

## Unified Modeling Language

Khái niệm về hướng đối tượng và UML

Giảng viên: Mai Thúy Nga



### Nội dung môn học

Giới thiệu về PTTK hướng đối tượng với UML 1	Khái niệm về Hướng đối tượng trong UML 2	Đặc tả Yêu cầu với mô hình Use Case I 3
Đặc tả Yêu cầu với mô hình Use Case II 4	Tổng quan về Phân tích và Thiết kế 5	Phân tích Use Case I 6
Phân tích Use Case II 7	Mô hình hóa Thiết kế 8	Ôn tập 9

Concepts of Object Orientation and UML

2 - 3

### Nội dung trình bày

- ☆ Khái niệm về hướng đối tượng
  - ♦ Các nguyên lý trong hướng đối tượng
  - ♦ Một số khái niệm và ký hiệu trong UML

Concepts of Object Orientation and UML

2 - 4

### Công nghệ hướng đối tượng

- ♦ Là một tập hợp các nguyên lý trong việc xây dựng phần mềm bằng cách sử dụng ngôn ngữ lập trình, cơ sở dữ liệu và các công cụ hỗ trợ công nghệ này
  - (Object Technology: A Manager's Guide, Taylor, 1997)
- ♦ Công nghệ hướng đối tượng không chỉ đơn thuần là ngôn ngữ lập trình hướng đối tượng (OOP)
  - Cần phải có một quy trình, ngôn ngữ mô hình hóa (UML), công cụ và các phương pháp hỗ trợ quá trình mô hình hóa, xây dựng, kiểm thử và triển khai hệ thống

Concepts of Object Orientation and UML

2 - 5

---

---

---

---

---

---

---

### Ưu điểm của Công nghệ hướng đối tượng

- ♦ Cung cấp một cái nhìn thống nhất
  - Ngôn ngữ chung của người sử dụng, người phân tích, thiết kế và cài đặt
- ♦ Kiến trúc hoá bài toán và tăng tính sử dụng lại
- ♦ Mô hình hóa bài toán gần với thế giới thực
  - Mô tả thực thể
  - Phân chia theo quan hệ tự nhiên giữa chúng
- ♦ Tính Ổn định cao
  - Các thay đổi chỉ ảnh hưởng vài thành phần trong hệ thống
- ♦ Dễ phát triển để phù hợp với yêu cầu mới

Concepts of Object Orientation and UML

2 - 6

---

---

---

---

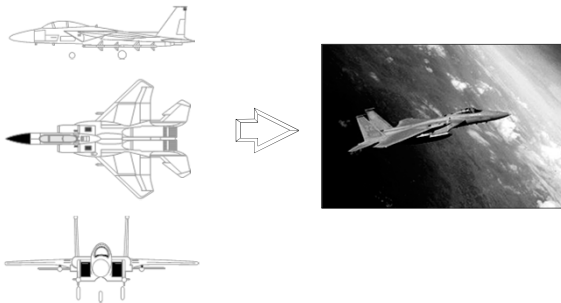
---

---

---

### Mô hình (Model)

- ♦ Là cách đơn giản hoá những gì có trong thực tế



Concepts of Object Orientation and UML

2 - 7

---

---

---

---

---

---

---

## Mục đích của mô hình

- ♦ Giúp chúng ta hiểu hơn về hệ thống sắp phát triển
- ♦ Hiểu rõ về một hệ thống phức tạp là rất khó, mô hình hóa giúp chúng ta đơn giản hóa bài toán và có thể hiểu nó được dễ dàng hơn
- ♦ Mục đích:
  - Cung cấp một cái nhìn về HT mà chúng ta mong muốn
  - Chỉ ra cấu trúc và hành vi của HT
  - Đưa cho một khuôn mẫu (Template) để hướng dẫn chúng ta trong việc phát triển HT
  - Ghi nhận các quyết định, yêu cầu mà chúng ta đưa ra

Concepts of Object Orientation and UML

2 - 8

---

---

---

---

---

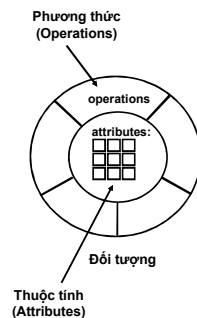
---

---

---

## Khái niệm về đối tượng (Object)

- ♦ Đối tượng là một thực thể được mô tả rõ ràng về vai trò, phạm vi và định danh của nó
- ♦ Một đối tượng bao gồm trạng thái và hành vi:
  - Trạng thái được biểu diễn bởi các thuộc tính và mối quan hệ giữa chúng
  - Hành vi được biểu diễn bởi các thao tác, phương thức



Concepts of Object Orientation and UML

2 - 9

---

---

---

---

---

---

---

---

## Trạng thái của đối tượng

- ♦ Trạng thái của đối tượng là một trong những điều kiện mà đối tượng có thể tồn tại
- ♦ Trạng thái này thường thay đổi theo thời gian



Tên: Biệt tuốt  
Mã: 567138  
Ngày tuyển dụng: 25/07/1991  
Trạng thái: Cơ hữu  
Môn dạy: UML  
Số lớp có thể dạy đồng thời: 3



Giáo viên "Biệt tuốt"

Concepts of Object Orientation and UML

2 - 10

---

---

---

---

---

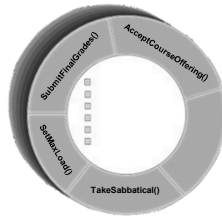
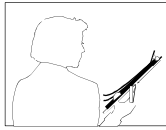
---

---

---

## Hành vi của đối tượng

- ♦ Hành vi chỉ ra đối tượng có thể hoạt động và tương tác như thế nào
- ♦ Các hành vi của 1 đối tượng được mô hình hoá bằng tập các thông điệp mà nó có thể được kích hoạt (hàm)



Các hành động của giáo viên "Biết tốt"

- . Nộp điểm
- . Xác nhận lớp sẽ dạy
- . Nghỉ phép
- . Xác nhận số lớp có thể dạy đồng thời

Concepts of Object Orientation and UML

2 - 11

---

---

---

---

---

---

---

---

## Định danh của đối tượng

- ♦ Mỗi đối tượng có một định danh duy nhất ngay cả khi trạng thái của nó giống hệt với trạng thái của các đối tượng khác



Giáo viên "Biết tốt"  
dạy môn UML



Giáo viên "Biết tốt"  
dạy môn UML

Concepts of Object Orientation and UML

2 - 12

---

---

---

---

---

---

---

---

## Biểu diễn các đối tượng trong UML

- ♦ Mỗi đối tượng được biểu diễn trong UML bằng 1 ký hiệu là hình chữ nhật với định danh được gạch dưới
  - Thực tế, ký hiệu có thể thay đổi tùy theo vai trò của đối tượng đó



Giáo viên "Biết tốt"

Biet\_tuot : Professor

Đối tượng đã có tên

: Professor

Đối tượng chưa có tên

Concepts of Object Orientation and UML

2 - 13

---

---

---

---

---

---

---

---

## Nội dung trình bày

- ♦ Khái niệm về hướng đối tượng
- ☆ Các nguyên lý trong hướng đối tượng
- ♦ Một số khái niệm và ký hiệu trong UML

Concepts of Object Orientation and UML

2 - 14

---

---

---

---

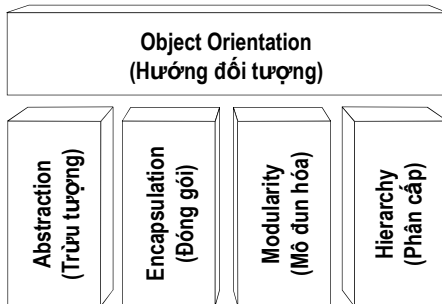
---

---

---

---

## Các nguyên lý cơ bản trong hướng đối tượng (OO)



Concepts of Object Orientation and UML

2 - 15

---

---

---

---

---

---

---

---

## Khái niệm về Trừu tượng hóa

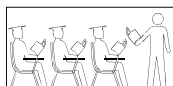
- ♦ Các đặc điểm cơ bản của một thực thể để phân biệt được nó với các thực thể khác



Sinh viên



Giáo viên



Lớp học  
(UML, Tiết 4, 5, Thứ 2, 4, 6)



Môn học (UML, CNPM)

Concepts of Object Orientation and UML

2 - 16

---

---

---

---

---

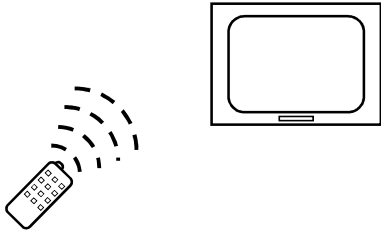
---

---

---

## Khái niệm về Đóng gói

- ◆ Che dấu sự cài đặt (lập trình) với thành phần tương tác
  - Chỉ thể hiện thông qua giao diện



Object Orientation (Hướng đối tượng)  
 Abstraction (Trừu tượng)  
 Encapsulation (Đóng gói)  
 Inheritance (Di truyền)  
 Polymorphism (Đa hình)

Concepts of Object Orientation and UML 2 - 17

---

---

---

---

---

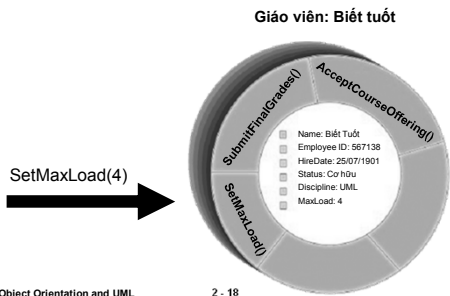
---

---

---

## Minh hoạ đóng gói

- ◆ Giáo viên “Biết tuốt” có thể dạy 4 lớp trong kì sắp tới



Giáo viên: Biết tuốt

Attributes:

- Name: Biết Tuốt
- Employee ID: 567138
- HireDate: 25/07/1901
- Status: Có hưu
- Discipline: UML
- MaxLoad: 4

Methods:

- SubmitFinalGrade()
- AcceptCourseOffering()
- SetMaxLoad()

SetMaxLoad(4)

Concepts of Object Orientation and UML 2 - 18

---

---

---

---

---

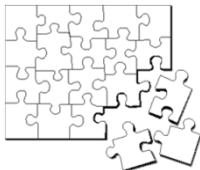
---

---

---

## Khái niệm về Module hóa

- ◆ Mô đun hoá là việc tách một sự việc phức tạp thành những mảnh nhỏ có thể quản lý được
- ◆ Mô đun hoá giúp mọi người hiểu được một hệ thống phức tạp dễ dàng hơn



Object Orientation (Hướng đối tượng)  
 Abstraction (Trừu tượng)  
 Encapsulation (Đóng gói)  
 Inheritance (Di truyền)  
 Polymorphism (Đa hình)

Concepts of Object Orientation and UML 2 - 19

---

---

---

---

---

---

---

---

**Module hóa: Ví dụ**

♦ Ví dụ tách một hệ thống phức tạp thành các module nhỏ hơn

The diagram illustrates the decomposition of a large puzzle piece labeled 'Hệ thống quản lý khóa học' into three smaller puzzle pieces labeled 'Hệ thống thanh toán', 'Hệ thống quản lý môn học', and 'Hệ thống quản lý sinh viên'.

Concepts of Object Orientation and UML 2 - 20

---

---

---

---

---

---

---

---

**Khái niệm về Phân cấp**

The diagram shows a hierarchy of assets (Tài sản) categorized into three main types: Tài khoản ngân hàng, Thẻ chấp, and Bất động sản. These are further subdivided into smaller categories: Tiết kiệm, Séc, Cổ phần, and Trái phiếu. A legend on the right indicates the levels of abstraction: Tài sản (highest), Tài khoản ngân hàng, Thẻ chấp, Bất động sản, and then the sub-categories (lowest).

Tăng mức độ trừu tượng

Giảm mức độ trừu tượng

Các thành phần ở cùng một mức phân cấp sẽ có mức độ trừu tượng hóa giống nhau.

Concepts of Object Orientation and UML 2 - 21

---

---

---

---

---

---

---

---

**Nội dung trình bày**

- ♦ Khái niệm về hướng đối tượng
- ♦ Các nguyên lý trong hướng đối tượng

☆ Một số khái niệm và ký hiệu trong UML

Concepts of Object Orientation and UML 2 - 22

---

---

---

---

---

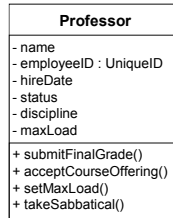
---

---

---

## Khái niệm về Lớp (Class)

- ♦ Một lớp là một tập các đối tượng có cùng các thuộc tính, thao tác, mối quan hệ
  - Một đối tượng là thể hiện của một lớp
- ♦ Một lớp là sự trừu tượng hoá mà trong đó:
  - Nhấn mạnh các đặc điểm quan trọng
  - Bỏ qua các đặc điểm không cần thiết khác
- ♦ Thường được biểu diễn bởi hình chữ nhật chia ngăn



Concepts of Object Orientation and UML

2 - 23

---

---

---

---

---

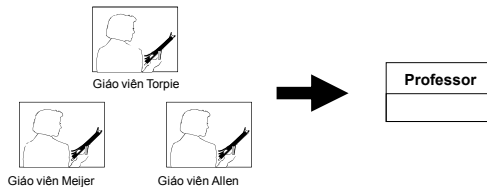
---

---

---

## Mối quan hệ giữa lớp và đối tượng

- ♦ Lớp là khái niệm trừu tượng của đối tượng
  - Nó định nghĩa cấu trúc và hành vi của mỗi đối tượng trong lớp
  - Nó cung cấp 1 mẫu để tạo các đối tượng
- ♦ Đối tượng là thể hiện của một lớp



Concepts of Object Orientation and UML

2 - 24

---

---

---

---

---

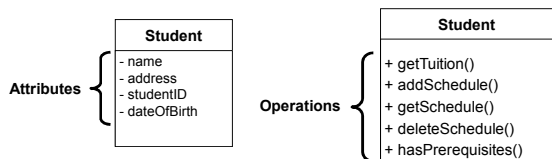
---

---

---

## Thuộc tính và Phương thức

- ♦ Thuộc tính - đặc tính được đặt tên của 1 lớp
  - Đối tượng sẽ chứa giá trị cụ thể cho các thuộc tính
- ♦ Phương thức – 1 dịch vụ mà lớp cung cấp



Concepts of Object Orientation and UML

2 - 25

---

---

---

---

---

---

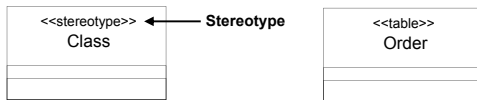
---

---



## Khái niệm về Stereotypes

- ♦ Stereotypes định ra một thành phần mô hình mới mà các mô hình chuẩn chưa cung cấp
- ♦ Có thể được coi như là kiểu "Đặc biệt"
- ♦ Thường được sử dụng để minh họa như là một lớp đặc biệt

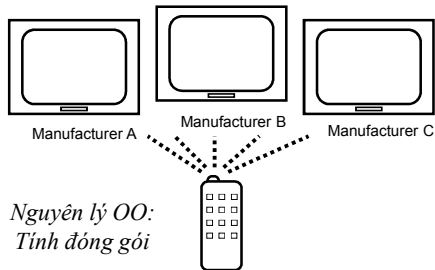


Concepts of Object Orientation and UML

2 - 26

## Khái niệm về Tính đa hình

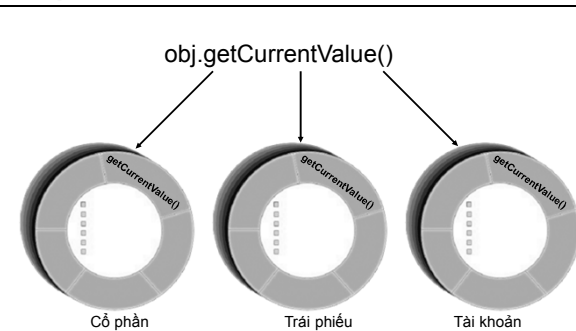
- ♦ Là cách che dấu các cài đặt khác nhau trong 1 giao diện



Concepts of Object Orientation and UML

2 - 27

## Ví dụ về tính đa hình

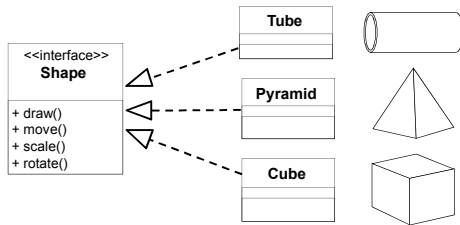


Concepts of Object Orientation and UML

2 - 28

## Khái niệm về Giao diện

- ♦ Giao diện là cách thức tìm cách thể hiện một loạt các hành vi mà chưa cung cấp cài đặt cho nó
- ♦ Giao diện cung cấp tính đa hình cho một nhóm đối tượng và hỗ trợ kiến trúc "Plug & Play"



Concepts of Object Orientation and UML

2 - 29

---

---

---

---

---

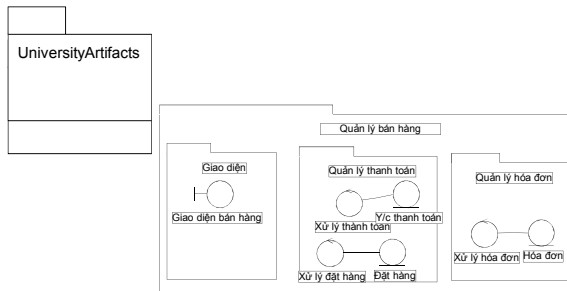
---

---

---

## Khái niệm về Gói (package)

- ♦ Gói là một cách thức gộp một số thành phần khác nhau trong cùng một nhóm



Concepts of Object Orientation and UML

2 - 30

---

---

---

---

---

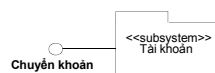
---

---

---

## Khái niệm về Hệ thống con (Subsystem)

- ♦ Subsystem là việc nhóm các thành phần khác nhau như lớp, thành phần, subsystem khác như package.
- ♦ Tuy nhiên Subsystem thể hiện một hành vi nào đó cho hệ thống thực
  - Subsystem = Gói + Hành vi (Package + Behavior)
- ♦ Subsystem thực hiện một hoặc nhiều interface để thể hiện hành vi đó của hệ thống



Concepts of Object Orientation and UML

2 - 31

---

---

---

---

---

---

---

---

## Khái niệm về Thành phần (Component)

- ♦ Là một phần gần như độc lập và có thể thay thế bởi một thành phần khác mà không ảnh hưởng đến toàn bộ hệ thống
- ♦ Cung cấp 1 chức năng tương đối hoàn chỉnh cho hệ thống

Thành phần mã nguồn	
Thành phần thư viện	
Thành phần mã nhị phân	

Concepts of Object Orientation and UML 2 - 32

---

---

---

---

---

---

---

---

## Subsystem và Component

- ♦ Hệ thống con mang tính khái niệm, mô tả hoạt động của một hành vi (chức năng) cụ thể của hệ thống
  - Được sử dụng trong giai đoạn thiết kế để thể hiện một thành phần
- ♦ Thành phần mang tính thực tế, vật lý
  - Được phát triển từ thực thể trừu tượng trong quá trình thiết kế

Mô hình thiết kế

Mô hình thực thi

Concepts of Object Orientation and UML 2 - 33

---

---

---

---

---

---

---

---

## Relationship, Multiplicity, Navigability

- ♦ Quan hệ (Relationship)
  - Các đối tượng kết nối đến các đối tượng khác như thế nào
- ♦ Bội số (Multiplicity)
  - Bội số cho phép chỉ ra số lượng của một đối tượng cần thiết để quan hệ với số lượng của một đối tượng khác
- ♦ Chiều quan hệ (Navigability)
  - Lớp nào biết lớp nào trong quan hệ hiện có

Concepts of Object Orientation and UML 2 - 34

---

---

---

---

---

---

---

---

## Bội số của quan hệ (Multiplicity)

- ♦ Bội số cho phép chỉ ra số lượng của 1 đối tượng cần thiết để quan hệ với số lượng của 1 đối tượng khác

Không cụ thể	
Một	1
Không hoặc nhiều	0..*
Không hoặc nhiều	*
Một hoặc nhiều	1..*
Không hoặc một	0..1
Khoảng cách cụ thể	2..4
Khoảng cách cụ thể	2, 4..6

Concepts of Object Orientation and UML

2 - 35

---

---

---

---

---

---

---

---

## Các kiểu quan hệ: Aggregation, Dependency, Generalization

- ♦ Quan hệ tập hợp (association)
  - Lớp thành phần có thể vẫn tồn tại nếu lớp chứa bị mất đi
- ♦ Quan hệ kết hợp (aggregation/composition, "is part of")
  - Lớp thành phần sẽ bị mất đi nếu lớp chứa nó mất đi
- ♦ Quan hệ phụ thuộc (dependency, "uses")
  - Quan hệ sử dụng (phụ thuộc)
- ♦ Quan hệ tổng quát hóa (generalization, "is kind of")
  - Một lớp cha có thể thiết lập những gì chung nhất cho một nhóm các lớp con "kế thừa" nó

Concepts of Object Orientation and UML

2 - 36

---

---

---

---

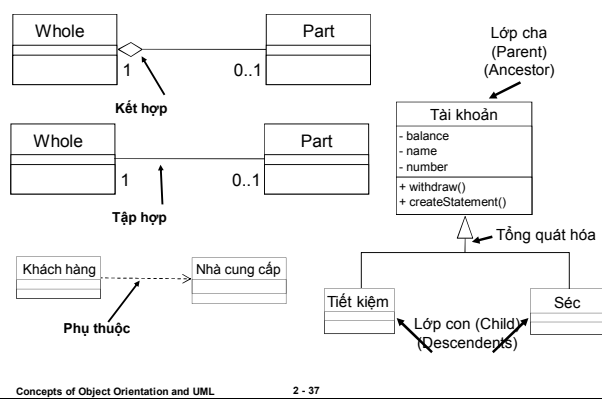
---

---

---

---

## Relationships: Aggregation, Dependency, Generalization



Concepts of Object Orientation and UML

2 - 37

---

---

---

---

---

---

---

---

## Khái niệm về Kế thừa

- ♦ Kế thừa các thuộc tính, hàm và quan hệ
- ♦ Lớp con (subclass) có thể:
  - Thêm các thuộc tính, hàm và quan hệ
  - Có thể thiết lập lại các hàm (!!!)
- ♦ Các thuộc tính, hàm, quan hệ chung nhất nên được thể hiện ở mức độ cao nhất có thể trong quan hệ phân cấp

2 - 38

---

---

---

---

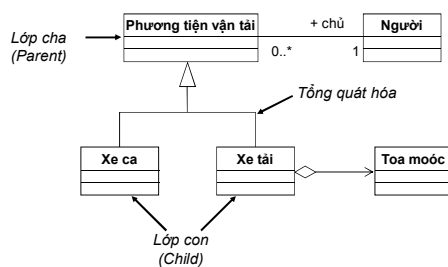
---

---

---

---

## Ví dụ về Kế thừa



Concepts of Object Orientation and UML

2 - 39

---

---

---

---

---

---

---

---

## Khái niệm về Note

- ♦ Có thể thêm các lời chú giải, nhận xét để bổ sung thêm các thông tin vào các mô hình và sơ đồ
- ♦ Có thể thêm vào với bất kỳ thành phần nào trong UML



2 - 40

---

---

---

---

---

---

---

---

### Tóm tắt

- ♦ Các nguyên lý trong hướng đối tượng
  - Trừu tượng
    - Các đặc điểm cơ bản để phân biệt thực thể này với các thực thể khác
  - Đóng gói
    - Che dấu sự cài đặt (lập trình) với thành phần tương tác
  - Module hóa
    - Tách một sự việc phức tạp thành những mảnh nhỏ có thể quản lý được
  - Phân cấp
    - Thể hiện mức độ phức tạp của bài toán thành các cấp độ khác nhau
- ♦ Thể hiện các thành phần OO bằng các sơ đồ UML

Concepts of Object Orientation and UML

2 - 41

---

---

---

---

---

---

---