



Môn học

Lập trình hướng đối tượng



Nội dung

❖ Chương 1: Tổng quan về Lập trình hướng đối tượng

- Các phương pháp lập trình
- Một số khái niệm lập trình hướng đối tượng
- Nguyên lý lập trình hướng đối tượng
- Ưu điểm của lập trình hướng đối tượng



1. Các phương pháp lập trình

❖ Lập trình tuyến tính (tuần tự)

- Chương trình gồm các câu lệnh được viết tuần tự và sẽ được thực hiện từ đầu đến cuối. Lúc này, người ta sử dụng lệnh **goto** và nhãn để thực hiện các “nhảy” lệnh theo những yêu cầu cụ thể. Assembly là một ví dụ cụ thể.
- Phương pháp lập trình này kém hiệu quả và khó kiểm soát lỗi.

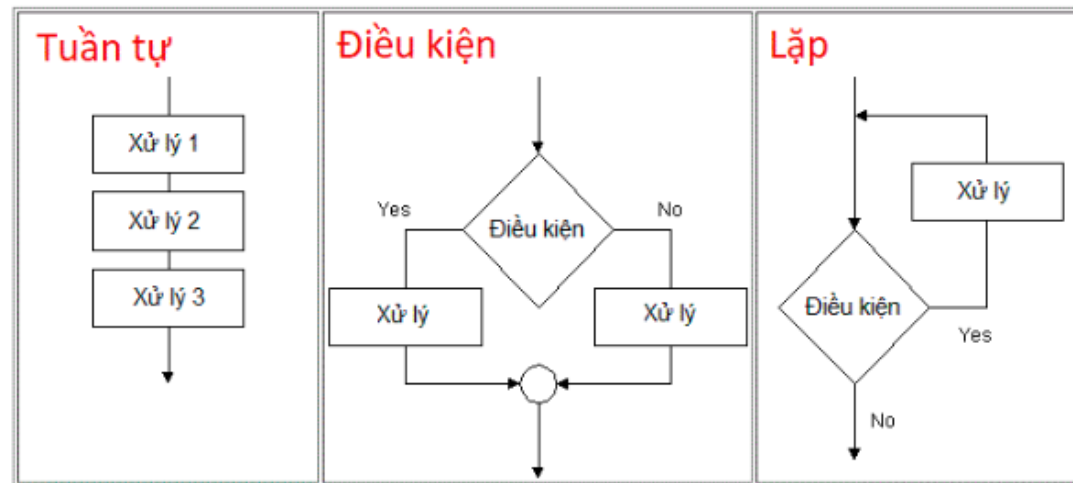
Linear execution	Conditional execution	
Step A	Step A	RELOAD EQU 0E6H ; defining reload constant for baudrate generation
Step B	Step B	ORG 0000H ; org directive
Step C	If ?? goto L1	SJMP START ; jump to main program
Step D	Step C	SENDCH: ORG 0023H ; ISR for RI/TI interrupt
Step E	Step D	CLR TI ; clear transmit flag
Step F	goto L2	MOV SBUF, #'A' ; send the ASCII value of 'A'
Step G	L1:	RETI ; return back to main program
	Step E	START: ANL PCON, #7FH ; Set SMOD=0
	Step F	ANL TMOD, #0FH ; Alter only the setting of Timer-1
	L2:	ORL TMOD, #20H ; Timer-1 in mode-2
	Step G	MOV TH1, #RELOAD ; Move the reload value to TH1
		SETB TR1 ; start Timer-1 for baud rate generation
		MOV SCON, #40H ; set serial port in mode-1
		ORL IE, #90H ; Enable serial port interrupt
		MOV SBUF, #'A' ; Transmit a character
		WAIT: SJMP WAIT ; wait till interrupt occurs
		END



1. Các phương pháp lập trình (t)

❖ Lập trình cấu trúc

- Cho phép chương trình được viết và thực thi theo cấu trúc logic dễ hiểu, dễ sửa đổi và hiệu quả hơn.
- Được đề xuất bởi hai nhà toán học Corrado Bohm và Guiseppe Jacopini. Họ đã chứng minh rằng, mọi chương trình máy tính được viết với ba cấu trúc: tuần tự, điều kiện, và vòng lặp.

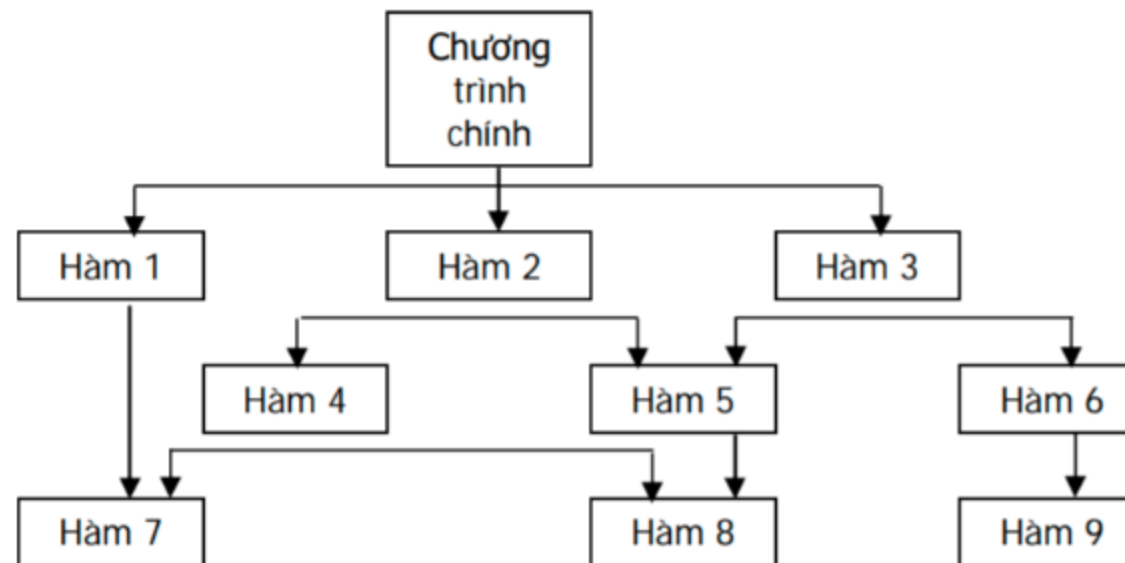




1. Các phương pháp lập trình (t)

❖ Lập trình cấu trúc

- Chương trình được tổ chức thành các hàm. Mỗi hàm đảm nhận xử lý một việc cụ thể trong toàn bộ hệ thống. Mỗi hàm có thể chia thành các hàm nhỏ hơn.

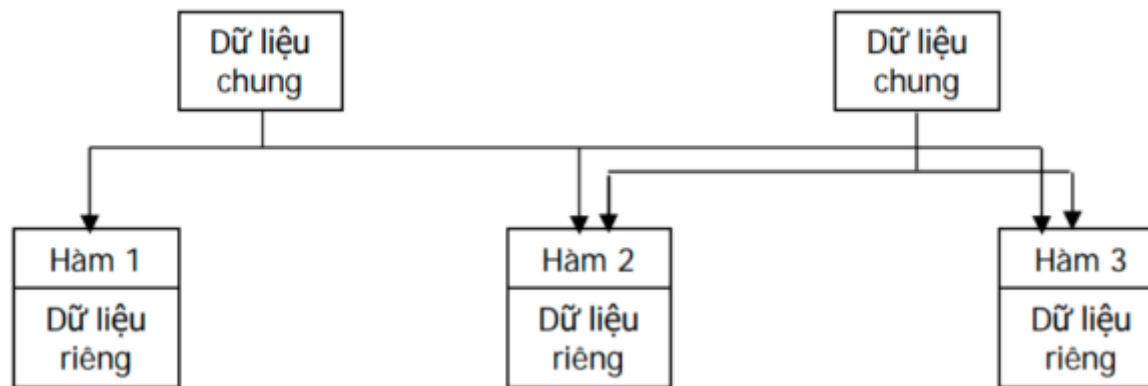




1. Các phương pháp lập trình (t)

❖ Lập trình cấu trúc

- Dữ liệu được chuyển đổi qua lại thông qua các tham số gọi hàm. Dữ liệu có tính chất dùng chung cho toàn bộ hệ thống sẽ được khai báo toàn cục.



- Tuy nhiên, trong nhiều trường hợp cần hiểu và xử lý dữ liệu một cách linh động, thì lập trình theo cấu trúc bậc lộ nhiều hạn chế.



1. Các phương pháp lập trình (t)

❖ Lập trình hướng đối tượng

- Đây là kiểu lập trình hướng vào sự tương tác giữa các đối tượng. Mỗi đối tượng có những thuộc tính (đặc điểm) và các phương thức (chức năng) của đối tượng.
- Đặc điểm:
 - Chương trình được chia thành các đối tượng độc lập
 - Cấu trúc dữ liệu được thiết kế sao cho đặc tả được các đối tượng
 - Dữ liệu được che giấu, bao bọc
 - Các đối tượng trao đổi thông qua các phương thức.



2. Một số khái niệm lập trình hướng đối tượng

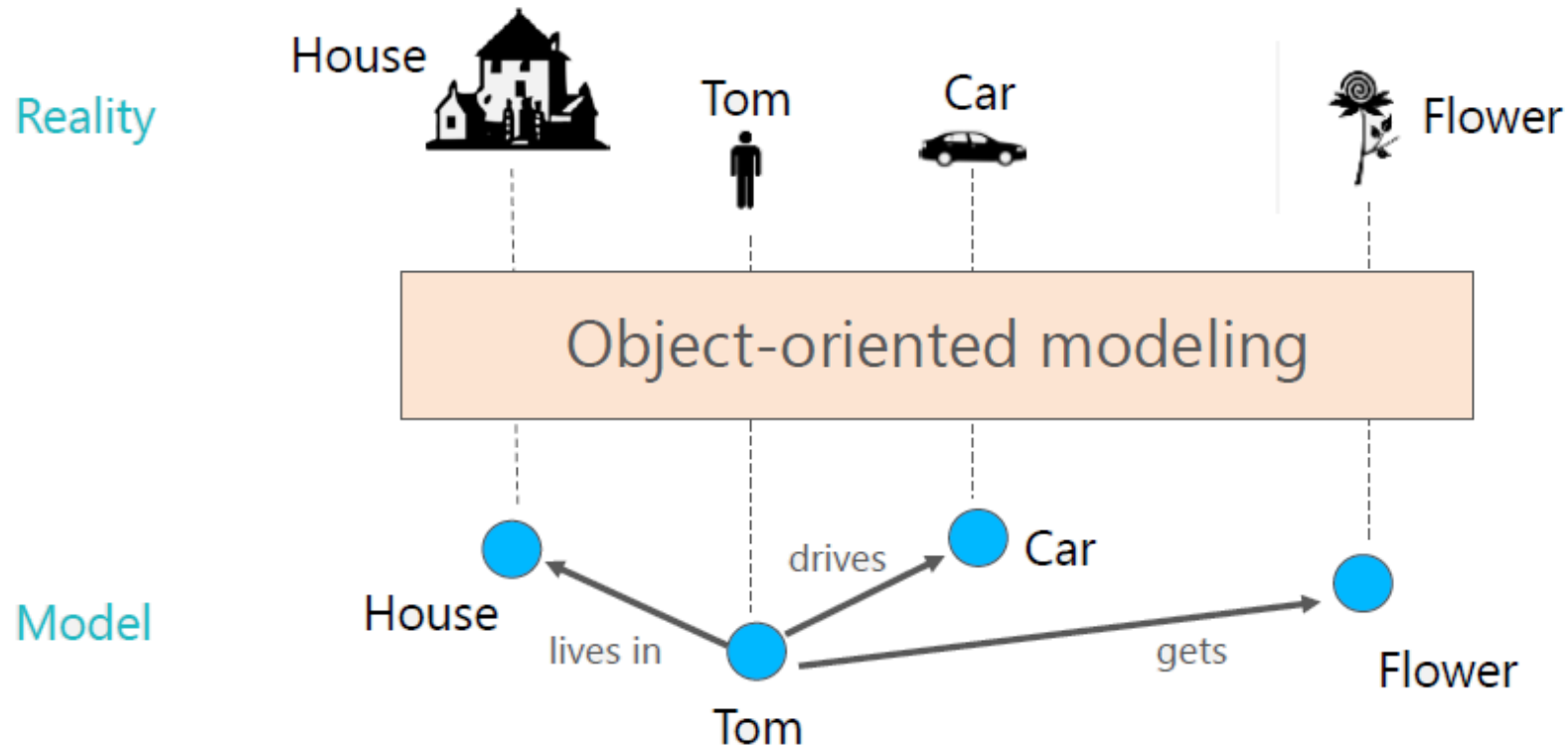
❖ Đối tượng

- Trong thế giới thực, đối tượng là những thực thể có thể tồn tại có trạng thái và hành vi.
- Ví dụ: Con người (sinh viên A, nhân viên B,...), đồ vật (Máy tính, máy in,...) chứng từ (hoá đơn HD001, đơn đặt hàng DDH001,...)



2. Một số khái niệm lập trình hướng đối tượng

❖ Đối tượng (ví dụ về một mô hình hướng đối tượng)





2. Một số khái niệm lập trình hướng đối tượng (t)

❖ Đối tượng

- Tiếp cận hướng đối tượng: Là kỹ thuật cho phép biểu diễn tự nhiên các đối tượng trong thực tế với các đối tượng bên trong chương trình.

	Trạng thái	Hành vi	
Con chó	Tên Màu sắc Giống chó Cảm xúc	Sủa Vẫy đuôi Ăn Chạy	
Ô tô	Hãng sản xuất Kích thước Màu sắc Giá tiền	Tăng tốc Giảm tốc Đâm	
Tài khoản	Tên tài khoản Số tài khoản Ngân hàng Số dư	Rút tiền Gửi tiền Kiểm tra số dư	



2. Một số khái niệm lập trình hướng đối tượng (t)

❖ Đối tượng vs Lớp đối tượng

- Lớp đối tượng (class): có thể được định nghĩa như là template mô tả trạng thái và hành vi mà loại đối tượng của lớp hỗ trợ.
- Đối tượng (object) nó là một thực thể (instance) của một lớp.

object

Một con mèo có
trạng thái: tên, màu
lông, giống và hành
vi: kêu, ăn uống



instance

Con mèo có
-Trạng thái: Tên: Mimi, màu
lông: Vàng nâu, giống: Mèo
thuần chủng
-Hành vi: kêu: meo meo, ăn
uống: ăn thức ăn dành cho
mèo

class

ConMeo
Tên
Mau_long
Giong
keu()
an_uong()



3. Nguyên lý lập trình hướng đối tượng

❖ Tính trừu tượng

- Từ những đối tượng giống nhau có thể trừu tượng hoá thành một lớp
- Loại bỏ tính chất phức tạp của đối tượng bằng cách chỉ ra các thuộc tính và phương thức cần thiết của đối tượng trong lập trình.

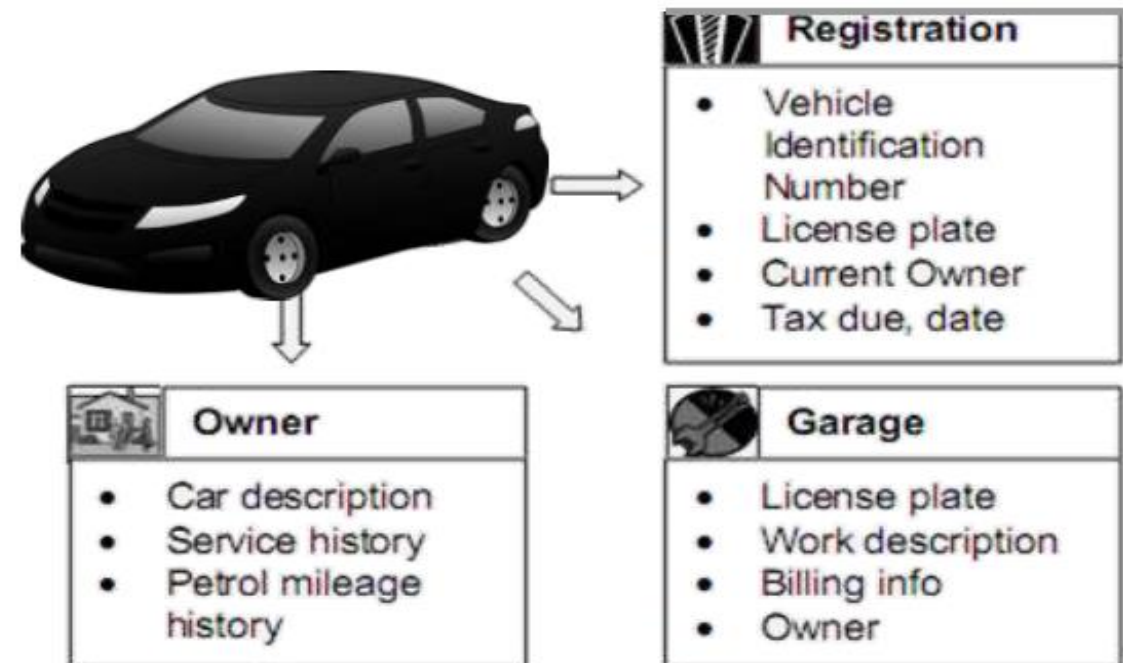
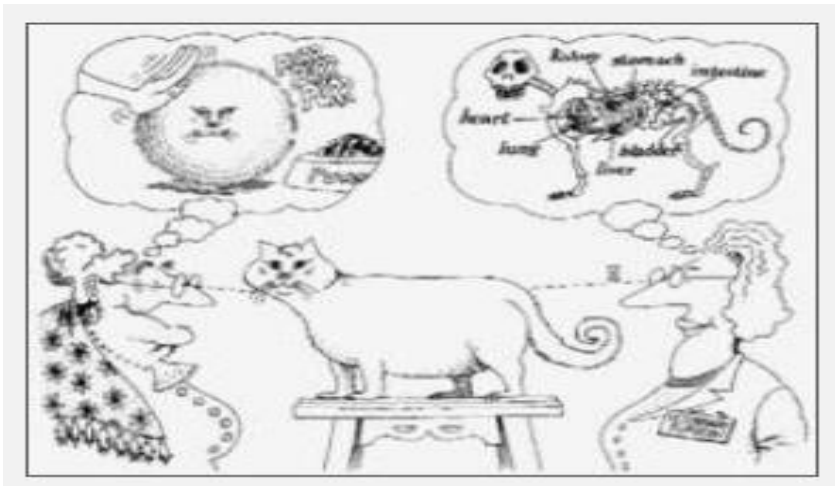




3. Nguyên lý lập trình hướng đối tượng (t)

❖ Tính trừu tượng

- Phụ thuộc vào góc nhìn

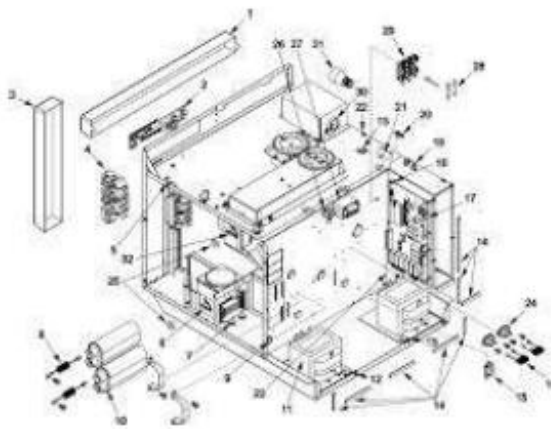




3. Nguyên lý lập trình hướng đối tượng (t)

❖ Đóng gói dữ liệu

- Tính đóng gói (Encapsulation): Mỗi lớp được xây dựng để thực hiện một nhóm chức năng đặc trưng của riêng lớp đó.
- Cho phép dấu thông tin của đối tượng bằng các kết hợp thông tin và các phương thức liên quan đến thông tin trong đối tượng

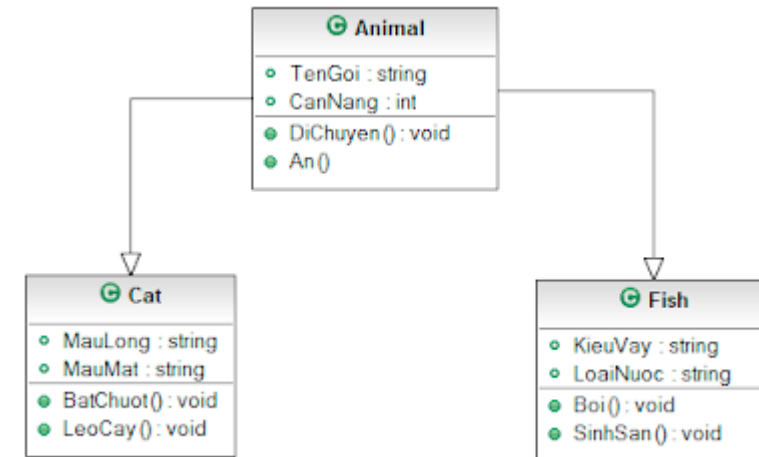
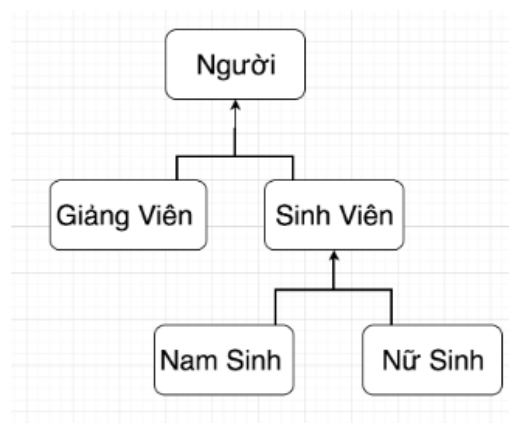




3. Nguyên lý lập trình hướng đối tượng (t)

❖ Kế thừa

- Tính kế thừa (Inheritance): Cho phép xây dựng một lớp mới dựa trên các định nghĩa của một lớp đã có.
 - Lớp đã có gọi là lớp Cha, lớp mới phát sinh gọi là lớp Con
 - Lớp Con kế thừa tất cả các thành phần của lớp Cha, có thể mở rộng các thành phần kế thừa và bổ sung thêm các thành phần mới.

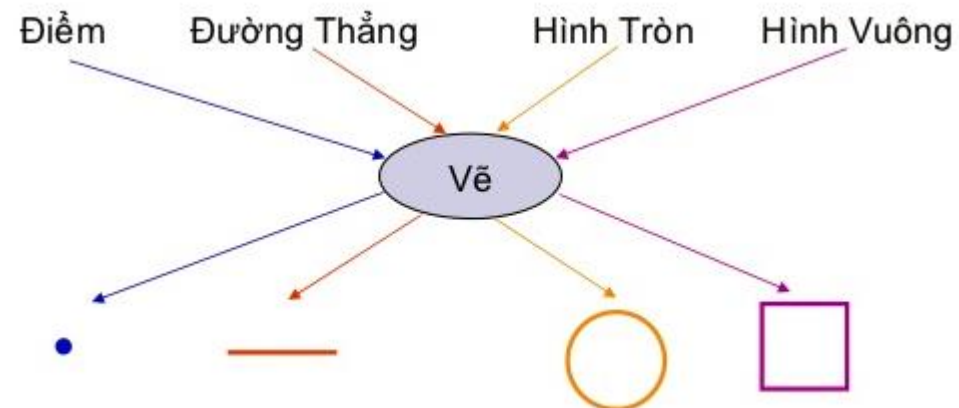




3. Nguyên lý lập trình hướng đối tượng (t)

❖ Đa hình

- Tính đa hình (Polymorphism): cùng hành vi (phương thức) có thể được sử dụng cho những mục đích khác nhau.
- Ví dụ 1: cùng một hành vi là lái xe (drive()) nhưng mỗi xe hơi sẽ có những cách lái khác nhau.
- Ví dụ 2: cùng một hành vi là Vẽ (draw()) nhưng tùy vào đối tượng được chọn là gì mà cách vẽ sẽ khác nhau.





4. Ưu điểm của lập trình hướng đối tượng

- ❖ Dữ liệu không bị thay đổi tự do trong chương trình như lập trình cấu trúc.
- ❖ Khi thay đổi cấu trúc dữ liệu của một đối tượng, không cần thay đổi các mã nguồn của các đối tượng khác mà chỉ cần thanh đối một một số hàm thành phần



4. Ưu điểm của lập trình hướng đối tượng (t)

- ❖ Có thể sử dụng lại mã nguồn qua kế thừa
- ❖ Phù hợp các phần mềm phức tạp, lớn.
- ❖ Được đánh giá là dễ học, năng suất, đơn giản, dễ bảo trì, dễ mở rộng...



Thank you!

Any questions?