# Mô hình hóa đối tượng

# Nội dung

**Phân tích theo Đối tượng**

Xác định các lớp

* Thuộc tính và Hoạt động

**UML Class Diagrams**

Sự kết hợp

* Đa dạng

Tập hợp

* Thành phần

Tổng quát

# Yêu cầu và kiểu tên miền

Mô hình phân tích của chúng ta nên ... ... đại diện cho con người, vật chất và các khái niệm quan trọng để hiểu biết của chúng ta về những gì đang xảy ra trong miền ứng dụng.

* … Hiển thị các kết nối và tương tác giữa những người, mọi thứ và khái niệm.
* … Cho thấy tình hình kinh doanh đủ chi tiết để đánh giá thiết kế có thể.
* … Được tổ chức để hữu ích sau này, trong quá trình thiết kế và thực hiện phần mềm.
* … Cho phép chúng tôi kiểm tra xem các chức năng chúng tôi sẽ bao gồm trong đặc tả sẽ đáp ứng các yêu cầu.
* … Kiểm tra sự hiểu biết của chúng ta về cách hệ thống mới sẽ tương tác với thế giới.

# Phân tích theo đối tượng

* Nền tảng
  + Mô hình các yêu cầu về mặt đối tượng và các dịch vụ mà họ cung cấp.
  + Phát triển từ thiết kế hướng đối tượng  Áp dụng để mô hình miền ứng dụng chứ không phải là chương trình. Motivation - Động lực
  + OO (Tự gọi là "tự nhiên") Trong một hệ thống phát triển, các chức năng nó thực hiện cần phải được thay đổi thường xuyên hơn các đối tượng mà chúng vận hành …
* … Một mô hình dựa trên các đối tượng (thay vì các chức năng) sẽ ổn định hơn theo thời gian …
* … Do đó, các thiết kế hướng đối tượng có thể duy trì được nhiều hơn.

## OO nhấn mạnh tầm quan trọng của các giao diện được xác định rõ ràng giữa các đối tượng. **Chú ý: OO áp dụng cho kỹ thuật yêu cầu bởi vì nó là một công cụ mô hình hóa. Nhưng chúng ta đang làm mẫu các đối tượng tên miền chứ không phải là thiết kế của hệ thống mới**

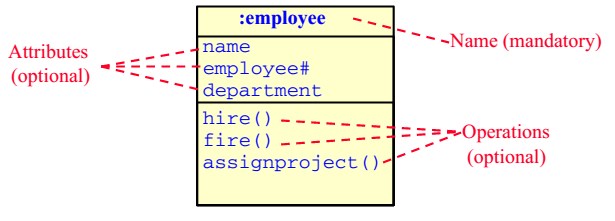
## Gần như mọi thứ có thể là một đối tượng

* Các đối tượng bên ngoài
  + … Tương tác với hệ thống được mô phỏng
* E.g Con người, thiết bị, các hệ thống khác
* Thứ
  + … Đó là các phần của miền được mô phỏng
* E.g. Báo cáo, hiển thị, tín hiệu, vv.
* Vai trò

sử dụng bởi những người tương tác với hệ thống

Lớp là gì?

* Một lớp mô tả một nhóm các đối tượng với các thuộc tính tương tự nhau (thuộc tính), các hành vi thường gặp (hoạt động), các mối quan hệ thường gặp với các đối tượng khác,
  + và ý nghĩa chung ("ngữ nghĩa")
* Ví dụ
  + **nhân viên**: có tên, nhân viên và bộ phận; Một nhân viên được thuê và sa thải; Một nhân viên làm việc trong một hay nhiều dự án**.**.



## Tìm lớp

* Tìm các lớp từ dữ liệu nguồn:

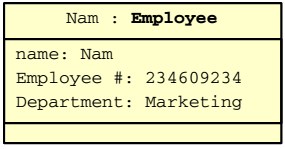
Tìm các danh từ và cụm danh từ trong mô tả của các bên liên quan về vấn đề.

* Tìm các lớp học từ các nguồn khác: Xem xét thông tin cơ bản; Người dùng và các bên liên quan khác;
* Tốt hơn để bao gồm nhiều lớp ứng cử viên lúc đầu Bạn có thể luôn luôn loại bỏ chúng sau này nếu họ phát hiện ra không phải là hữu ích.

# Chọn lựa lớp

* Loại bỏ các lớp học cho các khái niệm mà:
  + Không nằm trong phạm vi phân tích.  Nhân đôi các lớp khác.
  + Quá mơ hồ hoặc quá cụ thể.
* e.g Có quá nhiều hoặc quá ít trường hợp

# Đối tượng và lớp

* Các thể hiện của một lớp được gọi là các đối tượng Các đối tượng được trình bày như: 
  + Hai đối tượng khác nhau có thể có các giá trị thuộc tính giống nhau (như hai người có tên và địa chỉ giống hệt nhau)
* Đối tượng có kết hợp với các đối tượng khác E.g. Fred\_Bloggs:employee is associated with the KillerApp:project object.
  + Nhưng chúng ta sẽ nắm bắt những mối quan hệ này ở cấp lớp.
  + **Lưu ý: Đảm bảo các thuộc tính được kết hợp với đúng lớp**

# Sự kết hợp

* Các đối tượng không tồn tại độc lập với nhau Một mối quan hệ đại diện cho một sự kết nối giữa các sự vật.

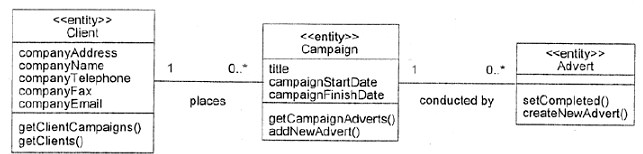
Trong UML, có nhiều loại mối quan hệ khác nhau.

Sự kết hợp Tổng hợp và Thành phần Tổng thể hoá Tính độc lập Thu chính**.**

**Lưu ý: Hai yếu tố cuối không hữu ích trong quá trình phân tích yêu cầu**.

# Sự kết hợp (cont.)

Biểu đồ lớp hiển thị các lớp và

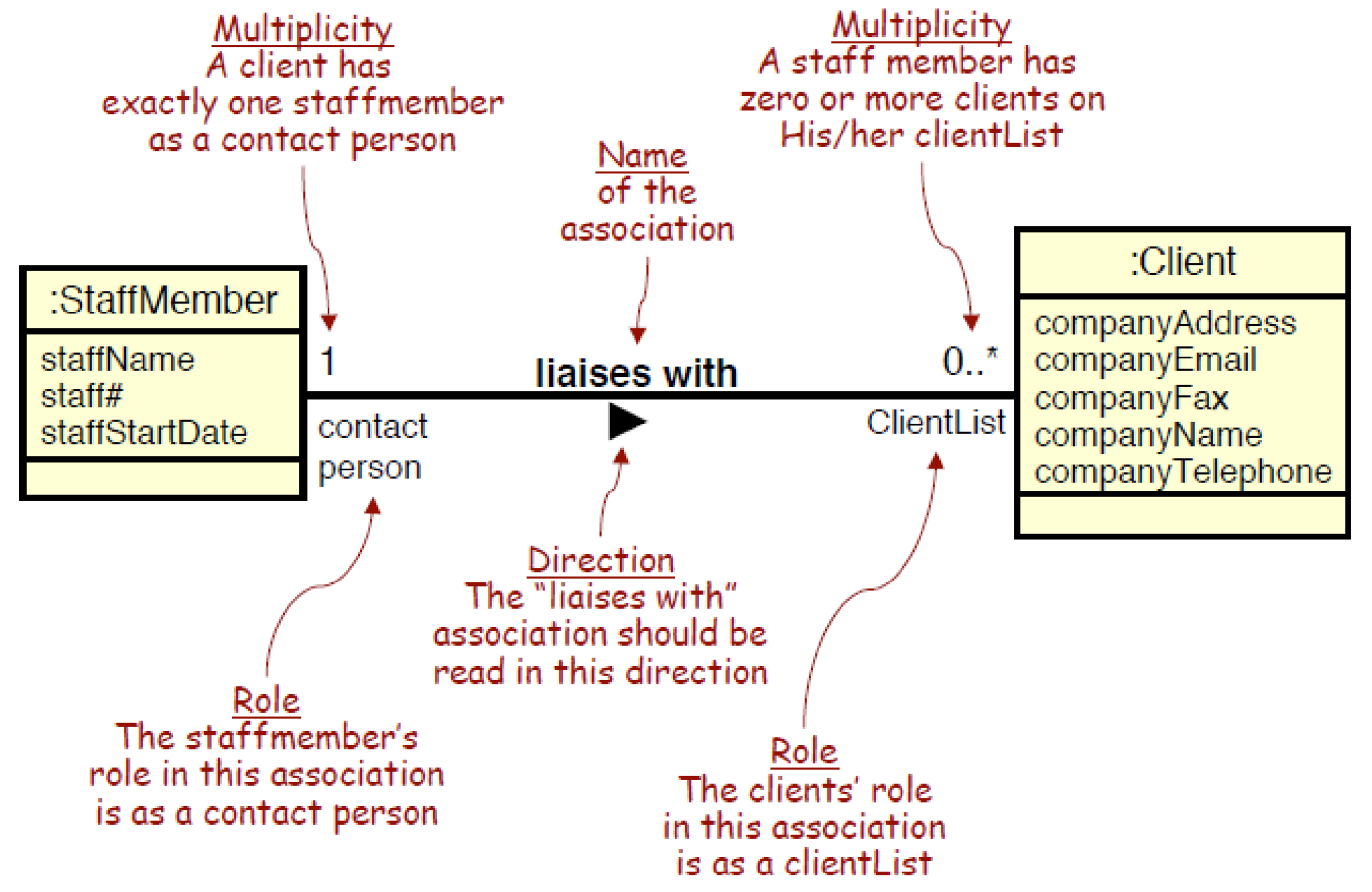
các mối quan hệ

# Nhiều sự kết hợp

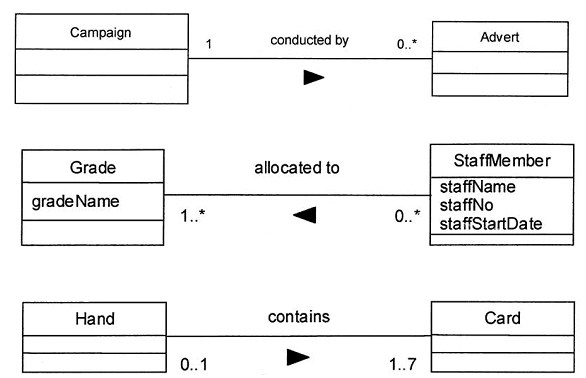
Chiến dịch có thể tồn tại mặc dù không có thành viên quản lý nó?

* Nếu có, thì kết hợp là tùy chọn ở cuối nhân viên - không hoặc nhiều hơn (**0..\***)
* Nếu không, thì nó không phải là tùy chọn - một hoặc nhiều (**1..\***)
* Nếu nó phải được quản lý bởi một và chỉ một nhân viên chính xác một (1)  Một số ví dụ về xác định tính đa dạng:

**Tùy chọn (0 or 1): 0..1** **Exactly one: 1 = 1..1** **Zero or more: 0..\* = \*** **One or more: 1..\*** **A range of values: 2..6**

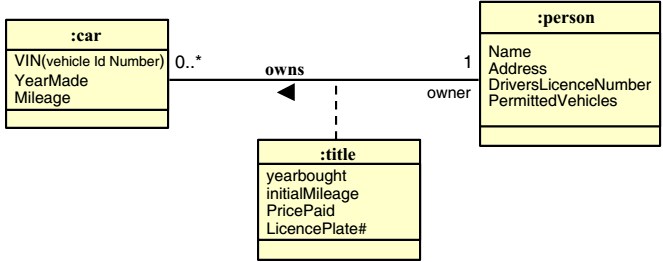


# More Examples



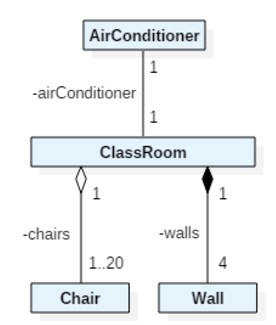
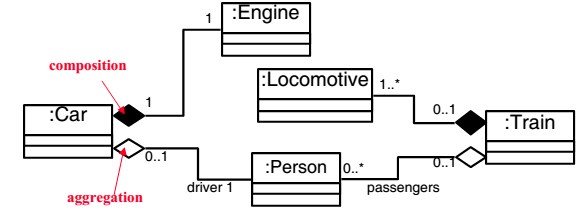
# Kết hợp các lớp

* Đôi khi sự kết hợp là một lớphọc
  + … Bởi vì chúng ta cần phải duy trì thông tin về sự kết hợp.
  + … Và thông tin đó không tự nhiên tồn tại trong các lớp.
* E.g. "Tiêu đề" là một đối tượng đại diện cho thông tin về mối quan hệ giữa chủ sở hữu và xe của cô ấy



# Tổng hợp và thành phần

* Tổng hợp Đây là mối quan hệ "Có-một" hoặc "Toàn bộ / một phần".
* Thành phần **Hình thức tập hợp mạnh mẽ ngụ ý về quyền sở hữu: Nếu toàn thể được loại bỏ khỏi mô hình, thì phần đó**.



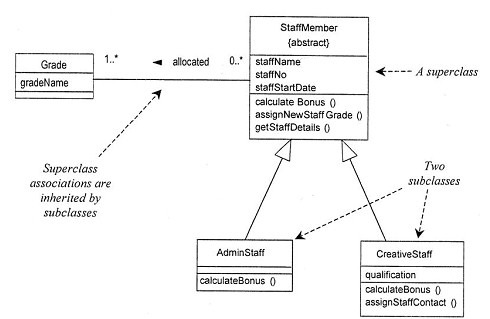
# Sự tổng quát

* Ghi chú:
  + Các lớp con kế thừa các thuộc tính, các sự kết hợp và các hoạt động từ nhiều lớp.
  + Một phân lớp có thể ghi đè lên một khía cạnh thừa kế
* e.g. **AdminStaff & CreativeStaff có các phương pháp khác nhau để tính tiền thưởng**.

Siêu lớp có thể được tuyên bố {abstract}, có nghĩa là nó không có trường hợp Ngụ ý rằng các lớp con bao gồm tất cả các khả năng.

* e.g. there are no other staff than **AdminStaff** and **CreativeStaff**

# Sự tổng quát (cont.)

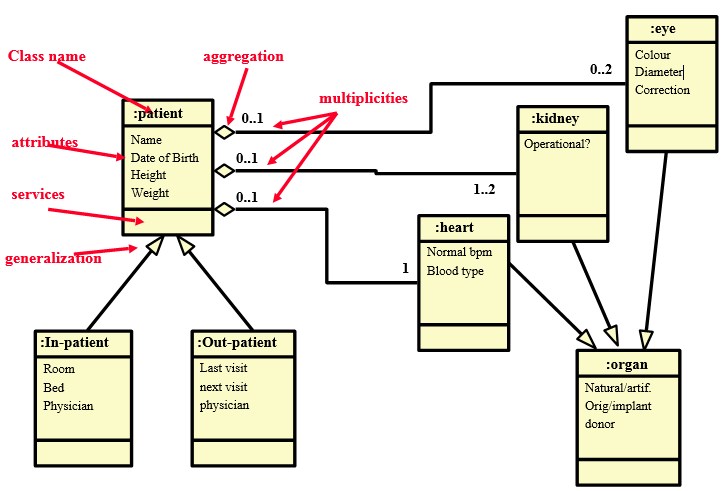


# Class Diagrams

* Một tập hợp các sơ đồ lớp đại diện cho toàn bộ hệ thống.
* Class diagrams hiển thị các đối tượng khác nhau sẽ được sử dụng trong các hệ thống và các mối quan hệ của chúng.
* Các yếu tố quan trọng:
  + **Lớp**
  + **Mối quan hệ (such as Inheritance, Composition, Aggregation,**

**Association, Dependency)** OCL (Object Constraint Language: Invariant, Precondition,

Postcondition) **Giao diện**



# Main references

* Prof Steve Easterbrook, lecture notes, University

of Toronto, Canada.

* https://github.com/imalitavakoli/learn-uml2

## Q&A