài liệu tham khảo

- *Giáo trình Ngôn ngữ lập trình*, Nguyễn Văn Linh, 2003,
- *Ngôn ngữ lập trình: các nguyên lý và mô hình*, Cao Hoàng Trụ, ĐH QG HCM, 2004
- *Concept of Programming Languages*, Robert W. Sebesta, Sixth Edition, 2004

Chương 1: Giới thiệu

Nội dung

- Giới thiệu
- Lịch sử phát triển của máy tính
- Lịch sử phát triển của NNLT
- Vai trò của NNLT
- Lợi ích của việc học NNLT
- Các tiêu chuẩn đánh giá NNLT

Giới thiệu

■ Lập trình máy tính

- ❑ Lập trình máy tính là một *cách nghĩ* về các vấn đề
- ❑ Biên dịch thành những chỉ thị để máy tính thực thi
- ❑ Sản phẩm là các chương trình

■ Chương trình

- ❑ Một tập hợp riêng biệt các hoạt động có thứ tự cho một máy tính thực hiện
- ❑ Chương trình bao gồm một chuỗi các chỉ thị lần lượt mà máy tính phải tuân theo

Giới thiệu (tt)

■ Lập trình

- ❑ Thao tác nhằm kiến tạo nên các chương trình máy tính thông qua các ngôn ngữ lập trình
- ❑ Còn gọi là quá trình mã hoá thông tin tự nhiên thành ngôn ngữ máy
- ❑ Quá trình này còn ãược gọi là là "viết mã"

■ Mã nguồn chương trình

- ❑ Những chỉ thị góp phần tạo nên chương trình được gọi là mã nguồn của chương trình
- ❑ Mỗi NNLT cũng chính là một chương trình, được dùng để tạo nên các chương trình khác

Ngôn ngữ lập trình

- NNLT là ngôn ngữ dùng để viết các chương trình cho máy tính
- NNLT có từ vựng, cú pháp và ngữ nghĩa (trình biên dịch)
- Ba loại NNLT:
 - Ngôn ngữ máy (nhị phân)
 - Hợp ngữ (NN cấp thấp)
 - Ngôn ngữ cấp cao (Pascal, C, C++, Java,...)

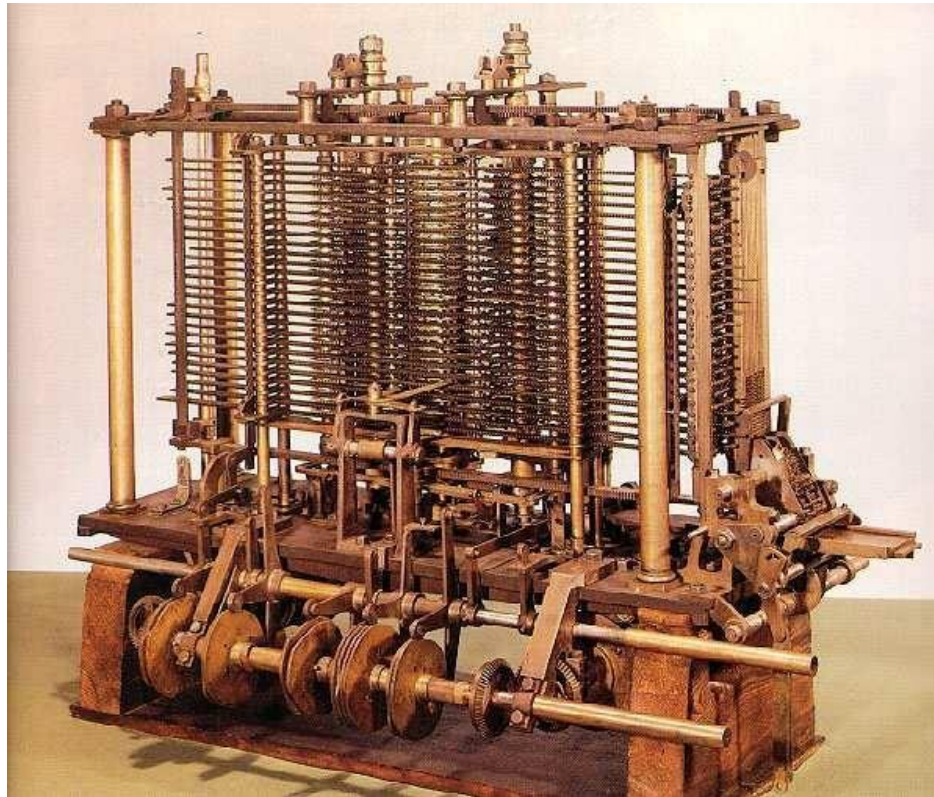
Đặc tính của ngôn ngữ máy

- Mã lệnh nhị phân để thực thi tác vụ
- Dòng điều khiển (control flow) dựa trên labels và rẽ nhánh có điều kiện (conditional branches)
- Tường minh về địa chỉ (location), e.g. registers, cho các biến và kết quả trung gian
- Tường minh về quản lý bộ nhớ (e.g stack)

Đặc tính của NNLT

- Diễn đạt ở mức phức tạp (toán học, logic)
- Có cấu trúc điều khiển (Loops, Selection, Cases)
- Sự kết hợp kiểu cơ bản (array, record, etc)
- Khai báo và kiểm tra kiểu
- Hỗ trợ nhiều cách lưu trữ (global/local/heap)
- Procedure/function
- Kiểu dữ liệu trừ tượng, module, object, etc

Lịch sử phát triển của máy tính

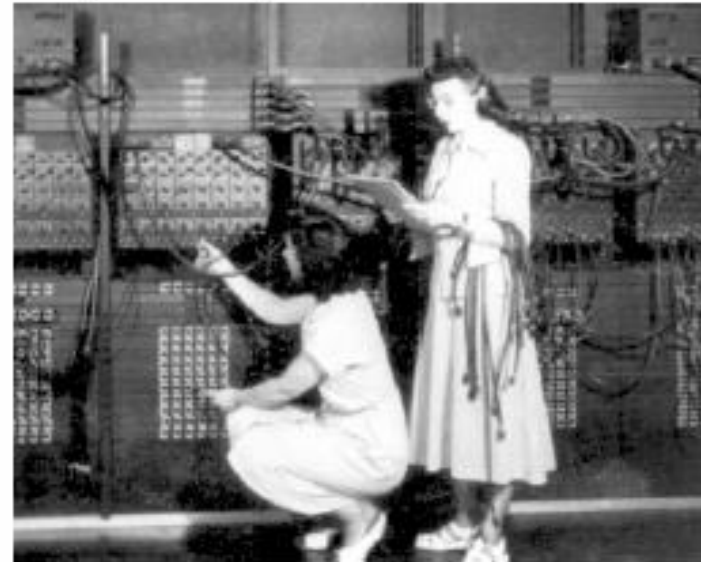
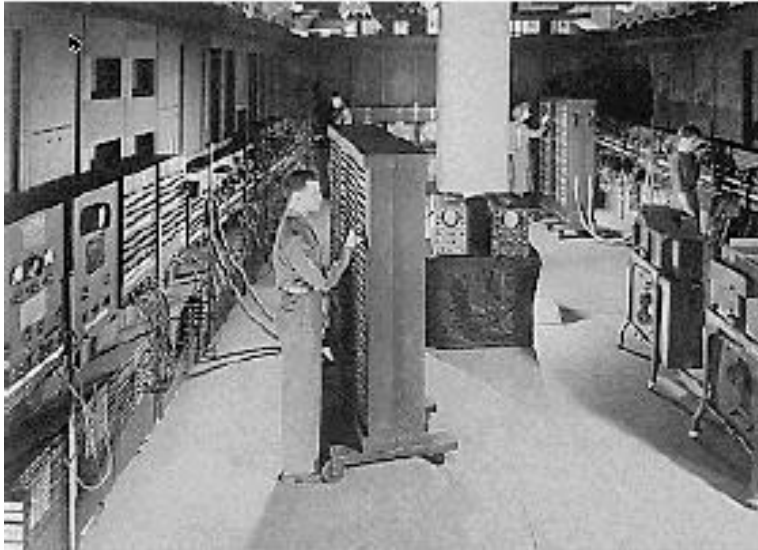


Máy tính của Babbage

- Năm 1837
 - Đặc điểm
 - Store (memory)
 - Mill (CPU)
 - Looping
 - Conditional
- => Không thành công

Lịch sử phát triển của máy tính (tt)

- ENIAC: plug and play circa 1945



Lịch sử phát triển của NNLT

- Sự phát triển của NNLT gắn liền với khoa học máy tính
 - ❑ 1920's: computer ~ «person»
 - ❑ 1936: Church's Lambda Calculus (= PL!)
 - ❑ 1937: Shannon's digital circuit design
 - ❑ 1940's: first digital computers
 - ❑ 1950's: FORTRAN (= PL!)
 - ❑ 1958: LISP (LISt Processing language)
 - ❑ 1960's: Unix
 - ❑ 1972: C Programming Language
 - ❑ 1981: TCP/IP
 - ❑ 1985: Microsoft Windows

Bạn biết được những NNLT nào?

- FORTRAN, COBOL, RPG, (Visual) BASIC, ALGOL-60, ALGOL-68, PL/I,
- C, C++, Java, C#, **Pascal**, Modula, Oberon, Lisp, Scheme, ML, Haskell,
- Ada, Prolog, Goedel, Curry, Snobol, ICON,
- . . .

Tại sao có rất nhiều NNLT

- Sự đa dạng về đặc điểm
 - e.g. **goto** vs **if-then, switch-case, while-do**
 - Quản lý bộ nhớ: programmer (C,C++) vs. language (Java through garbage collection)
- Sự ưa thích của lập trình viên
 - procedure vs. function
- Sự phát triển của phần cứng
- Mục tiêu đặc biệt của từng NNLT

Lĩnh vực ứng dụng của NNLT

- Nghiên cứu khoa học: Fortran, C, Matlab
- Lập trình hệ thống: C, C++
- Tính toán trong kinh doanh: Java, C#
- Trí tuệ nhân tạo: LISP
- Xử lý văn bản (Perl, Python)
- Mục đích đặc biệt cho NNLT (make, sh-shell)

Mô hình của NNLT

- Lập trình mệnh lệnh (Imperative)
 - Fortran, C, Pascal, etc
- Hàm
 - Lisp, scheme
- Hướng đối tượng
 - C++, Java, C#, etc
- Logic
 - Prolog
- Nhu cầu đặc biệt
 - Tex, Shell, make, HTML

Vai trò của NNLT

- Các giai đoạn để xây dựng 1 phần mềm
 - Xác định mục đích
 - Phân tích
 - Thiết kế
 - Cài đặt (coding)
 - Tích hợp và kiểm thử
 - Vận hành và khai thác

Lợi ích của việc nghiên cứu NNLT

- Cho phép lựa chọn một NNLT phù hợp với dự án thực tế
- Sử dụng một cách có hiệu quả các công cụ của ngôn ngữ lập trình
- Hiểu sâu sắc hơn ý nghĩa của các cài đặt bên trong NNLT
- Dễ dàng để học một ngôn ngữ lập trình mới
- Tăng vốn kinh nghiệm cho lập trình viên
- Thiết kế một NNLT mới (ngôn ngữ đơn giản)

Các tiêu chí đánh giá NNLT

- Dễ đọc: là sự dễ dàng đọc hiểu một chương trình
 - Sự giản dị: ngôn ngữ có ít thành phần cơ sở
 - VD `count = count + 1`, `count += 1`, `count++`, `++count`
 - Cấu trúc điều khiển: sáng sủa, dễ đọc, dễ hiểu
 - Kiểu và cấu trúc dữ liệu: góp phần nâng giá một ngôn ngữ có dễ ão hay không. VD **TRUE**, **FALSE**
 - Cú pháp: ảnh hưởng đến tính dễ đọc hiểu, VD độ dài tối đa của danh biểu, từ khóa

Các tiêu chí đánh giá NNLT

- Ví dụ: So sánh hai đoạn chương trình in ra màn hình 10 số tự nhiên đầu tiên

Viết bằng BASIC

```
10      i=1;  
  
20      IF i>10 THEN GOTO 60;  
  
30      PRINT i;  
  
40      i=i+1;  
  
50      GOTO 20;  
  
60      PRINT "In xong";
```

Viet bang C

```
...  
for (i =1; i<=10; i++ )  
printf ("%d",i);  
...
```

Các tiêu chí đánh giá NNLT (tt)

- Dễ viết: khả năng sử dụng ngôn ngữ để viết một chương trình cho một vấn đề nào đó một cách dễ dàng hay không
 - Tính diễn đạt cao: cung cấp cho người lập trình những công cụ để có thể nghĩ sao thì viết chương trình như vậy
 - 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
 - Hỗ trợ trừu tượng: khả năng định nghĩa và sử dụng các cấu trúc hoặc các phép toán phức tạp cho phép bỏ qua các chi tiết

Các tiêu chí ñánh giá NNLT (tt)

- Độ tin cậy: khả năng của ngôn ngữ hỗ trợ người lập trình tạo ra các chương trình đúng đắn
 - Kiểm tra kiểu
 - Xử lý ngoại lệ: phát hiện và sửa chữa lỗi và tiếp tục thực hiện mà không phải dừng chương trình
 - Sự lấm tên (Aliasing): Có hai hay nhiều tên cùng liên kết tới một ô nhớ. Sự lấm tên có thể làm giảm ñộ tin cậy

Các tiêu chí ñánh giá NNLT (tt)

- Chi phí (time & money): một tiêu chuẩn để đánh giá ngôn ngữ, được hiểu là cả tiền bạc và thời gian
 - ❑ Chi phí ñào tạo chương trình viên: phụ thuộc vào sự giản dị của ngôn ngữ
 - ❑ Chi chí lập trình: phụ thuộc vào tính dễ viết của ngôn ngữ
 - ❑ Chi phí thực hiện chương trình
 - ❑ Chi phí cài ñặt chương trình
 - ❑ Chi phí bảo trì chương trình

Phân loại ngôn ngữ lập trình

- Theo tiến trình lịch sử phát triển
 - Ngôn ngữ máy, Hợp ngữ, Ngôn ngữ cấp cao, Ngôn ngữ cấp rất cao
- Theo mục tiêu của ngôn ngữ
 - đa dụng, nhúng, lập trình web, ...
- Theo cách thức
 - Mã máy, thông dịch, lai
- Theo phương pháp
 - Lập trình tuyến tính, lập trình có cấu trúc, lập trình hướng đối tượng, lập trình trực quan, lập trình logic, ...

Phân loại ngôn ngữ lập trình (tt)

- Phân loại NNLT theo tiến trình lịch sử phát triển
 - Ngôn ngữ máy (machine language)
 - Hợp ngữ (assembly language)
 - Ngôn ngữ cấp cao (HLL - High level language)
 - Ngôn ngữ cấp rất cao (VHLL – Very High Level Language): HDL –High Description Language

Phân loại ngôn ngữ lập trình (tt)

- Phân loại theo mục tiêu của ngôn ngữ
 - ❑ đa dụng: như C, perl, python, C++, Java, ...
 - ❑ Nhúng: Như Ch, Ada, ...
 - ❑ Lập trình web: PHP, Java Script, Cold Fusion, ...
 - ❑ Riêng: Như AutoLisp, Mathematica, ...

Phân loại ngôn ngữ lập trình (tt)

- Phân loại theo cách thức
 - Mã máy: Thường gặp như C, C++, ...
 - Thông dịch: Giữ nguyên dưới dạng file văn bản, ãuợc thông dịch sang mã máy khi chạy như Perl, PHP, ...
 - Lai: Ví dụ Java biên dịch ra Java Byte code, khi chạy mới thông dịch sang mã máy, như vậy tạo ra khả năng "viết một nơi chạy mọi nơi" của Java. Một ví dụ khác là .NET

Phân loại ngôn ngữ lập trình (tt)

- Phân loại theo phương pháp

- Hướng đối tượng:

- Xem mọi thứ là đối tượng, quá trình giải quyết công việc là quá trình các đối tượng tương tác với nhau
 - Ví dụ: C++, Java, Ada, ...

- Lập trình logic:

- Cho chương trình biết sẽ làm cái gì chứ không phải làm như thế nào
 - Toàn bộ các câu lệnh chỉ đơn giản là đúng hoặc sai
 - Ví dụ: ngôn ngữ Prolog

Phân loại ngôn ngữ lập trình (tt)

- Phân loại theo phương pháp (tt)
 - Lập trình hàm:
 - Toàn bộ các lệnh và mọi thứ đều được đưa về hàm.
 - Cũng là ngôn ngữ khai báo (declarative)
 - Ví dụ: ngôn ngữ LISP
 - Lập trình trực quan:
 - Hỗ trợ tạo mã nguồn một cách trực quan.
 - Ví dụ: ngôn ngữ Visual Basic
 - Ngôn ngữ đánh dấu:
 - Ví dụ: ngôn ngữ HTML

Một số NNLT thông dụng

- FORTRAN (FORmula TRANslation)
 - ❑ được thiết kế bởi John Backus năm 1957
 - ❑ Ngôn ngữ cấp cao đầu tiên
 - ❑ Thích hợp cho các ứng dụng trong khoa học và kỹ thuật
 - ❑ Cấu trúc ngữ pháp khá đơn giản do việc nhập chương trình chủ yếu thông qua các phiếu bấm lỗ

Một số NNLT thông dụng (tt)

- LISP (LISt Processing)
 - ❑ được thiết kế bởi John McCarthy năm 1959
 - ❑ Xử lý trên các ký hiệu
 - ❑ Ngôn ngữ *lập trình hàm* với việc sử dụng nhiều các phép gọi đệ quy
 - ❑ Một trong những ngôn ngữ cơ bản cho ngành Trí tuệ Nhân tạo (Artificial Intelligence)

Một số NNLT thông dụng (tt)

- COBOL (COmmon Business Oriented Language)
 - được thiết kế bởi Grace Hopper & CODASYL năm 1959
 - Ngôn ngữ đầu tiên dùng cho các ứng dụng thương mại
 - Cấu trúc ngữ pháp giống tiếng Anh giúp COBOL dễ học hiểu nhưng khá dài dòng

Một số NNLT thông dụng (tt)

- ALGOL 60 (ALGO^rithmic Language)
 - Peter Naur & European Committee, 1960
 - Ngôn ngữ ãầu tiên sử dụng *cấu trúc khối*
 - Sử dụng chủ yếu cho toán và khoa học
 - Là ngôn ngữ tính toán khoa học chuẩn cho ãều ãầu những năm 1970

Một số NNLT thông dụng (tt)

■ Pascal

- ❑ Nicholas Wirth, 1970
- ❑ Ngôn ngữ chuẩn cho *lập trình cấu trúc*
- ❑ Là ngôn ngữ chính cho ngành khoa học máy tính trong thập niên 1970
- ❑ Dùng ngôn ngữ sát với ngôn ngữ tự nhiên nên nó thân thiện với người lập trình
- ❑ Gọn, dễ nhớ, khả năng truy cập cấp thấp, và các cấu trúc dữ liệu ãa dạng

Một số NNLT thông dụng (tt)

■ C

- ❑ Dennis Ritchie, 1972
- ❑ đến nay, ANSI ban hành chuẩn về C
- ❑ Là ngôn ngữ cấp trung vì nó cho phép thao tác trên các bit, byte, và địa chỉ
- ❑ Sử dụng cho các ứng dụng như hệ điều hành, phát triển ngôn ngữ máy tính khác hay xử lý văn bản
- ❑ Là ngôn ngữ cấp tương đối thấp để lập trình hiệu quả nhưng có các ưu điểm của ngôn ngữ cấp cao
- ❑ Do vậy, C tỏ ra thích hợp với lập trình hệ thống

Một số NNLT thông dụng (tt)

- PROLOG (PROgramming in LOGic)
 - Alain Colmerauer & Philippe Roussel, 1972
 - Ngôn ngữ mô tả (declarative language) để đặc tả các vấn đề ở dạng
 - Lập trình luận lý (logic programming): một chương trình bao gồm các sự kiện và qui luật
 - được xem như nền tảng của việc phát triển phần mềm cho dự án máy tính thế hệ thứ năm của Nhật

Một số NNLT thông dụng (tt)

■ C++

- ❑ Bjarne Stroustrup, 1986
- ❑ Mở rộng ngôn ngữ C với lập trình hướng đối tượng (object-oriented) nhưng vẫn duy trì lập trình cấu trúc
- ❑ Nó mang đầy đủ các đặc tính của C
- ❑ Là ngôn ngữ lập trình hướng nôi tượng
- ❑ Nó có tính kế thừa, tính đóng gói và tính đa hình
- ❑ Dựa vào tính kế thừa và có nhiều thư viện hỗ trợ mạnh được phát triển nên ñược sử dụng rộng rãi trong thập niên 1990

Một số NNLT thông dụng (tt)

■ Java

- ❑ Sun Microsystems, 1995, cú pháp giống C++
- ❑ Kế thừa nhưng tinh giản hơn các tính năng của C++
- ❑ JAVA là ngôn ngữ hướng đối tượng hoàn toàn
- ❑ Có nhiệm vụ viết phần mềm hệ thống nhúng vào các sản phẩm nhện tử
- ❑ Đơn giản: JAVA đã được lược bỏ các tính năng khó nhất của C++ như con trỏ, không hỗ trợ chồng toán tử
- ❑ Cho phép xây dựng ứng dụng mạng, Web với độ an toàn cao
- ❑ độc lập với platform