#### Systems Analysis & Design Phân tích và thiết kế hệ thống CS183 Spring Semester 2008

Dr. Jonathan Y. Clark

Dịch: kuam aka

Email: j.y.clark@surrey.ac.uk

Course Website: www.computing.surrey.ac.uk/personal /st/J.Y.Clark/teaching/sad/cs183.html



## Course Textbook: Systems Analysis and Design With UML 2.0

An Object-Oriented Approach, Second Edition Sách Giáo Khoa:

Phân tích và thiết kế hệ thống với UML 2.0

theo Hướng đối tượng, tái bản lần thứ hai

#### **Chapter 7:**

**Structural Modelling** 

Chương 7:

Mô hình hóa Cấu trúc

Alan Dennis, Barbara Wixom, and David Tegarden

© 2005

John Wiley & Sons, Inc.



### Adapted from slides © 2005 John Wiley & Sons, Inc.



#### Key Ideas Ý chính

- A structural or conceptual model describes the structure of the data that supports the business processes in an organization..
- Mô hình cấu trúc hay mô hình ý tưởng mô tả cấu trúc của dữ liệu cần thiết trong nghiệp vụ của tổ chức..
- The structure of data used in the system is represented through CRC cards, class diagrams, and object diagrams.
- Cấu trúc dữ liệu dùng trong hệ thống có thể được biểu diễn qua thẻ CRC, biểu đô lớp, và biểu đô đối tượng.



## STRUCTURAL MODELS MÔ HÌNH CẦU TRÚC



## MụC đích của Mô hình Cấu

- Reduce the "semantic gap" between the real world and the world of software
- Giảm sự "nhập nhằng ngữ nghĩa" giữa thế giới thực và thế giới ảo.
- Create a vocabulary for analysts and users
- Tạo bộ từ vựng chung cho phân tích gia và người dùng.
- Represent things, ideas, and concepts of importance in the application domain
- Biểu diễn vật chất, ý tưởng, và các khái niệm quan trọng trong phạm vi ứng dụng.



#### Classes and Objects Nhắc lại Lớp và Đối tượng

- Class Template to define specific instances or objects
- Lớp Mô hình để định nghĩa các bản thể hay các đối tượng cụ thể
- Object Instantiation of a class
- Đối tượng Thể hiện của lớp
- Attributes Describes the object
- Thuộc tính Mô tả đối tượng
- Behaviours specify what an object can do
- Hành vi cho biết đối tượng đó có thể làm gì



## Basic Characteristics of Object Oriented Systems Các Thành phần Cơ bản của OOS

- Classes and Objects
- Lớp và Đối tượng
- Methods and Messages
- Phương thức và Thông điệp
- Encapsulation and Information Hiding
- Đóng gói và Che giấu Thông tin
- Inheritance
- Kế thừa
- Polymorphism
- Đa hình



#### Helpful Hint....'Compile'

#### Mẹo để nhớ..."Compile"

- C Classes
- O Objects
- M Methods and Messages
- P Polymorphism
- I Inheritance
- (Last, but not least)
- E Encapsulation

- C Lóp
- O Đối tượng
- M Phương thức và Thông điệp
- P Đa hình
- I Kế thừa
- (L Cuối cùng, nhưng không kém phần quan trọng)
- E Đóng gói



### Classes and Objects

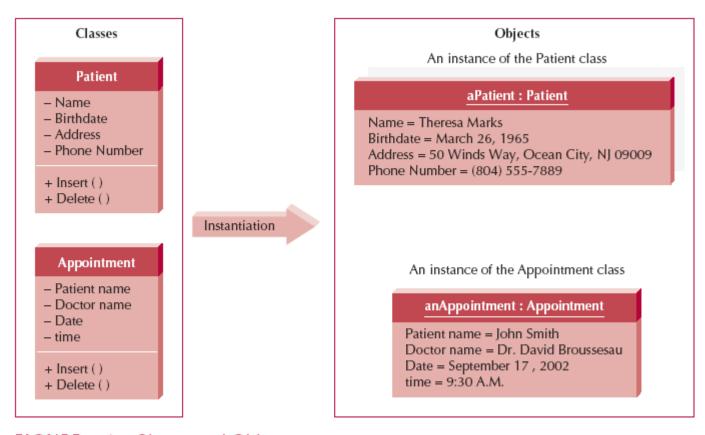


FIGURE 2-1 Classes and Objects



## Lớp và Đối tượng

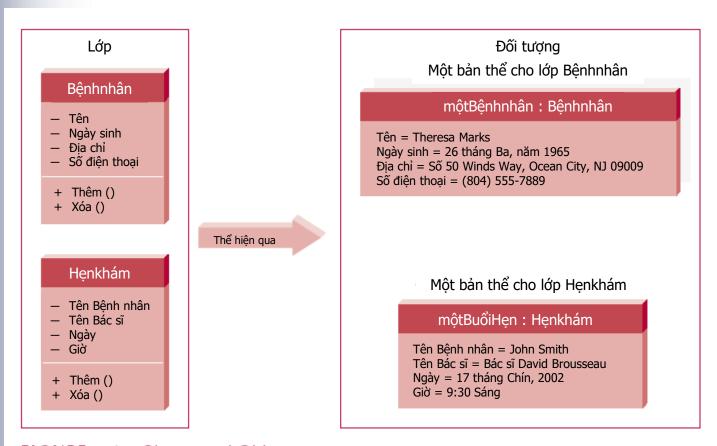


FIGURE 2-1 Classes and Objects



## Encapsulation and Information Hiding Đóng gói và che giấu thông tin

- Encapsulation
- Đóng gói
  - combination of data and process into an entity
  - Hợp nhất dữ liệu và quy trình thành một thực thể
- Information Hiding
- Che giấu thông tin
  - Only the information required to use a software module is published to the user
  - Chỉ những thong tin cần thiết để sử dụng phần mềm mới được công khai.
- Reusability is the Key Point
- Khả năng tái sử dụng là điểm mấu chốt
  - an object is used by calling methods
  - Một đối tượng được dung bằng cách gọi phương thức

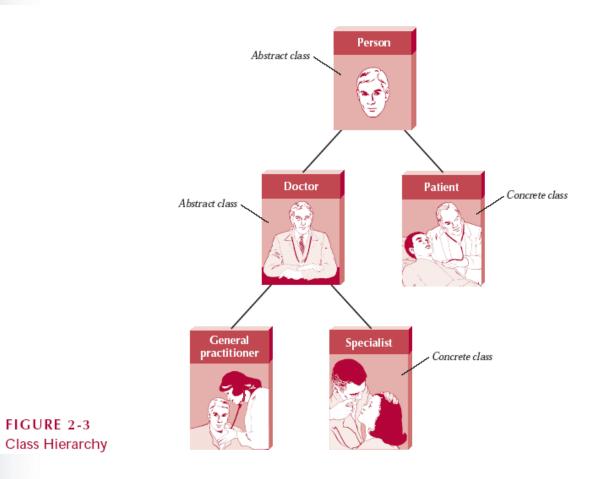


#### Inheritance Kế thừa

- Superclasses or general classes are at the top of a hierarchy of classes
- Lớp cha hay lớp khái quát sẽ đứng trên đỉnh của cây kế thừa.
- Subclasses or specific classes are at the bottom
- Lớp con hay lớp cụ thể sẽ nằm ở đáy
- Subclasses inherit attributes and methods from classes higher in the hierarchy
- Lớp con sẽ kế thừa các thuộc tính và phương thực của lớp đứng cao hơn trong cây kế thừa.

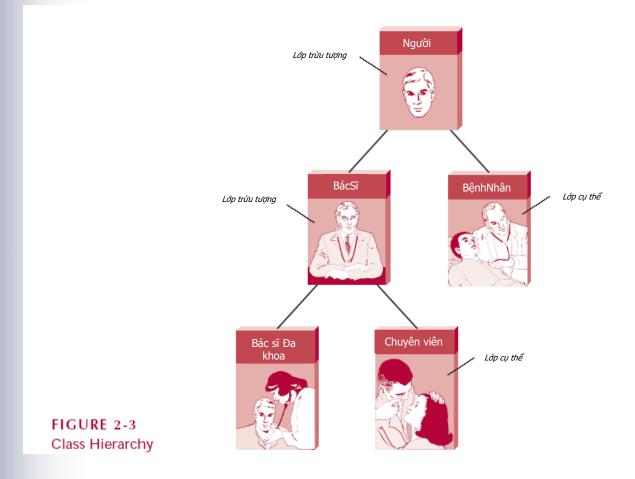


### **Class Hierarchy**



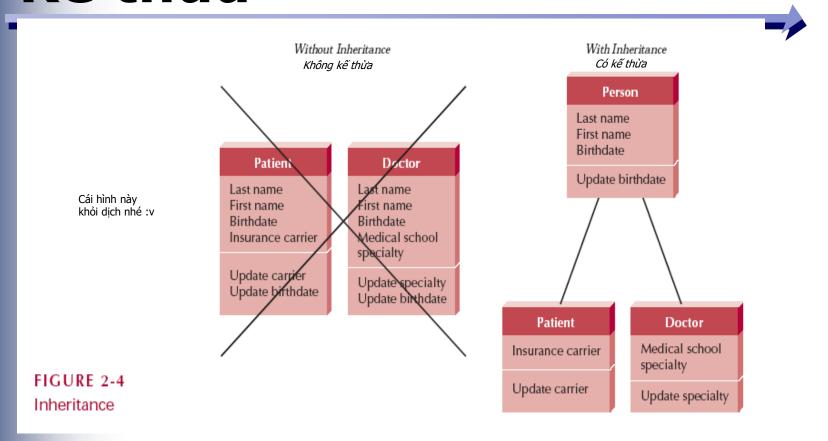


## Cây kế thừa





### Inheritance Kế thừa





## Classes Lớp



- Mô hình để tạo nên bản thể và đối tượng
  - Concrete (can have real instances)
  - Cụ thể (có thể có bản thể thật)
  - Abstract (only exists to hold subclasses)
  - Trừu tượng (chỉ tồn tại để tạo lớp con)
- Typical examples:
- Ví dụ điển hình:
  - Application domain, user interface, data structure, file structure, operating environment, document, and multimedia classes
  - Các lớp chỉ phạm vi ứng dụng, giao diện người dùng, cấu trúc dữ liệu, cấu trúc tệp, môi trường thực thi, thư mục và đa phương tiện.



## Attributes Thuộc tính

- Units of information <u>relevant</u> to the description of the class
- Các đơn vị thông tin thích hợp để mô tả lớp
- Only attributes <u>important</u> to the task should be included
- Chỉ những thuộc tính quan trọng tới các tác vụ mới được nhắc tới.



## Operations (Methods) Phương thức

- Action that instances/objects can take
- Hoạt động của các bản thể/ đối tượng
- Focus on relevant problem-specific operations (at this point)
- Tập trung vào các phương thức thích hợp cho từng vấn đề cụ thể (ở phạm vi này)



## Relationships Quan hệ

- Generalization
- Sinh (kế thừa)
  - Enables <u>inheritance</u> of attributes and operations [...is a <u>kind</u> of...]
  - Cho phép kế thừa các thuộc tính và phương thức của lớp khác [...là một <u>loại</u> của...]
- Aggregation
- Kết tập
  - Relates parts to the whole [..is a part of..]
  - Là một phần của toàn thể [...là một <u>phần</u> của...]
- Association
- Liên hợp (liên kết)
  - Miscellaneous relationships between classes
  - Các quan hệ khác nhau giữa hai lớp



#### CLASS-RESPONSIBILITY-COLLABORATION CARDS THỂ LỚP-TRÁCH NHIỆM-CỘNG TÁC



## Responsibilities and Collaborations Trách nhiệm và Cộng tác

- Responsibilities
- Trách nhiệm
  - Knowing
  - Nhận thức
  - Doing
  - Hành động
- Collaboration
- Cộng tác
  - Objects working together to service a request
  - Các đối tượng làm việc với nhau để xử lý yêu cầu



#### A CRC Card

Front:			
Class Name: Patient	ID: 3		Type: Concrete, Domain
Description: An Individual that needs to receive of medical attention		or has received	Associated Use Cases: 2
Responsibilities			Collaborators
Make appointment		Appointment	
Calculate last visit			
Change status			
Provide medical history		Medical history	

#### **Back of CRC Card**

Attributes:	
Amount (double)	
Insurance carrier (text)	
Relationships:	
Generalization (a-kind-of):	Person
Aggregation (has-parts):	Medical History
Other Associations:	Appointment



## **Một Thẻ CRC**

<b>「ên lớp:</b> Bệnh nhân	<b>ID:</b> 3		<b>Loại:</b> Cụ thể, trong phạm vi
Mô tả: Một cá nhân cần nhận hoặc được nhận sự quan		n tâm về y tế.	Ca sử dụng liên kết: 2
Trách nhiệm			Cộng tác
Tạo Buổi hẹn		Buổi hẹn	
Tính toán kể từ lần gặp cuối			
Đổi trạng thái			
Cung cấp Lịch sử khám bệnh		Lịch sử khám l	bệnh

## Mặt sau

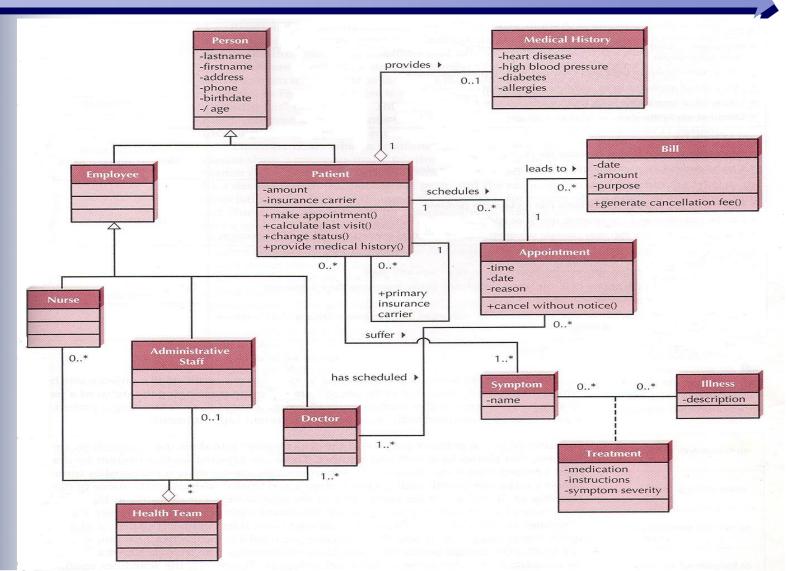
Chi phí (double)	
Nhà bảo hiểm (text)	
an hệ	
Kế thừa (một-loại-của):	Người —
Kết tập (có-các-phần):	Lịch sử khám bệnh
net tập (có các phán).	



#### CLASS DIAGRAMS BIỂU ĐỒ LỚP

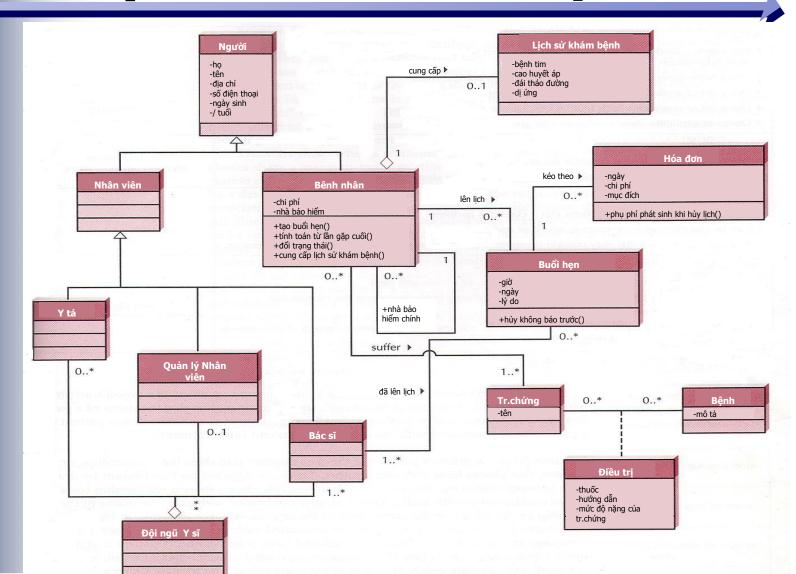


### **Example Class Diagram**





## Ví dụ về Biểu đồ Lớp





### **Class Diagram Syntax**

A CLASS	Class 1
	-attribute
	+operation ()
AN ATTRIBUTE	Attribute name /derived attribute name
AN OPERATION	operation name ()
AN ASSOCIATION	1* 01 verb_phrase



## Cú pháp trong Biểu đồ Lớp

MỘT LỚP	Lớp 1
	-thuộc tính
	+phương thức ()
MỘT THUỘC TÍNH	tên thuộc tính /thuộc tính kéo theo
MỘT PHƯƠNG THỨC	tên phương thức ()
MỘT LIÊN KẾT	1* 01 <u>cụm động từ</u>



## **More on Attributes Tiếp về Thuộc tính**

- Derived attributes
- Thuộc tính kéo theo
  - /age, for example can be calculated from birth date and current date
  - Ví dụ,/tuổi có thể được tính được từ ngày sinh.
- Visibility
- Tầm nhìn
  - + Public (not hidden)
  - +Công khai (không che đậy)
  - # Protected (hidden from all except immediate subclasses)
  - # Được bảo vệ (chỉ những lớp con trực tiếp mới dùng được)
  - Private (hidden from all other classes)
  - Không công khai (bị giấu đi hoàn toàn)



## More on Operations Tiếp về Phương thức

- Constructor
- Phương thức khởi dựng
  - Creates object (ie. creates instance)
  - Tạo đối tượng (cũng như tạo bảo thể)
- Destructor
- Phương thức hủy
  - Removes object (ie. removes instance)
  - Xóa đối tượng (cũng như xóa bản thể)
- Query
- Phương thức truy vấn
  - Makes information about state available
  - Tạo thông tin về trạng thái hiện tại
- Update
- Phương thức cập nhật
  - Changes values of some or all attributes
  - Thay đổi các giá trị của một số hoặc tất cả các thuộc tính



# Aggregation and Kết tập

- Generalization shows that a subclass inherits from a superclass
- Quan hệ kế thừa cho ta thấy sự kế thừa của một lớp con với lớp cha
  - Doctors, nurses, admin personnel are kinds of employees
  - Bác sĩ, y tá, quản lý cán bộ là một **dạng** nhân viên
- Aggregation classes comprise other classes
- Lớp kết tập **bao hàm** các lớp khác
  - Health team class comprised of doctor, nurses, admin personnel classes
  - Lớp đôị ngũ y bác sĩ **bao hàm** lớp bác sĩ, y tá, quản lý cán bộ.



### More on Relationships Tiếp về Quan hệ

- Class can be related to itself
- Một lớp có thể tự liên kết tới chính nó.
- Multiplicity
- Tính bội
  - Exactly one (1), zero or more (0..\*), one or more (1..\*), zero or one (0..1), specified range (eg. 2..4), multiple disjoint ranges (eg. 1..3,5)
  - Chỉ một (1), không hoặc nhiều hơn (0..\*), một hoặc nhiều hơn (1..\*), không hoặc một (0..1), phạm vi cụ thể (vd 2..4), đa nhánh (vd 1..3,5)
- Association class (class describing a relationship)
- Lớp liên kết (lớp dùng đề mô tả một quan hệ)

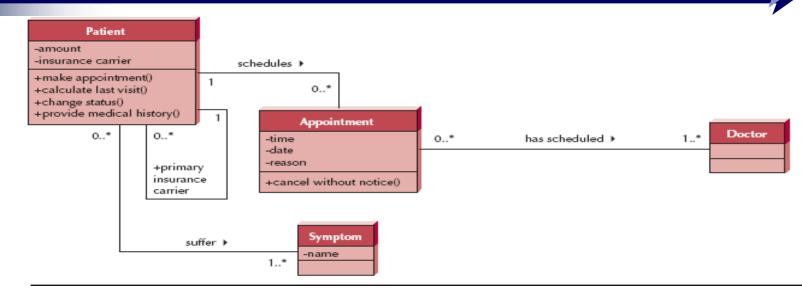


#### Simplifying Class Diagrams Đơn giản hóa Biểu đồ Lớp

- A view mechanism can show a subset of information
- Chỉ vẽ cơ chế sẽ chỉ cho ta thấy một phần thông tin
  - Eg. A use-case view that shows only that part of the diagram referring to a particular use case.
  - Ví dụ. Vẽ một ca sử dụng chỉ cho ta thấy một phần của biểu đồ tham chiếu tới ca sử dụng đó.
  - Eg. A view showing only aggregations
  - Ví dụ. Bản vẽ chỉ thể hiện sự kết tập
  - Eg. A view showing only the class name and attributes
  - Ví dụ. Bản vẽ chỉ thể hiện tên lớp và thuộc tính



## Object Diagrams Biểu đồ Đối tượng





Symptom1: Symptom

name = "muscle pain"

Khỏi dịch nhé :v

khoa

Clinic: Phòng khám tư, chuyên

FIGURE 7-5 Example Object Diagram



## CREATING CRC CARDS AND CLASS DIAGRAMS TẠO THỂ CRC VÀ VỀ BIỂU ĐỒ LỚP



## Object Identification Xác định Đối tượng

- Textual analysis of use-case information
- Phân tích các thông tin từ ca sử dụng nguyên bản
  - Nouns suggest classes
  - Các danh từ gợi ý cho lớp
  - Verbs suggest operations
  - Các động từ gợi ý cho phương thức
- Creates a rough first cut
- Thực hiện vẽ phác thảo
- Common object list
- Danh sách đối tượng phổ biến
- Incidents
- Các liên thuộc
- Roles
- Các vai trò

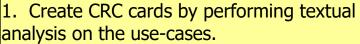


#### Patterns Các Mẫu hình

- Useful groupings of classes that recur in various situations
- Nhóm các lớp có thể xuất hiện trong nhiều tình huống khác nhau
- Contain groups of classes that collaborate or work together
- Chứa các nhóm lớp có thể hợp tác hay làm việc với nhau
- Enable reusability
- Có thể tái sử dụng



## Steps for Object Identification and Structural Modelling Các bước Xác định Đối tượng và Mô hình hóa Cấu trúc



- Brainstorm additional candidate classes, attributes, operations, and relationships by using the common object list approach.
- 3. Role-play each use-case using the CRC cards.
- 4. Create the class diagram based on the CRC cards.
- Review the structural model for missing and/or unnecessary classes, attributes, operations, and relationships.
- 6. Incorporate useful patterns.
- 7. Review the structural model.

- In các thẻ CRC sau khi phân tích các ca sử dung.
- 2. Họp để bàn luận về các "ứng cử viên" lớp, thuộc tính, phương thức, và quan hệ bằng cách dùng danh sách đói tượng phổ biến.
- 3. "Đóng vai" các lớp trong ca sử dụng dùng thẻ CRC.
- 4. Vẽ biểu đồ lớp dưa vào các thẻ CRC.
- 5. Đánh giá xem mô hình cấu trúc có thiếu và/hay thừa lớp, thuộc tính, phương thức, và quan hệ nào không.
- 6. Kết hợp với các mẫu hình hợp lý.
- 7. Đánh giá lại mô hình vừa vẽ.



## Summary Tổng kết

- CRC cards capture the essential elements of a class.
- Thẻ CRC chứa các thành phần cốt yếu của lớp.
- Class and object diagrams show the underlying structure of an object-oriented system.
- Biểu đồ lớp và đối tượng cho ta thấy cấu trúc cơ bản của một hệ thống hướng đối tượng.
- Constructing the structural model is an iterative process involving: textual analysis, brainstorming objects, role playing, creating the diagrams, and incorporating useful patterns.
- Xây dựng mô hình cấu trúc là một quy trình lặp lại bao gồm: phân tích nguyên bản, họp đối tượng, nhập vai, vẽ biểu đồ, và kết hợp với mẫu hình.

