



## Sơ đồ UML (Ngôn ngữ Mô hình hóa Thống nhất)

---

UML là một ngôn ngữ mô hình hóa tiêu chuẩn được sử dụng để trực quan hóa, đặc tả, xây dựng và tài liệu hóa các tạo phẩm của một hệ thống phần mềm.





## Sơ đồ Lớp (Class Diagram):

---

- Cho thấy cấu trúc của hệ thống về các lớp, thuộc tính, phương thức/thao tác và các mối quan hệ giữa các lớp.
- Biểu diễn tổ chức nội bộ của các cấu trúc dữ liệu.
- Bao gồm các khái niệm như thuộc tính/phương thức công khai (khả năng hiển thị +), phương thức riêng tư (khả năng hiển thị -), kế thừa (được hiển thị bằng mũi tên) và tổng quát hóa.
- Các thuộc tính dẫn xuất (có giá trị được tính từ các thuộc tính khác) được hiển thị với tiền tố '/'.
- Đa số (Multiplicity) chỉ ra số lượng thể hiện tham gia vào một mối quan hệ (ví dụ: 0.., 1, 0..1, i..j, 3..).



## Sơ đồ Trình tự (Sequence Diagram):

---

- Biểu diễn sự tương tác giữa các đối tượng theo thời gian.
- Bao gồm các đối tượng, đường đời (Lifeline), thời gian kích hoạt và thông điệp.
- Đường đời là đường nét đứt dọc bên dưới một đối tượng, biểu thị sự tồn tại của nó theo thời gian trong tương tác.
- Thông điệp là các lời gọi hoặc tín hiệu được gửi giữa các đối tượng.
- Sơ đồ trình tự hữu ích để cho thấy cách các đối tượng hợp tác, minh họa các lời gọi phương thức cho một lớp và tạo các đoạn mã.



## Sơ đồ Thành phần (Component Diagram):

---

- Biểu diễn các đơn vị mã vật lý.
- Một thành phần có thể có nhiều hơn một giao diện.
- Được sử dụng để biểu diễn các mối quan hệ vật lý giữa các thành phần phần mềm và phần cứng trong một hệ thống.



## Sơ đồ Trường hợp sử dụng (Use Case Diagram):

- Biểu diễn sự tương tác giữa các thực thể bên ngoài (tác nhân) và hệ thống (trường hợp sử dụng).
- Các thành phần chính là tác nhân, trường hợp sử dụng, các mối quan hệ và ranh giới hệ thống.
- **Tác nhân** là bất kỳ thực thể bên ngoài nào (người, hệ thống khác) tương tác với hệ thống đang được thiết kế. Tên của tác nhân thường là một danh từ hoặc cụm danh từ.
- **Trường hợp sử dụng** biểu diễn một tập hợp các bước mô tả sự tương tác giữa một tác nhân và hệ thống để đạt được một mục tiêu. Nó nên mô tả hành vi của hệ thống từ góc nhìn của người dùng.
- **Các mối quan hệ giữa các trường hợp sử dụng bao gồm:**
  - **<<include>>**: Được sử dụng khi một trường hợp sử dụng kết hợp chức năng của một trường hợp sử dụng khác. Trường hợp sử dụng được bao gồm thường được chạy bất cứ khi nào trường hợp sử dụng cơ sở chạy.
  - **<<extend>>**: Được sử dụng khi một trường hợp sử dụng thêm chức năng vào một trường hợp sử dụng khác trong các điều kiện cụ thể.
- Tổng quát hóa có thể tồn tại giữa các tác nhân hoặc giữa các trường hợp sử dụng.
- Trong sơ đồ trường hợp sử dụng, mỗi trường hợp sử dụng phải có mối quan hệ trực tiếp với ít nhất một tác nhân.



## Sơ đồ Triển khai (Deployment Diagram):

---

- Được sử dụng để biểu diễn vị trí vật lý của các thành phần phần mềm trên các nút phần cứng.
- Cho thấy các nút (đại diện cho các thiết bị hoặc môi trường thực thi như máy chủ hoặc máy khách) và các thành phần được triển khai trên chúng.



## Các loại Sơ đồ khác:

---

- **Sơ đồ Luồng Dữ liệu (DFD - Data Flow Diagram):** Biểu diễn các chức năng biến đổi luồng dữ liệu. Một chức năng thường được biểu diễn bằng một hình tròn hoặc hình bầu dục.
- **Sơ đồ Thực thể - Mối quan hệ (ERD - Entity Relationship Diagram):** Biểu diễn các mối quan hệ giữa các đối tượng dữ liệu.
- **Sơ đồ Chuyển trạng thái (State Transition Diagram):** Cho thấy cách hệ thống hoặc một đối tượng thay đổi trạng thái để phản ứng với các sự kiện bên ngoài.
- **Sơ đồ Luồng điều khiển (Control Flow Diagram):** Được sử dụng để mô hình hóa các hệ thống hướng sự kiện.
- **Từ điển Dữ liệu (Data Dictionary):** Chứa mô tả của từng đối tượng dữ liệu phần mềm. Phân tích ngữ pháp của các câu lệnh quy trình là bước đầu tiên tốt để tạo ra một từ điển dữ liệu.
- Sơ đồ Mối quan hệ không phải là một loại sơ đồ UML tiêu chuẩn.





## Nguyên tắc Mô hình hóa

---

Khi một kỹ sư phần mềm (hoặc kỹ sư hệ thống nói chung) bắt đầu xây dựng một **mô hình hệ thống** (ví dụ: sơ đồ UML, biểu đồ luồng dữ liệu, v.v.), điều cốt yếu là họ phải xem xét kỹ lưỡng các **giả định** và **ràng buộc**.

### 1. Giả định (Assumptions):

**Giả định** là những yếu tố mà các kỹ sư **cho là đúng** hoặc **sẽ đúng** trong môi trường hoạt động của hệ thống, mặc dù chúng chưa được xác nhận hoặc kiểm chứng đầy đủ tại thời điểm mô hình hóa.

### 2. Ràng buộc (Constraints):

**Ràng buộc** là những **giới hạn** hoặc **điều kiện bắt buộc** mà hệ thống phải **tuân thủ**.