Logic lập trình

Phần mềm:

Ngôn ngữ lập trình

Môi trường thực thi

Quy tắc/tưd khoá

Thư viện

Giải quyết vấn đề

Hiểu rõ vấn đề

Thu thập thông tin

Xử lý thông tin

Đi đến giải pháp

Mã giả pseudocode viết ra để mô tả chương trình

Lưu đồ flowcharts

Người viết phải hiểu, người đọc cũng phải hiểu logic, ký hiệu.

# 01-8-2025

Biến là một kí hiệu do lập trình viên đặt tên theo quy chuẩn

Kí hiệu này đại diện/quản lý 1 giá trị sử dụng trong ct, có thể thay đổi giá trị

Lưu dữ liệu để sử dụng trong ct, thay đổi được theo logic ct

Làm cho ct có giá trị sử dụng.

Tên biến:

không dấu,

không dấu cách,

không kí tự đặc biệt,

không trùng từ khoá của ngôn ngữ

không bắt đầu bằng số

phân biệt chữ hoa chữ thường

kiểu ký tự

int: 16 bits (2 bytes) %d

float: 32 bits (4 bytes) %f

double: 64 bits (8 bytes) %lf

char: 8 bits (1 byte) %c

# 04-8-2025

Github.com

Branch:

Gitlab.com, bitbucket

Expression:

Sự kết hợp giữa toán tử (các kí hiệu toán học) và toán hạng (các giá trị)

Dấu bằng là một toán tử gán giá trị

Biểu thức không sử dụng dấu bằng để tính toán vì nó gán giá trị của biểu thức vào biến.

Biến = biểu thức

**Số học:**

+, -, \*, /, %

Kết quả là giá trị dạng số học

So sánh:

Kết quả trả về là đúng hoặc sai, có hoặc không

* **AND (&&)** chỉ đúng khi cả hai đều đúng
* **OR (||)** chỉ cần 1 đúng thì sẽ đúng
* **NOT (!)** đảo ngược giá trị

Mức độ ưu tiên: NOT 🡪 AND 🡪 OR

Đ && S || Đ || !Đ -🡪 Đ && S || Đ || S 🡪 S || Đ || Đ (từ trái sang phải)

* Ưu tiên số học trước 🡪 so sánh 🡪 logic (a+b>c && a+c>b && b+c>a)

**++ -> x++ ⬄ x = x + 1**

**‘-- -> x-- ⬄ x=x-1**

**+= -> x+=2 ⬄ x = x+2**

**-= -> x-=2 ⬄ x = x-2**

**\*= -> x\*=2 ⬄ x = x\*2**

**/= -> x/=2 ⬄ x = x/2**

Convert data: nguyên lý xuôi

Ô nhớ bé chuyển thành ô nhớ to

Cast: ép kiểu

Ô nhớ bé 🡪 to

Local variable:

Biến địa phương lúc nào cần thì khai báo, chỉ tồn tại trong 1 block code.

# 08 - 8 -2025

# LOOP

Tác dụng của vòng lặp là như thế nào?

Nếu không dung vòng lặp có giải quyết được vấn đề hay không? Tại sao? (có thực sự cần vòng lặp)

Tài khoản 4000$

Lãi suất 8%/năm (lãi kép)

Cần bao nhiêu năm để có 1M $?

1 – xác định xem có cần vòng lặp hay không?

2 – xác định điều kiện chạy vòng lặp

3 – logic trong vòng lặp

4 – kết thúc vòng lặp cần sử dụng gì từ vòng lặp đã xong

Nhập 1 số nguyên n, tìm số thứ n trong dãy số Fibonacci

0 1 1 2 3 5

x1 x2 x3

x1 di chuyen den x2

x2 di chuyen x2

x3 moi = x1 moi + x2 moi

# 11-8-205

break;

dừng vòng lặp

continue;

bỏ qua nếu gặp điều kiện

# 18 – 8 -2025

Search and Sort

Sắp xếp 5 người:

Cần 5 ô nhớ để chứ danh sách đã sắp xếp.

Tìm số bé nhất (lớn nhất) x 4 lần

=🡺 mảng lớn thì khó, mất thời gian

Insert sort:

J để cố định số

I để so sánh

Temp

Tháp chi phí trong thiết kế phần mềm

Bubble sort tính ổn định cao, cho các mảng hỗn loạn, tính được thời gian.

Insert sort tốc độ ko ổn định, cần mảng đã tương đối sắp xếp. Nếu dữ liệu đã được sắp xếp thì insert sort là tối ưu khi có dữ liệu mới them vào.

Tìm kiếm

Linear search:

Tạo trạng thái để kiểm tra

Càng nhiều thì thời gian tìm kiếm càng nhiều

Binary search

Mảng đã được sắp xếp

So sánh với số chính giữa, nếu nhỏ hơn thì bỏ bên phải