**TÌM HIỂU VỀ MÔ HÌNH MVC**

**I. Tổng quan mô hình MVC**

| Thành phần | Vai trò | Mô tả |
| --- | --- | --- |
| Model | Quản lý dữ liệu và logic nghiệp vụ | Là nơi xử lý và lưu trữ dữ liệu, logic nghiệp vụ, truy xuất từ cơ sở dữ liệu. |
| View | Giao diện người dùng | Hiển thị dữ liệu cho người dùng, lấy thông tin nhập từ người dùng. |
| Controller | Điều phối hoạt động giữa Model và View | Nhận sự kiện từ View, xử lý và cập nhật dữ liệu cho Model và View. |

**II. Mối quan hệ giữa các thành phần**

Người dùng <--> View <--> Controller <--> Model

↑ ↓

<--- Dữ liệu --->

**View** nhận đầu vào từ người dùng (ví dụ: click nút).

**Controller** tiếp nhận sự kiện từ View, xử lý yêu cầu.

**Model** thực hiện xử lý logic, truy vấn dữ liệu, và cập nhật lại View thông qua Controller.

**III. Ví dụ minh họa**

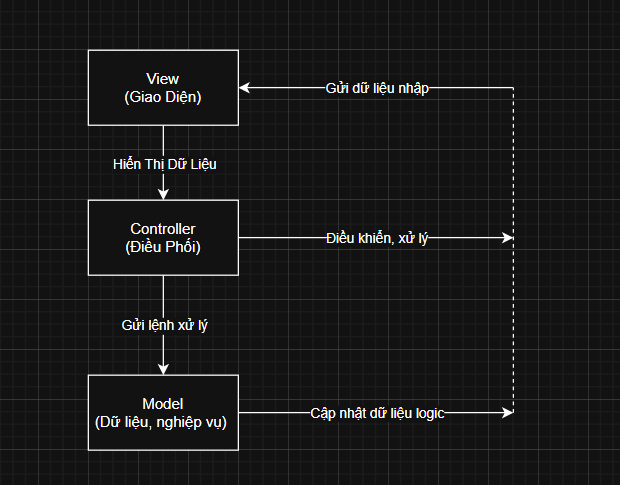
Ví dụ bạn đang tạo ứng dụng quản lý nhân viên:

* Model: Lớp NhanVien.java chứa thông tin nhân viên (Mã NV, Tên, Ngày sinh, v.v.).
* View: Form frmQLNhanVien.java giao diện người dùng để nhập/sửa/xem nhân viên.
* Controller: Lớp QuanLyNhanVien.java điều phối giữa form và xử lý dữ liệu (thêm/sửa/xóa).

**IV. Ưu điểm của mô hình MVC**

* **Tách biệt giao diện với logic nghiệp vụ → dễ bảo trì.**
* **Tái sử dụng mã nguồn: thay View mà không ảnh hưởng Model.**
* **Dễ kiểm thử từng thành phần riêng biệt.**
* **Thường được dùng trong Java Swing, JavaFX, ASP.NET, Django, Laravel,...**

**V. Cấu trúc tổng quát mô hình MVC**

****

**Chi tiết :**

**1. Model (M) — *Dữ liệu và nghiệp vụ***

* Chứa lớp đại diện cho dữ liệu (ví dụ: NhanVien, PhongBan, Luong,...).
* Chịu trách nhiệm lưu trữ, cập nhật và xử lý dữ liệu.
* Kết nối với cơ sở dữ liệu.

**2. View (V) — *Giao diện người dùng***

* Hiển thị dữ liệu cho người dùng.
* Nhận thao tác người dùng (nhập liệu, click nút,...).
* Không chứa logic xử lý nghiệp vụ.
* Ví dụ: frmQLNhanVien.java trong Java Swing

**3. Controller (C) — *Điều phối, xử lý yêu cầu***

* Nhận sự kiện từ View (VD: người dùng nhấn nút "Thêm").
* Gọi đến Model để xử lý dữ liệu (VD: thêm nhân viên vào DB).
* Cập nhật lại View nếu cần (VD: làm mới bảng hiển thị).
* Ví dụ: QuanLyNhanVien.java