

BỘ MÔN CÔNG NGHỆ PHẦN MỀM  
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG  
TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

**LÝ THUYẾT NGÔN NGỮ HƯỚNG ĐỐI TƯỢNG**

**Lecture 11 - Class Diagrams**

Nguyễn Thị Thu Trang  
trangntt@soict.hust.edu.vn

2

## Mục đích

- Mô tả khung nhìn tĩnh của hệ thống
- Minh họa cách đọc và hiểu một biểu đồ lớp
- Mô hình hóa các mối quan hệ giữa hai lớp: liên kết (association), kết tập (aggregation) và cấu thành
- Mô hình tổng quát hóa (generalization) trên một biểu đồ lớp.

3

## Nội dung

- ⇒ 1. Biểu đồ lớp (Class diagram)
- 2. Liên kết (Association)
- 3. Kết tập (Aggregation)
- 4. Tổng quát hóa (Generalization)

4

## 1.1. Lớp (Class)

- Sử dụng hình chữ nhật gồm 3 thành phần
  - Tên lớp
  - Các thuộc tính
  - Các phương thức

Class_Name
attribute1 attribute2 attribute3
method1() method2() method3()

5

## Biểu diễn thuộc tính

- Chỉ ra tên, kiểu và giá trị mặc định nếu có
  - `attributeName : Type = Default`
- Tuân theo quy ước đặt tên của ngôn ngữ cài đặt và của dự án.
- Kiểu (type) nên là kiểu dữ liệu cơ bản trong ngôn ngữ thực thi
  - Kiểu dữ liệu có sẵn, kiểu dữ liệu người dùng định nghĩa, hoặc lớp tự định nghĩa.

6

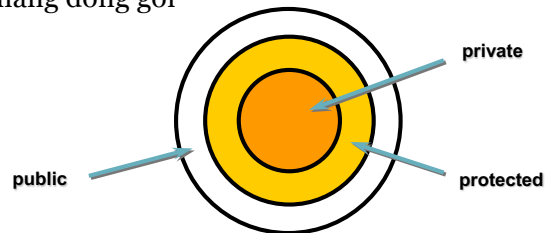
## Mô tả phương thức

- Tên phương thức:
  - Mô tả kết quả
  - Sử dụng góc nhìn của đối tượng khách (client – đối tượng gọi)
  - Nhất quán giữa các lớp
- Chữ ký của phương thức:
  - `operationName([direction] parameter:class,...):returnType`
  - Direction: **in** (mặc định), **out** hoặc **inout**

7

## Phạm vi truy cập (Visibility)

- Phạm vi truy cập được sử dụng để thực hiện khả năng đóng gói



## Phạm vi truy cập được biểu diễn như thế nào?

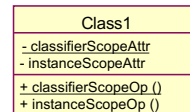
- Các ký hiệu sau được sử dụng:
  - `+` Public access
  - `#` Protected access
  - `-` Private access

Class1
- privateAttribute
+ publicAttribute
# protectedAttribute
- privateOperation ()
+ publicOperation ()
# protecteOperation ()

9

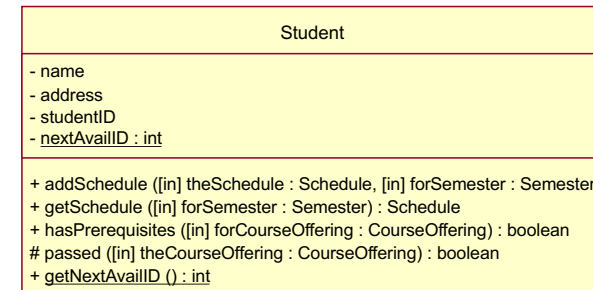
## Phạm vi (Scope)

- Xác định số lượng thể hiện của thuộc tính/thao tác:
  - **Instance:** Một thể hiện cho mỗi thể hiện của mỗi lớp
  - **Classifier:** Một thể hiện cho tất cả các thể hiện của lớp
- Phạm vi Classifier được ký hiệu bằng cách gạch dưới tên thuộc tính/thao tác.



10

## Ví dụ: Scope



11

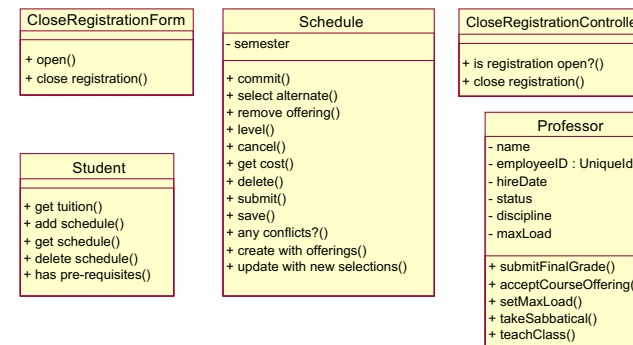
## 1.2. Biểu đồ lớp là gì?

- Biểu đồ lớp chỉ ra sự tồn tại của các lớp và mối quan hệ giữa chúng trong bản thiết kế logic của một hệ thống
  - Chỉ ra cấu trúc tĩnh của mô hình như lớp, cấu trúc bên trong của chúng và mối quan hệ với các lớp khác.
  - Chỉ ra tất cả hoặc một phần cấu trúc lớp của một hệ thống.
  - Không đưa ra các thông tin tạm thời.
- Khung nhìn tĩnh của một hệ thống chủ yếu hỗ trợ các yêu cầu chức năng của hệ thống.

12

## Biểu đồ lớp (Class Diagram – CD)

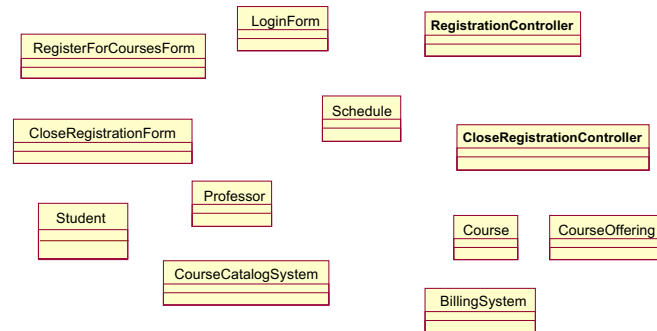
- Khung nhìn tĩnh của hệ thống



13

## Ví dụ Biểu đồ lớp

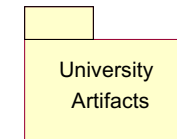
- Có cách nào tốt hơn để tổ chức biểu đồ lớp?



14

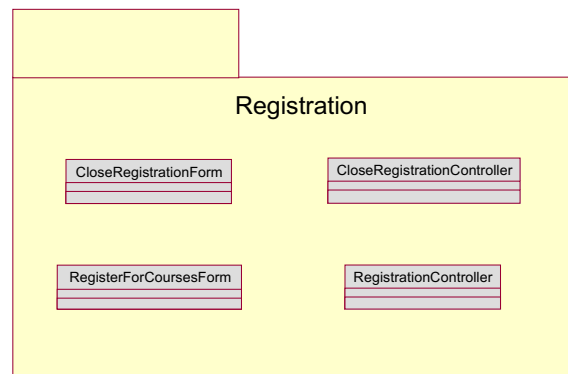
## Gói (package)

- Một cơ chế chung để tổ chức các phần tử thành nhóm.
- Một phần tử trong mô hình có thể chứa các phần tử khác.



15

## Ví dụ: Registration Package



16

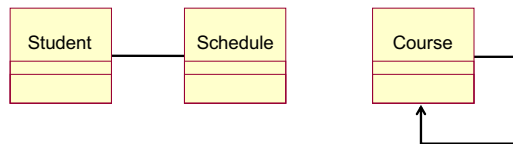
## Nội dung

1. Biểu đồ lớp (Class diagram)
- ⇒ 2. Liên kết (Association)
3. Kết tập (Aggregation)
4. Tổng quát hóa (Generalization)

17

## Liên kết (association) là gì?

- Mỗi liên hệ ngữ nghĩa giữa hai hay nhiều lớp chỉ ra sự liên kết giữa các thể hiện của chúng
- Mỗi quan hệ về mặt cấu trúc chỉ ra các đối tượng của lớp này có kết nối với các đối tượng của lớp khác.



18

## Bội số quan hệ (Multiplicity)

- Bội số quan hệ là số lượng thể hiện của một lớp liên quan tới MỘT thể hiện của lớp khác.
- Với mỗi liên kết, có hai bội số quan hệ cho hai đầu của liên kết.
  - Với mỗi đối tượng của Professor, có nhiều Course Offerings có thể được dạy.
  - Với mỗi đối tượng của Course Offering, có thể có 1 hoặc 0 Professor giảng dạy.



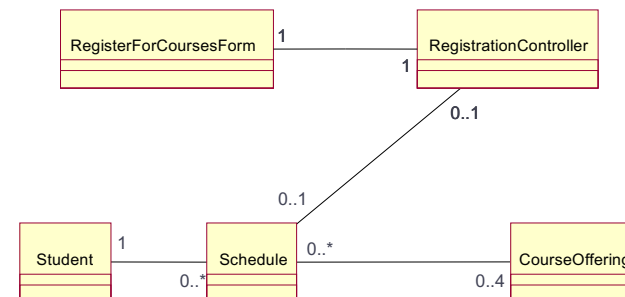
19

## Biểu diễn bội số quan hệ

Unspecified	
Exactly One	1
Zero or More	0..*
Zero or More	*
One or More	1..*
Zero or One (optional value)	0..1
Specified Range	2..4
Multiple, Disjoint Ranges	2, 4..6

20

## Ví dụ về bội số quan hệ



21

## Nội dung

1. Biểu đồ lớp (Class diagram)
2. Liên kết (Association)
- ⇒ 3. Kết tập (Aggregation)
4. Tổng quát hóa (Generalization)

22

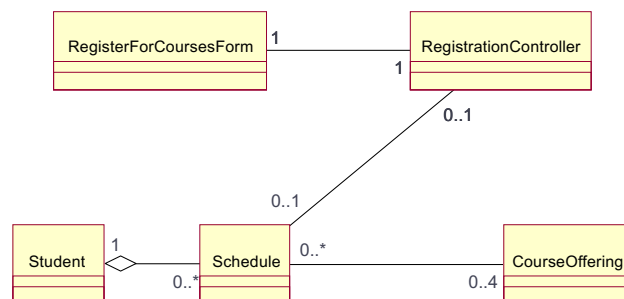
## Kết tập (aggregation) là gì?

- Là một dạng đặc biệt của liên kết mô hình hóa mối quan hệ toàn thể-bộ phận (whole-part) giữa đối tượng toàn thể và các bộ phận của nó.
  - Kết tập là mối quan hệ “là một phần” (“is a part-of”).
- Bội số quan hệ được biểu diễn giống như các liên kết khác



23

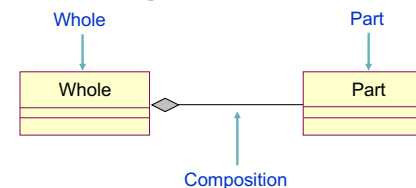
## Ví dụ về kết tập



24

## Cấu thành (Composition) là gì?

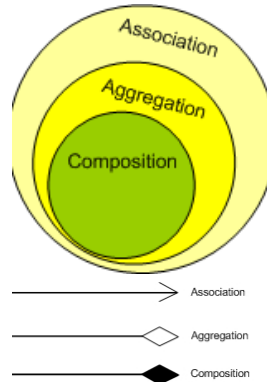
- Một dạng của kết tập với quyền sở hữu mạnh và các vòng đời trùng khớp giữa hai lớp
  - Whole sở hữu Part, tạo và hủy Part.
  - Part bị bỏ đi khi Whole bị bỏ, Part không thể tồn tại nếu Whole không tồn tại.



25

## Association, Aggregation and Composition

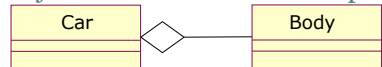
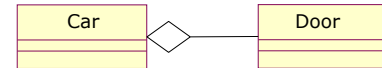
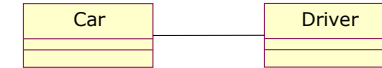
- Mọi quan hệ giữa các lớp (relationship)
  - Liên kết (Association)
    - Sử dụng (use-a)
  - Kết tập (Aggregation)
    - Strong association
    - has-a/is-a-part
  - Cấu thành (Composition)
    - Strong aggregation
    - Share life-time



26

## Examples for Ass, Agg & Comp

- Association
  - use-a
  - Objects of one class are associated with objects of another class
- Aggregation
  - has-a/is-a-part
  - Strong association, an instance of one class is made up of instances of another class
- Composition
  - Strong aggregation, the composed object can't be shared by other objects and dies with its composer
  - Share life-time



27

## Nội dung

1. Biểu đồ lớp (Class diagram)
2. Liên kết (Association)
3. Kết tập (Aggregation)
- ⇒ 4. Tổng quát hóa (Generalization)

28

## Tổng quát hóa (Generalization)

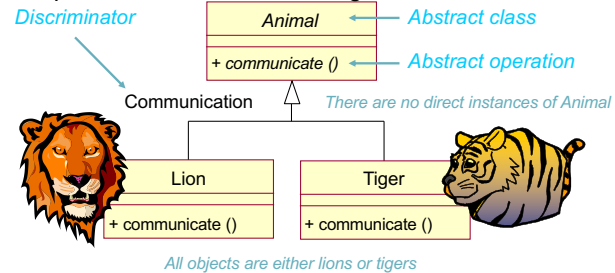
- Mọi quan hệ giữa các lớp trong đó một lớp chia sẻ cấu trúc và/hoặc hành vi với một hoặc nhiều lớp khác
- Xác định sự phân cấp về mức độ trừu tượng hóa trong đó lớp con kế thừa từ một hoặc nhiều lớp cha
  - Đơn kế thừa (Single inheritance)
  - Đa kế thừa (Multiple inheritance)
- Là mối liên hệ “là một loại” (“is a kind of”)

29

## Lớp trừu tượng và lớp cụ thể (Abstract and Concrete Class)

- Lớp trừu tượng không thể có đối tượng
  - Chứa phương thức trừu tượng
  - Chữ nghiêng

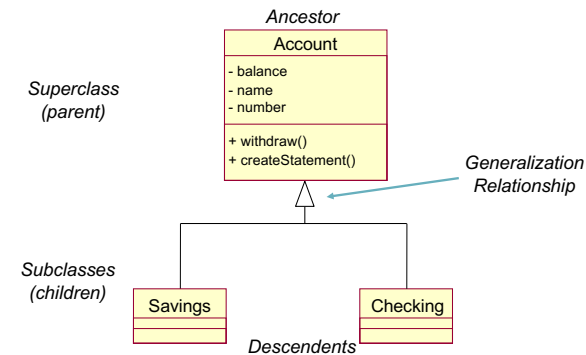
- Lớp cụ thể có thể có đối tượng



30

## Ví dụ về Đơn kế thừa

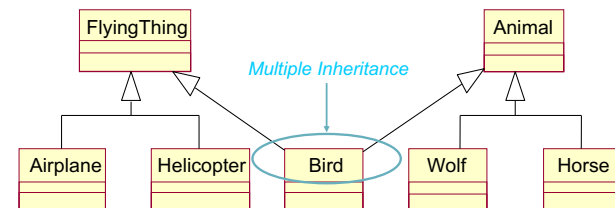
- Một lớp kế thừa từ MỘT lớp khác



31

## Ví dụ về Đa kế thừa

- Một lớp có thể kế thừa từ nhiều lớp khác

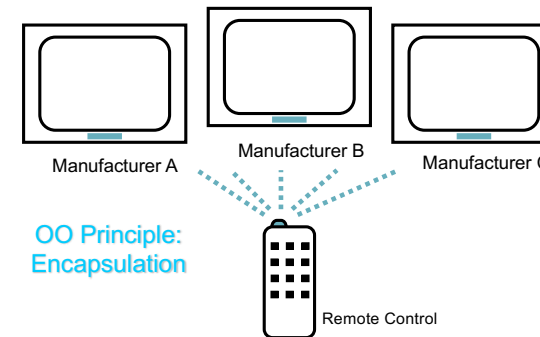


Sử dụng đa kế thừa chỉ khi cần thiết và luôn luôn phải cẩn thận!

32

## Đa hình (Polymorphism) là gì?

- Khả năng che giấu các thực thi khác nhau dưới một giao diện duy nhất.



OO Principle:  
Encapsulation



