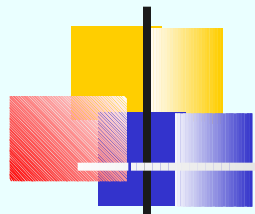


Lập trình mạng

Thiết kế theo mô hình MVC



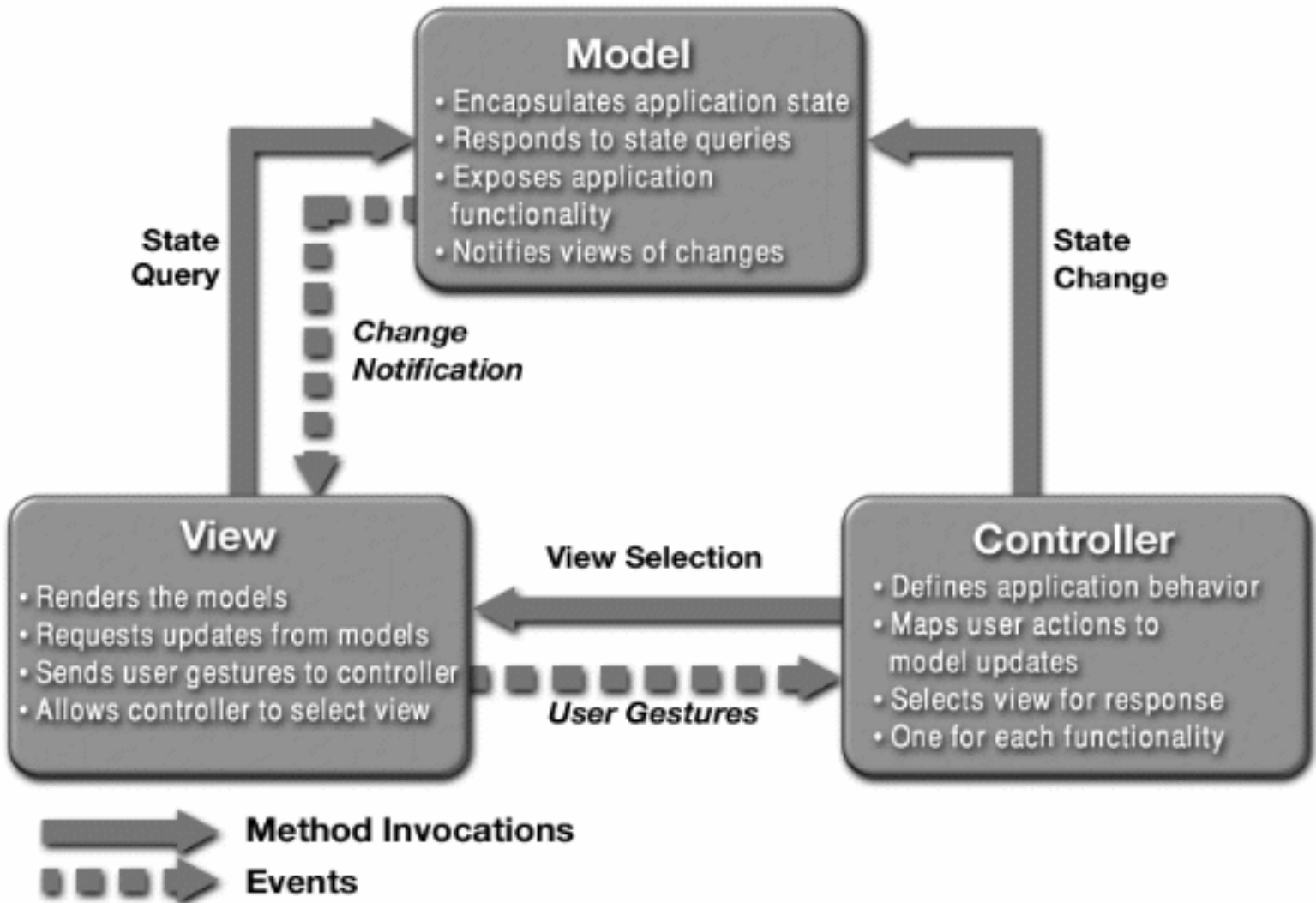
Nội dung

- Mô hình MVC tổng quan
- Mô hình MVC cải tiến
- Ví dụ
- Bài tập



Mô hình MVC

Mô hình MVC (1)





Mô hình MVC (2)

M - model:

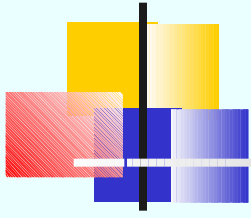
- Đóng gói dữ liệu, thông tin
- Chức năng biểu diễn, vận chuyển thông tin để trình diễn (view) và xử lý (control)



Mô hình MVC (3)

C - control:

- Định nghĩa các hành vi, hoạt động, xử lí của hệ thống
- Đối chiếu hành động của user (nhận từ view), vào tập chức năng để xử lí, đồng thời chọn hành động đưa view ra để show

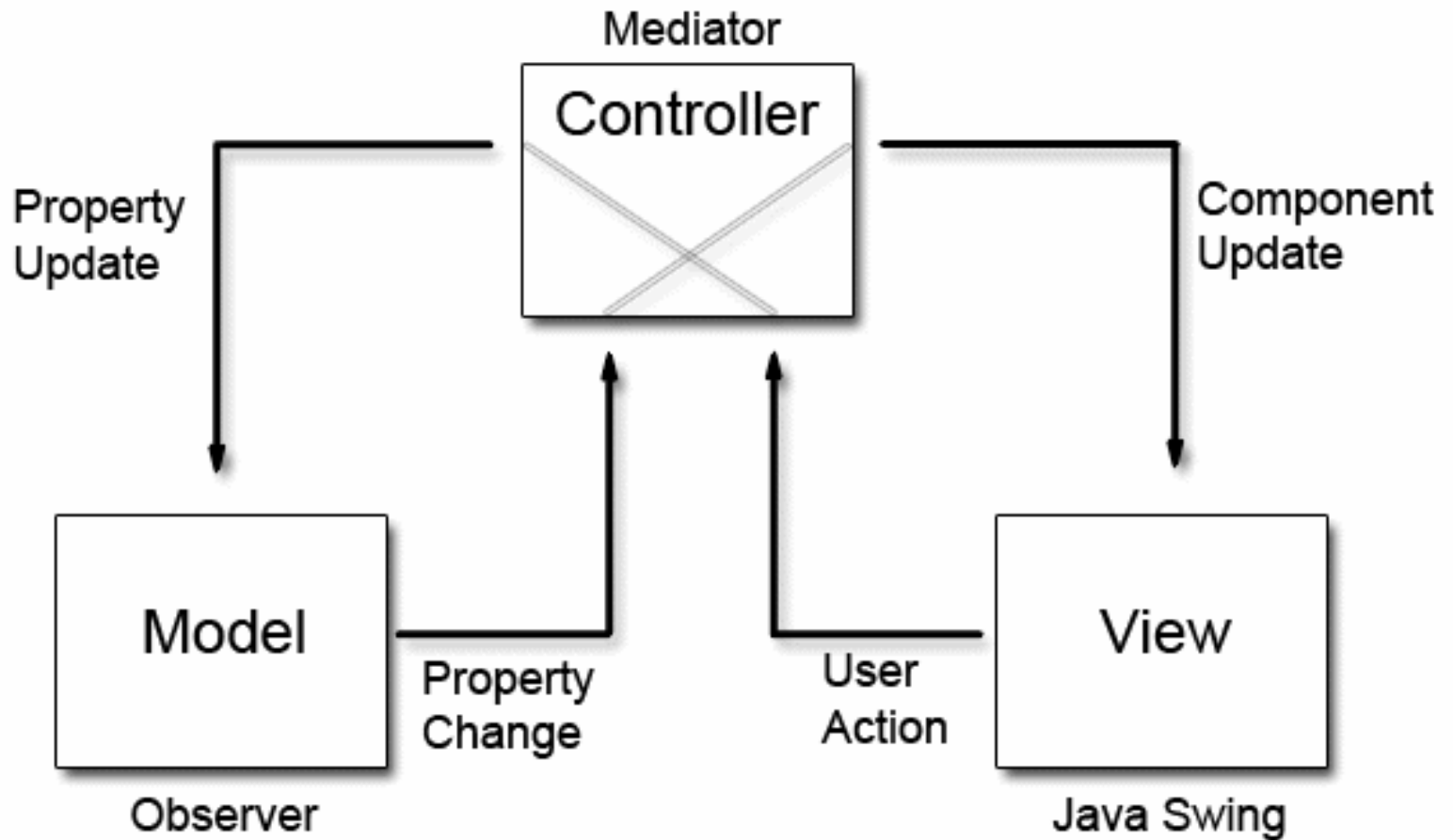


Mô hình MVC (4)

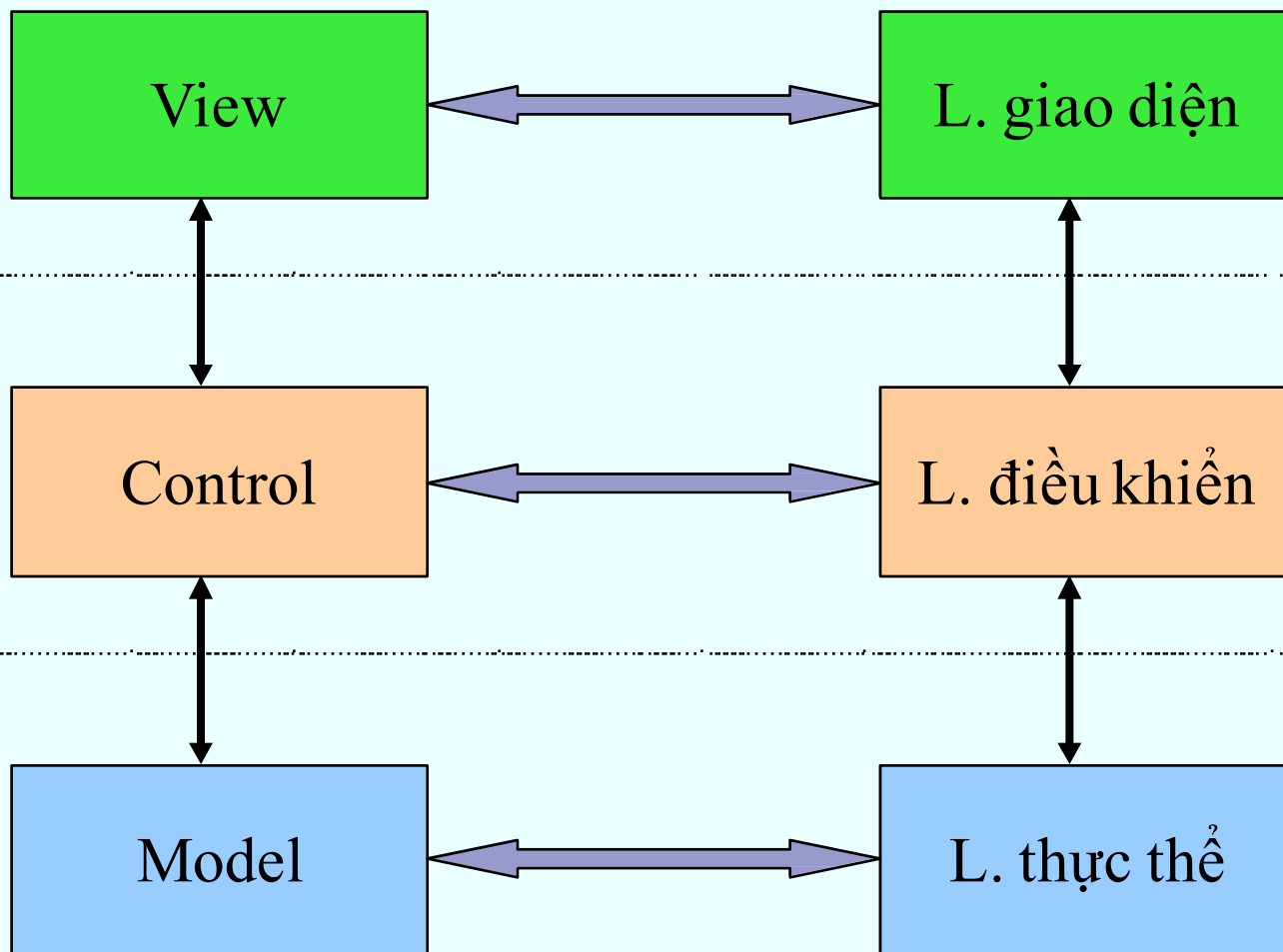
V - view:

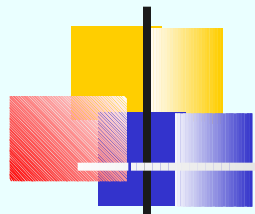
- Giao diện với người sử dụng
- Show các kết quả xử lý của tầng control
- Thu nhận các hoạt động, yêu cầu của người sử dụng và chuyển cho tầng control xử lý

MVC cải tiến (1)



MVC cải tiến (2)





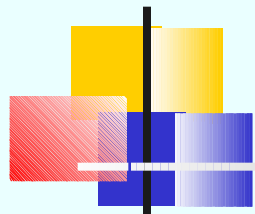
Các lớp thực thể

- Đóng gói dữ liệu, thông tin
- Chỉ chứa các thuộc tính và các phương thức truy cập các thuộc tính (javaBean)
- Chức năng biểu diễn, vận chuyển thông tin để trình diễn (view) và xử lý (control)



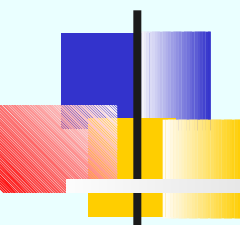
Các lớp điều khiển

- Cập nhật thông tin vào DB (thông tin chứa trong các thực thể)
- Thực hiện các tính toán, xử lý trung gian
- Đối chiếu hành động của user (nhận từ view), vào tập chức năng để xử lý, đồng thời chọn hành động đưa view ra để show

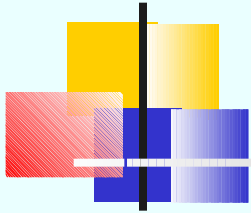


Các lớp giao diện

- Các frame, cửa sổ của ứng dụng (javaSwing)
- Các trang giao diện web: html, jsp
- Các bảng, mẫu biểu, báo cáo in ra

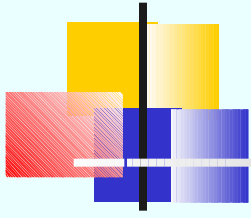


Ví dụ:
điều khiển đăng nhập từ dòng lệnh



Login: Model

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



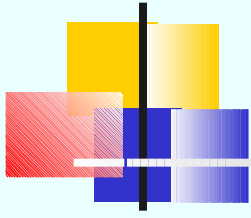
Login: View (1)

```
import java.io.DataInputStream;
import java.io.IOException;

public class LoginView {
    LoginModel user;

    public LoginView(LoginModel user) {
        this.user = user;
    }

    public void showMessage(String smg) {
        System.out.println(smg);
    }
}
```



Login: View (2)

```
public void getUserInfo() {  
    try {  
        Scanner input = new Scanner(System.in);  
  
        System.out.print("Username: ");  
        user.setUserName(input.nextLine());  
        System.out.print("Password: ");  
        user.setPassword(input.nextLine());  
  
        input.close();  
    } catch (IOException e) {  
        System.out.println(e);  
    }  
}
```

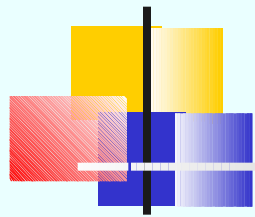



Login: Control (1)

```
public class LoginControl {
    LoginModel user;
    LoginView view;

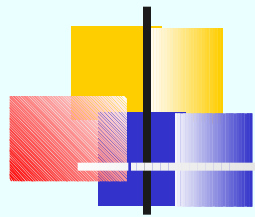
    public LoginControl(LoginModel user, LoginView view){
        this.user = user;
        this.view = view;

        while(true){
            view.getUserInfo();
            if(checkLogin()){
                view.showMessage("success!");
                break;
            }else{
                view.showMessage("wrong username or password!");
            }
        }
    }
}
```



Login: Control (2)

```
private boolean checkLogin(){  
    if ( (user.getUserName().equals("sa"))  
        &&(user.getPassword().equals("sa")) ){  
        return true;  
    }  
    return false;  
}  
}
```



Login: main

```
public class LoginMVC {  
  
    public static void main(String[] args) {  
        LoginModel user = new LoginModel();  
        LoginView view = new LoginView(user);  
        LoginControl control = new LoginControl(user, view);  
    }  
}
```