





Bộ môn Công nghệ Phần mềm
Viện CNTT & TT
Trường Đại học Bách Khoa Hà Nội


LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG
Bài 12. Biểu đồ lớp



Mục đích

- Mô tả khung nhìn tĩnh của hệ thống và cách đưa nó vào trong một mô hình.
- Minh họa cách đọc và hiểu một biểu đồ lớp.
- Mô hình hóa mối liên kết (association) và kết tập (aggregation) và chỉ ra cách mô hình chú ng vào biểu đồ lớp.
- Mô hình tổng quát hóa (generalization) trên một biểu đồ lớp.


2



Nội dung

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

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

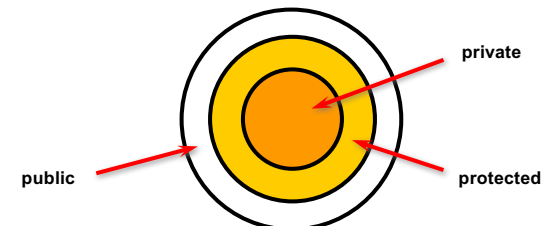
Mô tả phương thức

- 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



8

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 ()
protectedOperation ()

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.

Class1
- classifierScopeAttr
- instanceScopeAttr
+ classifierScopeOp ()
+ instanceScopeOp ()

10

Ví dụ: Scope

Student
- name
- address
- studentID
- nextAvailID : int
+ addSchedule ([in] theSchedule : Schedule, [in] forSemester : Semester)
+ getSchedule ([in] forSemester : Semester) : Schedule
+ hasPrerequisites ([in] forCourseOffering : CourseOffering) : boolean
passed ([in] theCourseOffering : CourseOffering) : boolean
+ getNextAvailID () : int

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

```

classDiagram
    class CloseRegistrationForm {
        +open()
        +closeRegistration()
    }
    class Schedule {
        -semester
        +commit()
        +selectAlternate()
        +removeOffering()
        +level()
        +cancel()
        +getCost()
        +delete()
        +submit()
        +save()
        +anyConflicts?()
        +createWithOfferings()
        +updateWithNewSelections()
    }
    class CloseRegistrationController {
        +isRegistrationOpen?()
        +closeRegistration()
    }
    class Student {
        +getTuition()
        +addSchedule()
        +getSchedule()
        +deleteSchedule()
        +hasPrerequisites()
    }
    class Professor {
        -name
        -employeeID : UniqueID
        -hireDate
        -status
        -discipline
        -maxLoad
        +submitFinalGrade()
        +acceptCourseOffering()
        +setMaxLoad()
        +takeSabbatical()
        +teachClass()
    }
  
```

13

Khi nào sử dụng biểu đồ lớp?

- Từ vựng của hệ thống (Vocabulary)
 - Khi trừu tượng hóa một phần hoặc bên ngoài hoặc biên của hệ thống.
 - Chỉ ra kết quả trừu tượng hóa và trách nhiệm của chúng
- Cộng tác (Collaboration)
 - Nhóm các lớp và các thành phần khác làm việc cùng nhau để thực hiện một công việc nào đó.
- Lược đồ CSDL logic (Logical database schema)
 - Tương tự như bản thiết kế khái niệm cho CSDL
 - Chứa các đối tượng cần lưu trữ lâu dài tức là cần lưu trong CSDL

14

Ví dụ Biểu đồ lớp

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

```

classDiagram
    package RegisterForCoursesForm
    package LoginForm
    package RegistrationController
    package CloseRegistrationForm
    package Schedule
    package CloseRegistrationController
    package Student
    package Professor
    package Course
    package CourseOffering
    package CourseCatalogSystem
    package BillingSystem
  
```

15

Gói (package)

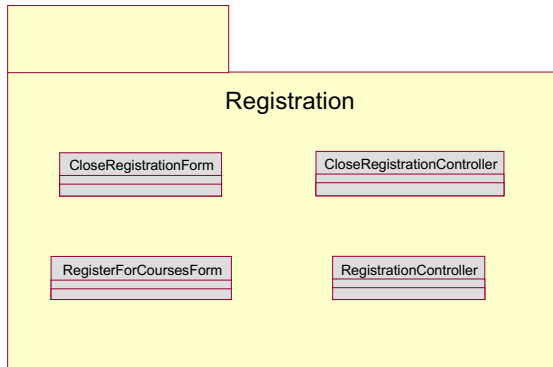
- Một cơ chế chung để tổ chức các phần tử thành nhóm.
- Một phần tử trong mô hình có thể chứa các phần tử khác.

```

classDiagram
    package University {
        Artifacts
    }
  
```

16

Ví dụ: Registration Package



17

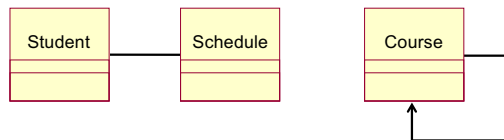
Nội dung

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

18

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.



19

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.



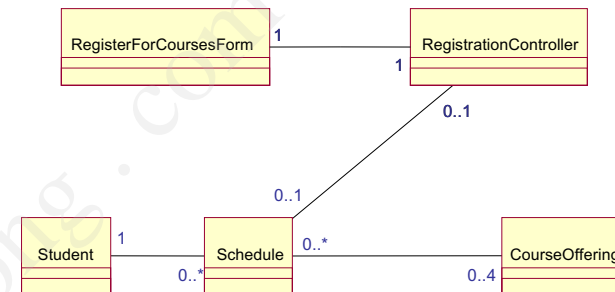
20

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

21

Ví dụ về bội số quan hệ



22

Nội dung

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

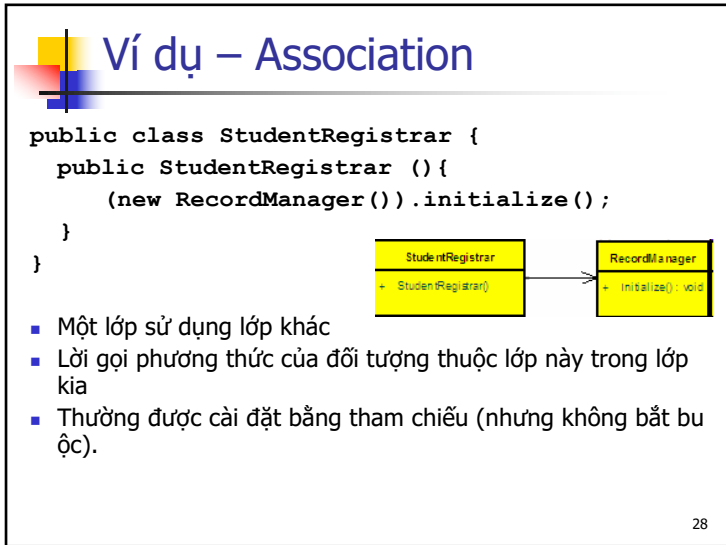
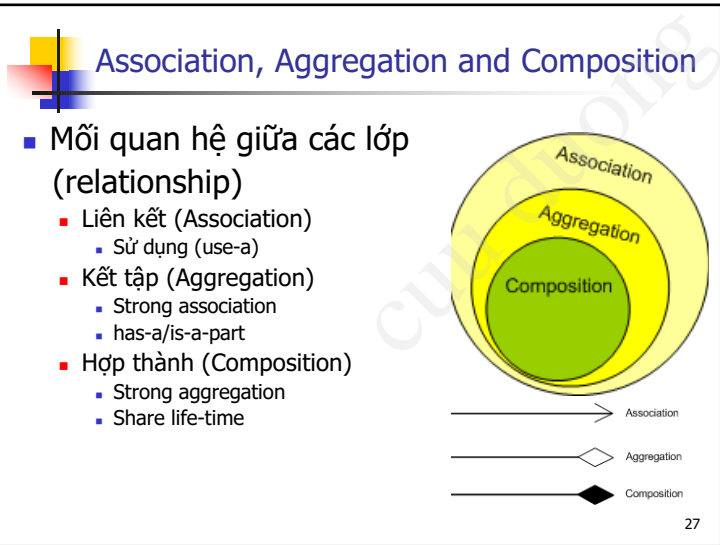
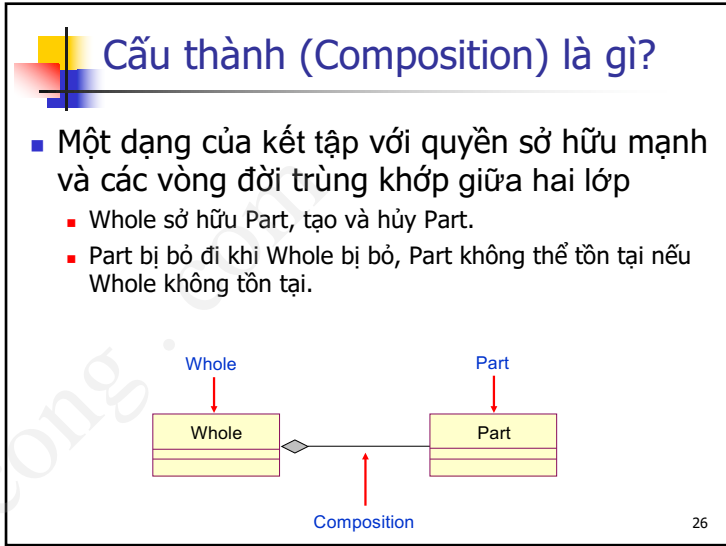
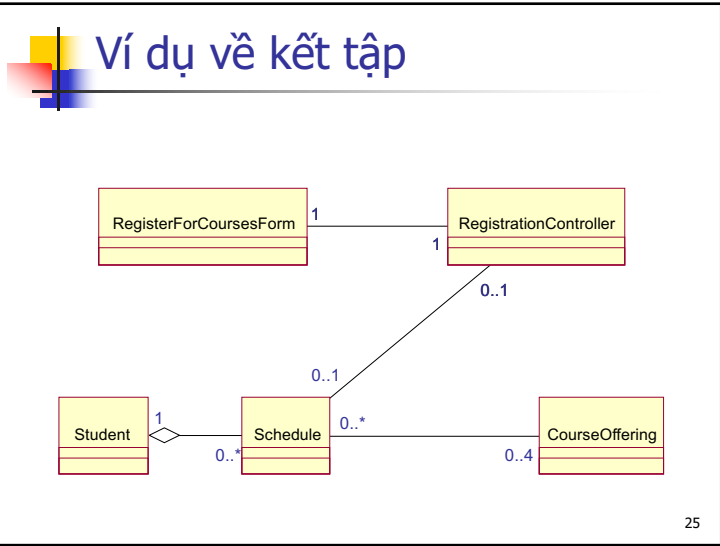
23

Kết tập (aggregation) là gì?

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

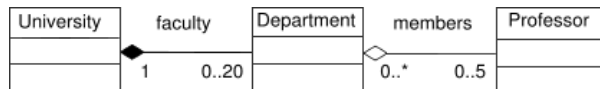


24



Ví dụ – Aggregation vs. Composition

- University chứa (own) nhiều Department
- Mỗi Department có 1 số các Professor



- Nếu hủy University:
 - Các phòng ban cũng không còn tồn tại
 - Nhưng các Professor trong các Department vẫn còn tồn tại
- Dấu hiệu khác:
 - 1 professor có thể làm trong nhiều Department
 - 1 Department chỉ thuộc về 1 University

29

Nội dung

- Biểu đồ lớp (Class diagram)
- Liên kết (Association)
- Kết tập (Aggregation)
- Tổng quát hóa (Generalization)**

30

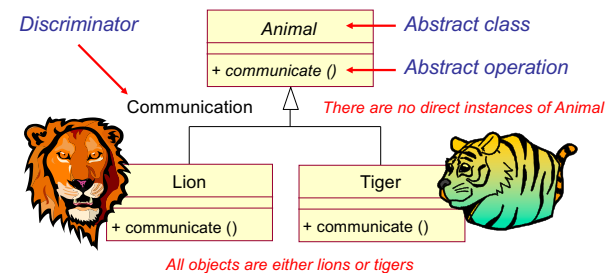
Tổng quát hóa (Generalization)

- Mỗi 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 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")

31

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

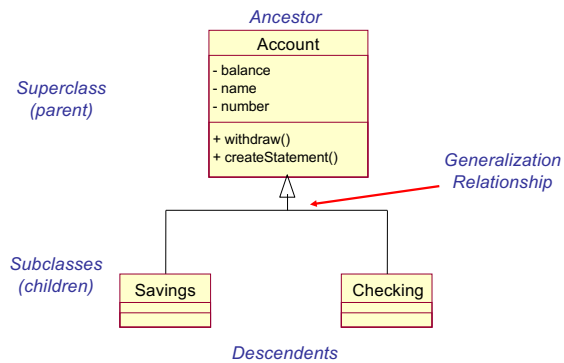
- Lớp trừu tượng không thể có đối tượng
 - Chứa phương thức trừu tượng
 - Chữ nghiêng
- Lớp cụ thể có thể có đối tượng



32

Ví dụ về Đơn kế thừa

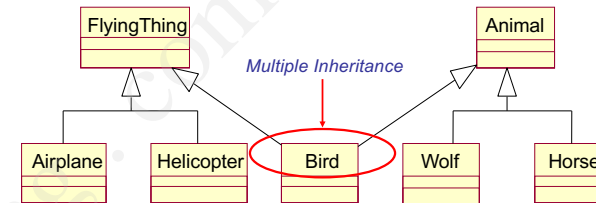
- Một lớp kế thừa từ MỘT lớp khác



33

Ví dụ về Đa kế thừa

- Một lớp có thể kế thừa từ nhiều lớp khác

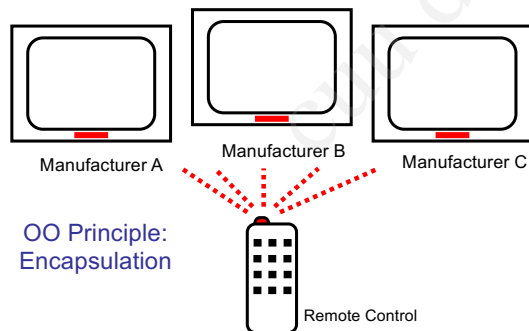


Sử dụng đa kế thừa chỉ khi cần thiết và luôn luôn phải cẩn thận!

34

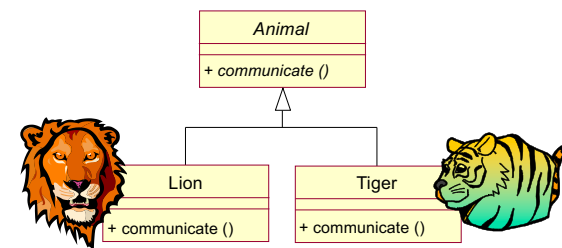
Đa hình (Polymorphism) là gì?

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



35

Tổng quát hóa: Thực thi đa hình



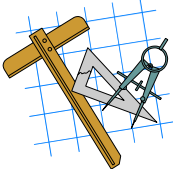
Without Polymorphism
 if animal = "Lion" then
 Lion communicate
 else if animal = "Tiger" then
 Tiger communicate
 end

With Polymorphism
 Animal communicate

36

Bài tập

- Given:
 - A set of classes and their relationships
- Draw:
 - A class diagram



37

- A class diagram containing the following classes: Personal Planner Profile, Personal Planner Controller, Customer Profile, and Buyer Record.
- Associations drawn using the following information:
 - Each Personal Planner Profile object can be associated with up to one Personal Planner Controller object.
 - Each Personal Planner Controller object must be related to one Personal Planner Profile.
 - A Personal Planner Controller object can be associated with up to one Buyer Record and Customer Profile object.
 - An instance of the Buyer Record class can be related to zero or one Personal Planner Controller.
 - Zero or one Personal Planner Controller objects are associated with each Customer Profile instance.

38