

# Lập trình hướng đối tượng

## Chương 1

# LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG VÀ CÁC KHÁI NIỆM CƠ BẢN

# Nội dung

- Tìm hiểu khái niệm về lập trình hướng đối tượng
- Khái niệm trừu tượng hóa dữ liệu
- Khái niệm lớp - đối tượng
- Các tính chất của lập trình hướng đối tượng

Tính trừu tượng

Tính đóng gói

Tính kế thừa

Tính đa hình

- Ngôn ngữ hỗ trợ lập trình hướng đối tượng
- Giới thiệu ngôn ngữ lập trình C++

## 1.1. Khái niệm lập trình hướng đối tượng

### CÁC PHƯƠNG PHÁP LẬP TRÌNH

**LẬP TRÌNH CẤU TRÚC**

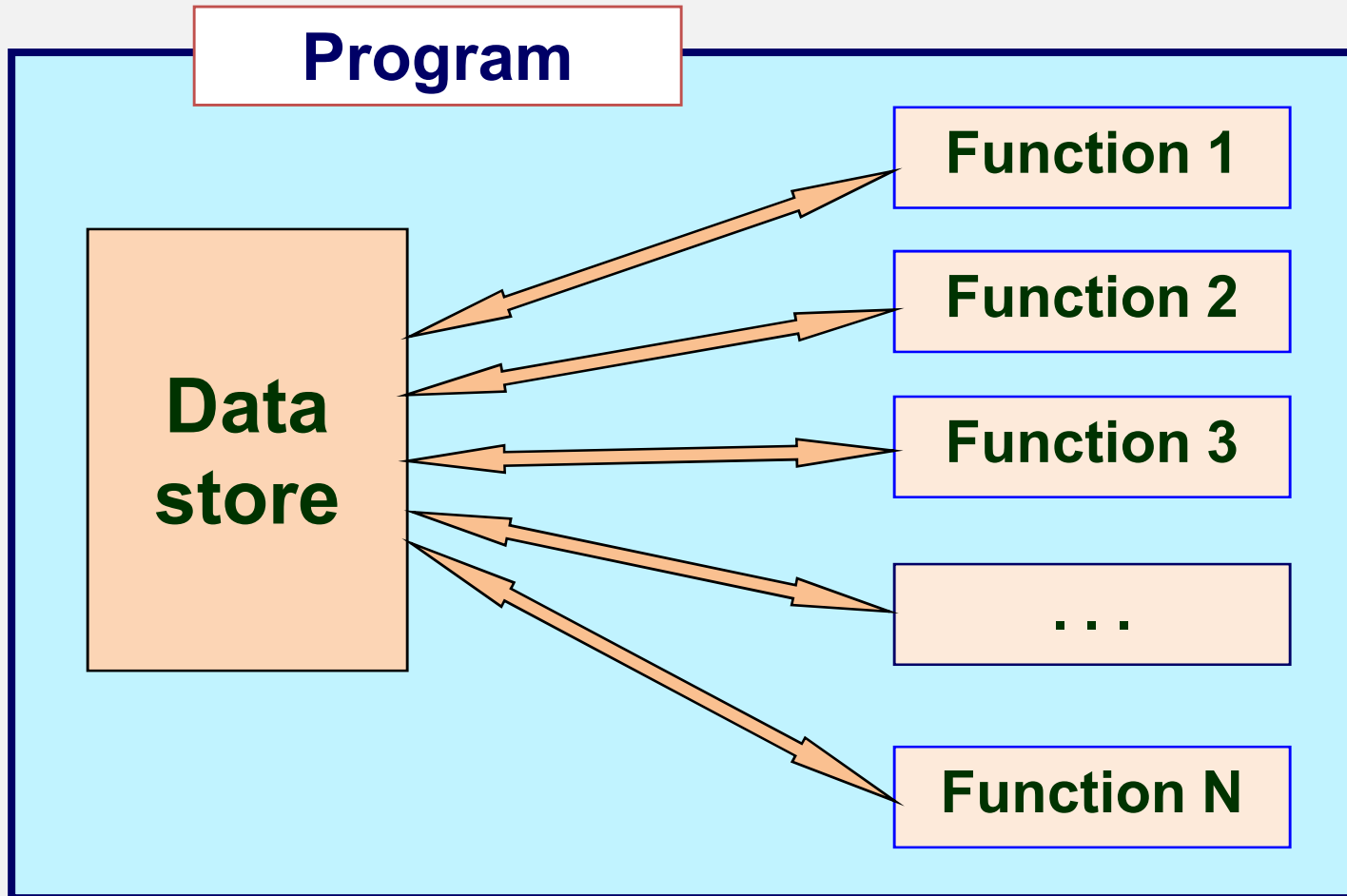
**LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG**

## 1.1.1. Lập trình cấu trúc

- Chương trình lớn được chia nhỏ thành các chương trình con.
- Mỗi chương trình con giải quyết một công việc tương đối độc lập, trọn vẹn.
- Chương trình được viết nhằm giải quyết một dãy tuần các việc cần được thực hiện.
- Mọi dữ liệu của chương trình được lưu trong một kho dữ liệu chung.
- Mọi hàm đều có thể truy nhập vào kho dữ liệu chung.

# Lập trình cấu trúc (tt)

**Chương trình = Cấu trúc dữ liệu + Thuật giải**



# Lập trình cấu trúc (tt)

- Việc thiết kế chương trình được thực hiện theo kiểu **TOP - DOWN**.
- Khi thiết kế chương trình thường tập trung vào các chức năng của bài toán hơn là tập trung vào dữ liệu của bài toán.
- Khi giải quyết một bài toán cần chỉ ra các cấu trúc dữ liệu phù hợp với bài toán và các giải thuật giải quyết các yêu cầu.

# Lập trình cấu trúc (tt)

- **Ưu điểm của lập trình cấu trúc**
  - Quen thuộc, dễ tiếp cận
  - Dễ thiết kế và cài đặt chương trình
  - Phù hợp với những chương trình vừa và nhỏ
- **Nhược điểm của lập trình cấu trúc**
  - Không thích hợp với hệ thống lớn do tính phụ thuộc của mã lệnh vào dữ liệu.
  - Khó mô tả được các đối tượng trong thế giới thực.
  - Việc bảo trì khó khăn vì nếu thay đổi cấu trúc dữ liệu, hoặc cấu trúc của một hàm có thể phải sửa đổi toàn bộ chương trình.

## 1.1.2. Lập trình hướng đối tượng

- Được phát triển dựa trên lập trình cấu trúc.
- Là phương pháp thiết kế và phát triển ứng dụng dựa trên kiến trúc lớp (**class**) và đối tượng (**object**).
- Chương trình được chia thành các đối tượng (đóng gói).

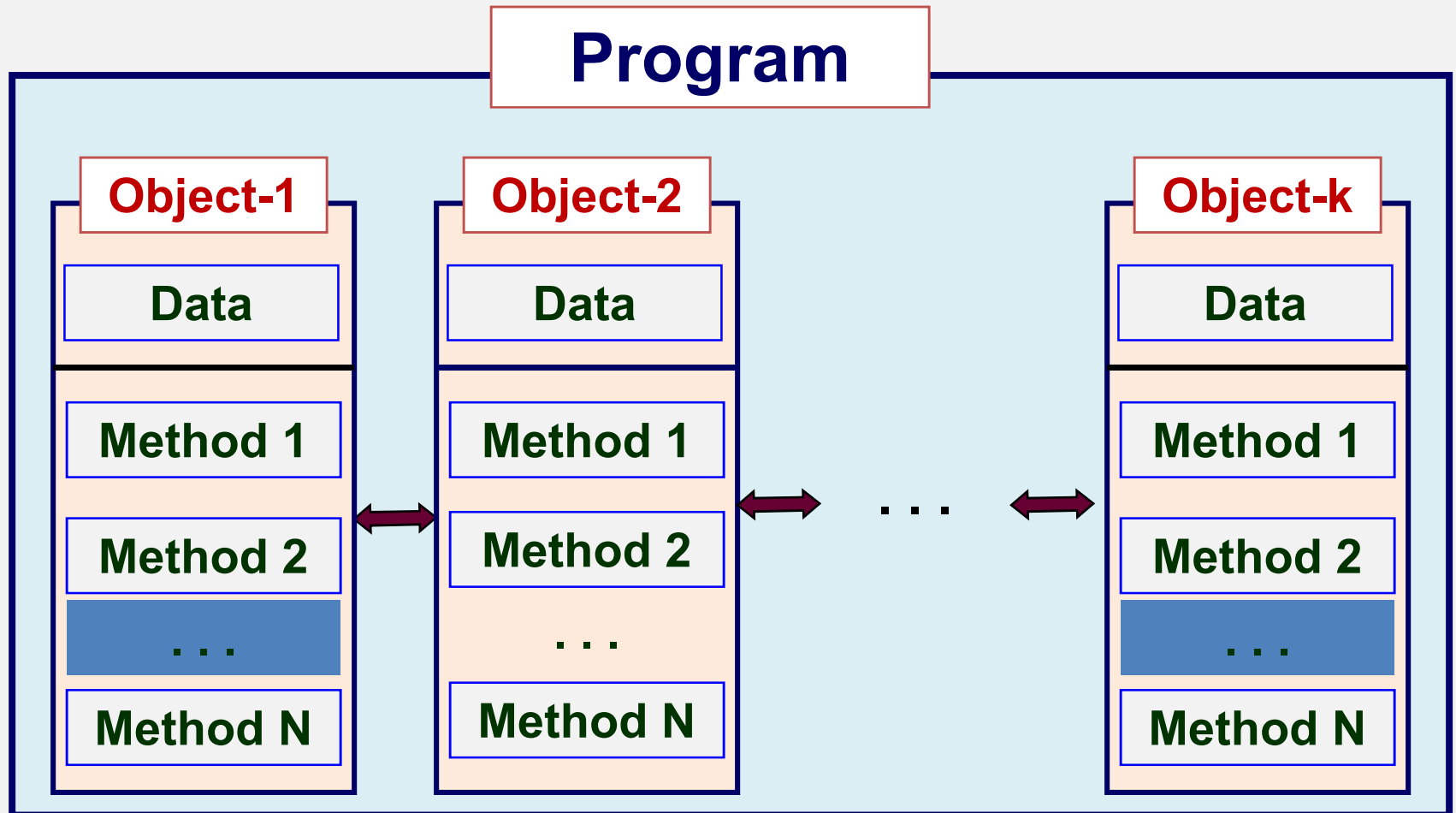
**Đối tượng = Phương thức + Dữ liệu**

- Cho phép liên kết cấu trúc dữ liệu với các thao tác.
- Các đối tượng trao đổi dữ liệu với nhau bằng thông điệp.
- Khi phát triển ứng dụng lấy đối tượng của bài toán làm trung tâm.



# Lập trình hướng đối tượng (tt)

**Object = data + methods**



# Lập trình hướng đối tượng (tt)

- **Ưu điểm của lập trình hướng đối tượng**
  - Mô tả các đối tượng trong chương trình phù hợp với các đối tượng trong thực tế.
  - Việc bảo trì ít khó khăn vì khi sửa đổi cấu trúc của một đối tượng ít ảnh hưởng đến đối tượng khác.
  - Phù hợp với những chương trình lớn.
- **Nhược điểm của lập trình hướng đối tượng**
  - Khó tiếp cận, ít công cụ hỗ trợ.
  - Việc thiết kế và cài đặt phức tạp.

## 1.2. Sự trừu tượng hóa dữ liệu

- Xem xét khái quát cấu trúc và hành vi của các đối tượng mà không quan tâm đến việc biểu diễn cụ thể chúng.
- Thực hiện mô hình hóa các đối tượng.
- Những ưu điểm của việc Trừu tượng hóa là:

**Tập trung vào vấn đề cần giải quyết.**

**Xác định những đặc tính thiết yếu và những hành động cần thiết.**

**Giảm thiểu những chi tiết không cần thiết, hoặc những giải thích.**

## 1.3. Khái niệm lớp - đối tượng

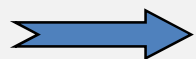
**Đối tượng - Object**

**Lớp - Class**

## 1.3.1. Khái niệm đối tượng - object

- **Đối tượng trong thế giới thực**

- Là những thực thể tồn tại khách quan trong thực tế .
- Ví dụ: quanh ta có thể nhìn thấy các thực thể như: chó, ô tô, xe máy, một bông hoa hay một người ...
- Mỗi thực thể đều có những đặc điểm, và các khả năng thực hiện một hành vi nào đó.
- Đặc điểm là thuộc tính hay các điểm đặc biệt mô tả thực thể, còn hành vi là các hoạt động hay các thao tác có liên quan đến đối tượng.



**Đối tượng là khái niệm mô tả các thực thể**

# Khái niệm đối tượng (tt)



## ○ *Có các đặc điểm:*

- Giống
- Màu lông
- Tuổi

## ○ *Các hành vi*

- Sữa
- Ăn
- Chạy

# Khái niệm đối tượng (tt)

- Xét hệ thống quản lý bán hàng gồm có hai đối tượng: **Thủ quỹ** và **Khách hàng**.

## Thủ quỹ

### Các thuộc tính:

- Tên
- Tuổi
- Chức vụ

### Các hành động

- Thu tiền
- In hóa đơn

## Khách hàng

### Các thuộc tính:

- Tên
- Tuổi
- Trọng lượng

### Các hành động

- Mua hàng
- Trả tiền

# Khái niệm đối tượng (tt) – trong phần mềm

- Là sự ánh xạ từ đối tượng trong thế giới thực.
- Đối tượng trong phần mềm ứng dụng có trạng thái (state) và hành vi (behavior).

Trạng thái của đối tượng phần mềm giống như các thuộc tính, đặc điểm.

Hành vi ứng xử giống như các hành vi, thao tác của đối tượng trong thế giới thực.

- Đối tượng phần mềm lưu trữ trạng thái các trong các trường (fields: các biến trong ngôn ngữ lập trình), các hành động được thể hiện qua các phương thức (method)



# Khái niệm đối tượng (tt)

- Trong chương trình hướng đối tượng thì đối tượng được hiểu theo nghĩa là một thực thể.
- Các dữ liệu và các lệnh tác động lên dữ liệu được đóng gói lại với nhau trong một đơn vị đầy đủ tạo nên đối tượng.
- Một đối tượng gồm có các *phương thức (method)* và các *thuộc tính (property)*

## 1.3.2. Khái niệm lớp – class

- Là một sự trừu tượng hóa của các đối tượng có cấu trúc dữ liệu và các phương thức giống nhau (hay nói cách khác, lớp là một tập các đối tượng cùng loại).
- Mỗi đối tượng sẽ là một thể hiện cụ thể (instance) của lớp đó. Trong lập trình, các đối tượng sẽ là các biến có kiểu lớp...
- Lớp được xem như là một kiểu dữ liệu có cấu trúc được định nghĩa trong chương trình mà các thành phần bên trong nó là các phương thức, các thuộc tính.

### 1.3.3. Lớp và đối tượng

- Lớp động vật, chứa các thuộc tính và hành vi của động vật.

Dong_vat
Chủng loại
Tên
Màu sắc
Di chuyển
Thở
Ăn

# Lớp và đối tượng (tt)

- Một lớp mô tả một tập thực thể, trong khi một đối tượng là một thực thể thật sự trong tập thực thể đó.
- Đối tượng là vật thật, trong khi lớp là một mô hình khái niệm, định nghĩa tất cả các trạng thái và hành vi cần thiết của một đối tượng.
- Lớp không thay đổi, trong khi dữ liệu chứa trong một đối tượng có thể thay đổi. (Với mỗi đối tượng dữ liệu ở cùng thuộc tính có thể là khác nhau)
- Mỗi đối tượng được gọi là một thể hiện của một lớp.

## 1.4. Các tính chất của lập trình hướng đối tượng

- **Tính trừu tượng – Abstraction**
- **Tính đóng gói – Encapsulation**
- **Tính kế thừa – Inheritance**
- **Tính đa hình – polymorphism**

## 1.4.1. Tính trừu tượng

- Từ những đối tượng giống nhau, chúng ta có thể trừu tượng hóa thành một lớp đối tượng.
- Tính trừu tượng cho phép chúng ta loại bỏ tính chất phức tạp của đối tượng bằng cách chỉ đưa ra các thuộc tính và phương thức cần thiết của đối tượng trong lập trình.

## 1.4.2. Tính đóng gói

- Tất cả các dữ liệu dùng để mô tả đối tượng cùng với các phương thức thao tác trên dữ liệu đó được đặt trong một khối (còn gọi là lớp).
- Với mỗi đối tượng người ta không thể truy nhập trực tiếp vào các thành phần dữ liệu của nó mà phải thông qua các thành phần chức năng (còn gọi là các phương thức) của chính đối tượng để làm việc đó.
- Tính đóng gói là tính chất đảm bảo sự toàn vẹn của đối tượng.

### 1.4.3. Tính kế thừa

- Đặc tính này cho phép xây dựng một lớp mới có thể có sẵn các đặc tính mà các lớp khác đã có thông qua kế thừa.
- Điều này cho phép các lớp chia sẻ hay mở rộng các đặc tính sẵn có mà không phải tiến hành định nghĩa lại.
- Tuy nhiên, không phải ngôn ngữ lập trình hướng đối tượng nào cũng có tính chất này.



## 1.4.4. Tính đa hình

- Tính đa hình xuất hiện khi có khái niệm kế thừa.
- Hành vi cụ thể của một đối tượng chỉ được xác định trong ngữ cảnh cụ thể.
- Nếu có một lớp **đa giác** và hai lớp kế thừa từ lớp **đa giác** là lớp **tam giác** và lớp **tứ giác** (tam giác và tứ giác sẽ có đầy đủ các thuộc tính và tính chất của một đa giác).

Lúc này một đối tượng thuộc lớp hình **tam giác** hay **tứ giác** đều có thể hiểu rằng nó là một hình **đa giác**.

Mỗi **đa giác** ta có thể tính diện tích, chu vi của nó. Như vậy làm thế nào mà một **đa giác** có thể sử dụng đúng công thức để tính diện tích phù hợp với nó là hình **tam giác** hay **tứ giác**. Ta gọi đó là tính đa hình.

## 1.5. Ngôn ngữ lập trình hướng đối tượng

- Là các ngôn ngữ được sử dụng để viết các chương trình được thiết kế theo hướng đối tượng.

**Pascal**

**C/C++**

**Java**

**V.V...**

## 1.6. Ngôn ngữ lập trình C++

- Bộ ký tự, các từ khóa, định danh trong C++
- Các kiểu dữ liệu
- Nhập xuất trong C++
- Các cấu trúc điều khiển
- Hàm

## 1.6.1. Bộ ký tự, từ khóa, định danh

- Bộ chữ cái latin, các chữ số, v.v...
- Các từ khóa để khai báo lớp: **class, private, public, protected, friend, operator, v.v...**
- Quy tắc định danh.

## 1.6.2. Các kiểu dữ liệu trong C++

- Kiểu số nguyên: **int, long, unsigned**
- Kiểu số thực: **float, double**
- Kiểu ký tự, chuỗi: **char**
- Kiểu mảng: **mảng 1 chiều, mảng hai chiều**

## 1.6.3. Nhập xuất dữ liệu trong C++

- Các lệnh nhập xuất: **cin>>**, **cout<<**
- Các hàm nhập xuất ký tự: **getch()**, **getchar()**, **getche()**, **putchar()**.
- Các hàm nhập xuất chuỗi: **gets()**, **puts()**.

## 1.6.4. Các cấu trúc điều khiển

- Cấu trúc **if ... else ...**
- Cấu trúc **switch ... case... default...**
- Cấu trúc lặp **for ...**
- Cấu trúc lặp **while ...**
- Cấu trúc lặp **do ... while ...**

## 1.6.5. Hàm

- Khái niệm hàm trong C++
- Cấu trúc hàm, khai báo hàm trong chương trình.
- Phân loại hàm
- Xây dựng và sử dụng hàm
- Truyền tham số cho hàm



## 1.7. Câu hỏi và bài tập

- **Viết một số chương trình đơn giản bằng C++**
  1. **Chương trình hello world**
  2. **Tính chu vi, diện tích tam giác**
  3. **Giải phương trình bậc hai**
  4. **Chương trình hiển thị thông tin sinh viên**
  5. **Chương trình kiểm tra số nguyên tố, hiển thị danh sách 20 số nguyên tố đầu tiên, tổng của chúng.**
  6. **Chương trình tính chu vi diện tích hình tròn, hình thang, hình hộp chữ nhật.**

## Câu hỏi và bài tập (tt)

1. Liệt kê những thuộc tính và những phương thức cần có để tính diện tích một tam giác.
2. Liệt kê các thuộc tính và các phương thức để thực hiện việc giải một phương trình bậc hai.
3. Liệt kê các thuộc tính và các phương thức để tính giai thừa của một số nguyên.
4. Liệt kê các thuộc tính và các phương thức để nhập và hiển thị thông tin của một sinh viên gồm mã, họ tên, năm sinh, điểm tổng kết.