



# Tổng quan

Môn học: Cơ sở lập trình [*Buổi 1*]

GV: Nguyễn Mai Huy

# I – Thuật toán – Giải thuật

*Nguyễn Mai Huy - [nmaihuy@bodua.com](mailto:nmaihuy@bodua.com)*

# Giới thiệu

*Thuật toán là một tập các quy tắc (hay quy trình cụ thể) để giải quyết một vấn đề nào đó với số lần thực hiện là hữu hạn, dựa trên các dữ liệu đầu vào được cung cấp cho quá trình xử lý của chương trình*

*VD: Thuật toán giải phương trình bậc nhất  $aX + b = c$   
(Với  $a, b, c$  là các số thực)*

*1 – Nếu  $a = 0$*

*\*  $b = c$  thì phương trình có vô số nghiệm (Nghiệm bất kỳ)*

*\*  $b \neq c$  thì phương trình vô nghiệm*

*2 – Nếu  $a \neq 0$*

*Phương trình có 1 nghiệm duy nhất  $x = (c - b) / a$*

# Đặc tính của thuật toán

- ❖ Chính xác, Đúng
- ❖ Tổng quát (*Luôn đúng dẫn trong nhiều tình huống khác nhau – Còn gọi là tính phổ dụng*)
- ❖ Rõ ràng
- ❖ Hữu hạn (*Số lần thực hiện các bước là xác định, có tính dừng*)

# Các phương pháp biểu diễn

- ❖ Biểu diễn bằng ngôn ngữ tự nhiên  
(*Native language*)
- ❖ Biểu diễn bằng mã giả  
(*Pseudo code*)
- ❖ Biểu diễn bằng lưu đồ  
(*Flow chart*)

# Minh họa bằng ngôn ngữ tự nhiên

**Khi biểu diễn thuật toán theo ngôn ngữ tự nhiên, người ta sử dụng ngôn ngữ thường ngày để liệt kê các bước của thuật toán**

*1 – Nhập các giá trị  $a$ ,  $b$ ,  $c$*

*2 – Nếu  $a = 0$*

*Nếu  $b = c$  thì thông báo phương trình có vô số nghiệm*

*nếu  $b \neq c$  thì thông báo phương trình vô nghiệm*

*3 – Nếu  $a \neq 0$*

*Thông báo: Phương trình có 1 nghiệm duy nhất*

$$x = (c - b) / a$$

# Minh họa bằng mã giả

Khi thể hiện thuật toán bằng mã giả, người ta thường *vay mượn* các cú pháp của một ngôn ngữ lập trình nào đó để thể hiện thuật toán. Việc dùng mã giả vừa tận dụng được các khái niệm trong ngôn ngữ lập trình, vừa giúp người cài đặt dễ dàng nắm bắt nội dung thuật toán

*Nhập  $a, b, c$*

*If  $a = 0$  then*

*if  $b = c$  then*

*write('có vô số nghiệm');*

*if  $b \neq c$  then*

*write('phương trình vô nghiệm');*

*Else*

*begin*

*$x := (c - b) / a;$*

*write('Phương trình có 1 nghiệm duy nhất  $x = , x$ );*

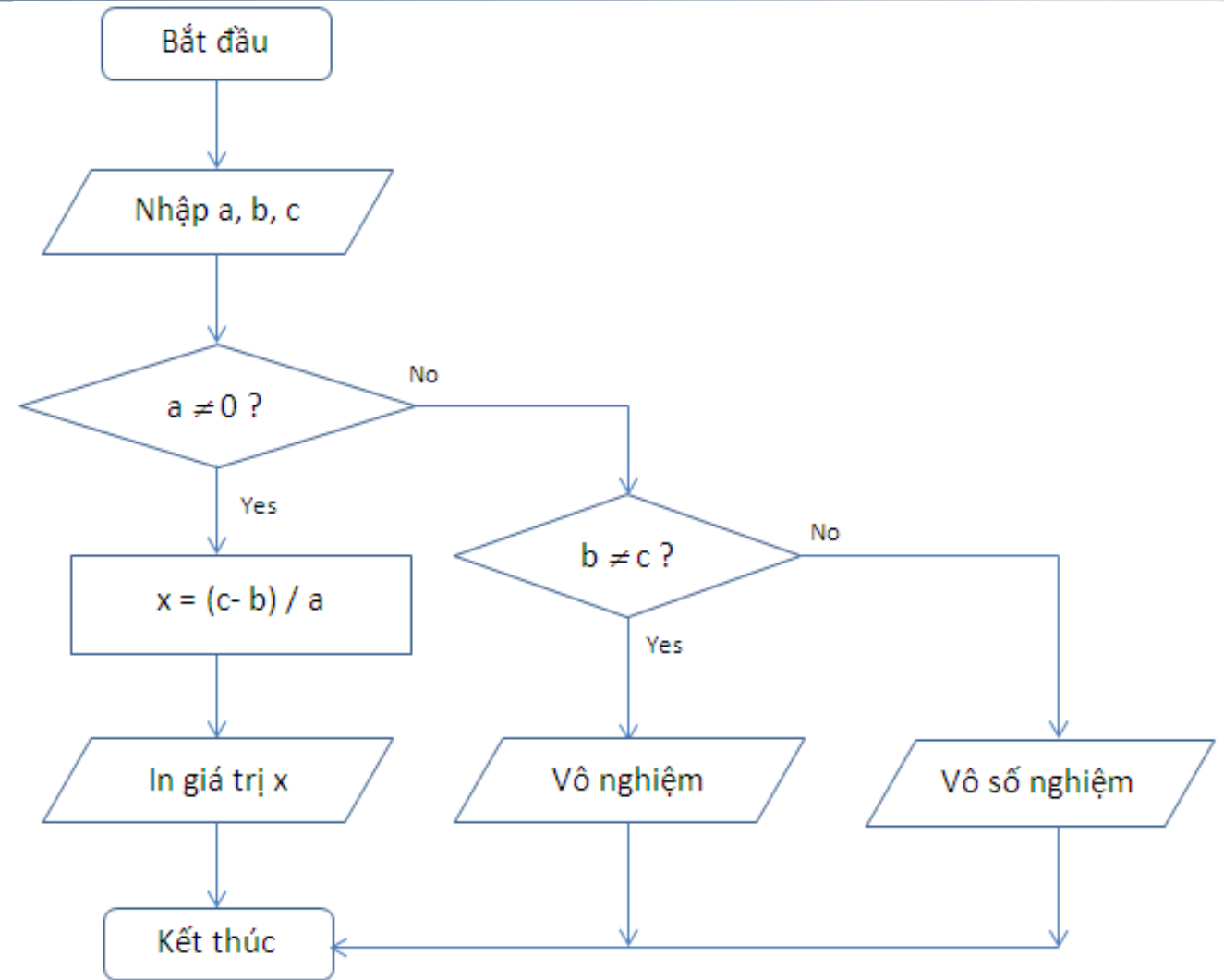
*end*

*Nguyễn Mai Huy - [nmaihuy@boduca.com](mailto:nmaihuy@boduca.com)*



# Minh họa bằng lưu đồ

Lưu đồ hay sơ đồ khối là một công cụ trực quan để diễn đạt các thuật toán. Biểu diễn thuật toán bằng lưu đồ sẽ giúp người đọc theo dõi được sự phân cấp các trường hợp và quá trình xử lý của thuật toán



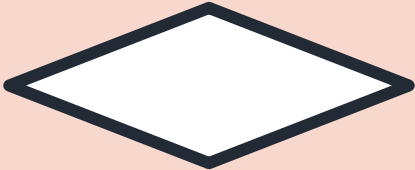
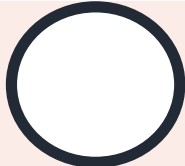
Nguyễn Mai Huy - [nmaihuy@bodua.com](mailto:nmaihuy@bodua.com)



# Biểu diễn thuật toán bằng lưu đồ

Lưu đồ thuật toán là công cụ dùng để biểu diễn thuật toán của chương trình, việc mô tả quá trình nhập (*input*), xuất dữ liệu (*output*) và các luồng xử lý (*processing*) đều được thể hiện dựa trên các ký hiệu hình học.

# Ký hiệu sử dụng trên lưu đồ

Ký hiệu - Biểu tượng	Ý nghĩa sử dụng trong lưu đồ
	Biểu thị bắt đầu hoặc kết thúc chương trình
	Biểu thị hướng xử lý trong một quá trình / chương trình
	Biểu thị thông tin vào hoặc thông tin ra ( <i>Nhập hoặc xuất dữ liệu</i> ) trong chương trình
	Biểu thị một hoạt động ( <i>Hay một quá trình</i> ) trong thuật toán
	Biểu thị một quyết định khi đứng trước một vấn đề logic cần phải lựa chọn ( <i>hoặc phân nhánh xử lý trong chương trình</i> ).
	Điểm nối trên lưu đồ ( <i>Sử dụng khi lưu đồ có kích thước lớn, phức tạp</i> )

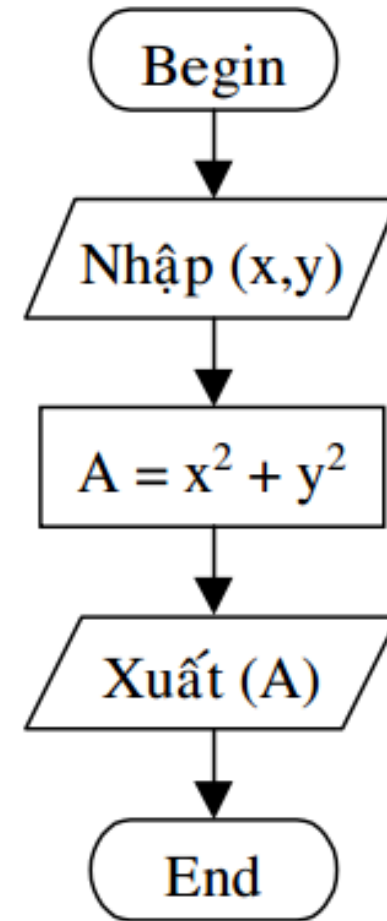
# Các hình thức biểu diễn

- ✓ Biểu diễn tuần tự
- ✓ Biểu diễn phân nhánh  
(*Có lựa chọn hướng xử lý*)
- ✓ Biểu diễn chu trình  
(*Cấu trúc lặp*)

# Cấu trúc tuần tự

## Yêu cầu:

- Nhập 2 số x, y. Sau đó tính tổng bình phương của 2 số và in kết quả ra màn hình

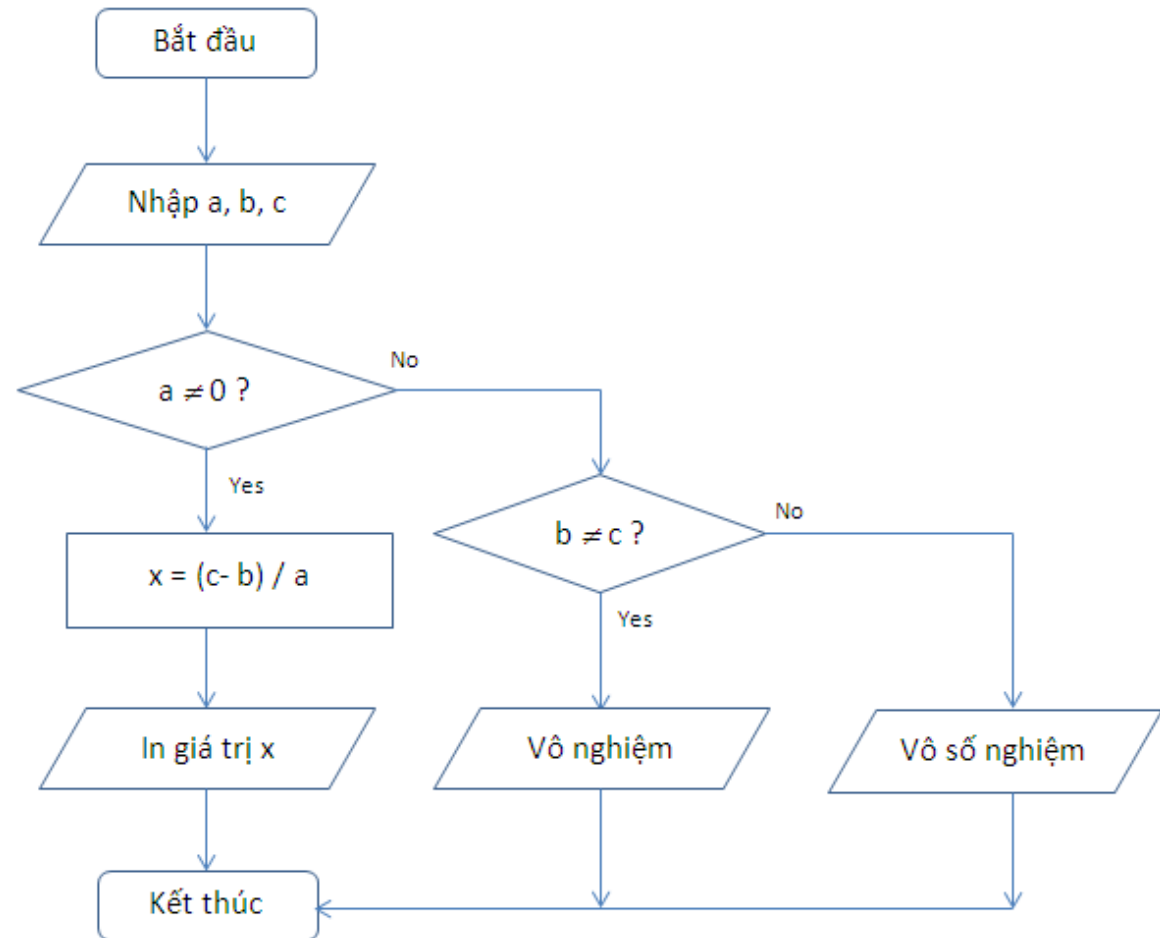


# Cấu trúc phân nhánh

## Yêu cầu:

- Giải phương trình

$$P(x): aX + b = c$$

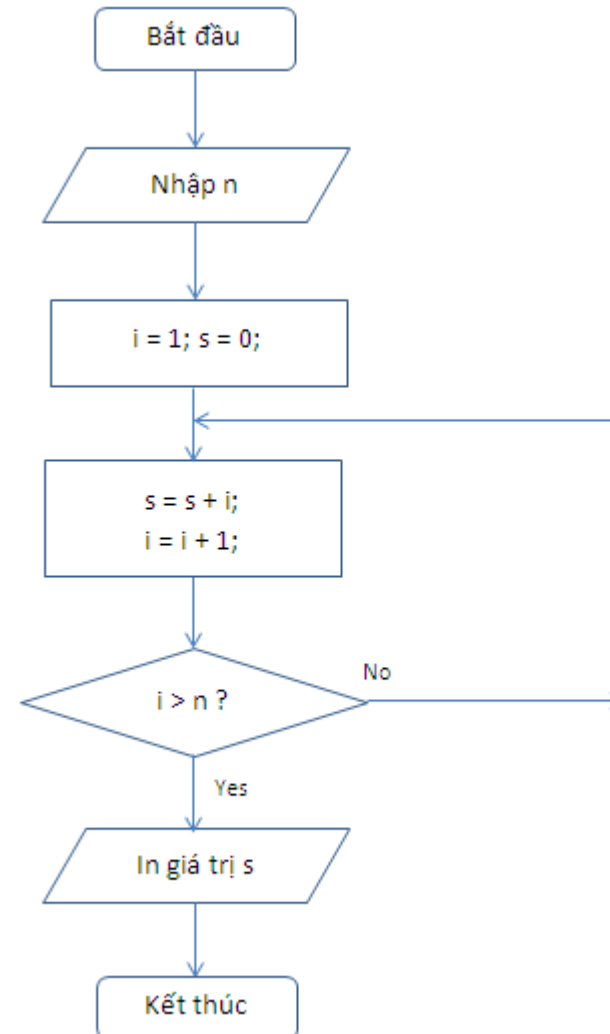


Nguyễn Mai Huy - [nmaihuy@boduca.com](mailto:nmaihuy@boduca.com)

# Lặp, xử lý theo chu trình

## Yêu cầu:

- Tính tổng các số tự nhiên từ 1 đến  $n$   
( $n$  được nhập từ bàn phím)



# Ngôn ngữ lập trình

*Nguyễn Mai Huy - [nmaihuy@bodua.com](mailto:nmaihuy@bodua.com)*



# Ngôn ngữ lập trình

**Ngôn ngữ lập trình** (*Programming language*) là một dạng ngôn ngữ được thiết kế và chuẩn hóa (*So với ngôn ngữ tự nhiên*) để truyền các chỉ thị cho máy tính (*hoặc các thiết bị khác có bộ xử lý: Smartphone, Tablet, Smart TV, ...*).

Ngôn ngữ lập trình thường được dùng để tạo ra các chương trình nhằm phục vụ cho việc điều khiển máy tính hoặc mô tả các thuật toán xử lý của chương trình máy tính.

Ngôn ngữ lập trình thường được chia làm 3 dạng

- ❖ **Ngôn ngữ máy** (*Machine language*)
- ❖ **Hợp ngữ** (*Assembly language*)
- ❖ **Ngôn ngữ lập trình cấp cao** (*Higher-Level language*)

# Chương trình dịch

Do máy tính (*Và các thiết bị cho phép lập trình*) chỉ có thể hiểu được ngôn ngữ máy, cho nên một chương trình sau khi đã được lập trình (*Viết bằng ngôn ngữ cấp cao, hợp ngữ*) cần phải chuyển đổi thành ngôn ngữ máy thì mới có thể thi hành được.

Những công cụ làm nhiệm vụ chuyển đổi cho mục đích này thường được gọi là ***chương trình dịch***.

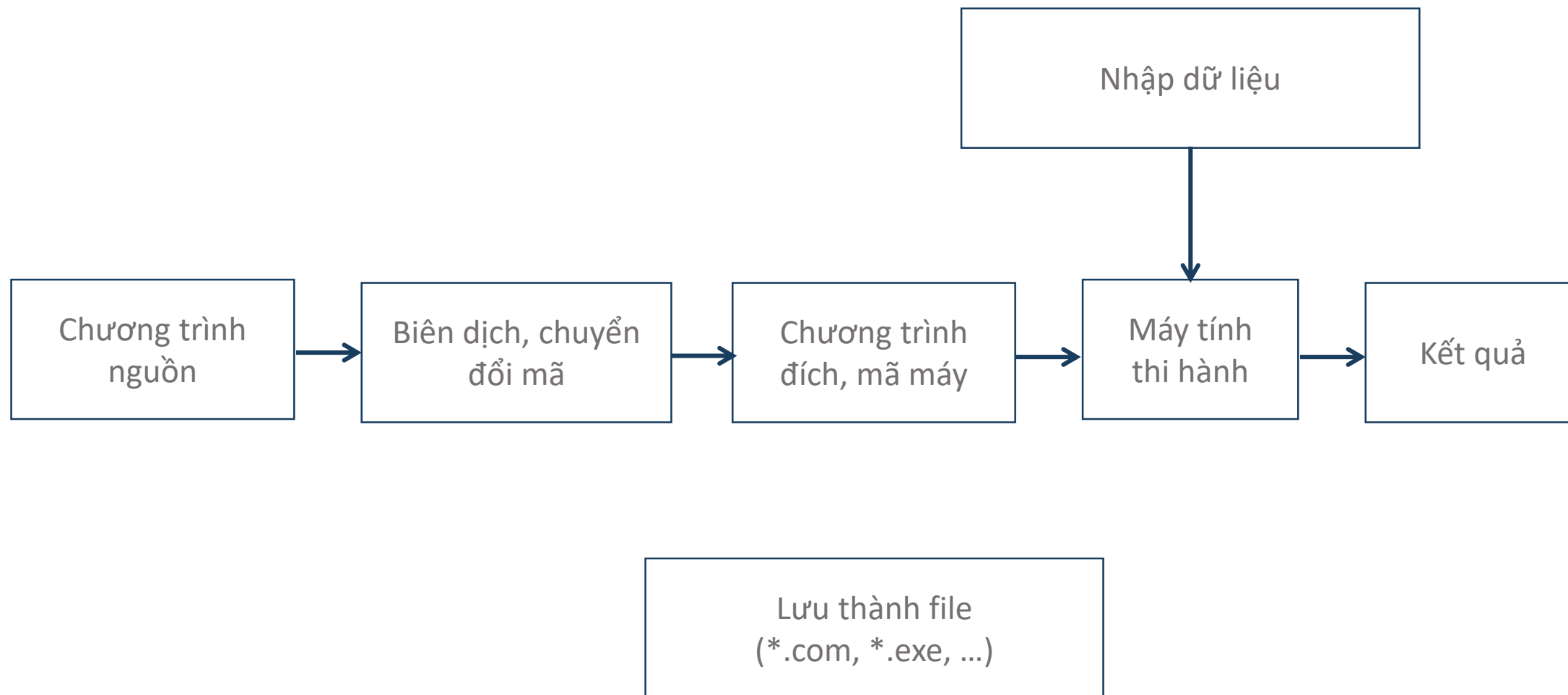
Chương trình dịch thường được phân biệt ở một trong hai dạng:

- ❖ **Trình thông dịch** (*Interpreter*)
- ❖ **Trình biên dịch** (*Compiler*)

# Trình biên dịch - Compiler

- Làm nhiệm vụ chuyển đổi một chương trình đã được viết bằng một ngôn ngữ lập trình nào đó (*Còn gọi là chương trình nguồn*) thành ngôn ngữ máy (*Chương trình đích*).
- Quá trình chuyển đổi từ chương trình nguồn thành chương trình đích thường được gọi là **thời gian dịch** (*Compile-time*) và thời gian thực thi chương trình sau khi đã biên dịch thành công được gọi là **thời gian thực thi** (*Run-time*)

# Cơ chế biên dịch



Nguyễn Mai Huy - [nmaihuy@boduca.com](mailto:nmaihuy@boduca.com)

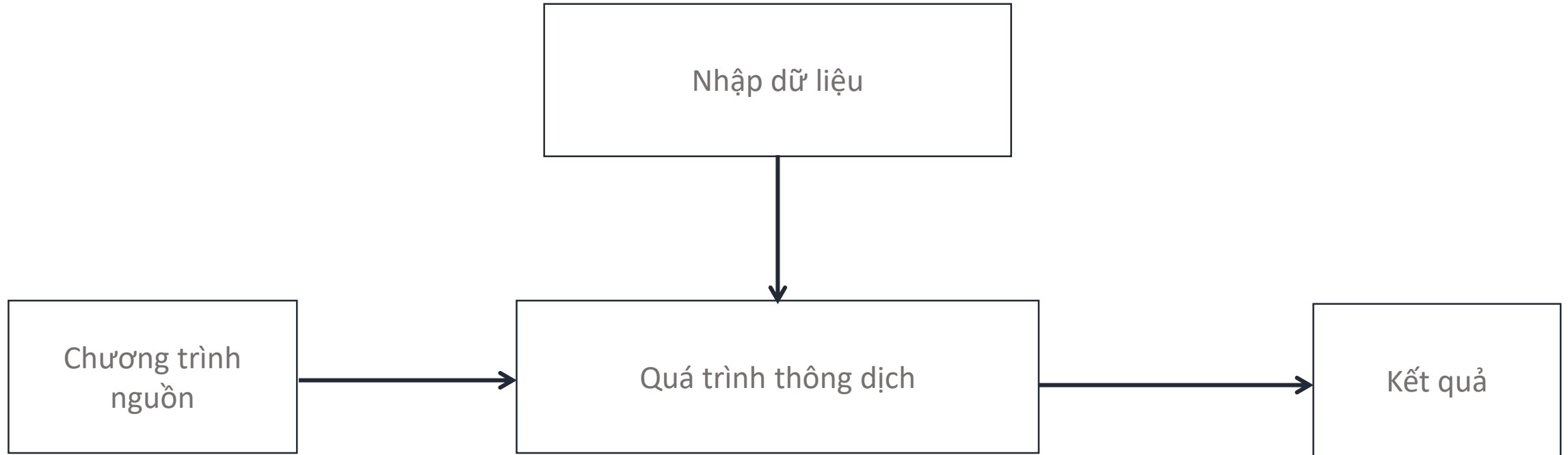
# Trình thông dịch - Interpreter

Những phần mềm có khả năng đọc và chuyển đổi mã nguồn của một chương trình đã được viết bằng ngôn ngữ lập trình ra mã máy để ra lệnh cho máy tính thi hành được gọi là trình thông dịch.

Khác với trình biên dịch, trình thông dịch sẽ dịch từng câu lệnh từ chương trình nguồn theo yêu cầu thực thi.

Như vậy, thời gian dịch diễn ra đồng thời với thời gian thực thi chương trình, quá trình này được gọi là **Thông dịch**

# Cơ chế thông dịch



# Bài tập

- Cho một số yêu cầu thực tế để sinh viên thực hiện vẽ lưu đồ cho mục tiêu cần giải quyết.
- VD:
  - ❖ Vẽ lưu đồ cho bài toán chia hai số nguyên dương (*input: 2 số*)
  - ❖ Vẽ lưu đồ cho bài toán tính tiền 1 loại sản phẩm trong siêu thị (*input: số lượng, đơn giá*)
  - ❖ Vẽ lưu đồ cho việc xếp loại học lực dựa trên điểm trung bình (*input: điểm trung bình*)
  - ❖ ...



# Nhớ gì ?!!!

- Các khái niệm thuật toán, giải thuật. Các đặc tính của thuật toán.  
Các phương pháp biểu diễn thuật toán
- Lưu đồ, ký hiệu sử dụng trên lưu đồ. Các hình thức biểu diễn trên lưu đồ.
- Ngôn ngữ lập trình, ngôn ngữ lập trình thường được chia làm bao nhiêu loại ?
- Khái niệm về chương trình dịch, các cơ chế biên dịch, thông dịch

# Tài liệu tham khảo

- Jose M. Garrido, “**Object-Oriented Programming: From Problem Solving to Java**”, Charles River Media
- Paul Deitel, Harvey Deitel, “**Java : How to program**”, 9<sup>th</sup> edition, 2012
- Oracle, “**The Java™ Tutorials**”,  
<https://docs.oracle.com/javase/tutorial/java/nutsandbolts/index.html>,  
6:20PM, 18/01/2018
- Java tutorial, <https://www.javatpoint.com/java-tutorial> , 6:20PM,  
18/01/2018