



## MÔ HÌNH MVC

# Lập trình JAVA

## MÔ HÌNH MVC



Đ.Huy

Nội Dung

**Đặt vấn đề**

**Giới thiệu tổng quan mô hình MVC**

**Một số Framework sử dụng MVC**

**Demo**

### Đặt vấn đề

- Khi làm việc trong 1 nhóm hoặc 1 dự án lớn sẽ rất khó tách biệt và phân công công việc giữa từng thành viên.
- Việc bảo trì chúng thường rất khó khăn, vì các đoạn code java lẫn lộn với các đoạn mã html.
- Khi có lỗi xảy ra, việc tìm ra lỗi và định vị lỗi cũng là một vấn đề khó khăn.
- Và còn nhiều những khó khăn khác ...

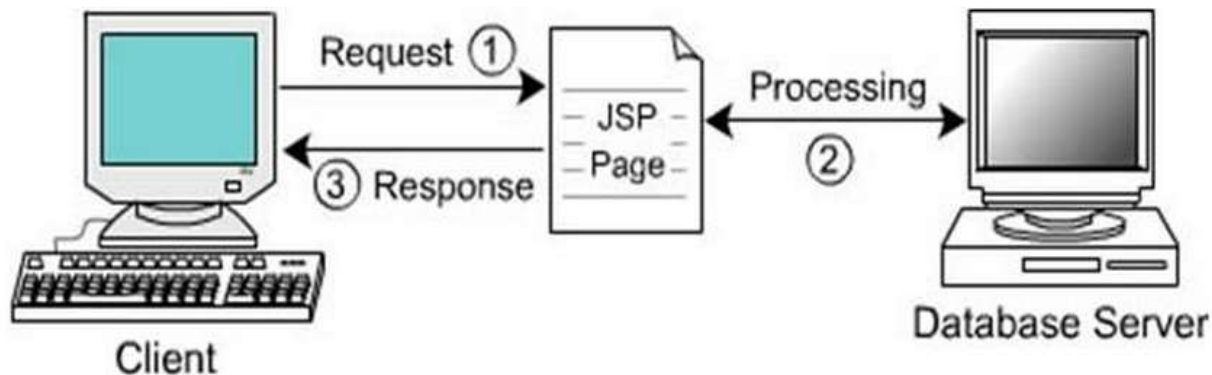
# Lập trình JAVA

## MÔ HÌNH MVC

Đ.Huy

### Đặt vấn đề

→ Việc lập trình như trên, người ta gọi là mô hình lập trình 1, hay là mô hình lập trình cổ điển.



### Đặt vấn đề

→ Để khắc phục các khó khăn trên người ta đưa ra mô hình 2 hay còn gọi là mô hình MVC (Model-View-Controller).

Tức là: Tương ứng với một trang JSP trong mô hình 1, bây giờ người ta tách nó ra làm ba thành phần:

Mô hình – Khung nhìn – Bộ điều khiển

### Giới thiệu tổng quan mô hình MVC

- Lịch sử hình thành.
- Khái niệm.
- Cơ chế hoạt động.
- So sánh MVC với 3-Layers (MH 3 lớp)
- Lợi ích.
- Ưu – Nhược điểm.

# Lập trình JAVA

## MÔ HÌNH MVC



Đ.Huy

Giới thiệu tổng  
quan mô hình MVC

Lịch sử hình thành

MVC được phát minh bởi 1 giáo sư người Na Uy tên là Trygve Reenskaug và được ông công bố, trình bày ngày 10/12/1979 tại phòng thí nghiệm Xerox PARC ở Palo Alto.

### Giới thiệu tổng quan mô hình MVC

### Khái niệm

- Ý niệm chính của mô hình MVC là tách biệt phần ánh xạ, lưu trữ và xử lý dữ liệu (model) với thành phần trình bày giao diện kết quả cho người dùng hay phần giao diện giúp đón nhận dữ liệu nhập xuất cho người dùng (View).
- Ý niệm trên cho phép người lập trình có thể tách biệt công việc trong quá trình xây dựng chức năng và xây dựng giao diện cho người dùng.
- Bên cạnh đó, ý niệm trên cho phép việc thay đổi thành phần của dữ liệu (Model) sẽ không ảnh hưởng nhiều đến giao diện (View) của người dùng.
- Tuy nhiên, một ứng dụng có thể có rất nhiều Model và nhiều View, do vậy, mô hình cần có một thành phần lựa chọn và kết nối các thành phần này lại với nhau theo cách hiệu quả nhất mà người ta gọi là Controller.



# Lập trình JAVA

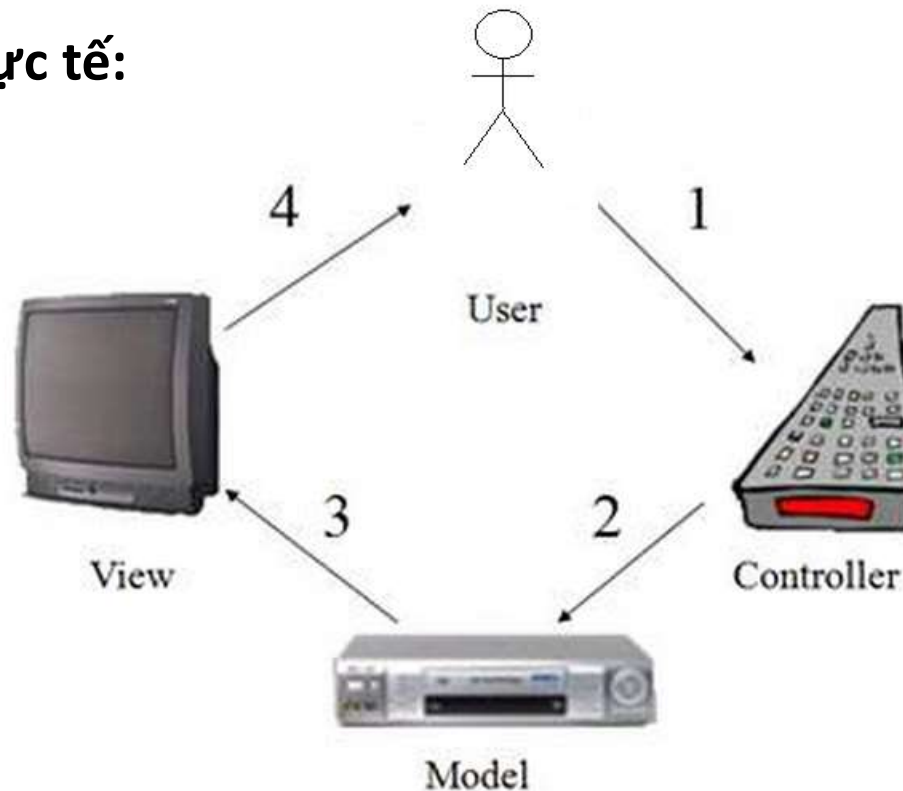
## MÔ HÌNH MVC

Đ.Huy

Giới thiệu tổng  
quan mô hình MVC

### Khái niệm

→ Ví dụ thực tế:



### Khái niệm

#### → Định nghĩa các thành phần trong mô hình MVC:

- Model (Mô hình): Mô hình là các lớp java có nhiệm vụ
  - + Nhận các yêu cầu từ View.
  - + Thực thi các yêu cầu đó (tính toán, kết nối CSDL ...).
  - + Trả về các giá trị tính toán cho View.
- View (Khung nhìn): Bao gồm các mã tương tự như JSP để hiển thị form nhập liệu, các kết quả trả về từ Model...
- Controller (Bộ điều khiển): Đồng bộ hoá giữa Khung nhìn (View) và Mô hình (Model). Tức là Controller sẽ xác định với một trang JSP này thì sẽ tương ứng với lớp java nào đó trong Model để xử lý nó và ngược lại, kết quả sẽ trả về trang jsp nào.

### Khái niệm

→ Như vậy chúng ta có thể tách biệt được các mã java ra khỏi mã html. Do vậy, nó đã giải quyết được các khó khăn đã nêu ra trong Mô hình 1. Người thiết kế giao diện và người lập trình java có thể mang tính chất độc lập tương đối. Việc debug hay bảo trì sẽ dễ dàng hơn, việc thay đổi các theme của trang web cũng dễ dàng hơn ...

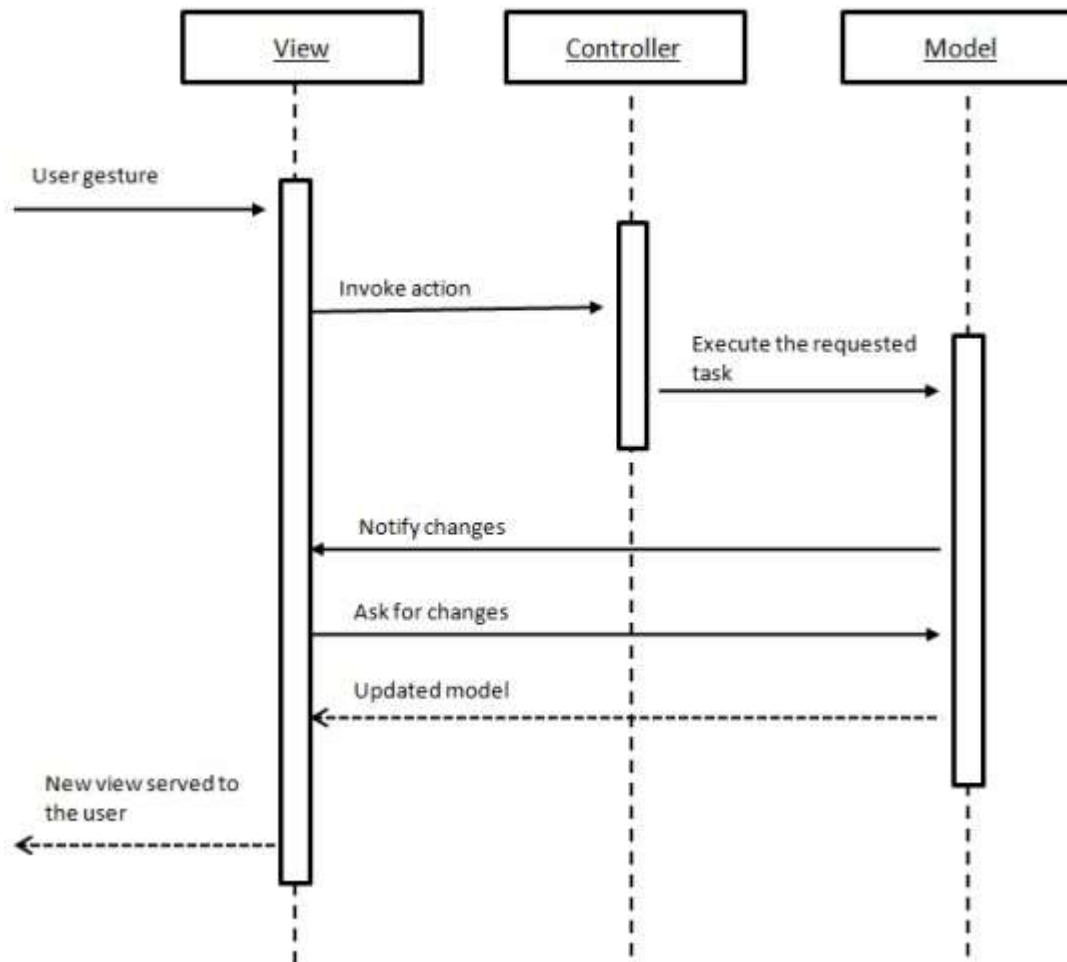
# Lập trình JAVA

## MÔ HÌNH MVC

Đ.Huy

Giới thiệu tổng  
quan mô hình MVC

### Cơ chế hoạt động



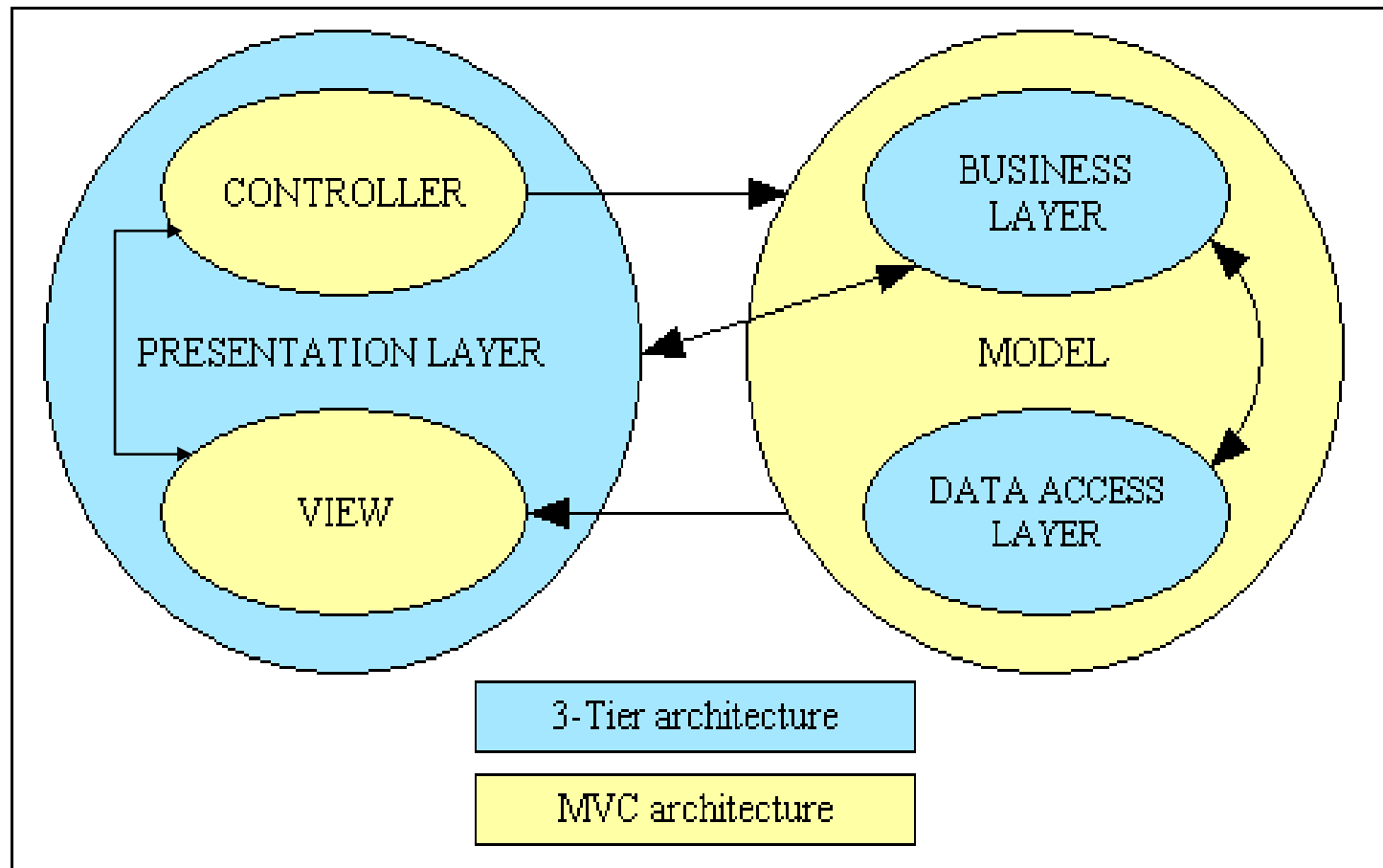
# Lập trình JAVA

## MÔ HÌNH MVC

Đ.Huy

Giới thiệu tổng  
quan mô hình MVC

So Sánh MVC và 3-Layers



#### Giống nhau:

- Cả hai đều để tách rời programming core/business logic ra khỏi những phụ thuộc về tài nguyên và môi trường.
- Trong một ứng dụng nhỏ, MVC thể hiện thế nào? Presentation thể hiện giống như chức năng của View và Controller. Business và Database thể hiện giống như chức năng của Model. Như thế nhìn ở góc độ này, thì MVC tương đương với 3-layer.

#### Khác nhau:

- Trong 3-layers, quá trình đi theo chiều dọc, bắt đầu từ Presentation Layer, sang Business Layer, rồi tới Data Access Layer, và từ Data Access Layer, chạy ngược lại Business Layer rồi quay ra lại Presentation Layer.
- Còn MVC xử lý theo mô hình tam giác, Model- Controller - View sẽ thực thi các thao tác xử lý với nhau mà không cần phải qua 1 lượt tất cả các tầng. Việc này tạo ra tính linh hoạt hơn.

# Lập trình JAVA

## MÔ HÌNH MVC



Đ.Huy

Giới thiệu tổng  
quan mô hình MVC

### Lợi ích

- + Tách biệt sự phụ thuộc giữa các thành phần.
- + Chuyên môn phát triển và tập trung.
- + Phát triển song song các nhóm riêng biệt.



# Lập trình JAVA

## MÔ HÌNH MVC



Đ.Huy

Giới thiệu tổng  
quan mô hình MVC

### Ưu – Nhược điểm

**Ưu điểm:** Phát triển phần mềm chuyên nghiệp hơn có thể chia công việc cho nhiều nhóm: nhóm thiết kế, nhóm lập trình, nhóm tổ chức database... giúp phát triển ứng dụng nhanh đơn giản dễ nâng cấp giúp phát hiện lỗi và dễ sửa lỗi.

**Nhược điểm:** Thích hợp với các dự án vừa và lớn còn đối với các dự án nhỏ việc áp dụng mô hình mvc sẽ gây cồng kềnh tốn thời gian trong quá trình phát triển.

# Lập trình JAVA

## MÔ HÌNH MVC



Đ.Huy

### Một số Framework sử dụng MVC

- + ASP.NET MVC Framework
- + Zend Framework
- + Spring web MVC framework
- + SMALLTALK

# Lập trình JAVA

## MÔ HÌNH MVC



Đ.Huy

**Demo**

# THE END



Xin cảm ơn

∧.∧~

