



# Chương 6

## Thuật toán và Ngôn ngữ lập trình

# NỘI DUNG CHƯƠNG 6

## 1. PHƯƠNG PHÁP GIẢI QUYẾT VẤN ĐỀ BẰNG MÁY TÍNH

## 2. THUẬT TOÁN

*2.1. Khái niệm thuật toán*

*2.2. Các tính chất của thuật toán*

*2.3. Độ phức tạp của thuật toán*

*2.4. Các cách diễn đạt thuật toán*

## 3. NGÔN NGỮ LẬP TRÌNH

*3.1. Khái niệm về ngôn ngữ lập trình*

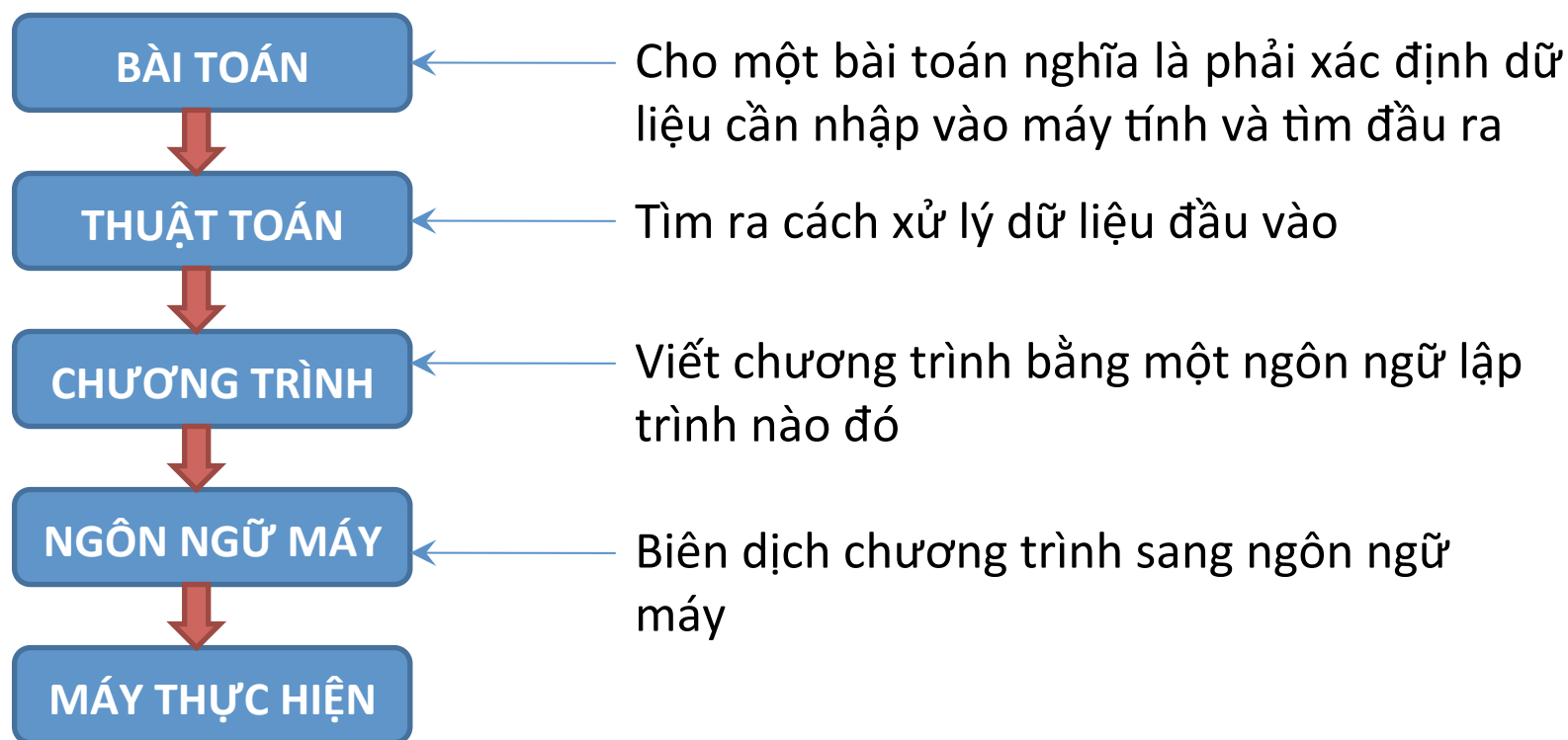
*3.2. Lịch sử phát triển của ngôn ngữ lập trình*

*3.3. Trình biên dịch và trình thông dịch*

*3.4. Các công việc của lập trình*

## 1. PHƯƠNG PHÁP GIẢI QUYẾT VẤN ĐỀ BẰNG MÁY TÍNH

- Phương pháp chung để giải quyết vấn đề (bài toán) bằng máy tính được thể hiện theo sơ đồ sau:



# NỘI DUNG CHƯƠNG 6

## 1. PHƯƠNG PHÁP GIẢI QUYẾT VẤN ĐỀ BẰNG MÁY TÍNH

## 2. THUẬT TOÁN

### 2.1. Khái niệm thuật toán

2.2. Các tính chất của thuật toán

2.3. Độ phức tạp của thuật toán

2.4. Các cách diễn đạt thuật toán

## 3. NGÔN NGỮ LẬP TRÌNH

3.1. Khái niệm về ngôn ngữ lập trình

3.2. Lịch sử phát triển của ngôn ngữ lập trình

3.3. Trình biên dịch và trình thông dịch

3.4. Các công việc của lập trình

## 2.1 Khái niệm thuật toán

- **Thuật toán (thuật giải, algorithms):** là tập hợp hữu hạn các thao tác, phép toán được thực hiện theo một trình tự xác định trên một số đối tượng dữ liệu nào đó để đạt được kết quả mong muốn.
- Để tìm thuật toán cho một bài toán ta cần xác định **dữ liệu vào (input) và dữ liệu ra (output)** cho bài toán.
- VD: Bài toán giải phương trình bậc 2  $ax^2 + bx + c = 0$ 
  - **Dữ liệu vào:** Giá trị của 3 hệ số a, b, c
  - **Dữ liệu ra:** Là nghiệm của phương trình

# NỘI DUNG CHƯƠNG 6

## 1. PHƯƠNG PHÁP GIẢI QUYẾT VẤN ĐỀ BẰNG MÁY TÍNH

## 2. THUẬT TOÁN

*2.1. Khái niệm thuật toán*

**2.2. Các tính chất của thuật toán**

*2.3. Độ phức tạp của thuật toán*

*2.4. Các cách diễn đạt thuật toán*

## 3. NGÔN NGỮ LẬP TRÌNH

*3.1. Khái niệm về ngôn ngữ lập trình*

*3.2. Lịch sử phát triển của ngôn ngữ lập trình*

*3.3. Trình biên dịch và trình thông dịch*

*3.4. Các công việc của lập trình*

## 2.2. Các tính chất của thuật toán

- Tính kết thúc
- Tính thực hiện được
- Tính kết quả
- Tính hiệu quả
- Tính duy nhất
- Tính hình thức

# NỘI DUNG CHƯƠNG 6

## 1. PHƯƠNG PHÁP GIẢI QUYẾT VẤN ĐỀ BẰNG MÁY TÍNH

## 2. THUẬT TOÁN

*2.1. Khái niệm thuật toán*

*2.2. Các tính chất của thuật toán*

***2.3. Độ phức tạp của thuật toán***

*2.4. Các cách diễn đạt thuật toán*

## 3. NGÔN NGỮ LẬP TRÌNH

*3.1. Khái niệm về ngôn ngữ lập trình*

*3.2. Lịch sử phát triển của ngôn ngữ lập trình*

*3.3. Trình biên dịch và trình thông dịch*

*3.4. Các công việc của lập trình*



### **2.3. Độ phức tạp của thuật toán**

- **Đánh giá một thuật toán dựa vào hai tiêu chí sau:**
    - Thời gian thực hiện: Đây là tiêu chí chủ yếu để đánh giá
    - Dung lượng bộ nhớ sử dụng
  - Đánh giá thời gian thực hiện thuật toán người ta dùng **“Độ phức tạp tính toán của thuật toán”**.
- => **Độ phức tạp tính toán của thuật toán** là thời gian thực hiện của thuật toán được đánh giá mà không phụ thuộc vào máy tính và các yếu tố liên quan, chỉ phụ thuộc vào kích thước của dữ liệu đầu vào.

## 2.3. Độ phức tạp của thuật toán

- Gọi  $n$  là kích thước của dữ liệu vào, thì thời gian thực hiện  $T$  của một giải thuật được biểu diễn như một hàm của  $n$ , **gọi là  $T(n)$ .**
- Nếu  $T(n) = Cn^2$  trong đó  $C$  là hằng số, thì ta nói độ phức tạp tính toán của thuật toán này có cấp  $n^2$ , kí hiệu là:  **$T(n) = O(n^2)$**
- **Tổng quát:**
  - Hàm  $f(n)$  có độ phức tạp tính toán cấp  $g(n)$  nếu hàm  $f(n)$  bị chặn bởi  $Cg(n)$ , với  $C$  là hằng số. Kí hiệu là  $f(n) = O(g(n))$
- Các hàm thể hiện độ phức tạp tính toán của giải thuật có các dạng sau:  **$n^n, n!, 2^n, n^3, n^2, n \log_2 n, n, \log_2 n$** . Các hàm đó đã được sắp theo thứ tự ưu tiên giá trị giảm dần

# NỘI DUNG CHƯƠNG 6

## 1. PHƯƠNG PHÁP GIẢI QUYẾT VẤN ĐỀ BẰNG MÁY TÍNH

## 2. THUẬT TOÁN

*2.1. Khái niệm thuật toán*

2.2. Các tính chất của thuật toán

2.3. Độ phức tạp của thuật toán

**2.4. Các cách diễn đạt thuật toán**

## 3. NGÔN NGỮ LẬP TRÌNH

*3.1. Khái niệm về ngôn ngữ lập trình*

*3.2. Lịch sử phát triển của ngôn ngữ lập trình*

*3.3. Trình biên dịch và trình thông dịch*

*3.4. Các công việc của lập trình*

## 2.4. Các cách diễn đạt thuật toán

### **Cách 1:**

- **Liệt kê các bước bằng lời:** Sử dụng ngôn ngữ tự nhiên để liệt kê các công việc của thuật toán qua các bước: Bước 1, Bước 2, Bước 3...

**VD: Cho hai số  $m, n$  tìm  $d = \text{USCLN}(m, n)$**

Bước 1: Kiểm tra nếu  $m = n$  thì về bước 5, nếu không thực hiện tiếp bước 2

Bước 2: Nếu  $m > n$  thì về bước 4 nếu không thực hiện tiếp bước 3

Bước 3:  $m < n$ , bớt  $m$  đi một lượng bằng  $n$  và quay về bước 1

Bước 4: bớt  $m$  đi một lượng bằng  $n$  và quay về bước 1

Bước 5: Lấy  $d$  chính là giá trị chung của  $m$  và  $n$ . Kết thúc

## VÍ DỤ CÁC BƯỚC CỦA THUẬT TOÁN EUCLID

m	n	So sánh
15	21	$m > n$
15	6	$m > n$
9	6	$m > n$
3	6	$m < n$
3	3	$m = n$

$$\text{USCLN}(15,21) = 3$$

Bước 1: Kiểm tra nếu  $m = n$  thì về bước 5, nếu không thực hiện tiếp bước 2

Bước 2: Nếu  $m > n$  thì về bước 4 nếu không thực hiện tiếp bước 3

Bước 3:  $m < n$ , bớt  $m$  đi một lượng bằng  $n$  và quay về bước 1

Bước 4: bớt  $m$  đi một lượng bằng  $n$  và quay về bước 1

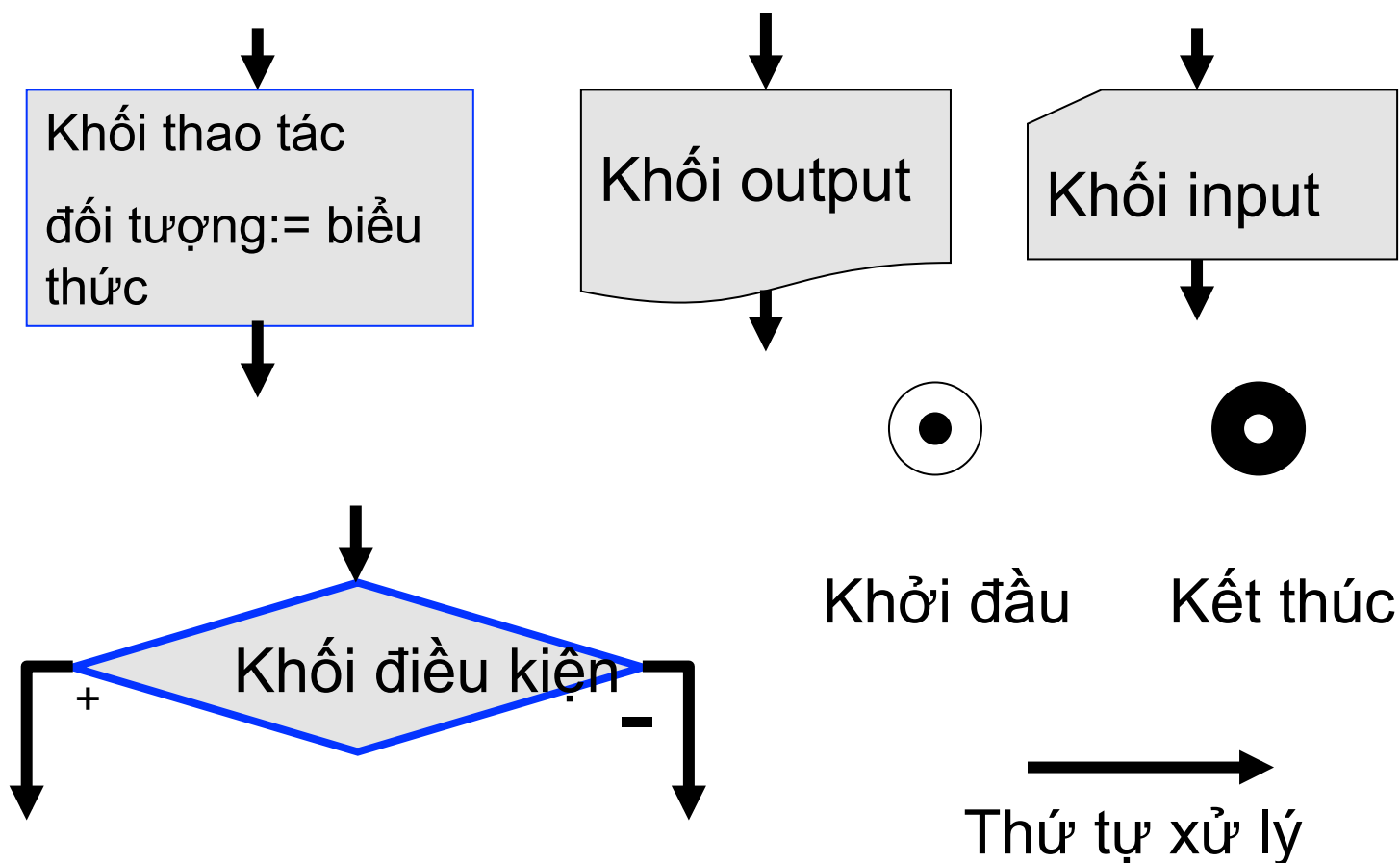
Bước 5: Lấy  $d$  chính là giá trị chung của  $m$  và  $n$ . Kết thúc

## 2.4. Các cách diễn đạt thuật toán

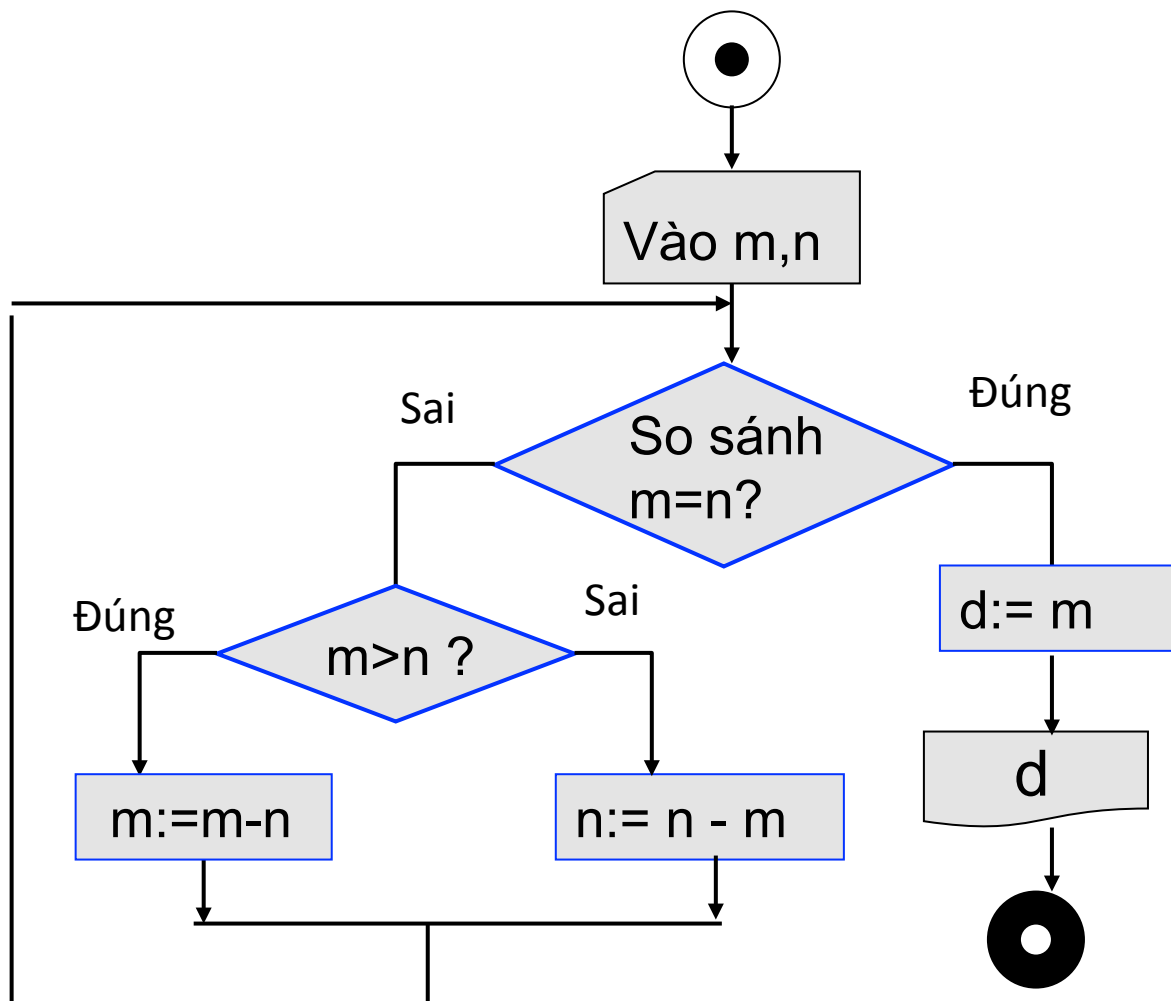
### **Cách 2:**

- **Dùng lưu đồ thuật toán:** Sử dụng các hình vẽ cơ bản để vẽ hình có hướng đi thể hiện các công việc và trình tự thực hiện thuật toán.
- **Các hình cơ bản gồm có:** Bắt đầu, Kết thúc, Vào/Ra dữ liệu, Thực hiện một công việc A, Kiểm tra điều kiện đúng/sai.

## Các hình cơ bản gồm có:



## VD: BIỂU DIỄN BẰNG LƯỚI ĐỒ THUẬT TOÁN EUCLID



Bước 1: Kiểm tra nếu  $m = n$  thì về bước 5, nếu không thực hiện tiếp bước 2

Bước 2: Nếu  $m > n$  thì về bước 4 nếu không thực hiện tiếp bước 3

Bước 3:  $m < n$ , bớt  $m$  đi một lượng bằng  $n$  và quay về bước 1

Bước 4: bớt  $m$  đi một lượng bằng  $n$  và quay về bước 1

Bước 5: Lấy  $d$  chính là giá trị chung của  $m$  và  $n$ . Kết thúc



## 2.4. Các cách diễn đạt thuật toán

### Cách 3:

- **Sử dụng giả ngôn ngữ lập trình (giả mã):** Sử dụng ngôn ngữ tự nhiên kết hợp với một số từ khóa và cấu trúc điều khiển trong ngôn ngữ lập trình bậc cao để diễn tả các công việc của thuật toán.
- VD: Viết thuật toán tìm USCL của 2 số nguyên dương

## VD: Viết thuật toán tìm USCL của 2 số nguyên dương

Trong khi  $m \neq n$  thì lặp lại khối sau:

Nếu  $m > n$  thì

Bớt  $m$  đi một lượng là  $n$

Nếu ngược lại thì

Bớt  $n$  đi một lượng là  $m$

Cho tới khi  $m = n$  thì kết luận USCLN chính là giá trị chung của  $m$  và  $n$

```
read(m,n);  
while m <> n do  
  if m>n then  
    m:=m-n  
  else  
    n:= n-m;  
write(m);
```

Chương trình  
trong PASCAL

# NỘI DUNG CHƯƠNG 6

## 1. PHƯƠNG PHÁP GIẢI QUYẾT VẤN ĐỀ BẰNG MÁY TÍNH

## 2. THUẬT TOÁN

*2.1. Khái niệm thuật toán*

2.2. Các tính chất của thuật toán

2.3. Độ phức tạp của thuật toán

2.4. Các cách diễn đạt thuật toán

## **3. NGÔN NGỮ LẬP TRÌNH**

**3.1. Khái niệm về ngôn ngữ lập trình**

*3.2. Lịch sử phát triển của ngôn ngữ lập trình*

*3.3. Trình biên dịch và trình thông dịch*

*3.4. Các công việc của lập trình*

## 3.1. Khái niệm về ngôn ngữ lập trình

- **Ngôn ngữ lập trình (*programming language*)** : Tập hợp các ký hiệu và các quy tắc viết các lệnh để thể hiện thuật toán
- Ngôn ngữ lập trình gồm 2 loại chính:
  - **Ngôn ngữ lập trình bậc thấp (*hợp ngữ, assembly*):**
    - Có cấu trúc lệnh rất giống với ngôn ngữ máy, chỉ khác là dùng mã chữ thay cho mã nhị phân.
    - Ví dụ: Lệnh ADD AX, BX cộng (Addition) nội dung thanh ghi AX và BX, kết quả để trong AX.

## 3.1. Khái niệm về ngôn ngữ lập trình

### *– Ngôn ngữ lập trình bậc cao (ngôn ngữ thuật toán):*

- Là ngôn ngữ có các lệnh rất gần với ngôn ngữ con người và ngôn ngữ toán học.
- Các ngôn ngữ này chỉ nhằm vào thể hiện thuật toán nên người ta còn gọi là các ngôn ngữ thuật toán.

# NỘI DUNG CHƯƠNG 6

## 1. PHƯƠNG PHÁP GIẢI QUYẾT VẤN ĐỀ BẰNG MÁY TÍNH

## 2. THUẬT TOÁN

*2.1. Khái niệm thuật toán*

2.2. Các tính chất của thuật toán

2.3. Độ phức tạp của thuật toán

2.4. Các cách diễn đạt thuật toán

## 3. NGÔN NGỮ LẬP TRÌNH

3.1. Khái niệm về ngôn ngữ lập trình

**3.2. Lịch sử phát triển của ngôn ngữ lập trình**

*3.3. Trình biên dịch và trình thông dịch*

*3.4. Các công việc của lập trình*

## **3.2. Lịch sử phát triển của ngôn ngữ lập trình**

- **Thế hệ 1** (đầu năm 1950): Lập trình ở mức mã máy điển hình là hợp ngữ (assembly).
- **Thế hệ 2** (Từ cuối năm 1950 đến hết năm 1960):
  - Các lệnh của hợp ngữ được gộp lại thành các câu lệnh có cấu trúc.
  - Các ngôn ngữ lập trình: FORTRAN, COBOL, ALGOL và cao hơn một chút là BASIC.

## 3.2. Lịch sử phát triển của ngôn ngữ lập trình

- **Thế hệ 3:** Đây là thế hệ của các ngôn ngữ lập trình hiện đại, có tính cấu trúc mạnh mẽ. Các ngôn ngữ lập trình trong thế hệ này chia thành 3 lớp:
  - **Ngôn ngữ lập trình cấp cao vạn năng:** Gồm có PL/1, Pascal, Modula-2, C và Ada.
  - **Ngôn ngữ lập trình hướng đối tượng:** Là các ngôn ngữ lập trình cài đặt được các nội dung của phương pháp lập trình hướng đối tượng. Điển hình là C++, Smalltalk và Eiffel.
  - **Ngôn ngữ lập trình chuyên dụng:** Là ngôn ngữ có dạng cú pháp bất thường được thiết kế riêng cho các ứng dụng. Ví dụ như LISP, PROLOG, APL, FORTH... LISP
- **Thế hệ 4:** Gồm có Ngôn ngữ hỏi, bộ sinh chương trình.



# NỘI DUNG CHƯƠNG 6

## 1. PHƯƠNG PHÁP GIẢI QUYẾT VẤN ĐỀ BẰNG MÁY TÍNH

## 2. THUẬT TOÁN

*2.1. Khái niệm thuật toán*

2.2. Các tính chất của thuật toán

2.3. Độ phức tạp của thuật toán

2.4. Các cách diễn đạt thuật toán

## 3. NGÔN NGỮ LẬP TRÌNH

3.1. Khái niệm về ngôn ngữ lập trình

3.2. Lịch sử phát triển của ngôn ngữ lập trình

**3.3. Trình biên dịch và trình thông dịch**

*3.4. Các công việc của lập trình*

### **3.3. Trình biên dịch và trình thông dịch**

- Máy tính chỉ hiểu được một ngôn ngữ duy nhất là ngôn ngữ máy. Bởi vậy, các chương trình viết bằng các ngôn ngữ lập trình (chương trình nguồn) phải được dịch sang ngôn ngữ máy.
- Có hai kiểu dịch: thông dịch và biên dịch.
  - **Thông dịch (Interpreter)** là kiểu dịch từng lệnh để hiểu được công việc phải làm và thực hiện luôn không cần tạo ra những đoạn mã tương ứng trong ngôn ngữ máy.

### **3.3. Trình biên dịch và trình thông dịch**

- **Biên dịch** là dịch toàn bộ chương trình nguồn thành một chương trình tương ứng trong ngôn ngữ máy (chương trình đích), sau đó nạp chương trình đích vào máy tính để thực hiện.
  - Một chương trình thực hiện việc biên dịch chương trình nguồn sang ngôn ngữ máy được gọi là **trình biên dịch**.
  - Mỗi ngôn ngữ lập trình có một trình biên dịch tương ứng. Ví dụ các ngôn ngữ lập trình có trình biên dịch là Pascal, C, C++, Java, C#...

# NỘI DUNG CHƯƠNG 6

## 1. PHƯƠNG PHÁP GIẢI QUYẾT VẤN ĐỀ BẰNG MÁY TÍNH

## 2. THUẬT TOÁN

2.1. *Khái niệm thuật toán*

2.2. Các tính chất của thuật toán

2.3. Độ phức tạp của thuật toán

2.4. Các cách diễn đạt thuật toán

## 3. NGÔN NGỮ LẬP TRÌNH

3.1. Khái niệm về ngôn ngữ lập trình

3.2. Lịch sử phát triển của ngôn ngữ lập trình

3.3. Trình biên dịch và trình thông dịch

**3.4. Các công việc của lập trình**

## 3.4. Các công việc của lập trình

- **Soạn thảo:**
  - Lưu tệp chương trình với phần mở rộng phù hợp với ngôn ngữ lập trình sử dụng,
  - ví dụ .pas cho Pascal, .c cho ngôn ngữ C hay .cpp cho ngôn ngữ C++...
  - Vd: Notepad++,
- **Biên dịch chương trình:**
  - dịch toàn bộ tệp chương trình nguồn sang tệp mã máy
- **Chạy thử chương trình**

## Câu hỏi và bài tập

1. Thuật toán là gì? Cho ví dụ.
2. Xác định input và output cho các thuật toán sau đây:
  - a. Rút gọn một phân số.
  - b. Kiểm tra xem ba số cho trước  $a$ ,  $b$  và  $c$  có thể là độ dài ba cạnh của một tam giác hay không?
3. Trình bày tính chất xác định của thuật toán và nêu rõ nghĩa của tính chất này
4. Cho tam giác ABC có góc vuông A và cho biết cạnh  $a$  và góc B. Hãy viết thuật toán để tính góc C, cạnh  $b$  và cạnh  $c$ .
5. Hãy phát biểu thuật toán để giải bài toán sau: "Có một số quả táo. Dùng cân hai đĩa (không có quả cân) để xác định quả táo nặng nhất"
6. Chỉ dùng phép cộng, tính bình phương của một số

## Ví dụ chạy trên chương trình Pascal

- Bài 1: Tìm USCLN của hai số nguyên dương
- Bài 2: Sắp xếp dãy số tăng dần
- Bài 3: Tìm vị trí số lớn nhất trong dãy số.