



LẬP TRÌNH MẠNG

MÔ HÌNH KIẾN TRÚC MVC

hungdn@ptit.edu.vn

Nội dung



- Mô hình kiến trúc MVC
- MVC với Java swing



Mô hình kiến trúc vs Mô hình thiết kế

architectural pattern

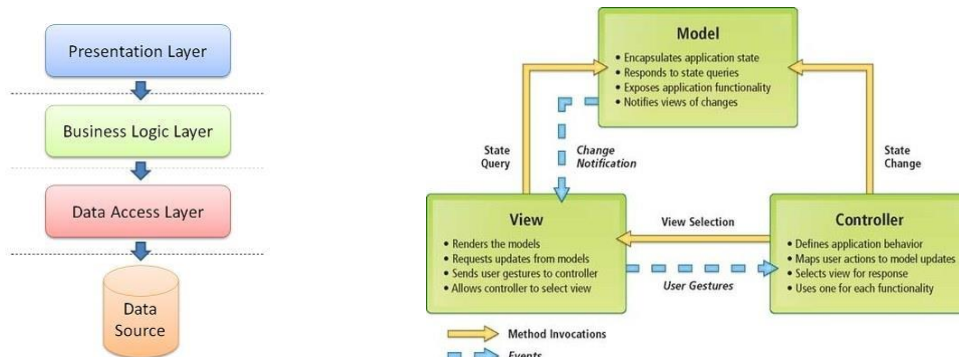
- **Phạm vi**
 - Framework, component ...
- **Sử dụng**
 - Đại diện cho nguyên tắc, nền tảng cần phải tuân theo
 - Giải quyết vấn đề về kiến trúc phần mềm
- **Ex**
 - N-Layer (Three-tier)
 - Microkernel
 - Model-View-Controller
 - Model-View-ViewModel

design pattern

- **Phạm vi**
 - Package, Class ...
- **Sử dụng**
 - Đại diện cho một cách để cấu trúc các lớp nhằm xử lý một vấn đề chung
 - Giải quyết vấn đề về kỹ thuật
- **Ex:**
 - Factory
 - Strategy
 - Skeleton
 - Facade.

-> Hoạt động ở các mức trừu tượng khác nhau

N-layer vs MVC

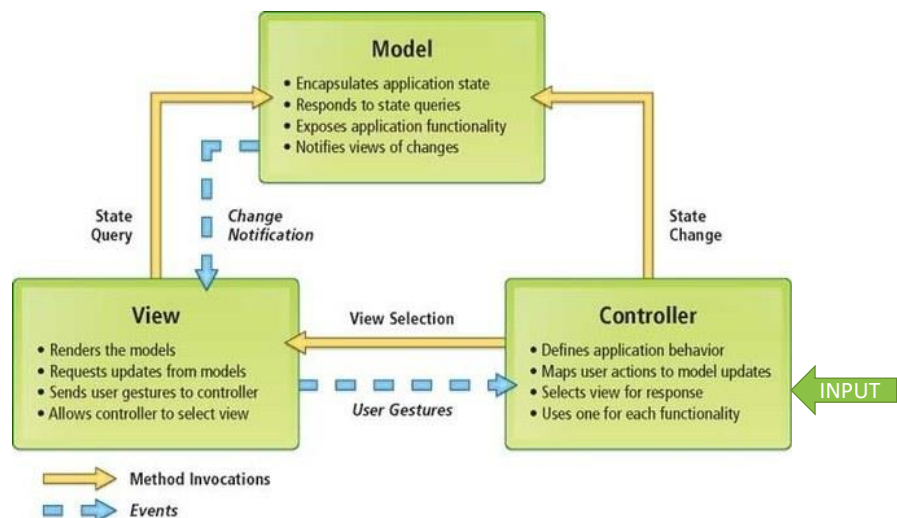


1. Mô hình kiến trúc MVC

MVC



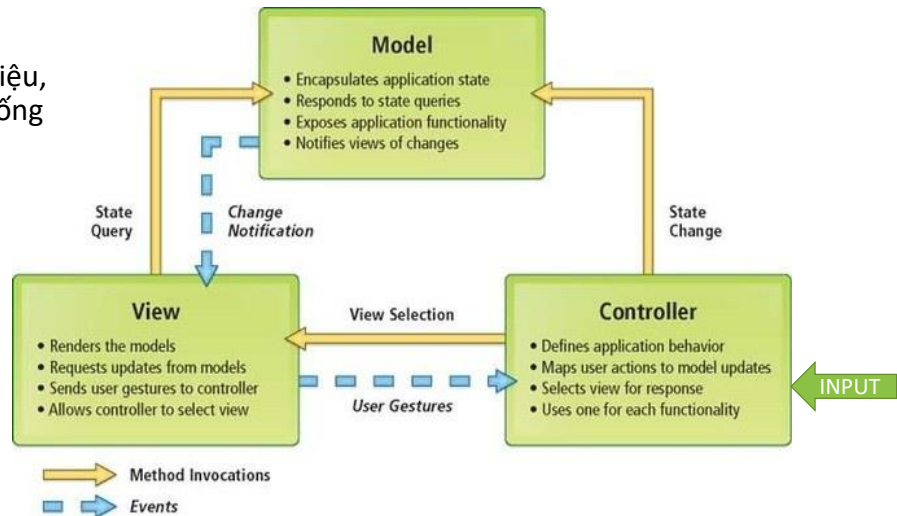
- **Model**
- **View**
- **Controller**



MVC - Model

• Model

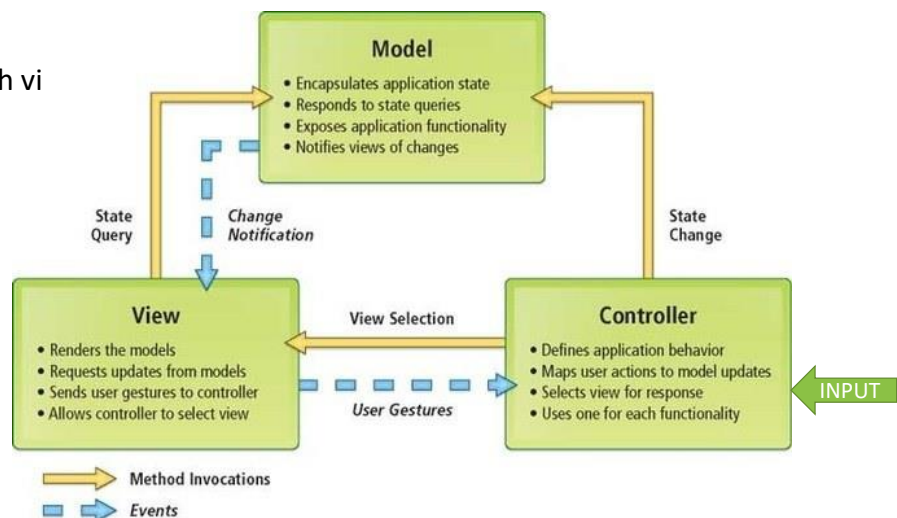
- Định nghĩa dữ liệu, thông tin hệ thống



MVC - Controller

• Controller

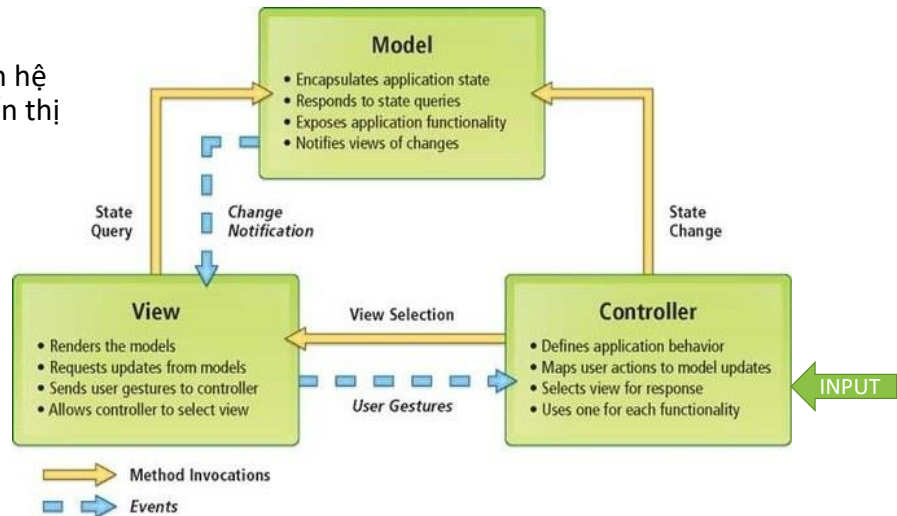
- Định nghĩa hành vi của hệ thống



MVC - View

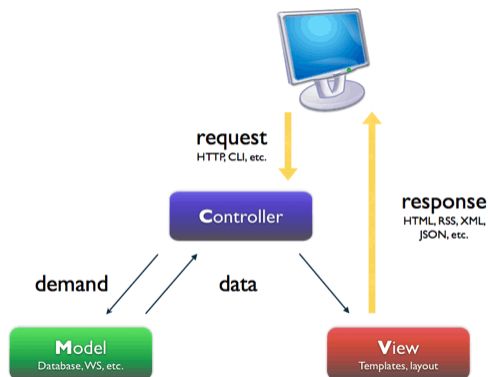
• View

- Định nghĩa cách hệ thống được hiển thị

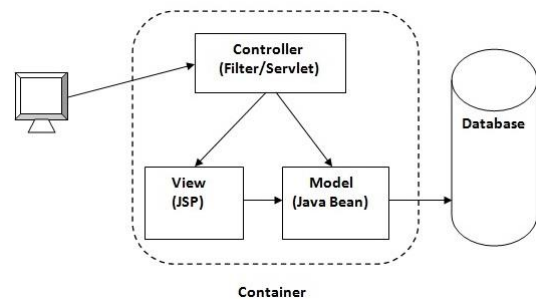


Ex: Web MVC

• Web MVC



• Ex: Java web

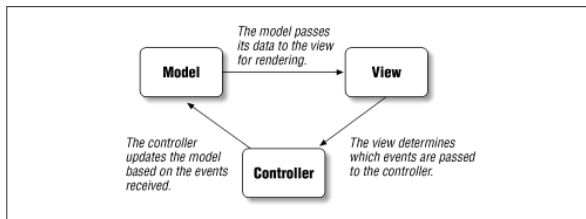


2. MVC với Java Swing

Java swing

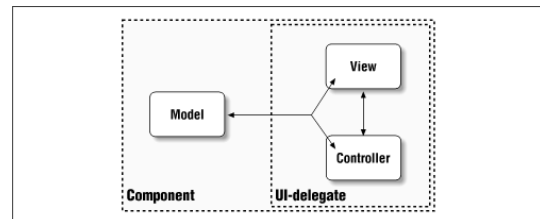


- MVC



- Mỗi thành phần cần các thành phần khác để giữ cho nó tiếp tục hoạt động

- Swing

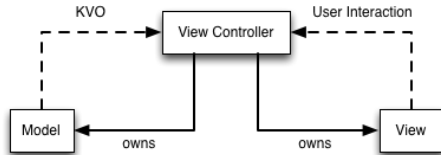


- Kết hợp view và controller vào một thành phần thực hiện việc hiển thị giao diện và xử lý sự kiện GUI (UI delegate)



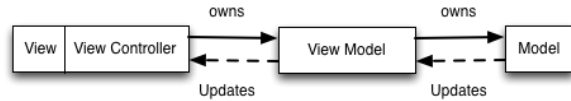
MVVM

• IOS MVC

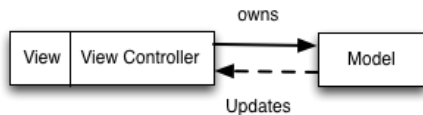


• MVVM

- Looking better!



• Problem



Ex: Model



• Model

- userName
- password

```

public class LoginModel {
    private String userName;
    private String password;
    public LoginModel() {
    }
    public LoginModel(String userName, String password) {
        this.userName = userName;
        this.password = password;
    }
    public String getUserName() {
        return userName;
    }
    public void setUserName(String userName) {
        this.userName = userName;
    }
    public String getPassword() {
        return password;
    }
    public void setPassword(String password) {
        this.password = password;
    }
}
    
```



Ex: View

```
public class LoginView {
    LoginModel user;
    private static int retry = 3;
    public LoginView(LoginModel user) {
        displayFormDetail(user);
    }
    public LoginView() {
        displayFormLogin();
    }
    private void displayFormLogin() {
        String strUser, strPass;
        System.out.println(" -- Login Form --");
        Scanner input = new Scanner(System.in);
        System.out.print("User: ");
        strUser = input.nextLine();
        System.out.print("Pass: ");
        strPass = input.nextLine();
        user = new LoginController().checkLogin(strUser, strPass);
        if(user!=null)
            displayFormDetail(user);
        else {
            System.out.println("wrong user or pass! " + retry--);
            if(retry>0)
                displayFormLogin();
            System.exit(retry);
        }
    }
    private void displayFormDetail(LoginModel user) {
        System.out.println(" -- Details Form -- ");
        System.out.println("user: " + user.getUserName());
    }
}
```

- View

- Login

- Details

Ex: Controller



- Controller

- checkLogin

```
public class LoginController {
    LoginModel user;

    public LoginController() {
    }

    public LoginController(LoginModel user) {
        this.user = user;
    }
    public LoginModel checkLogin(String strUser, String strPass)
    {
        if(strUser.equals("admin")&&strPass.equals("abc123"))
            user = new LoginModel(strUser, strPass);
        return user;
    }
}
```


Ex: MvcTest



- Main

```
public class MvcTest {  
    public static void main(String[] args) {  
        // LoginModel user = new LoginModel(); // 1) Login  
        LoginView login = new LoginView(new LoginModel("admin", "abc123")); // 2) Details  
    }  
}
```

- Bài tập : Cài đặt lại để input chương trình bắt đầu từ Controller

DEMO



- Login MVC

- Model (LoginModel)
 - Username
 - Password
- View (LoginView)
 - Login
 - Details
- Controller (LoginController)
 - checkLogin
 - ...
- MvcTest
 - View

- Console

- File
- DB



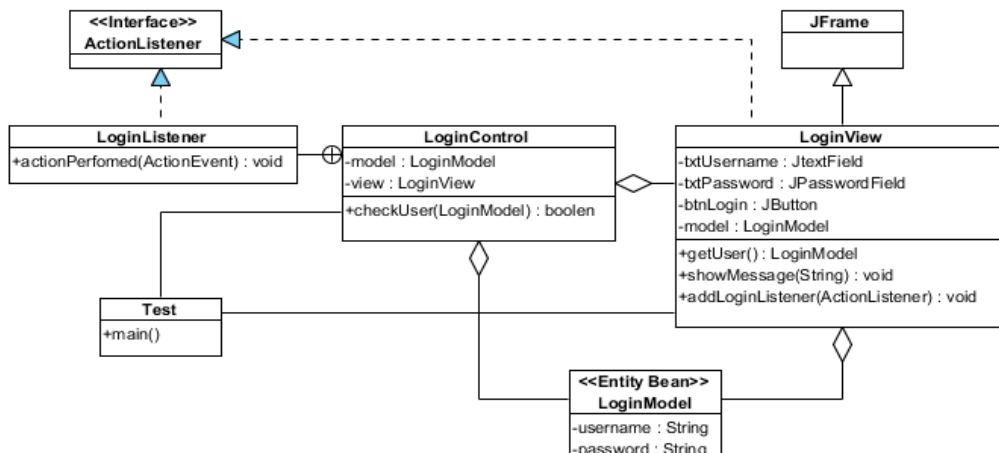
Case study: MVC với GUI

- Yêu cầu bài toán
 - Tạo một form đăng nhập gồm username, password và một nút login
 - Mỗi khi click vào nút login, chương trình phải kiểm tra thông tin đăng nhập có đúng không, nếu đúng thông báo thành công, nếu sai thông báo đăng nhập sai!
 - Xây dựng chương trình theo mô hình MVC
 - Dữ liệu chương trình với
 - File
 - DB

Case study: MVC với GUI



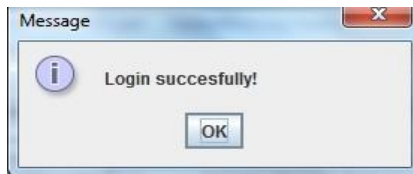
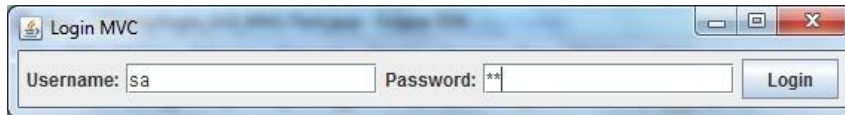
- Sơ đồ lớp



Case study: MVC với GUI



- Minh họa chương trình



Tổng kết



- **Mô hình kiến trúc MVC**
 - Vs N-Layer
 - Vs Mô hình thiết kế
- **Login MVC**
- **Calculator MVC**

Exercise



1. Mở rộng

- a) Sửa ứng dụng login để đọc và ghi user, password từ file text
- b) Viết thêm modul đăng kí, chỉnh sửa thông tin của user vào ví dụ trong bài
- c) Viết modul tìm kiếm user có username chứa một đoạn text nào đó, show kết quả lên GUI
- d) Viết một ứng dụng hoàn chỉnh quản lí người dùng của một hệ thống

1. Ứng dụng mô phỏng calculator theo mô hình kiến trúc MVC



Q & A