# Bài l Tổng quan Lập trình hướng đối tượng

Trịnh Thành Trung

trungtt@soict.hust.edu.vn

# Nội dung

- 1. Kỹ thuật lập trình
- 2. Kỹ thuật hướng đối tượng
- 3. Các khái niệm cơ bản
- 4. Các nguyên lý
- 5. Phân tích thiết kế hướng đối tượng

1

# Kỹ thuật lập trình

Lập trình hướng đối tượng là một kỹ thuật lập trình. Vậy kỹ thuật lập trình là gì?

## Kỹ thuật lập trình

- "Lập trình hướng đối tượng" là một kỹ thuật lập trình. Vậy "kỹ thuật lập trình" là gì?
- Kỹ thuật lập trình: <u>Kỹ thuật</u> thực thi một giải pháp phần mềm (cấu trúc dữ liệu + giải thuật) dựa trên nền tảng một <u>phương pháp luận</u> (methodology) và một hoặc nhiều <u>ngôn ngữ lập trình</u> phù hợp với yêu cầu đặc thù của ứng dụng.

## Kỹ thuật lập trình

"Phương pháp luận"

- Các mô thức lập trình
- Các ý tưởng, thuật toán để giải quyết vấn đề
- Phong cách trình bày trong lập trình
- Văn hóa lập trình

### Kỹ thuật lập trình

 Mô thức - nguyên tắc chung cơ bản

"Ngôn ngữ lập trình"

- Cú pháp xác định cái gì là hợp lệ trong mã nguồn
- Ngữ nghĩa ngữ pháp của ngôn ngữ lập trình

# Ngôn ngữ lập trình

- Ngôn ngữ?
  - Phương tiện để giao tiếp
  - Hệ thống ký hiệu để diễn đạt



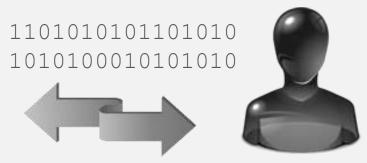




### Ngôn ngữ lập trình

- Ngôn ngữ máy
  - Các chỉ thị được thể hiện bằng các chữ số nhị phân 1 và
     0.





### Ngôn ngữ lập trình

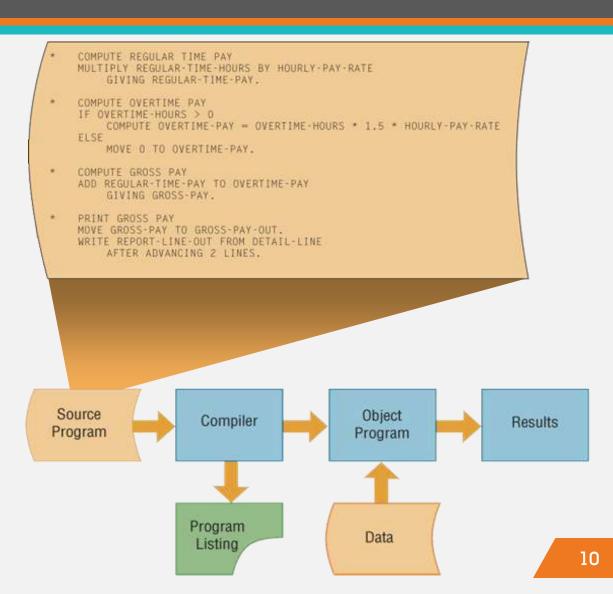
- Ngôn ngữ lập trình
  - Là ngôn ngữ được chuẩn hóa
  - Cả con người và máy tính có thể đọc và hiểu được
  - Sử dụng chương trình dịch tương ứng để giao tiếp với máy tính

```
tuning of the control of the control
```

#### Trình dịch

Trình dịch -Compiler

Là chương trình thực hiện biên dịch toàn bộ chương trình nguồn thành mã máy trước khi thực hiện



# Sự phát triển của ngôn ngữ lập trình



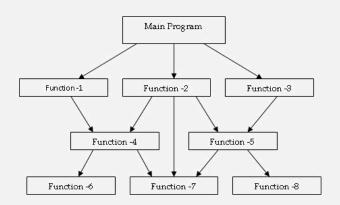
# Sự phát triển của ngôn ngữ lập trình

- Lập trình tuần tự
  - Mã nguồn được viết dưới dạng "đầu tiên làm thế này, sau đó làm thế kia"
  - Khó sửa lỗi, bảo trì.

```
; defining reload constant for baudrate generation
        RELOAD EOU 0E6H
        ORG 0000H
                          ; org directive
        SJMP START
                          ; jump to main program
        ORG 0023H
SENDCH:
                        ; ISR for RI/TI interrupt
        CLR TI
                         ; clear transmit flag
        MOV SBUF, #'A'
                        ; send the ASCII value of 'A'
        RETI
                          ; return back to main program
START:
        ANL PCON, #7FH
                        ; Set SMOD=0
                        ; Alter only the setting of Timer-1
        ANL TMOD, #0FH
        ORL TMOD, #20H ; Timer-1 in mode-2
        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
```

# Sự phát triển của ngôn ngữ lập trình

- Lập trình hướng thủ tục (Lập trình cấu trúc)
  - Xây dựng chương trình dựa trên các hàm/thủ tục/chương trình con
  - Dữ liệu và xử lý (hàm) tách rời nhau
  - Các hàm không bắt buộc phải tuân theo một cách thức chung truy cập vào dữ liệu



## Trừu tượng hóa

- Sự phát triển của các ngôn ngữ lập trình chính là sự phát triển của quá trình <u>trừu tượng hóa</u> (abstraction)
  - Assembly: Trừu tượng hóa kiểu dữ liệu/chỉ thị cơ bản
     MOV TH1, #RELOAD
  - Ngôn ngữ cấu trúc: Trừu tượng hóa điều khiển (control abstraction) + trừu tượng hóa chức năng (functional abstraction)

# 2

# Kỹ thuật hướng đối tượng

Giới thiệu về công nghệ đối tượng và kỹ thuật hướng đối tượng

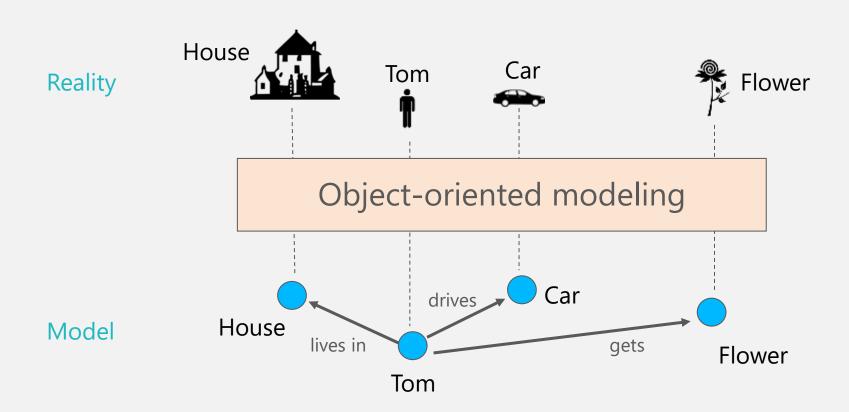
• Đối tượng là gì?

"Tất cả mọi thứ đều là đối tượng" -Alan Kay



# Lập trình hướng đối tượng

 Coi chương trình phần mềm là một tập hợp các đối tượng tương tác với nhau



## Lập trình hướng đối tượng

- Mỗi đối tượng trong chương trình có các dữ liệu độc lập của mình và chiếm bộ nhớ riêng của mình.
- Mỗi đối tượng đều có dạng đặc trưng của lớp các đối tượng đó.
- Tất cả các đối tượng thuộc về cùng một lớp đều có các hành vi giống nhau

### Trừu tượng hóa

- Sự phát triển của các ngôn ngữ lập trình chính là sự phát triển của quá trình <u>trừu tượng hóa</u> (abstraction)
- Lập trình hướng đối tượng
  - Trừu tượng hóa dữ liệu (data abstraction)
  - Coi các dữ liệu trong chương trình là các đối tượng

# 3

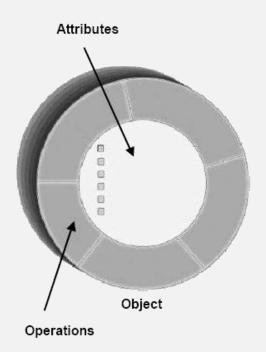
# Các khái niệm

Các khái niệm cơ bản trong lập trình hướng đối tượng

- Trong thế giới thực
  - VD: Sinh viên, ô tô, màu sắc...
- Mỗi đối tượng đều có:
  - Các thông tin, trạng thái
    - Ví dụ: Các thông tin về ô tô có màu sắc, tốc độ, năm sx...
  - Các hoạt động
    - Ví dụ: Tăng ga, phanh, giảm tốc, đi đến một địa điểm...

	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	ESPIEXPIESS Teller International

- Đối tượng là duy nhất. Không có hai đối tượng giống nhau dù cùng chia sẻ các tính chất, trạng thái
- Ví dụ
  - Sinh viên A và Màu tím
  - Sinh viên A và Con khủng long
  - Sinh viên A và Bill Gates
  - Sinh viên A và Sinh viên B



- Mô hình hóa vào trong lập trình: Đóng gói thành trạng thái (state) và hành vi (behavior).
  - Trạng thái được biểu diễn bởi các thuộc tính (attributes) và các mối quan hệ (relationships).
  - Hành vi được biểu diễn bởi các thao tác (operations) hay phương thức (methods).

### Trạng thái và hành vi

Trạng thái

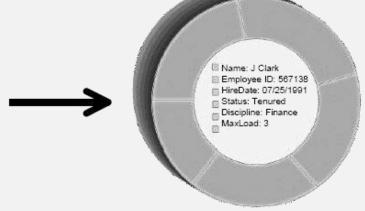
- Trạng thái của một đối tượng là một trong các điều kiện tại đó mà đối tượng tồn tại.
- Trạng thái của một đối tượng có thể thay đổi theo thời gian.



Name: J Clark Employee ID: 567138 Date Hired: July 25, 1991

Status: Tenured Discipline: Finance

Maximum Course Load: 3 classes



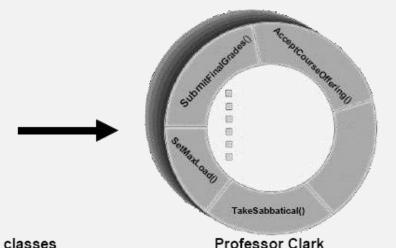
### Trạng thái và hành vi

Hành vi

- Hành vi quyết định đối tượng đó hành động và đáp trả như thế nào đối với bên ngoài.
- Hành vi nhìn thấy được của một đối tượng được mô hình thành một tập các thông điệp nó có thể đáp trả (các thao tác mà đối tượng đó thực hiện).

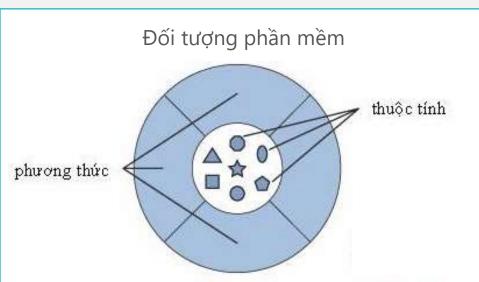


Professor Clark's behavior
Submit Final Grades
Accept Course Offering
Take Sabbatical
Maximum Course Load: 3 classes

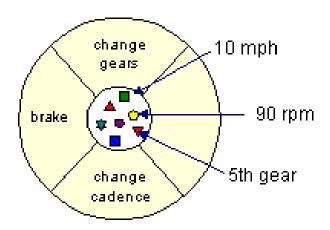


# Đối tượng phần mềm

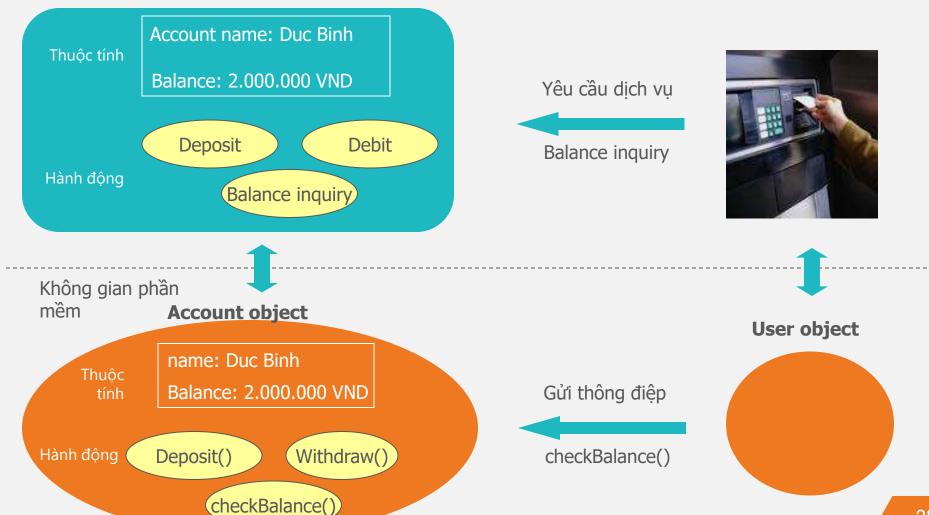
- Đối tượng (object) là một thực thể phần mềm bao bọc các thuộc tính và các phương thức liên quan.
  - Thuộc tính được xác định bởi giá trị cụ thể gọi là thuộc tính thể hiện.
  - Một đối tượng cụ thể được gọi là một thể hiện.



Đối tượng phần mềm Xe Đạp



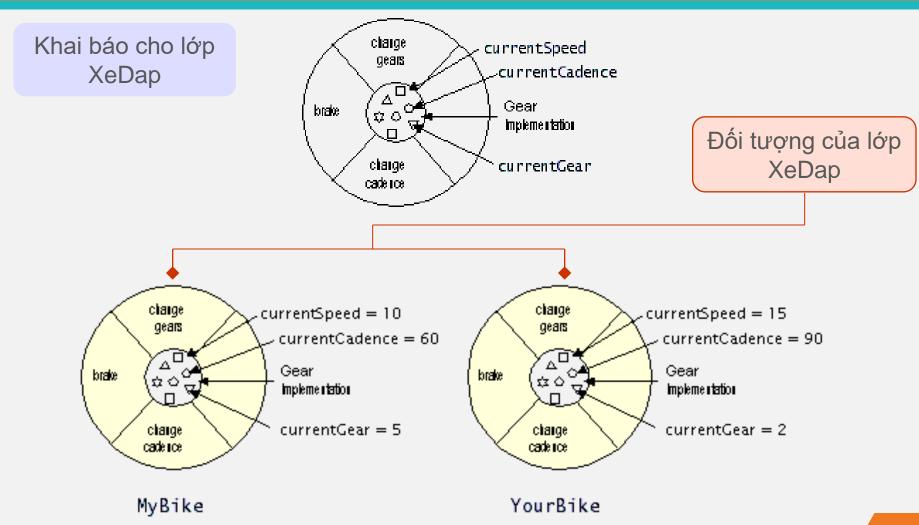
### Bài toán thực tiễn Quản lý tài khoản ngân hàng



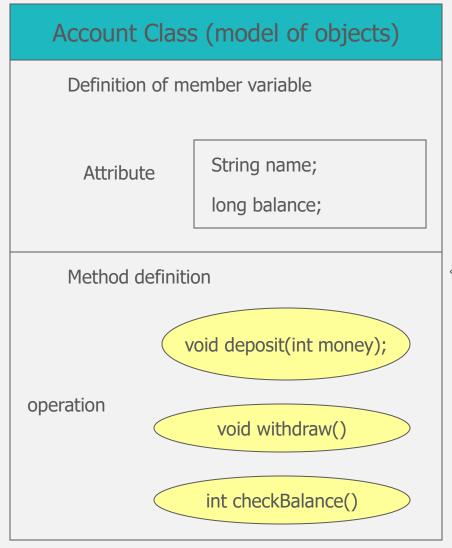
### Lóp

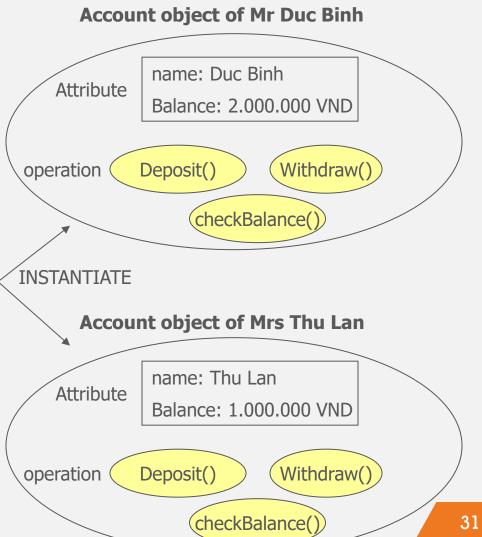
- Một lớp là một thiết kế (blueprint) hay mẫu (prototype) cho các đối tượng cùng kiểu
  - Ví dụ: lớp XeDap là một thiết kế chung cho nhiều đối tượng xe đạp được tạo ra
- Lớp định nghĩa các thuộc tính và các phương thức chung cho tất cả các đối tượng của cùng một loại nào đó
- Một đối tượng là một thể hiện cụ thể của một lớp.
  - Ví dụ: mỗi đối tượng xe đạp là một thể hiện của lớp XeDap
- Mỗi thể hiện có thể có những thuộc tính thể hiện khác nhau
  - Ví dụ: một xe đạp có thể đang ở bánh răng thứ 5 trong khi một xe khác có thể là đang ở bánh răng thứ 3.

## Ví dụ Lớp Xe đạp



# Lớp và Đối tượng



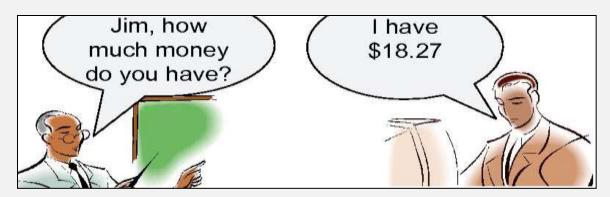


# Lớp và đối tượng



### Tương tác giữa các đối tượng

 Các đối tượng trong thế giới thực có thể tương tác được với nhau



- Trong lập trình
  - Các đối tượng giao tiếp với nhau bằng cách gửi thông điệp (message)
     Tương tự với gọi hàm?

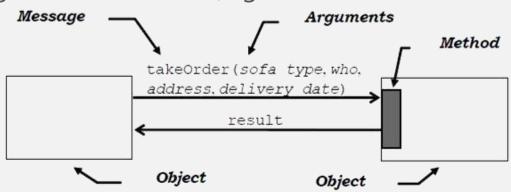


## Gọi hàm vs. Gửi thông điệp

- Gọi hàm (call function)
  - Chỉ ra chính xác đoạn mã nào sẽ được thực hiện.
  - Chỉ có duy nhất một sự thực thi của một hàm với một tên nào đó.
  - Không có hai hàm trùng tên
- Gửi thông điệp (send message)
  - Yêu cầu một dịch vụ từ một đối tượng và đối tượng sẽ quyết định cần phải làm gì
  - Các đối tượng khác nhau sẽ có các cách thực thi các thông điệp theo cách khác nhau.

## Thông điệp vs. Phương thức

- Thông điệp
  - Được gửi từ đối tượng này đến đối tượng kia, không bao gồm đoạn mã thực sự sẽ được thực thi
- Phương thức
  - Thủ tục/hàm trong ngôn ngữ lập trình cấu trúc
  - Là sự thực thi dịch vụ được yêu cầu bởi thông điệp
  - Là đoạn mã sẽ được thực thi để đáp ứng thông điệp được gửi đến cho đối tượng



4

# Các nguyên lý cơ bản

Trừu tượng hóa, đóng gói, module hóa, phân cấp.

## Các nguyên lý cơ bản

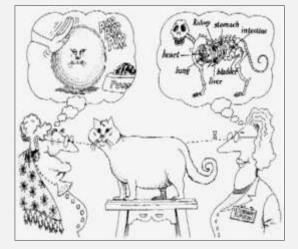


## Trừu tượng hóa

- Trừu tượng hóa
  - Là quá trình loại bỏ đi các thông tin cụ thể và giữ lại những thông tin chung.
  - Tập trung vào các đặc điểm cơ bản của thực thể, các đặc điểm phân biệt nó với các loại thực thể khác.
  - Phụ thuộc vào góc nhìn

Quan trọng trong ngữ cảnh này nhưng lại không có ý nghĩa

nhiều trong ngữ cảnh khác.

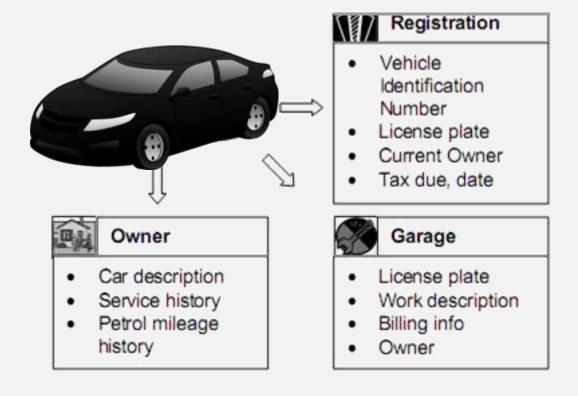


# Ví dụ

• Bài toán Quản lý sinh viên

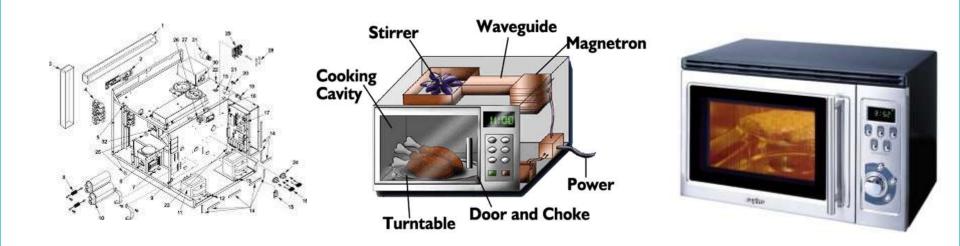
## Trừu tượng hóa

Phụ thuộc vào góc nhìn:



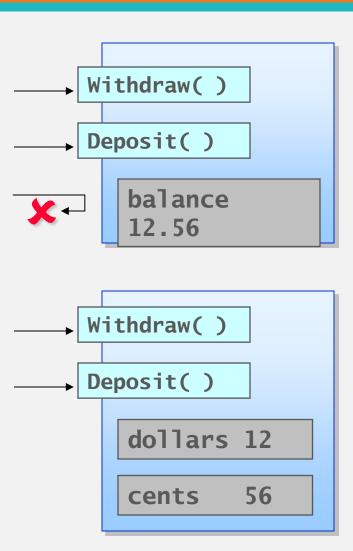
#### Đóng gói

- Đóng gói
  - Che giấu, ẩn đi chi tiết thực hiện bên trong
  - Cung cấp cho thế giới bên ngoài một giao diện
  - Người dùng không phụ thuộc vào việc sửa đổi sự thực thi bên trong



#### Đóng gói

- Cho phép điều khiển
  - Việc sử dụng đối tượng được kiểm soát thông qua các method public
- Hỗ trợ sự thay đổi
  - Việc sử dụng đối tượng không bị ảnh hưởng nếu dữ liệu nội tại (private) bị thay đổi



#### Module hóa

- Module hóa
  - Chia nhỏ hệ thống phức tạp thành những thành phần nhỏ có thể quản lý được.
  - Cho phép người dùng hiểu được về hệ thống.

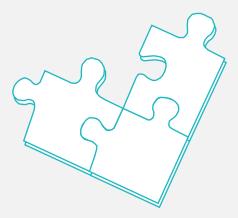


#### Module hóa

 Chia nhỏ một hệ thống phức tạp thành các mô đun nhỏ hơn.



Hệ thống quản lý xuất nhập sách



Hệ thống quản lý siêu thị sách



Hệ thống kế toán

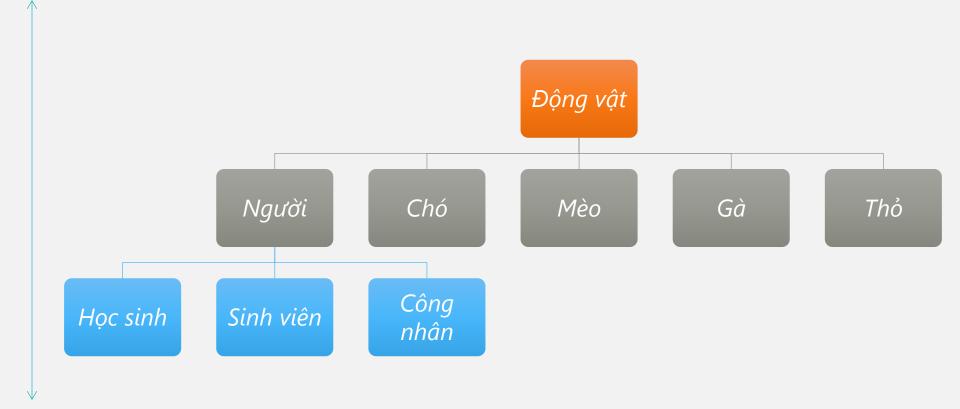
Hệ thống quản lý nhân viên

## Phân cấp

- Phân cấp
  - Xếp hạng hay xếp thứ tự các mức trừu tượng vào một cấu trúc cây
  - Tổ chức để phân loại. Sử dụng phân cấp rất dễ dàng nhận ra sử giống và khác nhau giữa các đối tượng

# Phân cấp

Tăng mức độ trừu tượng hóa



Giảm mức độ trừu tượng hóa

# 5

# Phân tích thiết kế hướng đối tượng

Object Oriented Analysis and Design (OOAD)

## Phân tích thiết kế hướng đối tượng

- Phương pháp luận (methodology) trong PT&TK phần mềm thông thường được định nghĩa như là một tập các quá trình và thao tác để tìm và khám phá cách có thể giải quyết được bài toán phần mềm.
- Một trong các phương pháp hiệu quả nhất để phát triển phần mềm.

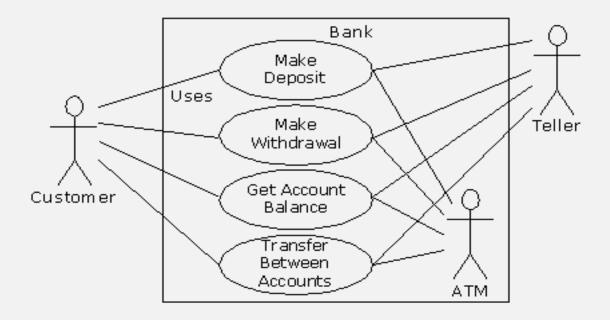
# Phát triển phần mềm

- Sáu giai đoạn
  - Giai đoạn 0: Lập kế hoạch (make a plan)
  - Giai đoạn 1: Xác định mục tiêu làm gì (what are we making)
  - Giai đoạn 2: Xác định cách làm thế nào (how to build it)
  - Giai đoạn 3: Xây dựng phần lõi Building the core
  - Giai đoạn 4: Lặp lại (hiệu chỉnh) các trường hợp sử dụng
  - Giai đoạn 5: Phát triển (evolution)

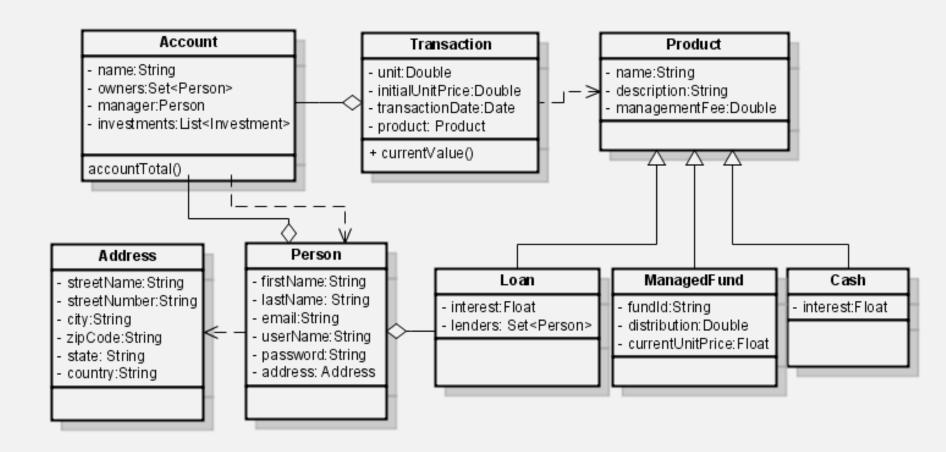
#### Xác định mục tiêu

- Giai đoạn 1: Xác định mục tiêu làm gì (what are we making)
- Trong giai đoạn này chúng ta có nhiệm vụ xác định cụ thể các mục tiêu, chức năng và nhiệm vụ mà phần mềm chúng ta cần xây dựng phải đáp ứng.
- Trong phương pháp lập trình cổ điển hướng thủ tục người ta gọi giai đoạn này là giai đoạn tạo ra "phân tích yêu câu và mô tả hệ thống" (requirements analysis and system specification).
- Trong PT&TK hướng đối tượng người ta sử dụng các ký pháp và kỹ thuật Use case để mô tả các công việc này.

# Biểu đồ use case



# Biểu đồ lớp



# Thank you!

Any questions?