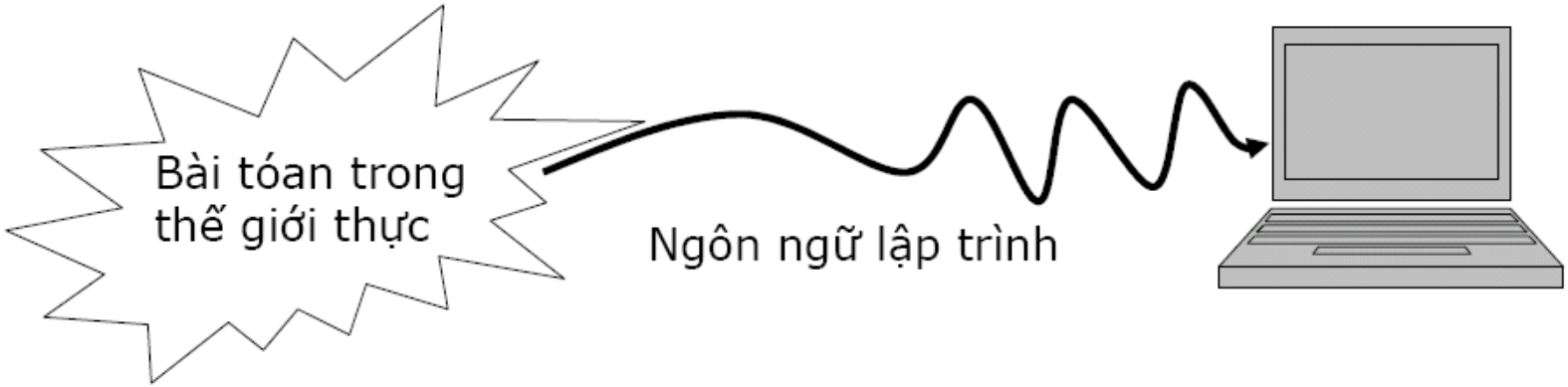
TỔNG QUAN VỀ LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG

Nội dung

1. Giới thiệu
2. Các phương pháp lập trình
3. Một số khái niệm cơ bản
4. Các đặc điểm quan trọng của OOP
5. Một số thuật ngữ OOP

Giới thiệu

- Mục tiêu của kỹ sư lập trình:
  - Tạo ra sản phẩm tốt một cách có hiệu quả
  - Nắm bắt được công nghệ



Giới thiệu

- Độ phức tạp và độ lớn ngày càng cao:
  - Một số hệ Unix chứa khoảng 4M dòng lệnh
  - MS Windows chứa hàng chục triệu dòng lệnh
  - Người dùng ngày càng đòi hỏi nhiều chức năng, đặc biệt là chức năng thông minh
  - Phần mềm luôn cần được sửa đổi
  - …

Giải pháp

- Cần kiểm soát chi phí:
  - Chi phí phát triển
  - Chi phí bảo trì
- Giải pháp chính là ***sử dụng lại*** (tái sử dụng):
  - Giảm chi phí và thời gian phát triển
  - Nâng cao chất lượng

Giải pháp

- Để sử dụng lại (mã nguồn):
  - Cần dễ hiểu
  - Được coi là chính xác
  - Có giao diện rõ ràng
  - Tính module hóa
  - *Không yêu cầu thay đổi khi sử dụng trong chương trình mới*

Mục tiêu của việc thiết kế một phần mềm

- Tính tái sử dụng (reusability): thiết kế các thành phần có thể được sử dụng trong nhiều phần mềm khác nhau
- Tính mở rộng (extensibility)
- Tính mềm dẻo (flexibility):
  - Có thể dễ dàng thay đổi khi thêm mới dữ liệu hay tính năng.
  - Các thay đổi không làm ảnh hưởng nhiều đến toàn bộ hệ thống

Các phương pháp lập trình

- Sự tiến hóa của các phương pháp lập trình:
  - Lập trình không có cấu trúc
  - Lập trình có cấu trúc (lập trình thủ tục), hướng chức năng
  - **Lập trình hướng đối tượng**

Lập trình không có cấu trúc

- Là phương pháp xuất hiện đầu tiên:
  - Các ngôn ngữ như Assembly, Basic
  - Sử dụng các biến toàn cục
  - Lạm dụng lệnh GOTO
- Nhược điểm?
  - Khó hiểu, khó bảo trì, hầu như không thể sử dụng lại
  - Chất lượng kém, Chi phí cao
  - Không thể phát triển các ứng dụng lớn

Lập trình không có cấu trúc

- Ví dụ:

```
10  k =1
20  gosub 100
30  if y > 120 goto 60
40  k = k+1
50  goto 20
60  print k, y
70  stop
100  y = 3*k*k + 7*k-3
110  return
```

Lập trình có cấu trúc

- Tổ chức thành các chương trình con (hay các module)
- Mỗi chương trình con đảm nhận xử lý một công việc nhỏ hay một nhóm công việc trong toàn bộ hệ thống.
- Mỗi chương trình con này lại có thể chia nhỏ thành các chương trình con nhỏ hơn.

Chương trình = Cấu trúc dữ liệu + Giải thuật

Lập trình có cấu trúc

- Sử dụng các lệnh có cấu trúc: for, do, while, if then else...
- Các ngôn ngữ: Pascal, C,...
- Chương trình là tập các hàm/thủ tục
- Ưu điểm?
  - Chương trình được module hóa, do đó dễ hiểu, dễ bảo trì hơn
  - Dễ dàng tạo ra các thư viện phần mềm

Lập trình có cấu trúc

- Ví dụ:

```
struct Date {
    int year, mon, day;
};
//...
void print_date(Date d) {
    printf("%d / %d / %d\n", d.day, d.mon, d.year);
}
```

Lập trình có cấu trúc

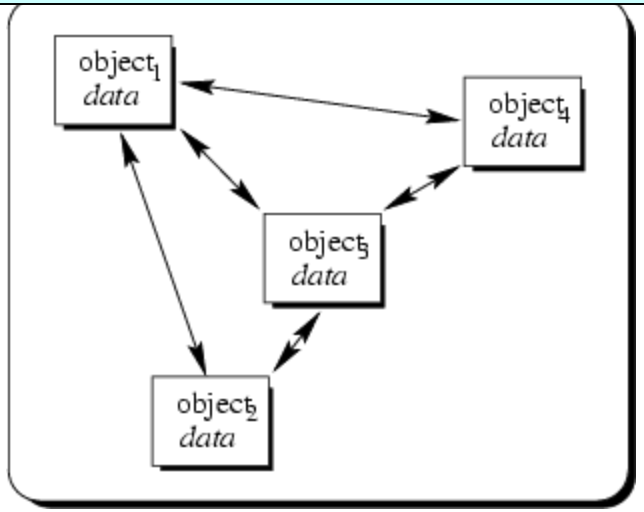
- Nhược điểm?
  - Dữ liệu và mã xử lý là tách rời
  - Người lập trình phải biết cấu trúc dữ liệu
  - Khi thay đổi cấu trúc dữ liệu →thuật toán phải thay đổi theo
  - Khó đảm bảo tính đúng đắn của dữ liệu
  - Không tự động khởi tạo hay giải phóng dữ liệu động
  - Không mô tả được đầy đủ, trung thực hệ thống trong thực tế

Lập trình hướng đối tượng

- Trong thế giới thực, chung quanh chúng ta là những đối tượng, đó là các thực thể có mối quan hệ với nhau.
  - Ví dụ: Các phòng trong một công ty
- Lập trình hướng đối tượng (Object Oriented Programming – LTHĐT)?
  - Là phương pháp lập trình *lấy đối tượng làm nền tảng để xây dựng thuật giải, xây dựng chương trình.*

Lập trình Hướng đối tượng

Lập trình hướng đối tượng là phương pháp lập trình dựa trên kiến trúc lớp (class) và đối tượng (object)



Một số khái niệm cơ bản

Đối tượng (object):

- Trong thế giới thực, đối tượng được hiểu như là một thực thể: người, vật hoặc một bảng dữ liệu…
- Mỗi đối tượng sẽ tồn tại trong một hệ thống và có ý nghĩa nhất định trong hệ thống.
- Đối tượng giúp biểu diễn tốt hơn thế giới thực trên máy tính
- Mỗi đối tượng bao gồm 2 thành phần: *thuộc tính và thao tác (hành động).*

Một số khái niệm cơ bản

- Ví dụ đối tượng: một người
  - Một người có các thuộc tính: *tên, tuổi, địa chỉ, màu mắt…*
  - Các hành động: *đi, nói, thở…*

Một đối tượng là 1 thực thể bao gồm thuộc tính và hành động

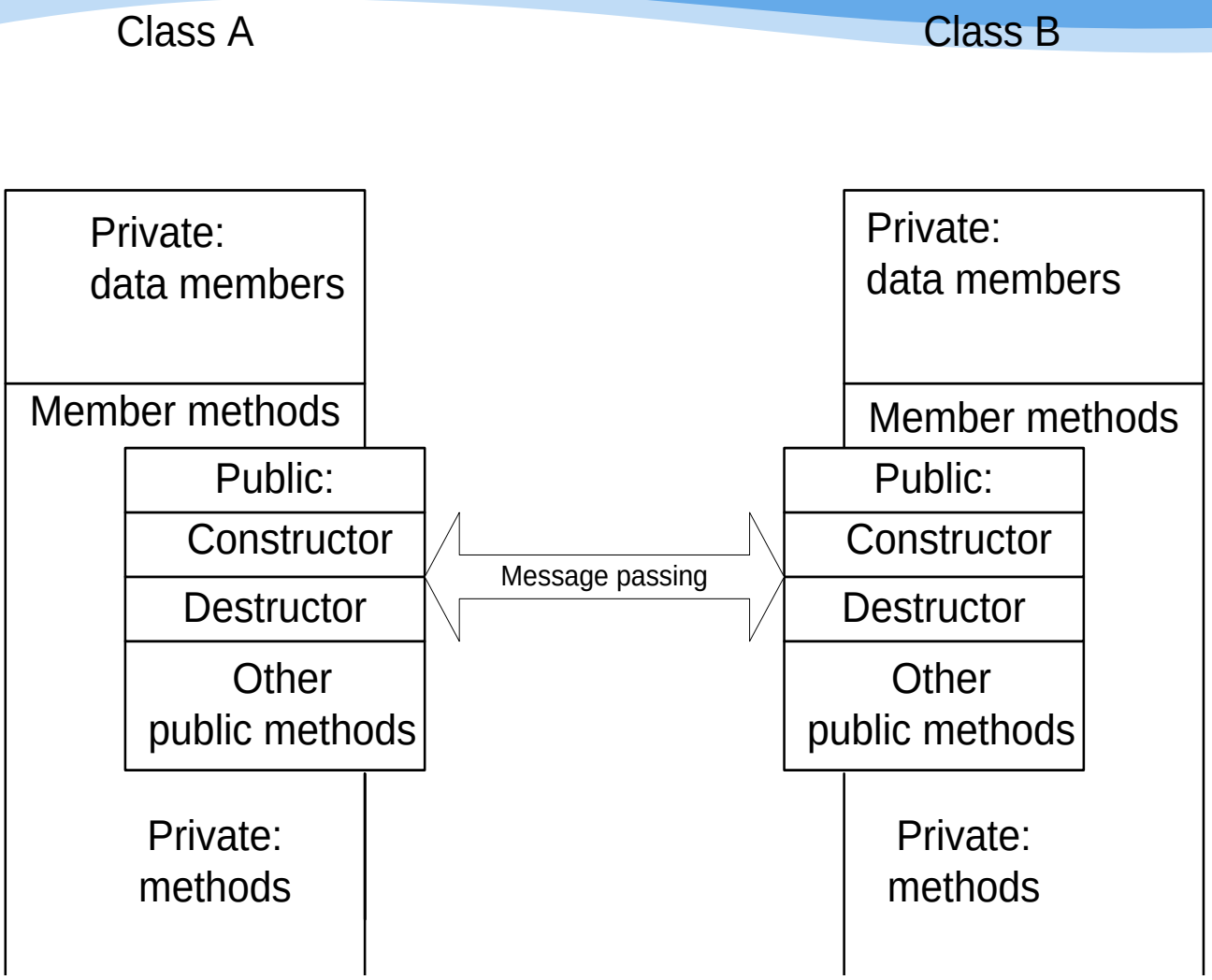
Một số khái niệm cơ bản

- Lớp (class):
  - Các đối tượng có các đặc tính tương tự nhau được gom chung thành lớp đối tượng. Một lớp đối tượng đặc trưng bằng các thuộc tính, và các hành động (hành vi, thao tác).
  - Thuộc tính (Attribute): Một thành phần của đối tượng, có giá trị nhất định cho mỗi đối tượng tại mỗi thời điểm trong hệ thống.
  - Thao tác (Operation): Thể hiện hành vi của một đối tượng tác động qua lại với các đối tượng khác hoặc với chính nó.

Một số khái niệm cơ bản

- Mỗi thao tác trên một lớp đối tượng cụ thể tương ứng với một cài đặt cụ thể khác nhau. Một cài đặt như vậy được gọi là một phương thức (method).
- Cùng một phương thức có thể được áp dụng cho nhiều lớp đối tượng khác nhau, một thao tác như vậy được gọi là có tính đa hình (polymorphism).
- Một đối tượng cụ thể thuộc một lớp được gọi là một thể hiện (instance) của lớp đó.
  - Ví dụ Joe Smith, 25 tuổi, nặng 58kg, là một thể hiện của lớp người.

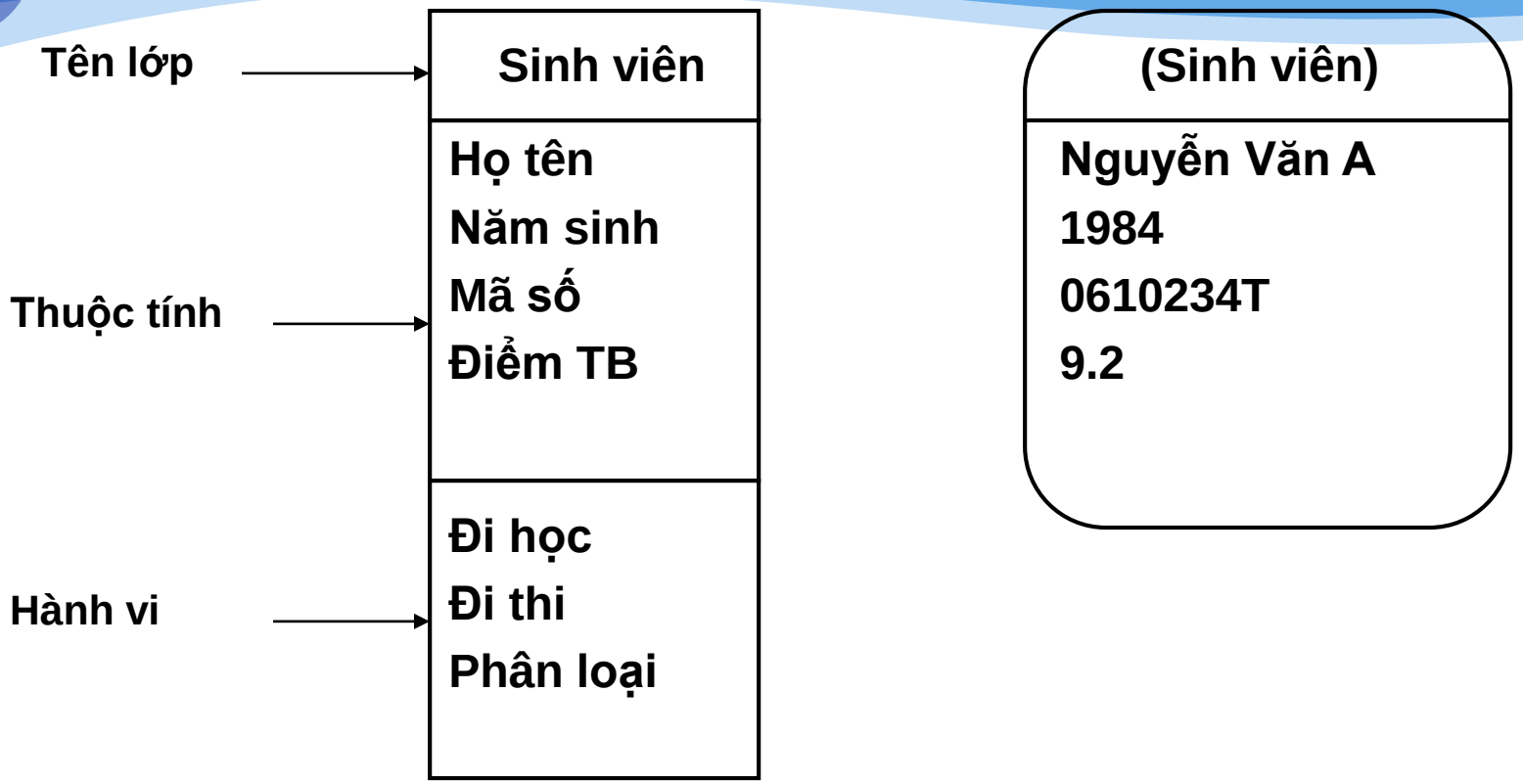
Interacting Objects



Sơ đồ đối tượng

- Ta dùng sơ đồ đối tượng để mô tả các lớp đối tượng. Sơ đồ đối tượng bao gồm sơ đồ lớp và sơ đồ thể hiện.
- Sơ đồ lớp mô tả các lớp đối tượng trong hệ thống, một lớp đối tượng được diễn tả bằng một hình chữ nhật gồm 3 phần:
  - Phần đầu chỉ tên lớp
  - Phần 2 mô tả các thuộc tính
  - Phần 3 mô tả các thao tác của các đối tượng trong lớp

Sơ đồ lớp và sơ đồ thể hiện



Sơ đồ lớp

Sơ đồ thể hiện

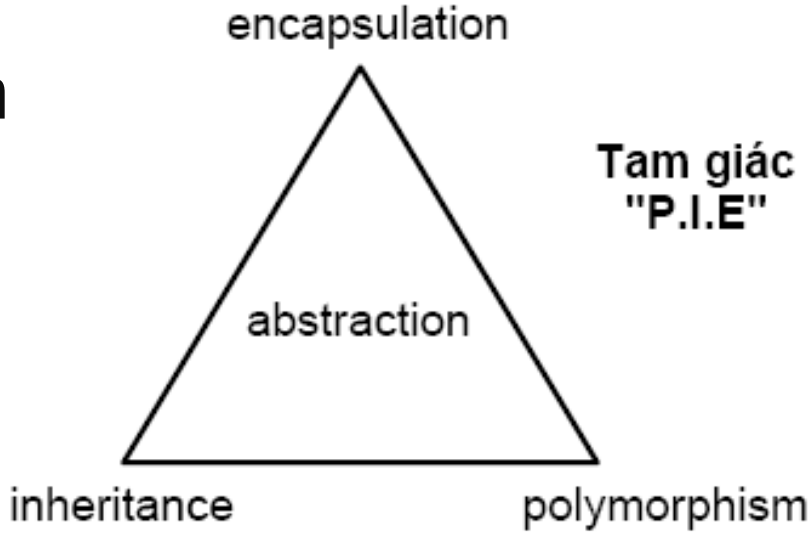
Đối tượng = Dữ liệu + Phương thức

Thiết kế theo hướng đối tượng

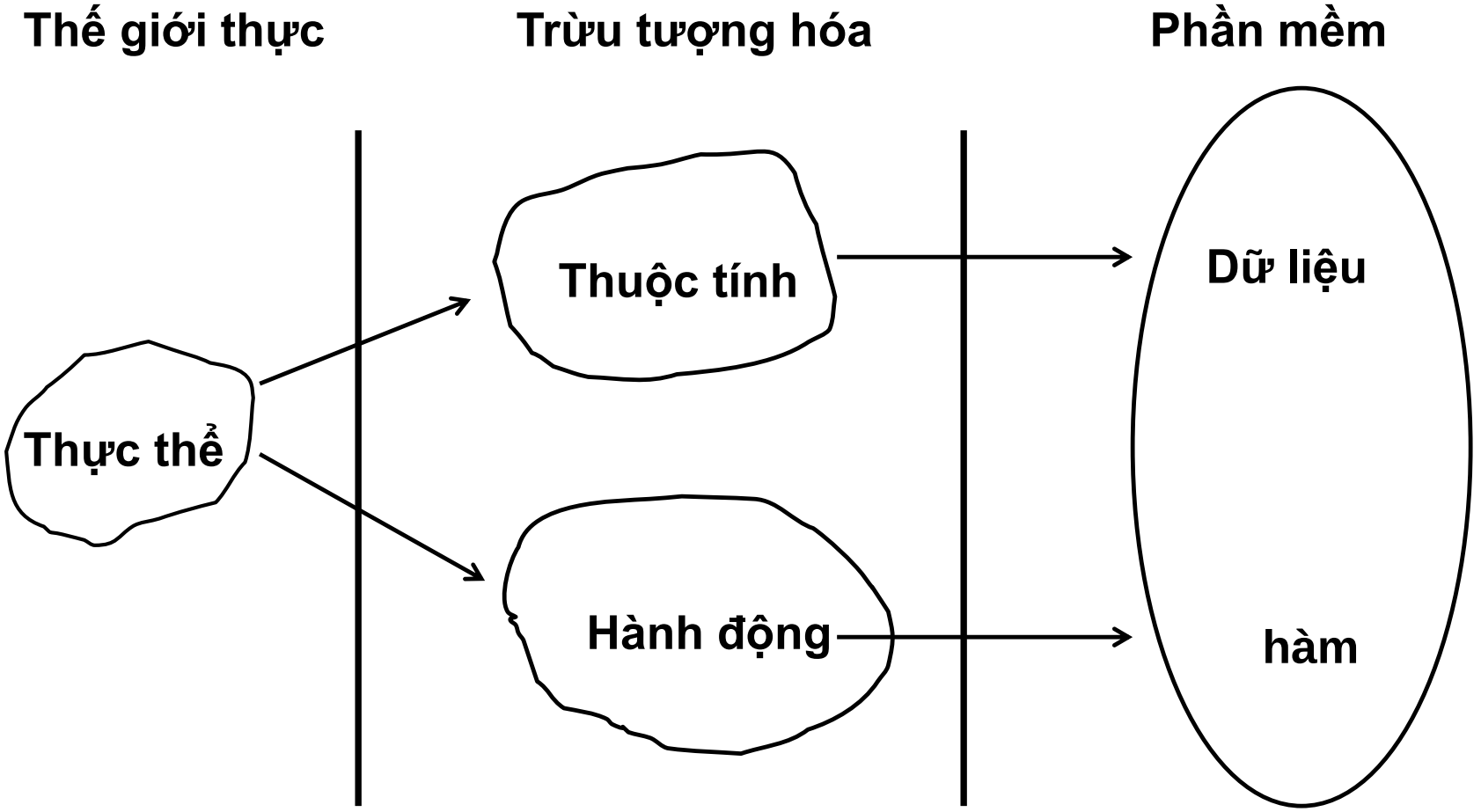
- Trừu tượng hóa dữ liệu và các hàm/thủ tục liên quan.
- Chia hệ thống ra thành các lớp/đối tượng.
- Mỗi lớp/đối tượng có các tính năng và hành động chuyên biệt.
- Các lớp có thể được sử dụng để tạo ra nhiều đối tượng cụ thể.

Các đặc điểm quan trọng của OOP

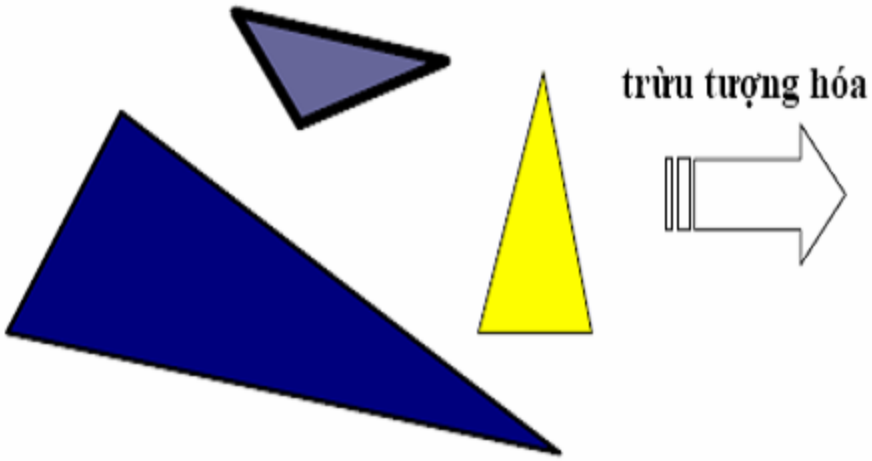
- Các lớp đối tượng - Classes
- Đóng gói - Encapsulation
- Thừa kế - Inheritance
- Đa hình - Polymorphism



Trừu tượng hóa



Trừu tượng hóa



các đối tượng

trừu tượng hóa

Tam giác:

cạnh1, cạnh2, cạnh3

mầu nền, mầu biên,

độ đậm biên

vẽ, tính diện tích, tính chu vi

lớp/kiểu dữ liệu

- Cách nhìn khái quát hóa về một tập các đối tượng có chung các đặc điểm được quan tâm (và bỏ qua những chi tiết không cần thiết).

Đóng gói – Che dấu thông tin

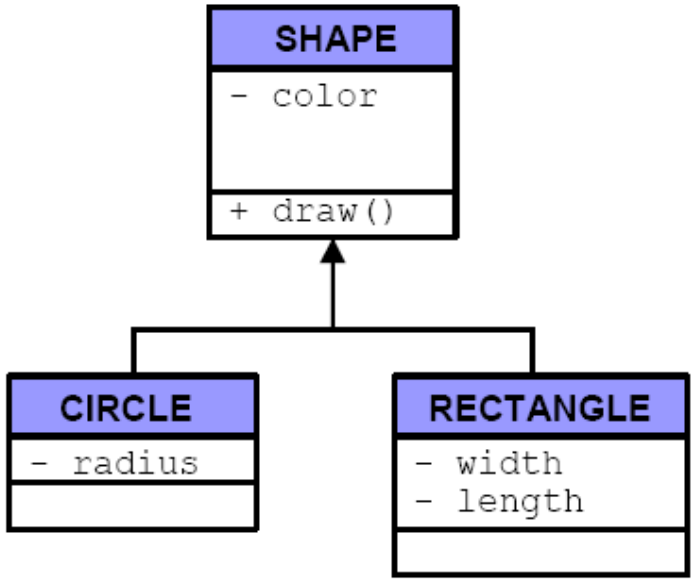
- Đóng gói: Nhóm những gì có liên quan với nhau vào làm một, để sau này có thể dùng một cái tên để gọi đến
  - Các hàm/ thủ tục đóng gói các câu lệnh
  - Các đối tượng đóng gói dữ liệu của chúng và các thủ tục có liên quan

Đóng gói – Che dấu thông tin

- Che dấu thông tin: đóng gói để che một số thông tin và chi tiết cài đặt nội bộ để bên ngoài không nhìn thấy
  - Che giấu những gì mà người dùng không cần.
  - Che giấu những gì mà mình cần giữ bí mật.

Thừa kế

- Là cơ chế cho phép một lớp D có được các thuộc tính và thao tác của lớp C, như thể các thuộc tính và thao tác đó đã được định nghĩa tại lớp D.
- Cho phép cài đặt nhiều đối tượng:
  - Đặc biệt hóa (“là”)
  - Khái quát hóa



Đa hình

- Là cơ chế cho phép một tên thao tác hoặc thuộc tính có thể được định nghĩa tại nhiều lớp và có thể có nhiều cài đặt khác nhau tại mỗi lớp trong các lớp đó.

Các ưu điểm của OOP

- Nguyên lý kế thừa: tránh lặp, tái sử dụng.
- Nguyên lý đóng gói – che dấu thông tin: chương trình an toàn không bị thay đổi bởi những đoạn chương trình khác
- Dễ mở rộng, nâng cấp
- Mô phỏng thế giới thực tốt hơn.

Các đặc tính chính của OOP

- Chương trình được chia thành các đối tượng.
- Các cấu trúc dữ liệu được thiết kế sao cho đặc tả được đối tượng.
- Các hàm thao tác trên các vùng dữ liệu của đối tượng được gắn với cấu trúc dữ liệu đó.

Các đặc tính chính của OOP

- Dữ liệu được đóng gói lại, được che giấu và không cho phép các hàm ngoại lai truy nhập tự do.
- Các đối tượng tác động và trao đổi thông tin với nhau qua các hàm.
- Có thể dễ dàng bổ sung dữ liệu và các hàm mới vào đối tượng nào đó khi cần thiết.
- Chương trình được thiết kế theo cách tiếp cận từ dưới lên (bottom-up).

Một số thuật ngữ OOP

- OOM (Object Oriented Methodology): Phương pháp luận hướng đối tượng
- OOA (Object Oriented Analysis): Phân tích hướng đối tượng.
- OOD: Object Oriented Design (Thiết kế hướng đối tượng).
- OOP: Object Oriented Programming (LTHĐT).
- Inheritance: Kế thừa
- Polymorphism: Đa hình
- Encapsulation: Tính đóng gói.

Ngôn ngữ OOP

- Cung cấp được những khả năng lập trình hướng đối tượng.
  - Cung cấp khả năng kiểm soát truy cập
  - Kế thừa
  - Đa hình

Q & A