LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG C++

Chương 2. Mở đầu về lập trình hướng đối tượng

Khảo sát các kỹ thuật lập trình

- **Lập trình không có cấu trúc (unstructured programming)**
- **Lập trình hướng thủ tục (procedure programming)**
- **Lập trình hướng mô-đun (modular programming)**
- **Lập trình hướng đối tượng (object-oriented programming)**

Lập trình không có cấu trúc

- Đây là kỹ thuật lập trình của những người mới bắt đầu học lập trình
- Không sử dụng hàm, viết tất cả trong một hàm main()
- Dữ liệu đều sử dụng chung, tất cả các biến đều là biến toàn cục

Chương trình

```
main()
{
    // Các biến dùng chung
}
```

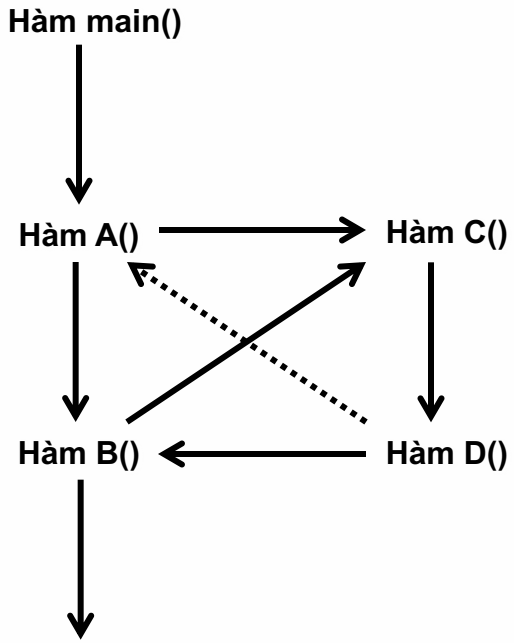
Lập trình không có cấu trúc

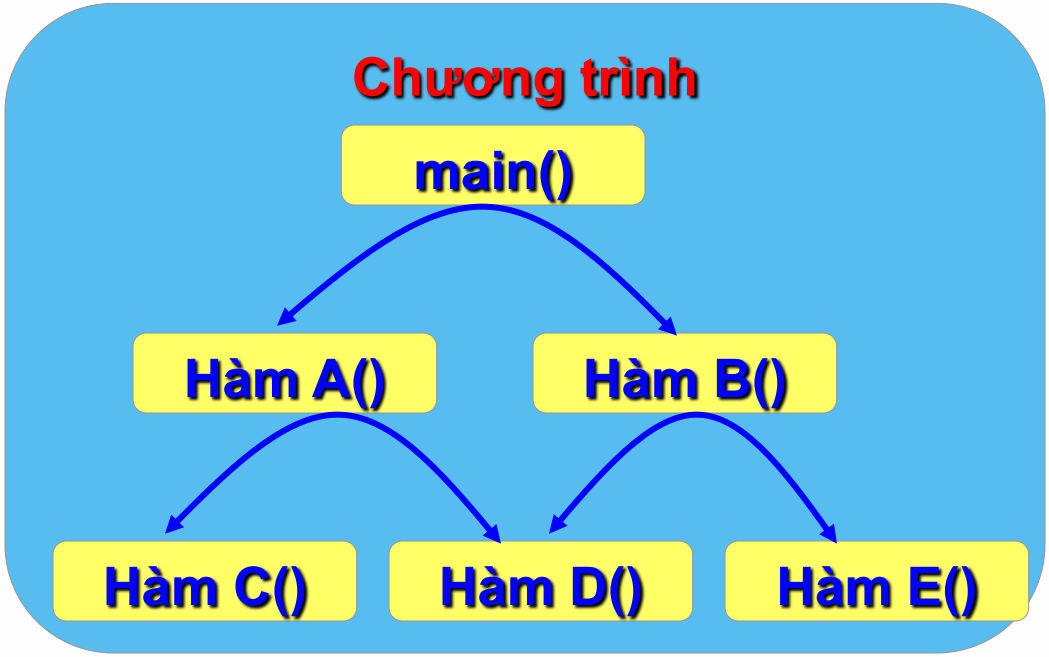
- **Nhược điểm:**
  - **Gặp khó khăn khi xây dựng các chương trình lớn.**
  - **Khi chương trình có những dòng lệnh được lặp lại thì buộc phải copy thành nhiều chỗ**
  - **Chương trình sẽ không khoa học, khó sửa chữa và bảo trì.**

Lập trình hướng thủ tục

- **Lấy các thủ tục (hàm) làm nền tảng xây dựng chương trình.**
- **Chương trình được phân nhỏ thành các thủ tục (hàm), mỗi thủ tục (hàm) sẽ có chức năng riêng biệt.**
- **Các thủ tục (hàm) có thể gọi qua lại lẫn nhau tạo thành một hệ thống hoạt động của chương trình**

Lập trình hướng thủ tục





- Xuất hiện khái niệm biến toàn cục, biến địa phương
- PP tiếp cận: Phân tích top-down

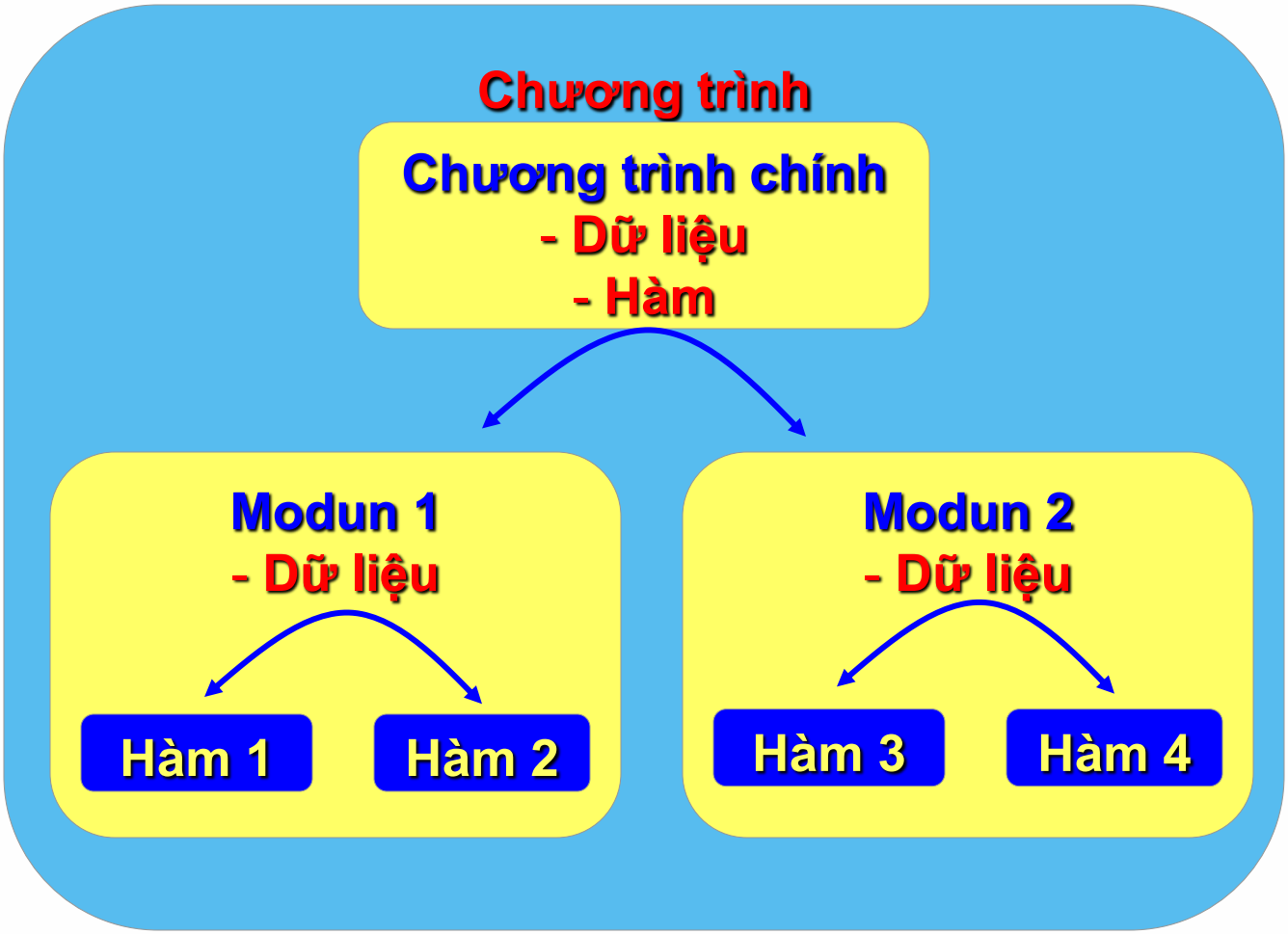
Lập trình hướng thủ tục

- **Ưu điểm:**
  - **Chương trình được tổ chức khoa học, dễ quản lý, bảo trì**
  - **Giải quyết được nhiều bài toán lớn**
- **Nhược điểm:**
  - **Cách tiếp cận đôi khi chưa phù hợp với các hoạt động trong thế giới thực**

Lập trình hướng mô-đun

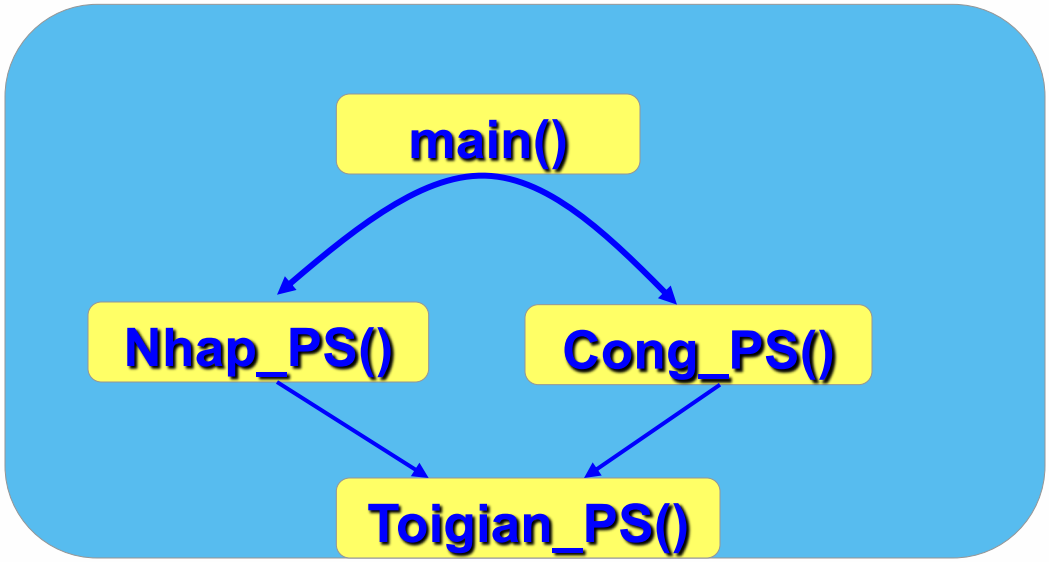
- Ý tưởng cũng tương tự như lập trình hướng thủ tục
- Ở đây, các hàm có chức năng gần giống nhau sẽ được gom lại vào trong các modun độc lập.
- Chương trình sẽ bao gồm nhiều modun chứ không còn đơn lẻ như trước

Lập trình hướng môđun



Lập trình hướng môđun

- **Ví dụ: Viết chương trình cho phép nhập vào hai phân số, sau đó cài đặt các phép toán cộng hai phân số**




```
typedef struct PS
{
    int tu, mau;
}

void Nhap_PS(PS &p);
PS Cong_PS(PS p1, PS p2);
void Toigian_PS(PS &p);
```

Lập trình hướng môđun

- **Một số điểm yếu của lập trình hướng modun:**
  - **Không thể khởi tạo ngầm định các thông tin về một đối tượng nào đó.**
    - **VD:**
      - **Khai báo và khởi tạo một số nguyên: int x = 5;**
      - **Khai báo và khởi tạo một phân số: ?????**
  - **Không thể tự động "dọn dẹp" bộ nhớ khi đối tượng bị hủy bỏ.**
    - **VD:**
      - **Khai báo một số nguyên: int x = 5, sau khi biến hết tác dụng sẽ tự động bị hủy bỏ**
      - **Khai báo và sử dụng một danh sách liên kết: khi không sử dụng danh sách nữa phải hủy bỏ bộ nhớ, nếu không bộ nhớ sẽ bị chứa nhiều "rác"**

Lập trình hướng môđun

- **Một số điểm yếu của lập trình hướng modun(tt):**
  - **Thiếu gắn kết giữa dữ liệu và các thao tác trên dữ liệu đó.**
    - **VD:**
      - **Hàm tối giản phân số: Toigian_PS(PS &p);**
      - **Nhận xét:**
        - » **Nếu tiếp cận theo cách trên, hàm Toigian_PS đóng vai trò chủ thể, phân số tham gia vào như là một đối số của hàm.**
        - » **Thực tế: Phân số mới là chủ thể, tối giản chỉ là một trong những thao tác của phân số**
        - » **Giải pháp: làm thế nào để phân số trở thành chủ thể, khi đó ta sẽ gọi: p.Toigian()**

Lập trình hướng đối tượng:
Cách tiếp cận mới!!!

Lập trình hướng đối tượng

- **Khái niệm:**
  - **Một cách tư duy mới, tiếp cận hướng đối tượng để giải quyết vấn đề bằng máy tính.**
  - **Một phương pháp thiết kế và phát triển phần mềm dựa trên kiến trúc lớp và đối tượng.**
- **Các bước tiến hóa của lập trình hướng đối tượng:**
  1. **Lập trình không có cấu trúc**
  2. **Lập trình có cấu trúc**
  3. **Sự trừu tượng hóa dữ liệu**
  4. **Lập trình hướng đối tượng**

Lập trình hướng đối tượng

- **Tại sao lại phải sử dụng PP lập trình hướng đối tượng :**
  - Loại bỏ những thiếu sót của tiếp cận theo thủ tục
  - Trong lập trình hướng đối tượng:
    - Dữ liệu được xem như một phần tử chính yếu và được bảo vệ
    - Hàm gắn kết với dữ liệu, thao tác trên dữ liệu
    - Phân tách bài toán thành nhiều thực thể (đối tượng) → xây dựng dữ liệu + hàm cho các đối tượng này.
  - Tăng cường khả năng sử dụng lại

Lập trình hướng đối tượng

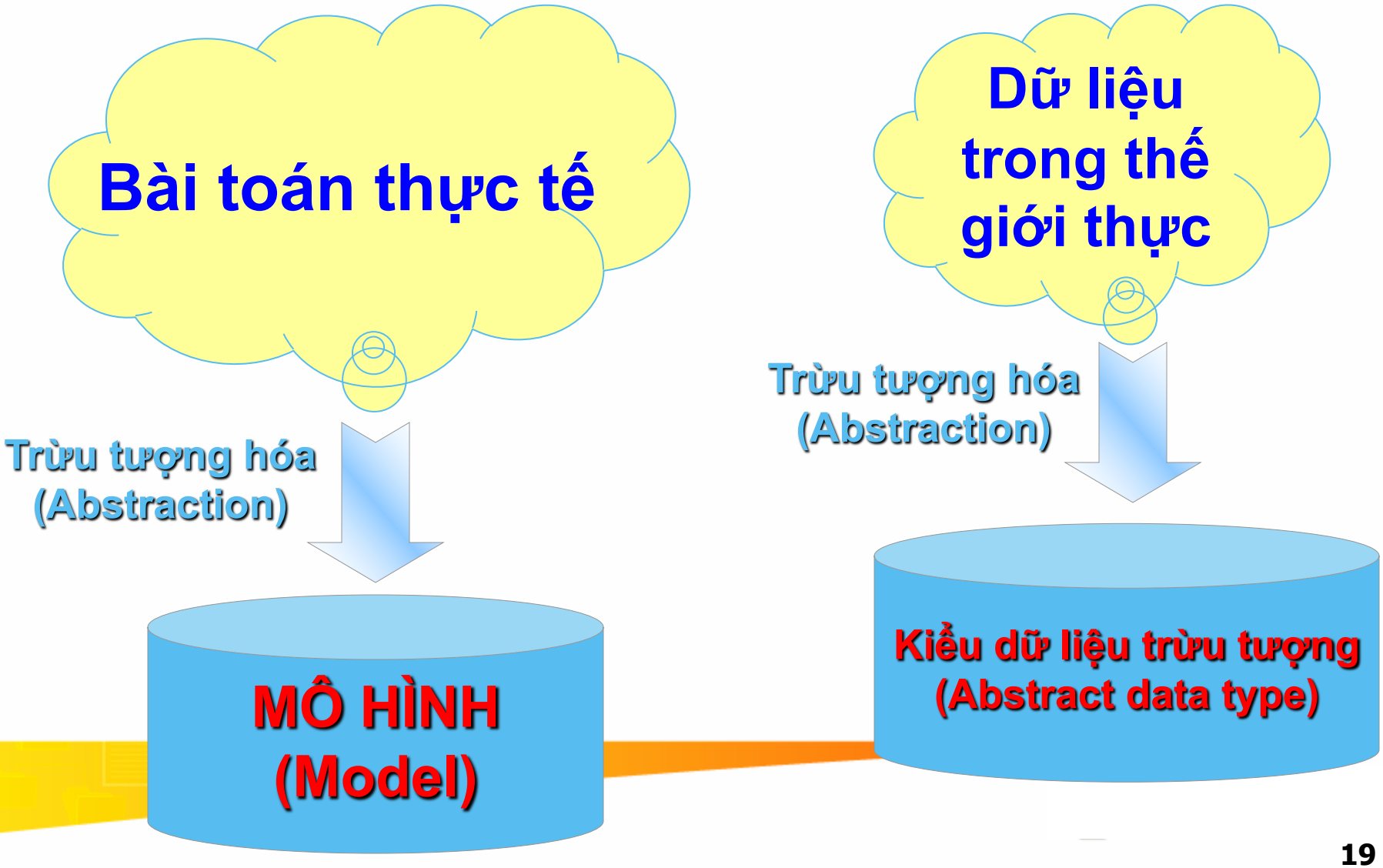
- **Một số đặc điểm nổi bật:**
  - Nhấn mạnh trên dữ liệu hơn là thủ tục
  - Chương trình được dựa trên các đối tượng
  - Dữ liệu được che giấu và không thể được truy xuất từ các hàm bên ngoài
  - Các đối tượng có thể giao tiếp với nhau thông qua các hàm
  - Dữ liệu hay các hàm mới có thể được thêm vào khi cần
  - Theo tiếp cận từ dưới lên (bottom up)

Lập trình hướng đối tượng

- **Thuận lợi so với các cách tiếp cận cổ điển:**
  - **So với các tiếp cận cổ điển thì LT HĐT có những thuận lợi sau:**
    - **LT HĐT cung cấp một cấu trúc module rõ ràng**
      - **Giao diện được định nghĩa tốt**
      - **Những chi tiết cài đặt được ẩn**
    - **LT HĐT giúp lập trình viên duy trì mã và sửa đổi mã tồn tại dễ dàng (các đối tượng được tạo ra với những khác nhau nhỏ so với những đối tượng tồn tại).**
    - **LT HĐT cung cấp một framework tốt với các thư viện mã mà các thành phần có thể được chọn và sửa đổi bởi lập trình viên.**

Trừu tượng hóa dữ liệu???

Trừu tượng hóa dữ liệu



Trừu tượng hóa dữ liệu

- **Ví dụ:**

Trừu tượng hóa
(Abstraction)

Trừu tượng hóa dữ liệu

- **Ví dụ: Xây dựng một chương trình quản lý nhân sự cho một công ty**
  - **Đối tượng: các nhân viên**
  - **Các dữ liệu:**
    - **Họ tên**
    - **Tuổi**
    - **Giới tính**
    - **Chiều cao**
    - **Cân nặng**
    - **Học vấn**
    - **Mức lương**
    - **Tình trạng gia đình**
    - **Màu da**
    - **Màu tóc**
    - **Sở thích ẩm thực**
    - **Sở thích âm nhạc**
    - **...**

Trừu tượng hóa dữ liệu

- **Ví dụ: (tt)**
  - **Các hành động:**
    - **Tuyển dụng một nhân viên mới**
    - **Sa thải một nhân viên**
    - **Tăng lương**
    - **Thưởng**
    - **Giao việc**
    - **...**

Trừu tượng hóa dữ liệu

- **Ví dụ: (tt)**
  - **Khi lập trình biểu diễn đối tượng nhân viên:**
    - **Không thể mô tả toàn bộ các dữ liệu liên quan đến một nhân viên -> chỉ giữ lại một số dữ liệu cần thiết.**
    - **Không thể mô tả toàn bộ hoạt động -> chỉ mô tả một số hoạt động liên quan đến ứng dụng**

Trừu tượng hóa dữ liệu

- **Đối tượng trong thế giới thực:**
  - **Là một thực thể cụ thể mà thông thường bạn có thể *sờ*, *nhìn thấy* hay *cảm nhận* được.**
  - **Đều có *dữ liệu* và *hành động* (phương thức) riêng.**

	Dữ liệu	Hành động	
Con chó	Tên Màu Giống	Sủa Vẫy tai Chạy Ăn	
Xe đạp	Bánh răng Bàn đạp Dây xích Bánh xe	Tăng tốc Giảm tốc Chuyển bánh răng ...	

Trừu tượng hóa dữ liệu

- **Đối tượng trong lập trình:**
  - **Dùng để mô tả, biểu diễn đối tượng trong thế giới thực**
  - **Đối tượng trong lập trình cũng có dữ liệu và hành động (phương thức, hàm) tác động trên các dữ liệu đó.**

Trừu tượng hóa dữ liệu

- **Trừu tượng hóa dữ liệu là quá trình tổ chức một bài toán phức tạp thành những đối tượng được cấu trúc chặt chẽ, trong đó các dữ liệu và hành động của đối tượng được định nghĩa. Trong đối tượng, dữ liệu và hành động được gắn kết chặt chẽ với nhau.**

Đối tượng???

- **Đối tượng** là chìa khóa để hiểu được kỹ thuật hướng đối tượng
- Trong hệ thống hướng đối tượng, mọi thứ đều là đối tượng

Củng cố

- **Các PP lập trình:**
  - **Lập trình không có cấu trúc**
  - **Lập trình hướng thủ tục**
  - **Lập trình hướng mođun**
  - **Lập trình hướng đối tượng**
- **Trừu tượng hóa dữ liệu**
- **Đối tượng (Object)**

Câu hỏi và thảo luận

Chân thành cảm ơn !