

# **PHÂN TÍCH THIẾT KẾ HỆ THỐNG THÔNG TIN**

## **Bài 3. UML**

**Giáo viên: TS. Trần Mạnh Tuấn**

**Bộ môn: Hệ thống thông tin**

**Khoa: Công nghệ thông tin**

**Email: [tmtuan@tlu.edu.vn](mailto:tmtuan@tlu.edu.vn)**

**Điện thoại: 0983.668.841**

# Nội dung

- ❖ UML - Overview
- ❖ UML – Khối xây dựng (Building Block)
- ❖ UML – Kiến trúc (Architecture)
- ❖ UML – Các loại mô hình hóa (Modeling Types)
- ❖ UML – Các ký hiệu căn bản (Basic Notations)
- ❖ UML – Các biểu đồ chuẩn (Standard Diagrams)

# Tổng quan về UML

- ❖ UML – Ngôn ngữ chuẩn cho đặc tả, hình ảnh hóa, xây dựng, và mô tả tài liệu cho các tác nhân của hệ thống phần mềm.
- ❖ UML – được tạo bởi Nhóm Quản lý Đối tượng (Object Management Group – OMG), phiên bản UML 1.0 được đưa ra vào tháng 01/1997.
- ❖ OMG tiếp tục phát triển UML như một tiêu chuẩn công nghiệp phần mềm thực sự.

# Tổng quan về UML

## ❖ Đặc điểm UML:

- UML viết tắt của Unified Modeling Language
- UML khác với ngôn ngữ lập trình: C/C++, Java, PHP, Cobol,...
- UML là ngôn ngữ hình ảnh sử dụng tạo ra các bản kế hoạch thiết kế.
- UML có thể được mô tả như ngôn ngữ mô hình hóa hình ảnh đa năng để hình ảnh hóa, đặc tả, xây dựng và văn bản hóa hệ thống phần mềm
- Mặc dù UML chuyên dành cho thiết kế, mô hình hóa phần mềm, nhưng nó có thể dùng để mô hình hóa các hệ thống không phải phần mềm. Ví dụ: luồng nghiệp vụ cho một đơn vị sản xuất.

# Tổng quan về UML

## ❖ Mục đích của UML:

- “A picture is worth a thousand words” – UML.
- Mục đích chính là UML như là cơ chế mô hình hóa đơn giản để mô hình tất cả các hệ thống tiện ích của môi trường thực tế phức tạp hiện nay.

# Tổng quan về UML

- ❖ Mô hình khái niệm là gì?
- ❖ Vì sao cần Mô hình khái niệm?

# Tổng quan về UML

- ❖ Mô hình khái niệm – mô hình được tạo lên bởi các khái niệm và các quan hệ giữa chúng.
- ❖ Mô hình khái niệm là bước đầu tiên vẽ các biểu đồ UML. Nó giúp hiểu được các thực thể trong thế giới thực và chúng tương tác lẫn nhau như thế nào.
- ❖ Để UML mô tả thế giới thực, nó cần phải tạo ra mô hình khái niệm và sau đó xử lý dần dần từng bước.
- ❖ Mô hình khái niệm của UML có thể mô tả 3 khái niệm chính:
  - Các khối xây dựng UML (UML Building Blocks)
  - Luật liên kết các khối xây dựng UML
  - Cơ chế chung của UML

# Tổng quan về UML

## ❖ Các khái niệm Hướng đối tượng

- Objects & Class
- Abstraction – Encapsulation – Inheritance – Polymorphism

## ❖ OO Analysis and Design

## ❖ Vai trò của UML trong OO Design

- UML là ngôn ngữ mô hình hóa các hệ thống phần mềm và hệ thống chung
- OOD là chuyển đổi thành các biểu đồ UML dựa trên các yêu cầu người dùng.



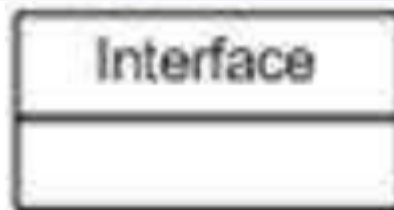
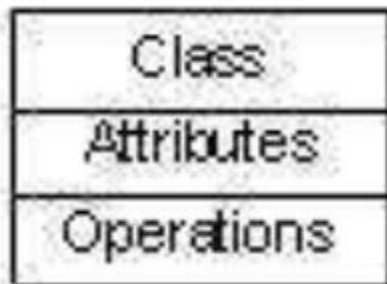
# Tổng quan về UML

## ❖ Building Blocks – bao gồm:

- Đồ vật (Things)
  - Cấu trúc (Structural)
  - Hành vi (Behavioral)
  - Nhóm (Grouping)
  - Lời giải thích (Annotation)
- Các mối quan hệ:
  - Phụ thuộc (Dependency)
  - Liên kết (Association)
  - Tổng quan (Generalization)
  - Hiện thực hóa (Realization)
- Biểu đồ:
  - Lớp (Class Diagram)
  - Đối tượng (Object Diagram)
  - Trường hợp sử dụng (Use case Diagram)
  - Liên tiếp (Sequence Diagram)
  - Hợp tác (Collaborate Diagram)
  - Hành động (Activity Diagram)
  - Biểu đồ trạng thái (Statechart Diagram)
  - Triển khai (Deploy Diagram)
  - Thành phần (Component Diagram)

# UML – Building Blocks

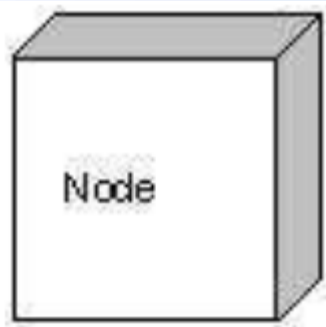
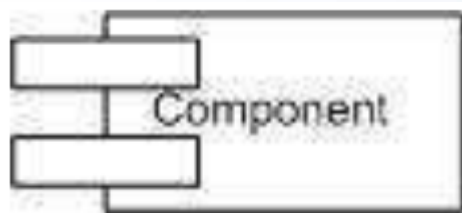
- **Đồ vật (Things) - Cấu trúc (Structural):** là phần tĩnh của mô hình. Bao gồm:
  - **Lớp (Class):** thể hiện 1 tập các đối tượng có cùng trách nhiệm tương tự nhau.
  - **Giao diện (Interface):** Thể hiện mộ tập các hoạt động chỉ ra các trách nhiệm của một lớp.
  - **Use case:** đại diện tập các hành động được thể hiện bởi hệ thống cho một mục đích nào đấy.



# UML – Building Blocks - Things

- ❖ **Thành phần (Component):** mô tả như là thành phần vật lý của hệ thống.
- ❖ **Node:** là phần tử vật lý tồn tại trong khi hệ thống đang chạy.

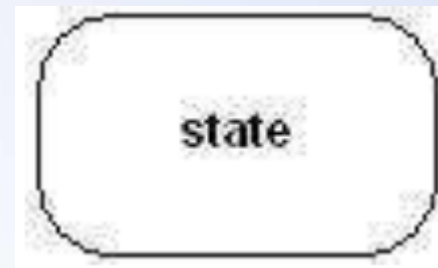
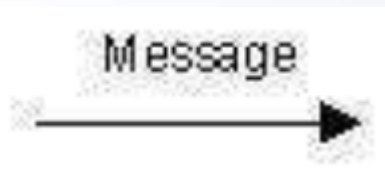
Sự hợp tác (Collaboration): xác định tương tác giữa các phần tử



# UML – Building Blocks - Things

## ❖ Hành vi

- Tương tác (Interaction): là một hành vi bao gồm nhóm các thông điệp (Messages) trao đổi giữa các phần tử để hoàn thành một nhiệm vụ nhất định
- Máy trạng thái (State machine): thể hiện các trạng thái của đối tượng trong vòng đời của đối tượng, mà các trạng thái này là quan trọng.



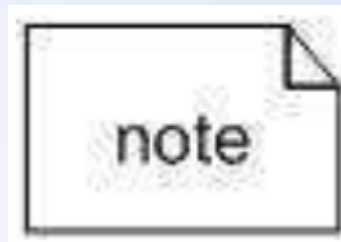
# UML – Building Blocks - Grouping

- ❖ **Nhóm (Grouping)** là cơ chế nhóm các phần tử của mô hình UML vào với nhau. Chỉ có một hình ảnh nhóm là Gói (Package).
- Package: là tập trung các cấu trúc và hành vi của một vật (Thing) vào với nhau



# UML – Building Blocks - Annotation

- ❖ **Note:** Đưa ra các lời chú giải, giảng buộc,... của một phần tử UML



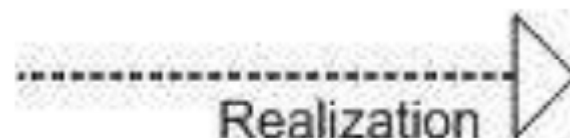
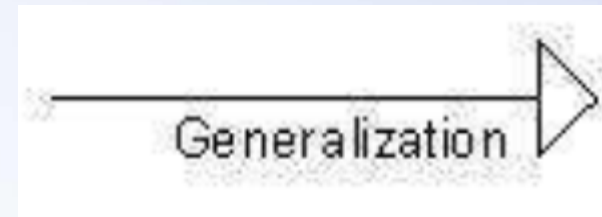
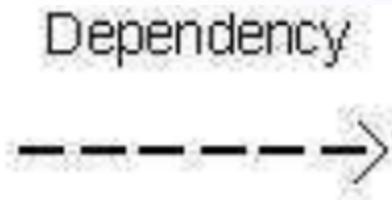
# UML – Building Blocks - Relationships

- ❖ **Các mối quan hệ (Relationships):** là một trong những Building Block quan trọng nhất của UML. Nó thể hiện các phần tử quan hệ với nhau như thế nào, và mỗi quan hệ này mô tả chức năng của ứng dụng.
- ❖ **Có bốn kiểu quan hệ:**
  - Phụ thuộc (Dependency)
  - Liên kết (Association)
  - Tổng quát hóa (Generalization)
  - Hiện thực hóa (Realization)



# UML – Building Blocks - Relationships

- ❖ Phụ thuộc – Dependency: là quan hệ khi một phần tử thay đổi cũng ảnh hưởng tới phần tử kia
- ❖ Liên kết – Association: căn bản là các mối liên kết các phần tử trong mô hình UML.
- ❖ Tổng quan hóa – Generalization: thể hiện liên kết một phần tử nhất định và phần tử sinh ra nó.
- ❖ Hiện thực hóa – Realization: thể hiện mối quan hệ hai phần tử liên kết. Một phần tử mô tả các trách nhiệm, nhưng không được cài đặt, còn phần tử còn lại cài đặt các trách nhiệm này. Interface là một trong các mqh này.





# Kiến trúc UML (UML – Architecture)

- ❖ **Kiến trúc chung:** Trước khi thiết kế cần phải đứng ở các phương diện người dùng khác nhau: LTV(Devevelopers), Testers, Kinh doanh (Business people), Nhà phân tích (analysts),... Dựa trên các quan điểm đó nên có kiến trúc chung của việc phát triển phần mềm:
  - Thiết kế (Design)
  - Cài đặt (Implementation)
  - Quy trình nghiệp vụ (Process)
  - Triển khai (Deployment)

# Kiến trúc UML (UML – Architecture)

- ❖ Design (thiết kế): Bao gồm các Lớp (Classes), Giao diện (Interfaces), và sự hợp tác (Collaborations) - Class Diagram và Object Diagram.
- ❖ Implementation (thực hiện): Các thành phần được kết hợp với nhau để tạo ra hệ thống vật lý – Component Diagram.
- ❖ Process (tiến trình): mô tả các luồng nghiệp vụ của hệ thống – Có một số phần tử dùng trong UML biểu diễn quan điểm này.
- ❖ Deployment (triển khai): Thể hiện các node vật lý của hệ thống – phần cứng – Deployment diagram.

# Một số mô hình trong UML

## ❖ Có ba loại mô hình hóa:

- Mô hình cấu trúc: Tính năng tĩnh của hệ thống.
  - Biểu đồ lớp
  - Biểu đồ đối tượng
  - Biểu đồ triển khai
  - Biểu đồ đóng gói
  - Biểu đồ cấu trúc hỗn hợp
  - Biểu đồ thành phần

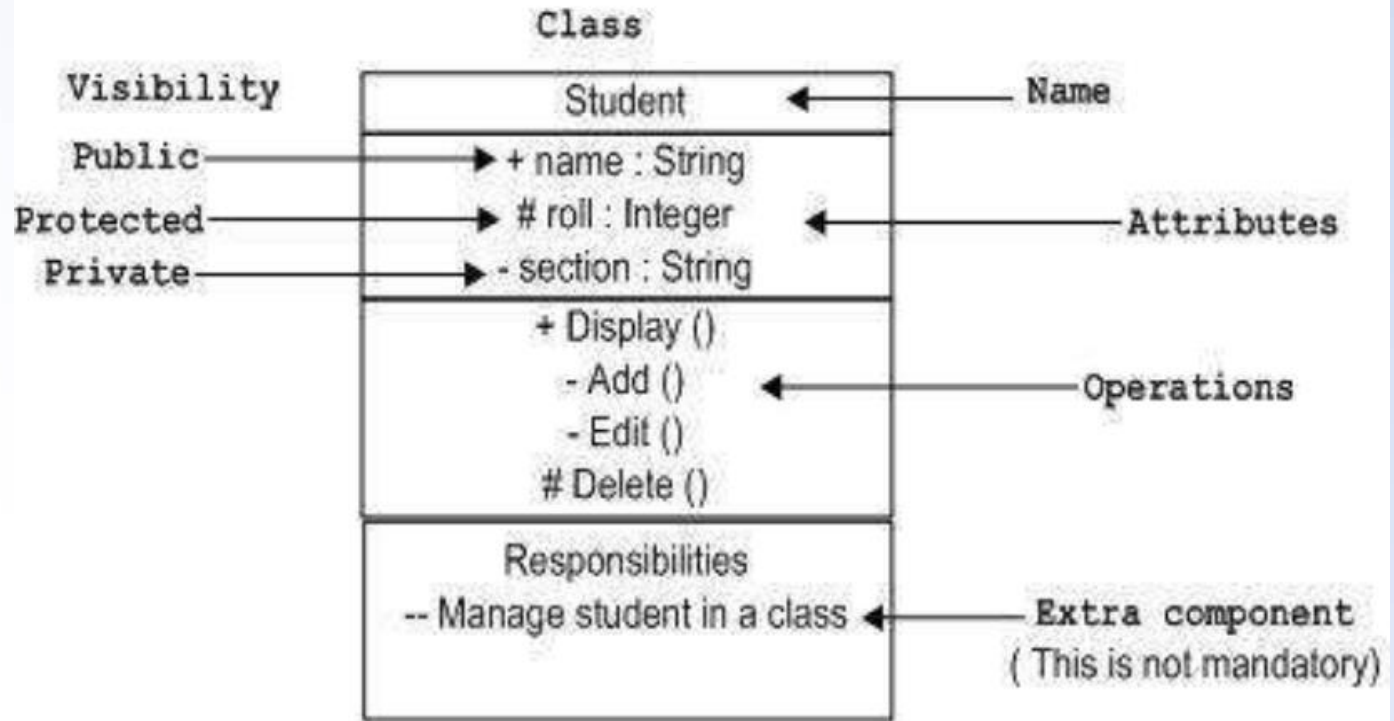
# Một số mô hình trong UML

- Mô hình hành vi: Tính năng tương tác của hệ thống
  - Biểu đồ hành động
  - Biểu đồ tương tác
  - Biểu đồ use case
- Mô hình kiến trúc: thể hiện khung chung cho toàn bộ hệ thống.
  - Biểu đồ đóng gói (Package Diagram)

# Các ký hiệu cơ bản trong UML

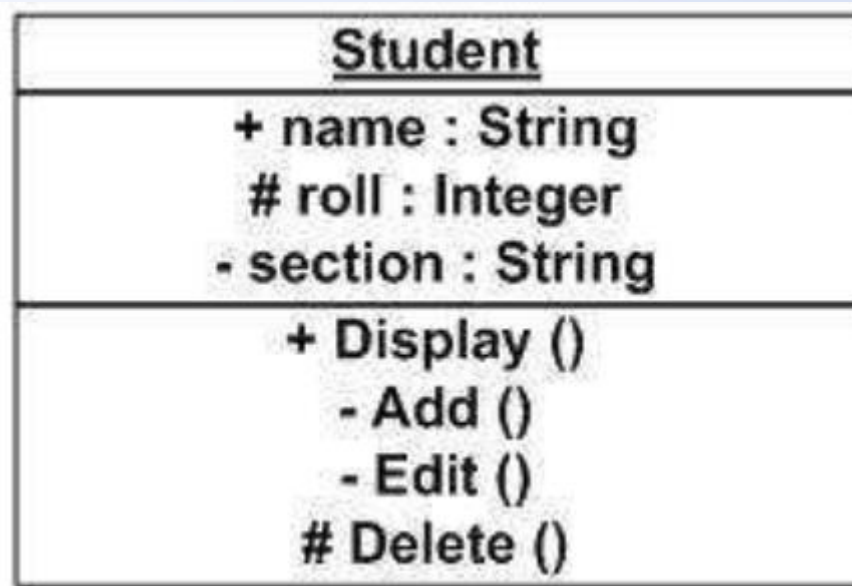
## ❖ Structural Things:

- Class: thể hiện cho đối tượng. Các đối tượng có thể có các thuộc tính và hành vi.



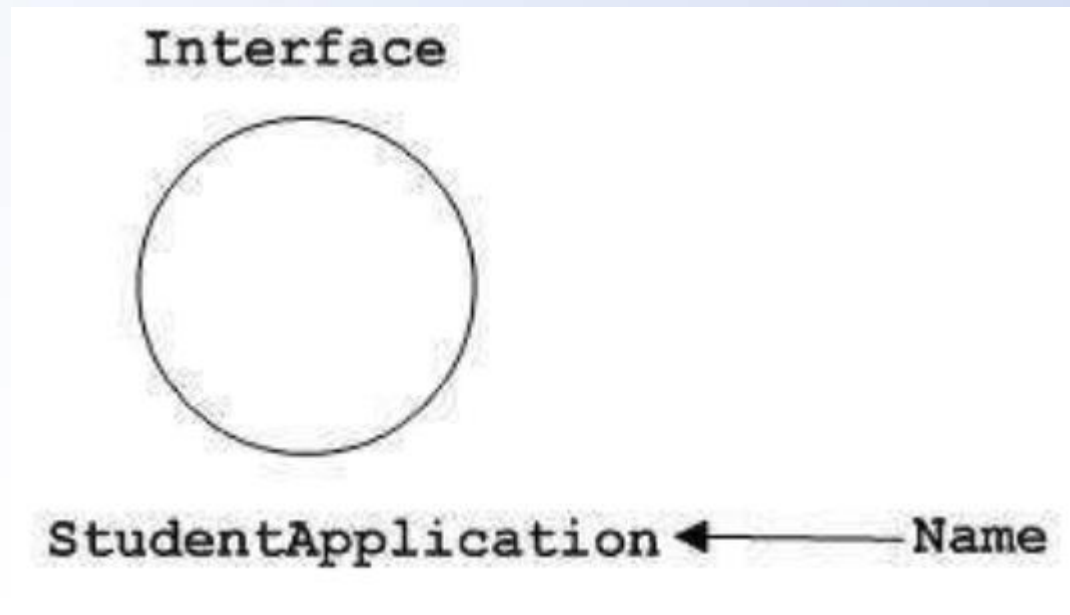
# Các ký hiệu cơ bản trong UML

- ❖ **Kí hiệu đối tượng (Object notation):**
  - Đối tượng: là sự cài đặt thực tế của lớp.



# Các ký hiệu cơ bản trong UML

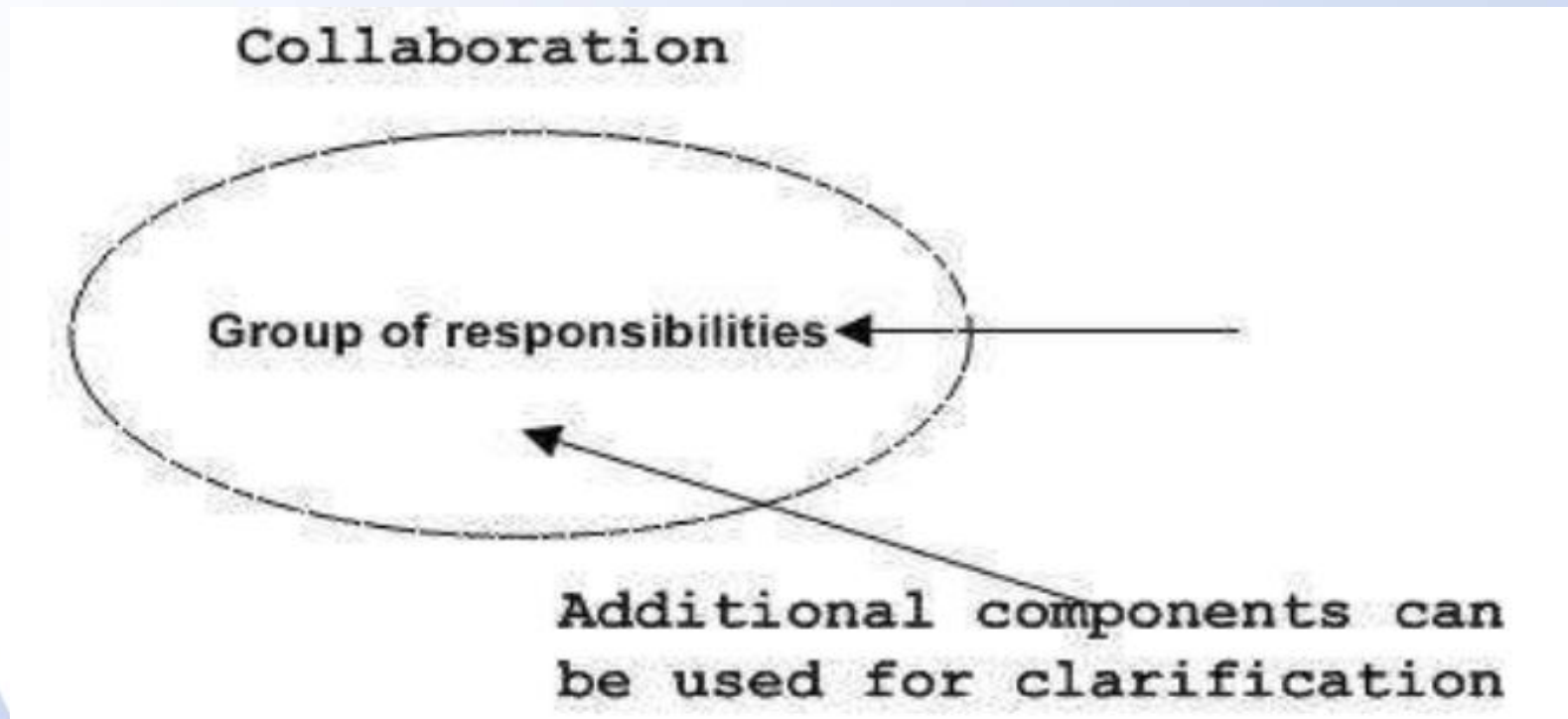
- ❖ Kiểu hiệu giao diện (interface Notation) – Thể hiện các chức năng chưa được cài đặt.





# Các ký hiệu cơ bản trong UML

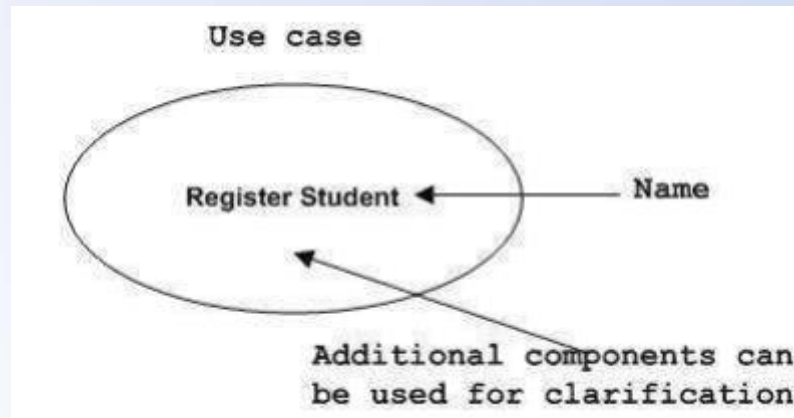
- ❖ Kiểu hiệu hợp (Collaboration Notation) – Thể hiện hành vi trách nhiệm, thường trong nhóm.



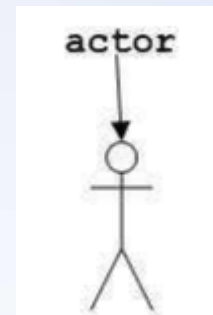


# Các ký hiệu cơ bản trong UML

## ❖ Use case

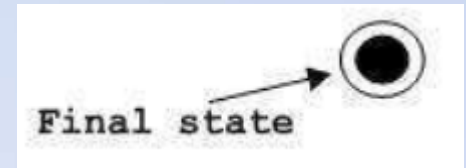
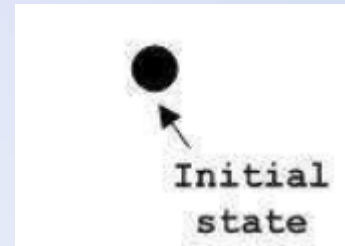


## ❖ Actor

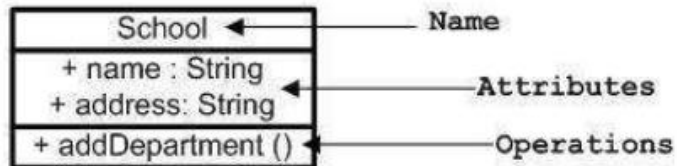


# Các ký hiệu cơ bản trong UML

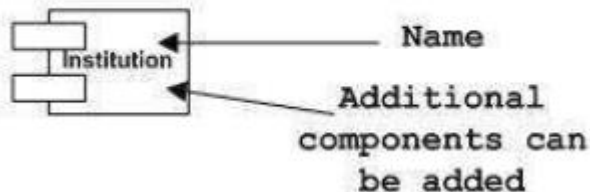
- ❖ Điểm khởi đầu (Initial state)
- ❖ Điểm kết thúc (Final state)
- ❖ Lớp hoạt động (Active Class)
- ❖ Component
- ❖ Node



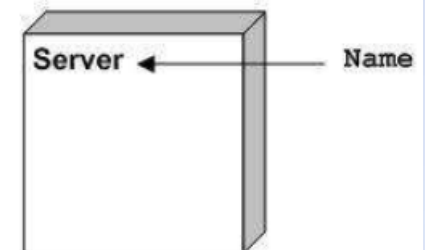
Active Class



Component



Node



# Các ký hiệu cơ bản trong UML

## ❖ Hành vi – (Behavioral Things)

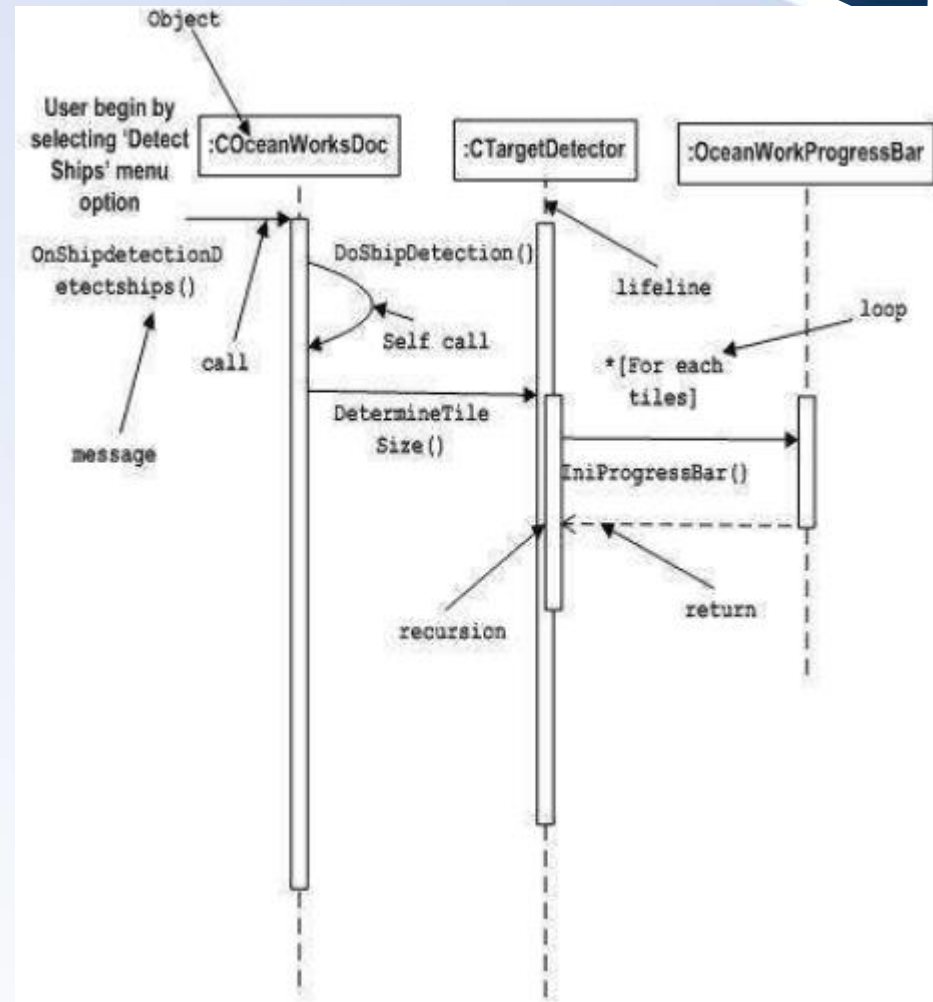
**Phần động của UML. Có hai thành phần động chính quan trọng trong UML:**

## ❖ Các tương tác(Interactions):

- Tuần tự (Sequential) thể hiện bởi biểu đồ Tuần tự (Sequential Diagram)
- Hợp tác (Collaboration) – Collaboration Diagram

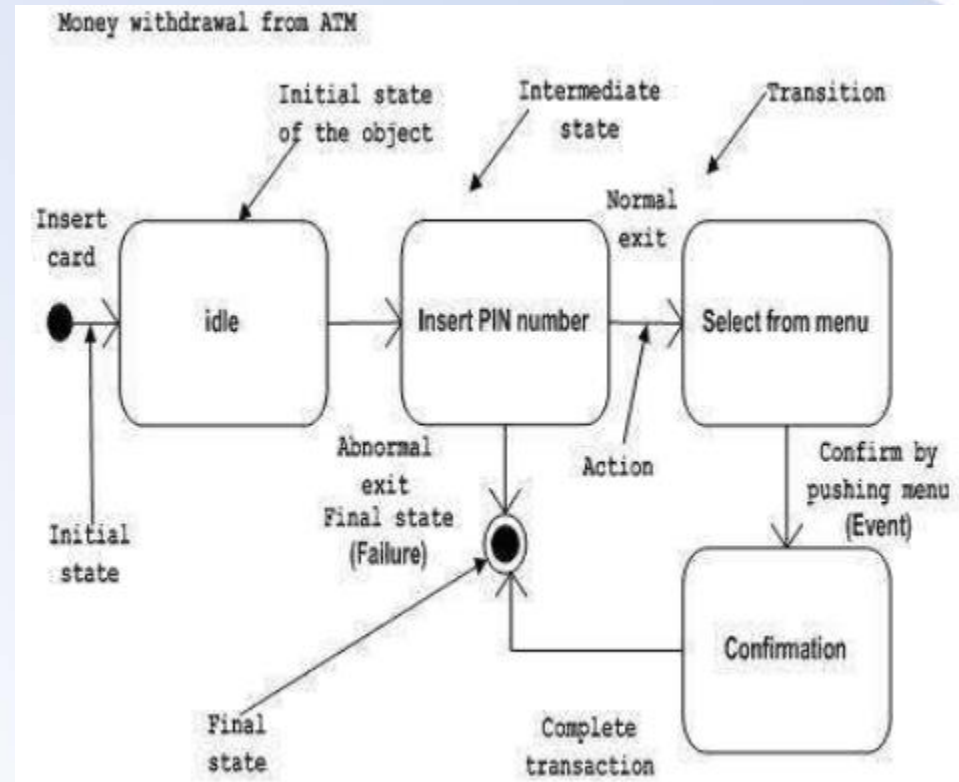
# Các ký hiệu cơ bản trong UML

- ❖ Các tương tác căn bản là các thông điệp giữa hai thành phần của UML.



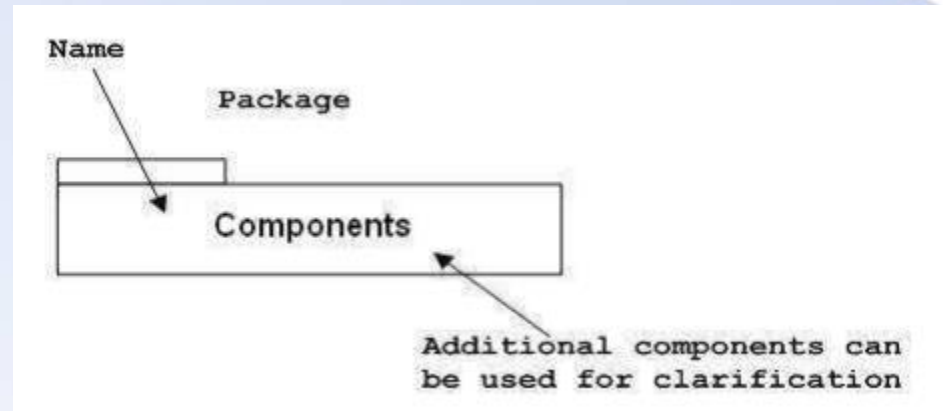
# Các ký hiệu cơ bản trong UML

- ❖ Statemachine là mô tả các trạng thái khác nhau của một thành phần trong vòng đời của nó.

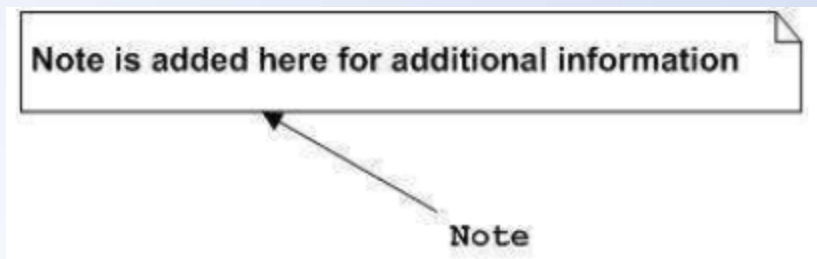


# Tính năng trong UML (UML – Grouping Things)

## ❖ Grouping Thing



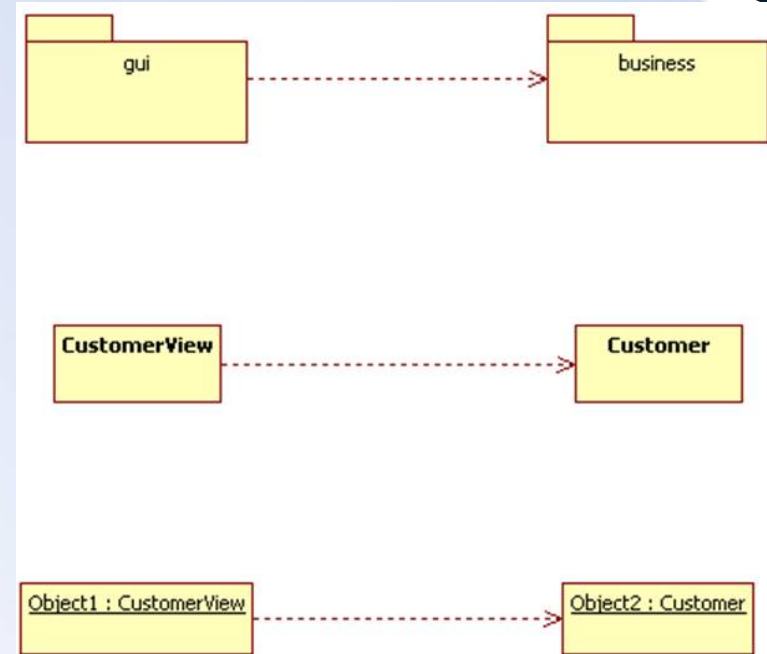
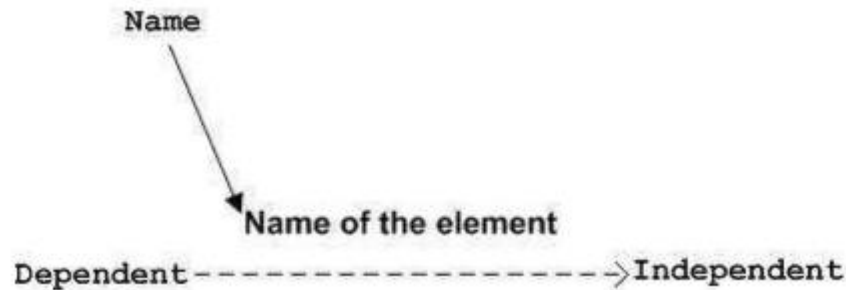
## ❖ Annotation Thing



# Tính năng trong UML (UML – Grouping Things)

## ❖ Relationships

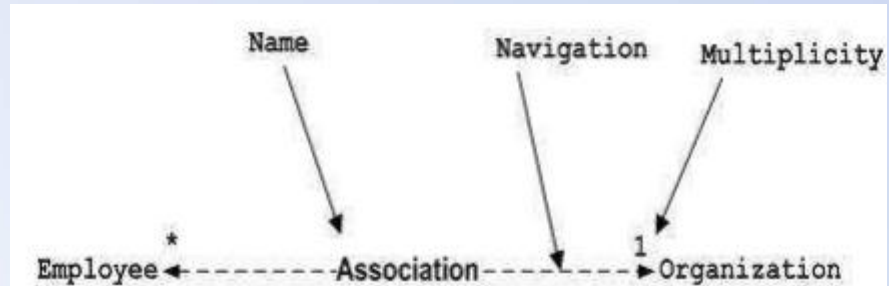
- Phụ thuộc (Dependency): thể hiện sự phụ thuộc giữa hai phần tử của một hệ thống.





# Tính năng trong UML (UML – Grouping Things)

❖ **Kết hợp (Association):** mối quan hệ chung giữa hai phần tử UML



❖ **Tổng quát (Generalization):** mối quan hệ cha con của hai phần tử UML

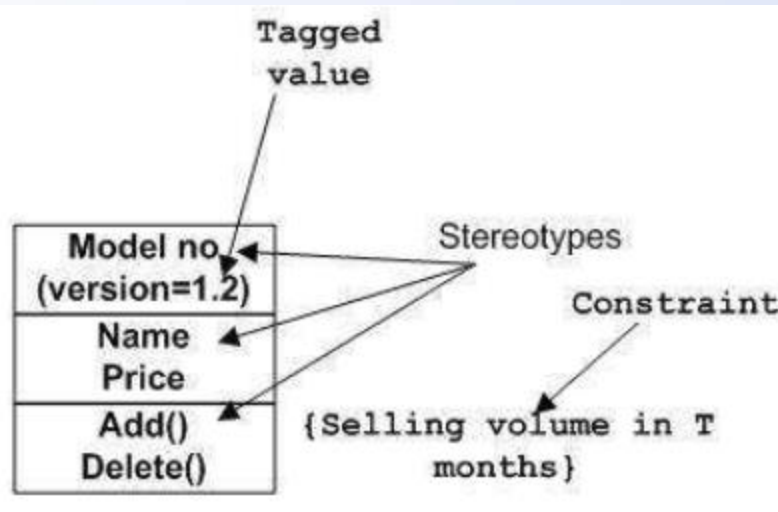




# Tính năng trong UML (UML – Grouping Things)

## ❖ Các ký hiệu mở rộng (Extensibility Notations)

- Stereotypes – thể hiện một phần tử mới
- Tagged values – thể hiện thuộc tính mới
- Constraints – thể hiện các giới hạn



**Trao đổi, câu hỏi?**