

Phân tích thiết kế hướng đối tượng

Bài 8: Đối tượng và lớp

TS. Nguyễn Hiếu Cường

Bộ môn CNPM, Khoa CNTT, Trường ĐH GTVT

Email: cuonggt@gmail.com

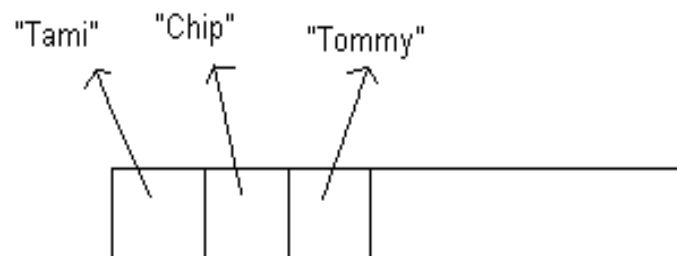
Phân tích hệ thống

- Mô hình ca sử dụng
 - Hệ thống cần làm gì dưới góc nhìn của người dùng
 - Các chức năng được yêu cầu của hệ thống
- Phần mềm gồm các đối tượng tương tác với nhau
- Xác định các đối tượng?

Ví dụ

- Suzy có:

- Một *chuồng* gồm nhiều *ngăn* để nuôi các con vật làm cảnh, mỗi ngăn chỉ nuôi 1 con, mỗi con vật đều có tên
- Nhiều loại con vật khác nhau



- Suzy cần:

- Xây dựng một chương trình để quản lý
 - Chương trình có nhiệm vụ gì?
 - *Xác định danh tính* con vật trong từng ngăn: tên gì, loại gì?
 - *Thêm* con vật mới mới vào chuồng, *lấy* một con vật ra khỏi chuồng
 - Có thể sửa chữa, mở rộng chương trình một cách dễ dàng
-

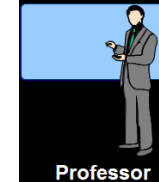
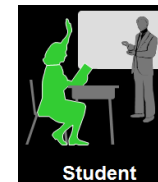
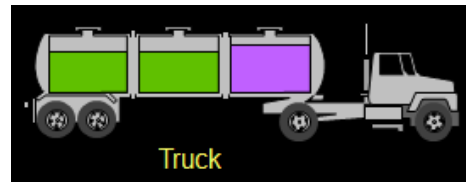
Phân tích

- Có những đối tượng gì trong bài toán?
 - **Các con vật** (giả thiết Suzi chỉ nuôi mèo và chó)
 - **Chuồng** (gồm nhiều ngăn, mỗi ngăn nuôi một con)
 - Các lớp cần có?
 - **Animal** Mô tả các đặc tính chung của các loại con vật
 - **Cat** Mô tả loại đối tượng con mèo, kế thừa lớp Animal
 - **Dog** Mô tả loại đối tượng con chó, kế thừa lớp Animal
 - **Kennel** Chuồng nuôi các con vật, gồm nhiều ngăn
-

Đối tượng

- Thế giới thực bao gồm các *đối tượng* (object)!

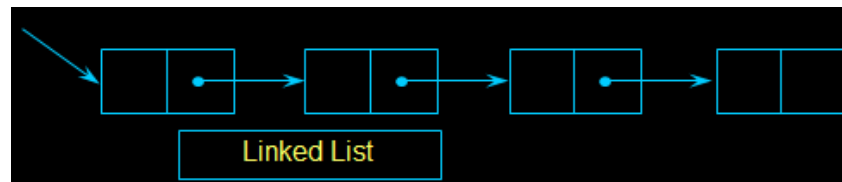
- Đối tượng vật lý



- Đối tượng khái niệm



- Đối tượng phần mềm



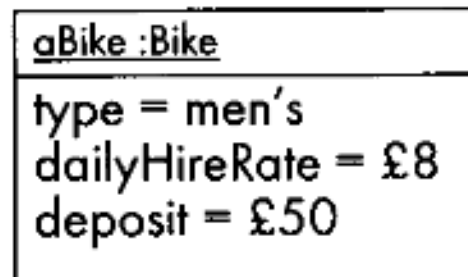
- Mỗi đối tượng gồm các *thuộc tính* và các *thao tác*.

Lớp

- Lớp (class) là một định nghĩa trừu tượng (abstract definition) của các đối tượng có cùng những đặc tính chung
- Đối tượng (object) là thể hiện cụ thể (instance) của một lớp
- Tác dụng của lớp?
 - Trừu tượng hoá dữ liệu (data abstraction)
 - Bao gói (encapsulation): dữ liệu + thao tác
 - Che giấu thông tin (information hiding)
- Liên hệ với các khái niệm đã biết: Kiểu dữ liệu và Biến?

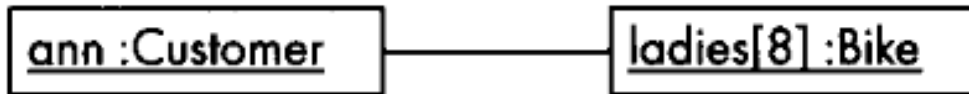
Đối tượng

- Đối tượng là một thực thể độc lập
- Một đối tượng gồm những gì?
 - Định danh (identity): mỗi đối tượng là duy nhất trong bộ nhớ
 - Trạng thái (state): định hình bởi giá trị các thuộc tính của đối tượng
 - Ứng xử (behaviour): thể hiện bởi các hành động có thể của đối tượng

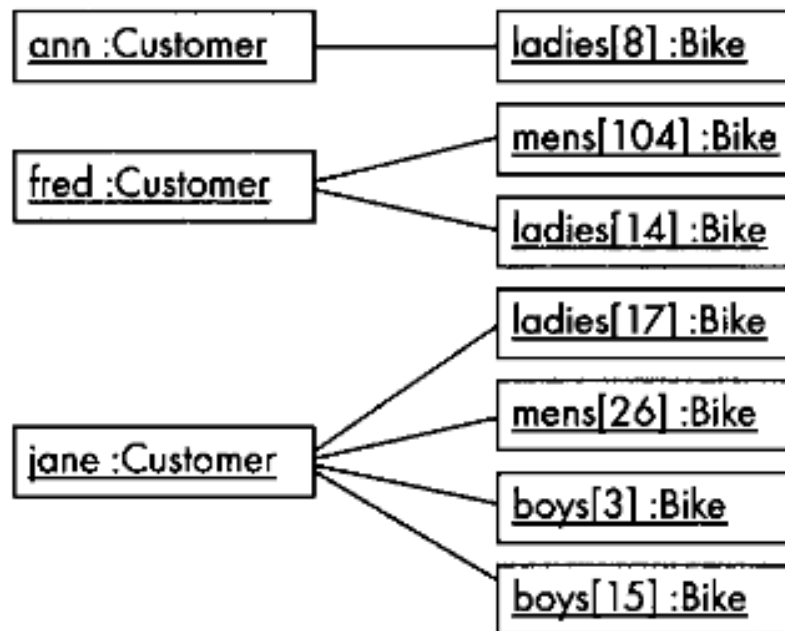


Sự phụ thuộc

- Hai thành phần là phụ thuộc vào nhau nếu có một thành phần sử dụng các dịch vụ cung cấp bởi thành phần kia
- Ví dụ: Khách hàng tên **ann** có thể thuê/trả xe đạp

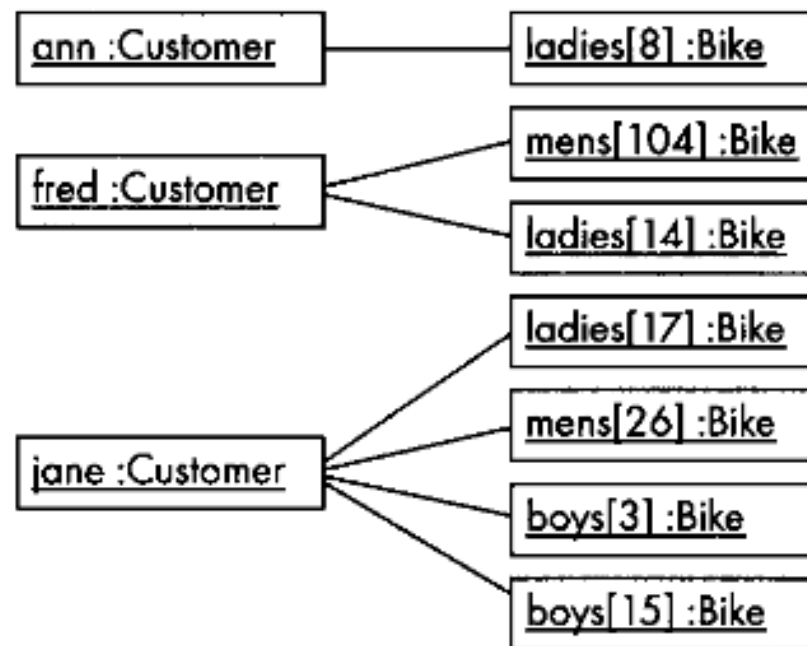


Biểu đồ đối tượng



Ưu nhược điểm của Biểu đồ đối tượng

- Ưu điểm?
 - Mô tả quan hệ giữa các đối tượng
 - Làm rõ hơn các hệ thống phức tạp
- Nhược điểm?



Biểu đồ lớp



Liên kết giữa các lớp

- Trong một hệ thống, các đối tượng có thể tương tác với nhau
- Giữa lớp A và lớp B có sự liên kết để thể hiện đối tượng của lớp A có thể tương tác được với đối tượng của lớp B



Các kiểu liên kết giữa các lớp

- Tác dụng của liên kết giữa hai lớp?
 - Thể hiện sự tương tác có thể giữa các đối tượng của các lớp đó
- Có nhiều loại liên kết giữa các lớp
 - Kết hợp (association)
 - Kết tập (aggregation)
 - Gộp (composition)
 - Kế thừa (inheritance)

Kết hợp (Association)

- Các đối tượng của hai lớp có thể tương tác với nhau (thông qua truyền thông điệp – message passing)
- Liên kết có thể được ghi rõ
- Đôi khi có thể ghi rõ vai trò (role name) ở mỗi đầu liên kết



Liên kết bội (Multiplicity)

- Xác định giới hạn số đối tượng có thể tham gia vào liên kết

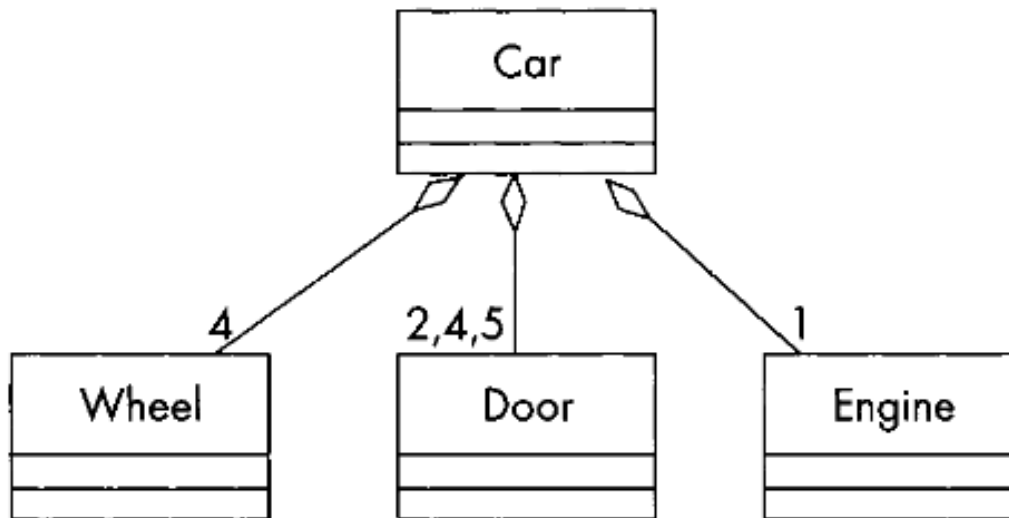


<i>Meaning</i>	<i>Example</i>	<i>Notation</i>
an exact number	exactly one exactly six	1 (or may be omitted) 6
many	zero or more one or more, lots of	0..* 1..*, *
a specific range	one to four, zero to six	1..4, 0..6,
a choice	two or four or five	2, 4, 5

Kết tập (Aggregation)

- Thể hiện quan hệ tổng thể - thành phần (whole-part)

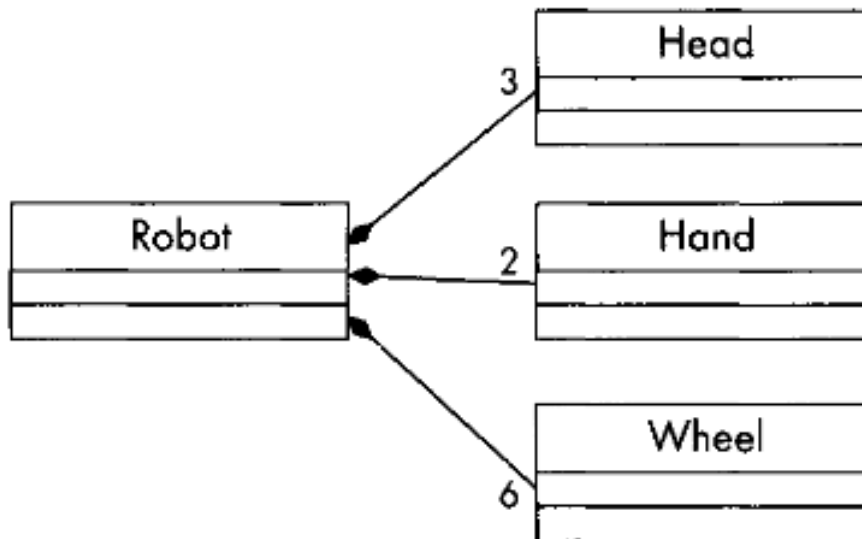
Trong mô tả thường có các cụm từ: “gồm có” (consist of), “có một” (has a), “là một phần của” (is a part of)



Gộp (Composition)

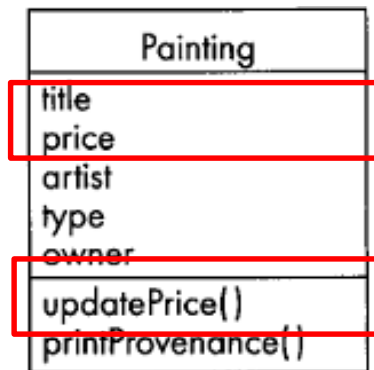
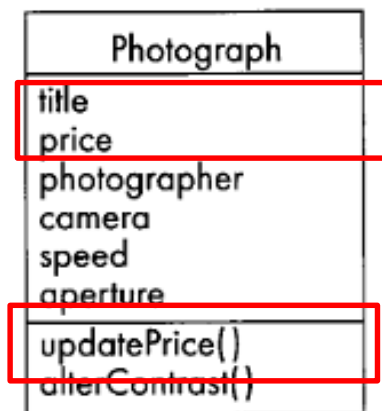
- Là một dạng quan hệ tổng thể - thành phần

Mạnh hơn quan hệ kết tập, trong đó các thành phần không tồn tại riêng rẽ với tổng thể



Kế thừa

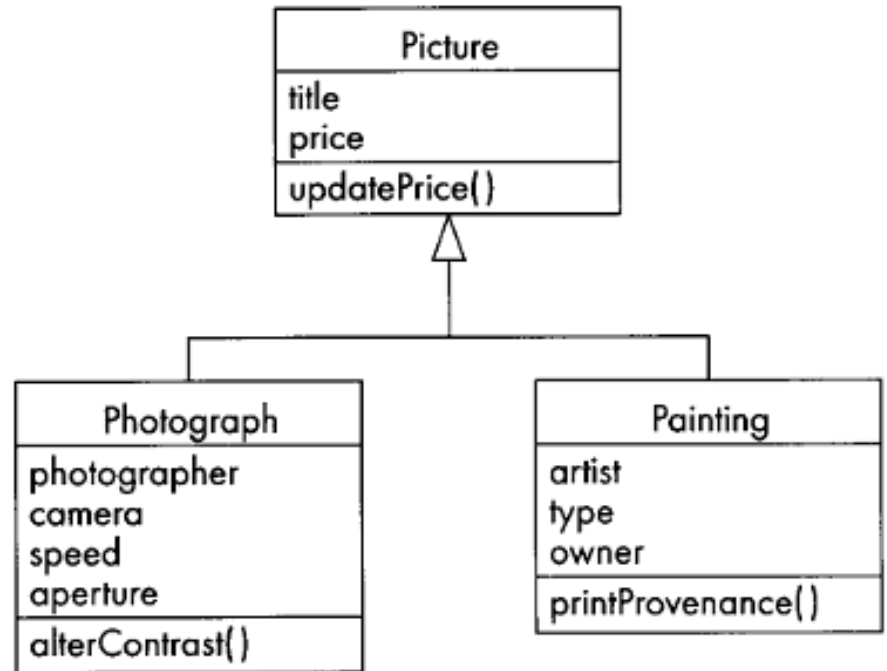
- Là quan hệ “is a”, hoặc “is a kind of”



Generalization

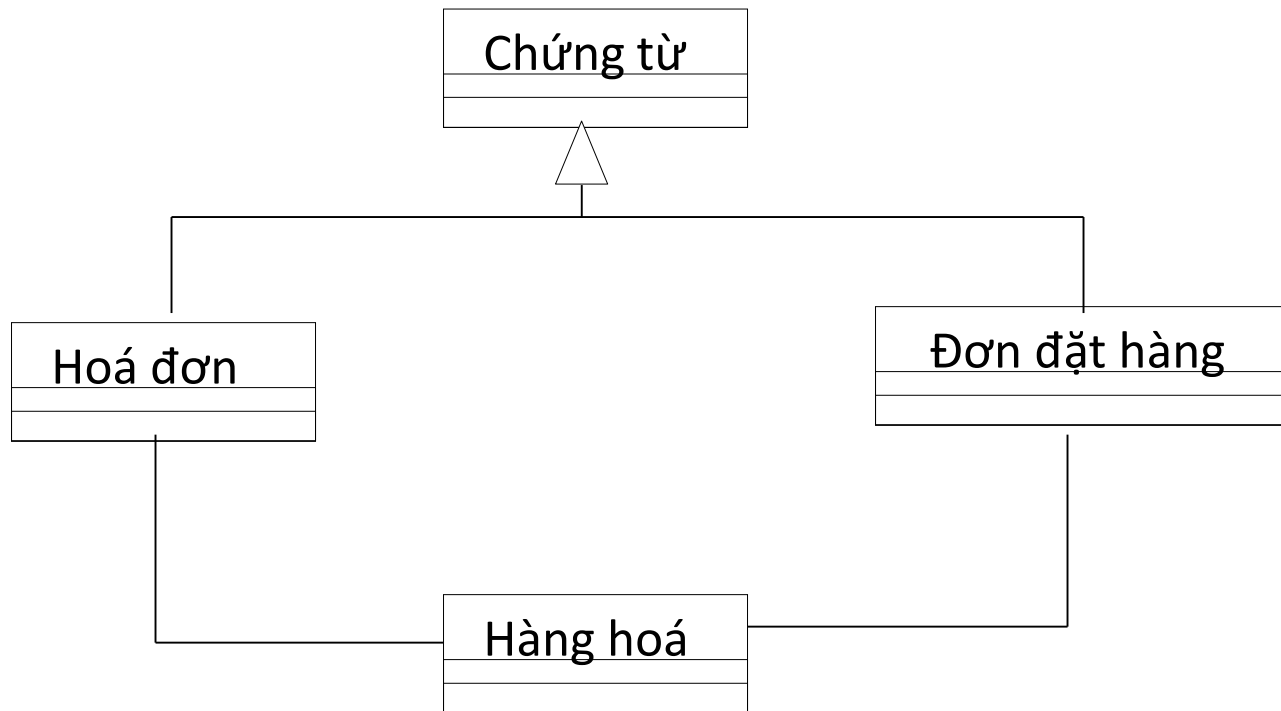


Specialization



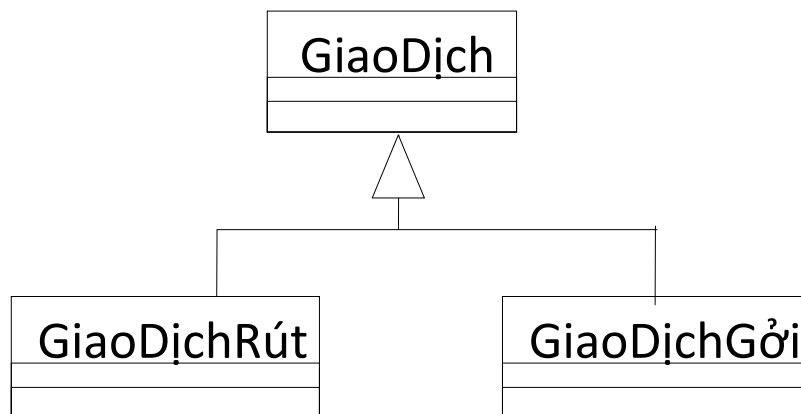
Ví dụ

Tiếp cận bottom-up (generalization)



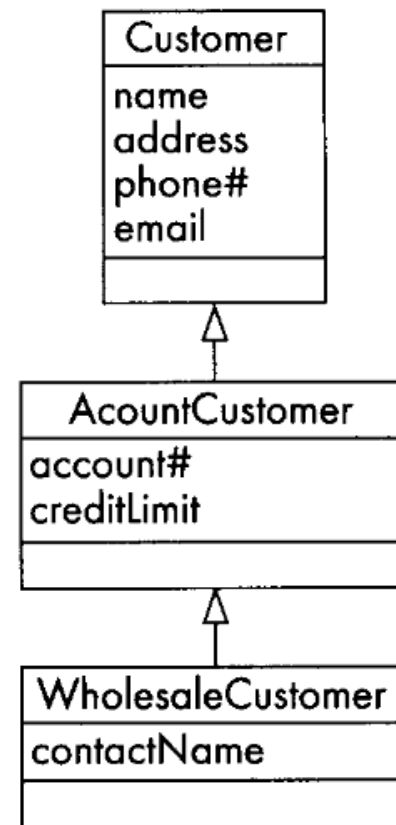
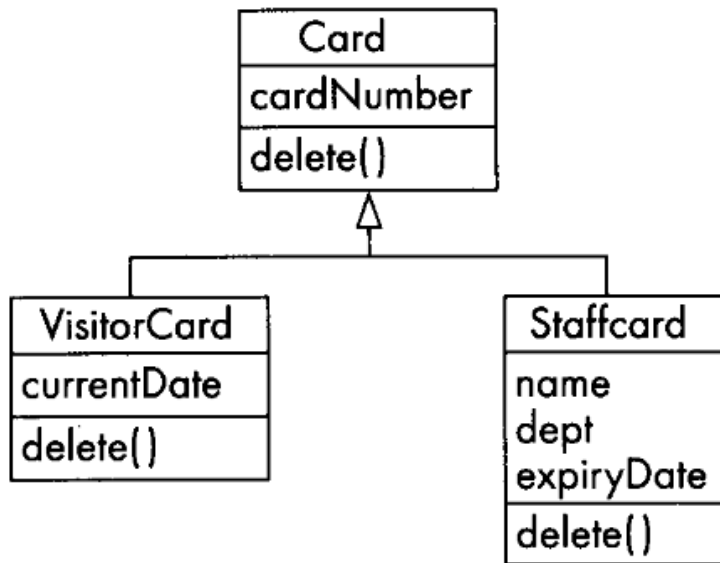
Ví dụ

Tiếp cận top-down (inheritance)



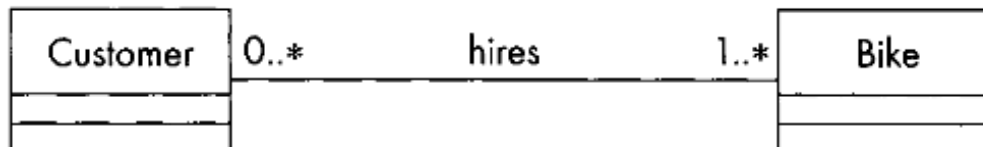
Ví dụ

- Trong biểu đồ sau, thuộc tính của các đối tượng Staffcard và WholesaleCustomer gồm những gì?



Lớp kết hợp

- Lớp kết hợp (association class) cho phép thêm các thuộc tính (attributes) vào liên kết
- Ví dụ



- Có nhận xét gì về liên kết “hires”?
- Liên kết “hires” có thể bao gồm nhiều thuộc tính (startDate, endDate)
- Trong trường hợp này nên chuyển nó thành một *lớp kết hợp*

Thế nào là một lớp tốt?

1. *Problem domain*: Trong quá trình phân tích, các lớp nên phản ánh đúng các đối tượng trong phạm vi của bài toán
2. *Functionality*: Một lớp cần có cả dữ liệu và các hành vi (ít nhất là trong quá trình phân tích). Nếu một lớp chỉ có các hành vi thì nó nên thuộc vào các lớp khác. Một lớp cũng không nên chỉ có các thuộc tính (dù có thể có thêm một số hàm set/get)
3. *Cohesion*: Mỗi lớp nên có tính cố kết cao, chỉ nên liên quan đến một việc chính
4. *Substituability*: Khi có kế thừa, đối tượng của lớp dẫn xuất (lớp con) cần có thể thay thế được cho một đối tượng của lớp cơ sở (lớp cha)

Câu hỏi

1. Các đặc trưng của một đối tượng?
2. Sự khác nhau giữa đối tượng và lớp? Ký hiệu UML của chúng thế nào?
3. Các đối tượng giao tiếp với nhau bằng cách nào?
4. Liệt kê 4 loại liên kết giữa các lớp?
5. Khác nhau giữa kết hợp (association) và kết tập (aggregation)
6. Khác nhau giữa kết tập (aggregation) và gộp (composition)?
7. Khi nào có thể mô hình một lớp là lớp con của một lớp khác?
8. Sự khác nhau giữa một hoạt động (operation) và một phương thức (method)?
9. Một số tiêu chuẩn thiết kế của một lớp được coi là tốt?

1. Hãy tìm các định nghĩa đúng cho mỗi khái niệm

<i>Concept</i>	<i>Definition</i>
aggregation ¹⁴	1 a relationship between two classes where one is a specialization of another
association ¹¹	2 the ability of one operation to be implemented by different methods
attribute ⁷	3 abstracting common features into a superclass
class ¹³	4 code implementing an operation
data hiding ⁵	5 concealing internal details of an object
encapsulation ¹⁰	6 creation of an object
generalization ³	7 data item defined as part of a class or object
inheritance ¹	8 instance of a class
instatiation ⁶	9 interface of a method
message ¹²	10 packaging together data and operations
method ⁴	11 relationship between classes
object	12 request for a service to be executed
operation ⁹	13 template for objects
polymorphism ²	14 whole-part relationship

Bài tập

2. Căn cứ vào tính chất của các khái niệm trong thực tế, hãy xác định quan hệ giữa các lớp trong từng trường hợp sau:
- a hotel room, booking, guest
 - b club member, adult member, junior member
 - c exam paper, instruction, question, solution
 - d animal, mammal, bird, reptile, dog, horse, parrot
 - e sentence, word, letter, punctuation
 - f academic staff, lecturer, professor, student.
-