

HỌC PHẦN: CÔNG NGHỆ JAVA

Giảng viên: Đào Thị Lệ Thủy

Email: thuydt0129@gmail.com

SĐT: 0946.921.976

Khoa Công nghệ Thông tin



GIỚI THIỆU MÔ HÌNH MVC

MVC là một design pattern (mẫu thiết kế) phổ biến được áp dụng trên hầu hết các ngôn ngữ lập trình hướng đối tượng hiện nay từ Java, C#, PHP, Visual C++

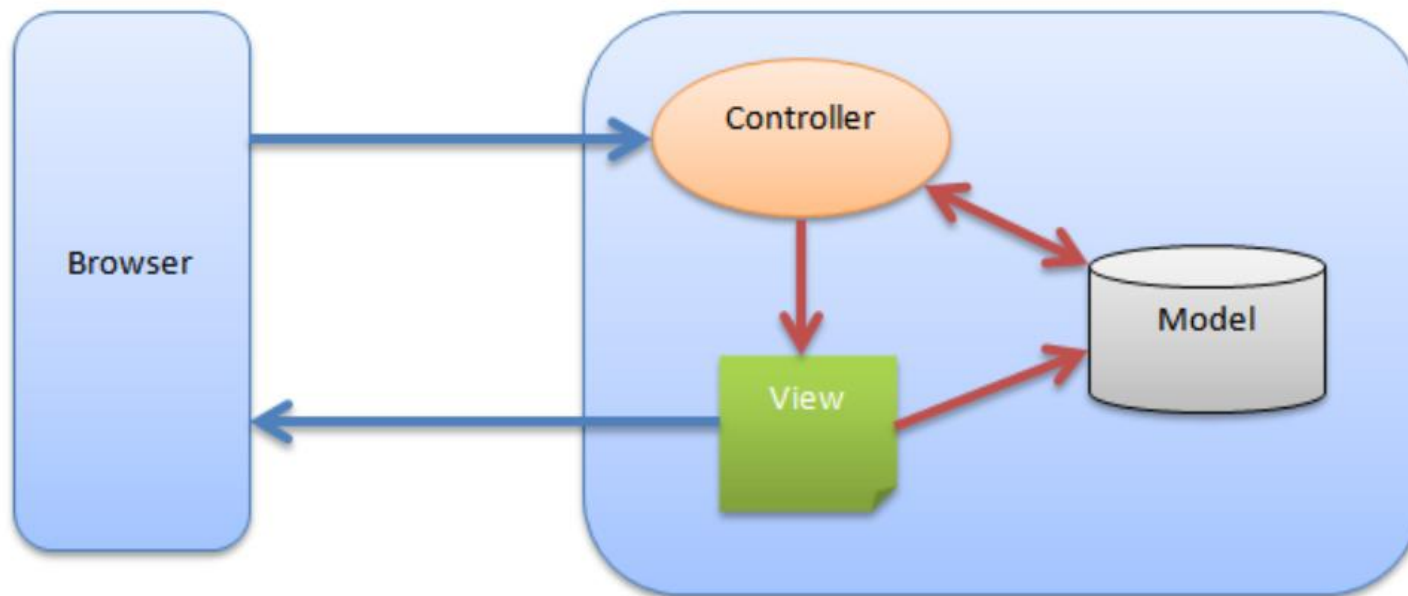
1. MVC là gì?

Mô hình MVC được giới thiệu lần đầu tiên bởi Trygve Reenskaug khi anh ta đang làm việc trong dự án Smalltalk ở công ty Xerox PARC ở California, Mỹ vào những năm 1970.

MVC được viết tắt bởi 3 từ: **M**odel – **V**iew – **C**ontroller, là một mẫu thiết kế nhằm mục tiêu chia tách phần giao diện và code để dễ quản lý, phát triển và bảo trì.

GIỚI THIỆU MÔ HÌNH MVC

- Model: quản lý, xử lý dữ liệu
- View: giao diện, hiển thị dữ liệu cho người dùng
- Controller: điều khiển sự tương tác giữa Model và View



Mô hình MVC giúp tách biệt giữa cách thức mà dữ liệu được xử lý phía nội hàm và phần dữ liệu hiển thị phía người dùng. Ở đây Controller nhận được tất cả các yêu cầu cho ứng dụng và sau đó làm việc với Model để chuẩn bị dữ liệu cần thiết cho View. View sau đó sử dụng các dữ liệu được chuẩn bị bởi Controller để hiển thị cho người dùng.

Các thành phần trong MVC

■ Model:

- Là bộ phận có nhiệm vụ quản lý dữ liệu của ứng dụng.
- Chức năng biểu diễn, vận chuyển thông tin để trình diễn (view) và xử lý (control)
- Chứa tất cả các nghiệp vụ logic, đối tượng mô tả dữ liệu, ...

■ View:

- Tương tác với người sử dụng.
- Show kết quả từ tầng Controller.
- Thu nhận các hoạt động, request của người sử dụng và chuyển cho tầng Controller xử lý.

■ Controller:

- Định nghĩa các hành vi, hoạt động, xử lý của hệ thống.
- Đối chiếu hành động của người sử dụng từ View. Đồng thời tương tác Model để gọi View và hiển thị thông tin tương ứng cho người dùng.

Ưu điểm và nhược điểm của MVC

■ Ưu điểm

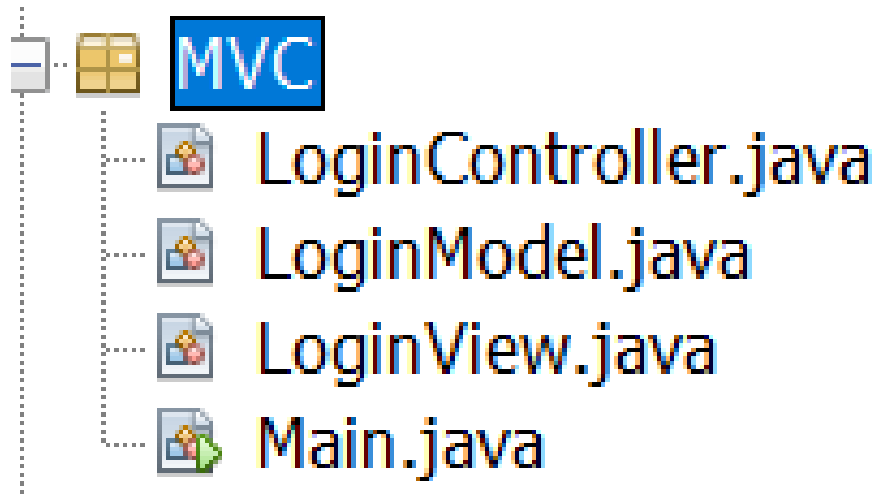
- Do được chia thành các thành phần độc lập nên Mô hình MVC giúp phát triển ứng dụng có code dễ đọc, dễ nâng cấp, bảo trì.
- Thể hiện tính chuyên nghiệp trong việc tạo ứng dụng.

■ Nhược điểm

- Đối với dự án nhỏ việc áp dụng mô hình MC gây cồng kềnh, tốn thời gian trong quá trình phát triển.
- Tốn thời gian trung chuyển dữ liệu của các thành phần.

VÍ DỤ

- Demo ứng dụng đăng nhập thông qua tài khoản, mật khẩu bằng mô hình MVC
- Cấu trúc cây thư mục



VÍ DỤ

```
4 package MVC;
5 public class LoginModel {
6     private String user;
7     private String pass;
8     public LoginModel() {}
9     public LoginModel(String _user, String _pass) {
10         this.user = _user;
11         this.pass = _pass;
12     }
13     public String getUser() {
14         return user;
15     }
16     public void setUser(String _user) {
17         this.user = _user;
18     }
19     public String getPass() {
20         return pass;
21     }
22     public void setPass(String _pass) {
23         this.pass = _pass;
24     }
25 }
```

VÍ DỤ

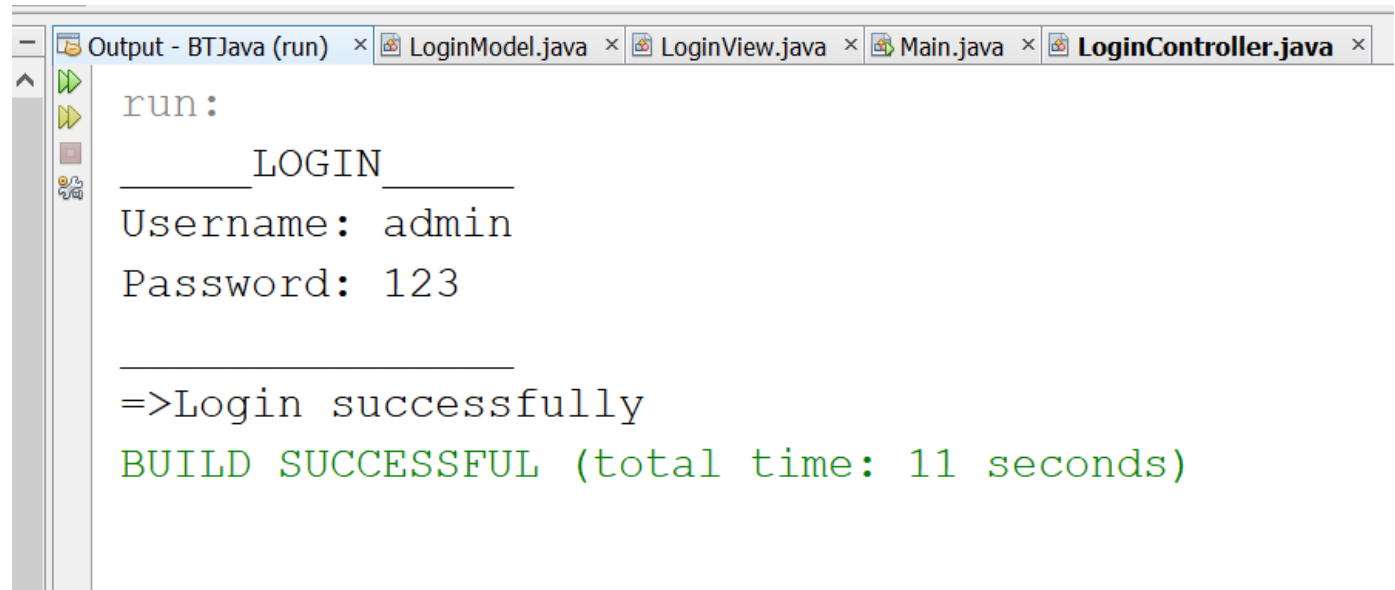
```
6 package MVC;
7 import java.util.Scanner;
8
9 public class LoginView {
10     public static Scanner scanner = new Scanner(System.in);
11
12     public void showMessage(String msg) {
13         System.out.println(msg);
14     }
15
16     public LoginModel getInfo() {
17         System.out.println("____LOGIN____");
18
19         LoginModel user = new LoginModel();
20         System.out.print("Username: ");
21         user.setUser(scanner.next());
22         System.out.print("Password: ");
23         user.setPass(scanner.next());
24         System.out.println("____");
25         return user;
26     }
27 }
```


VÍ DỤ

```
12 public class LoginController {
13     private LoginView view;
14
15     public LoginController(LoginView view) {
16         this.view = view;
17     }
18
19     private boolean check(LoginModel user) {
20         return (user.getUser().equals("admin"))
21             && (user.getPass().equals("123"));
22     }
23
24     public void login() {
25         while (true) {
26             LoginModel user = view.getInfo();
27             if (check(user)) {
28                 view.showMessage("=>Login successfully");
29                 break;
30             } else {
31                 view.showMessage("=>Wrong username or password");
32             }
33         }
34     }
35 }
```

VÍ DỤ

```
12 public class Main {  
13     public static void main(String[] args) {  
14         LoginView view = new LoginView();  
15         LoginController control = new LoginController(view);  
16  
17         control.login();  
18     }  
19 }  
20
```



```
run:  
_____  
LOGIN_____  
Username: admin  
Password: 123  
_____  
=>Login successfully  
BUILD SUCCESSFUL (total time: 11 seconds)
```