



# 第十四章 MVC架构

任课教师：武永亮

wuyongliang@edu2act.org



## ▣ 上节回顾

## ■ 课程内容

- MVC概述
- MVC实现细节
- MVC之设计模式
- 表现层演化MVP

## ■ 课程内容

- MVC概述
- MVC实现细节
- MVC之设计模式
- 表现层演化MVP

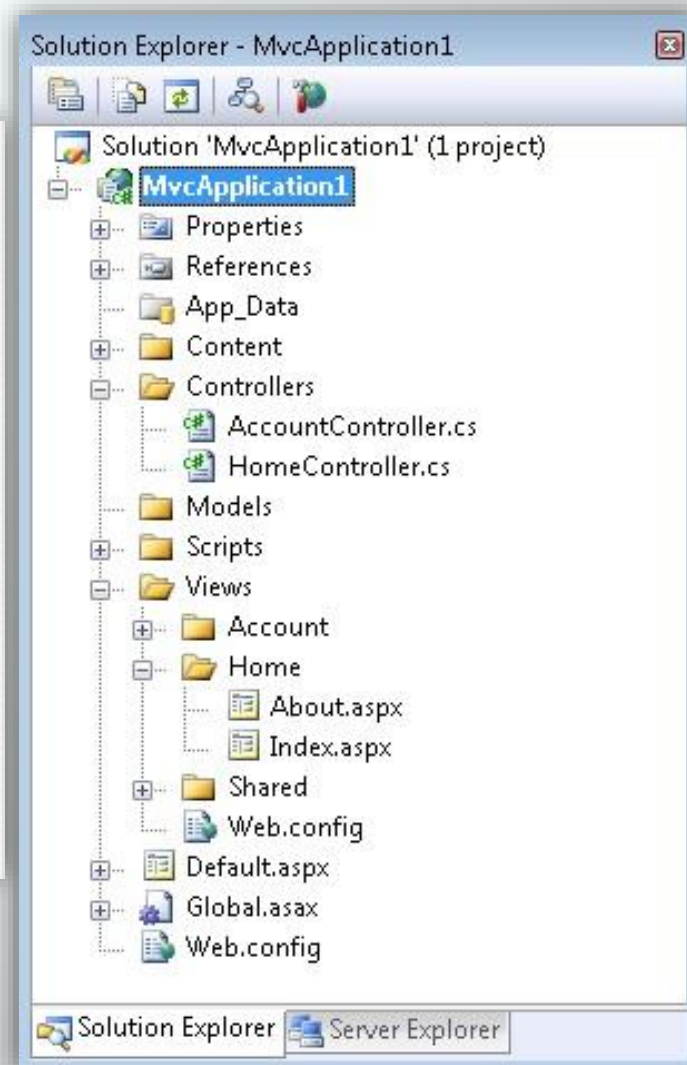
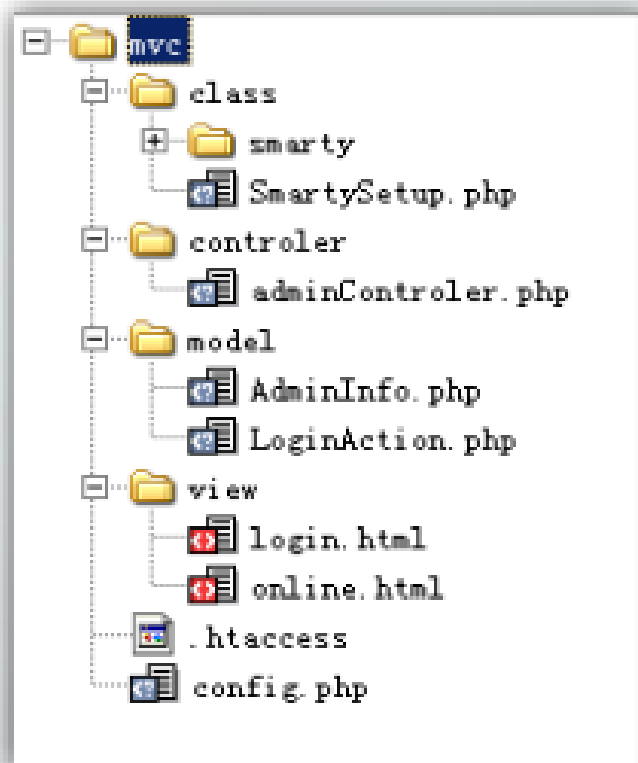
## ■ MVC的起源

- MVC架构最早是smalltalk语言研究团提出的，应用于用户交互应用程序中。

## ■ MVC的设计思想

- 把一个应用的输入、处理、输出流程按照Model、View、Controller的方式进行分离。

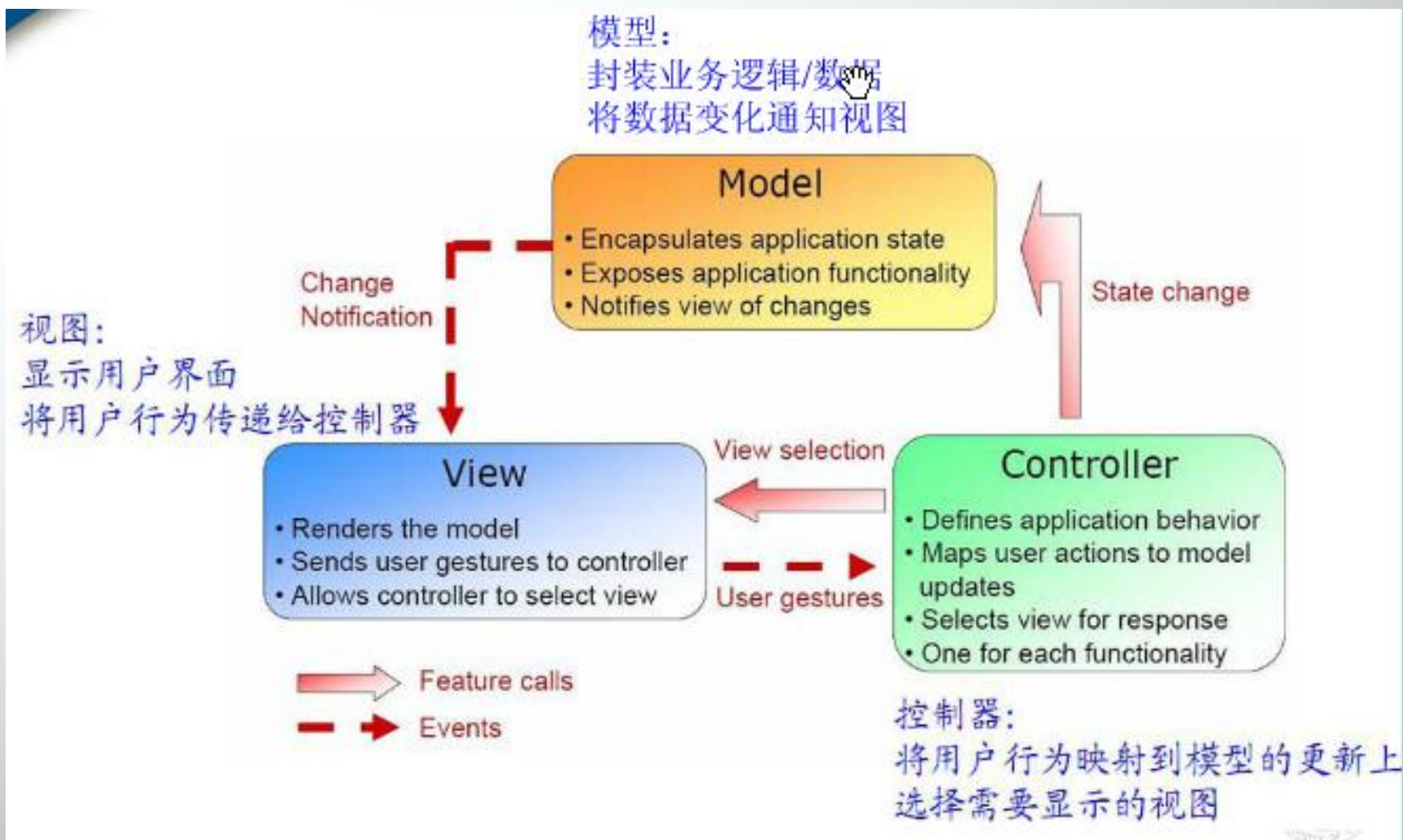
## ■ 我们知道的MVC.....



## ■ 课程内容

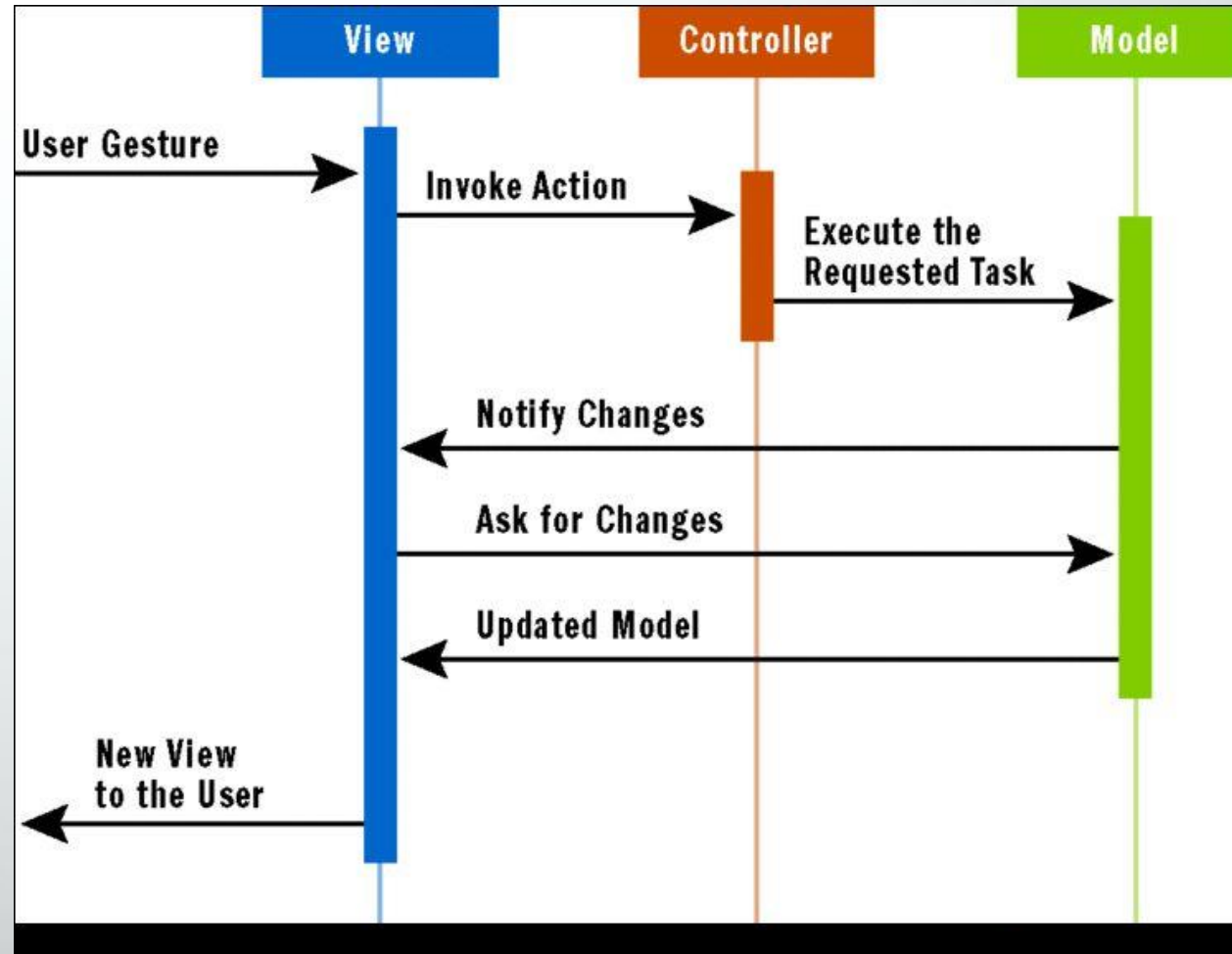
- MVC概述
- MVC实现细节
- MVC之设计模式
- 表现层演化MVP

## ■ MVC体系结构

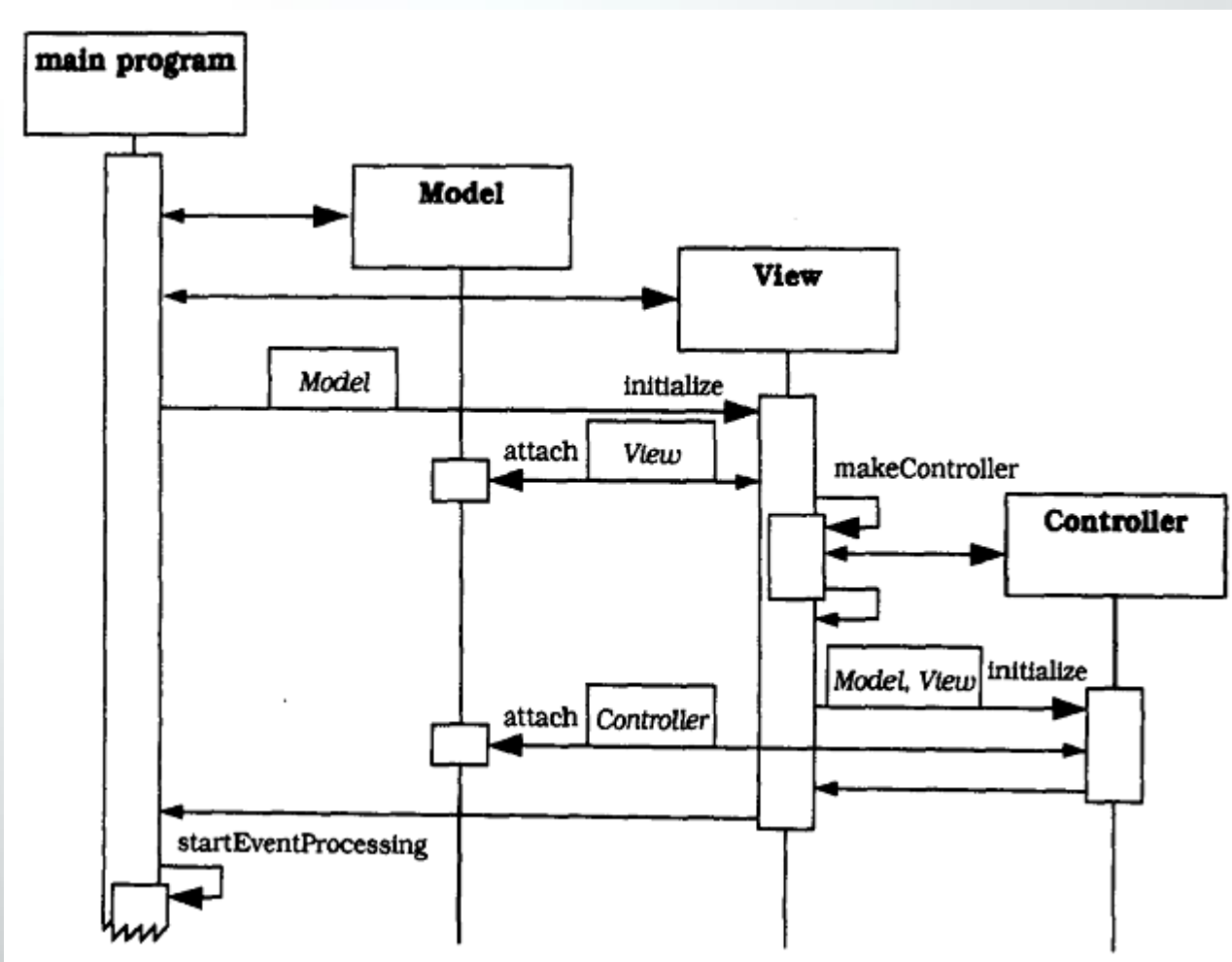




## ■ MVC的动态行为



## ■ MVC结构如何初始化



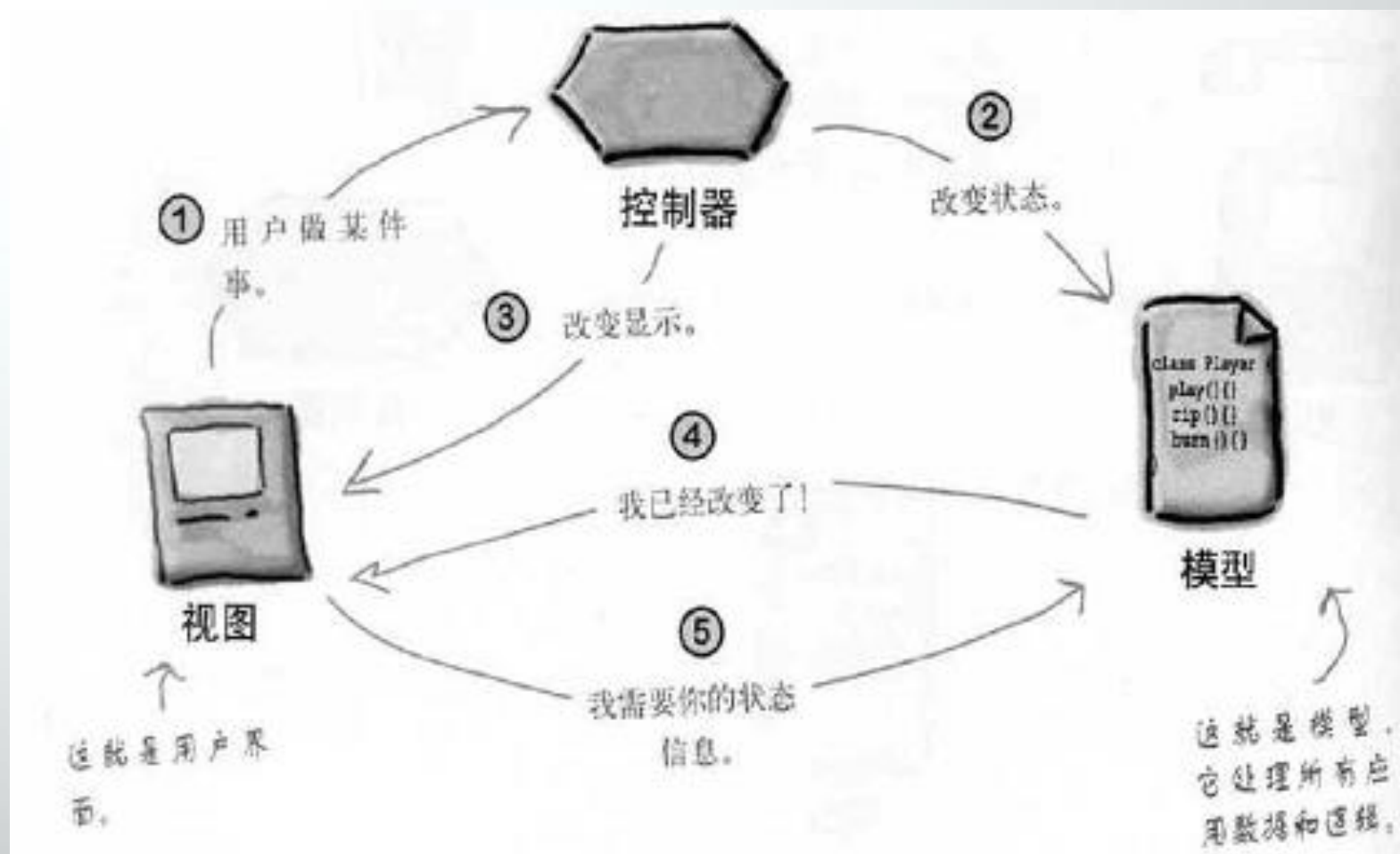
## ■ 课程内容

- MVC概述
- MVC实现细节
- MVC之设计模式
- 表现层演化MVP

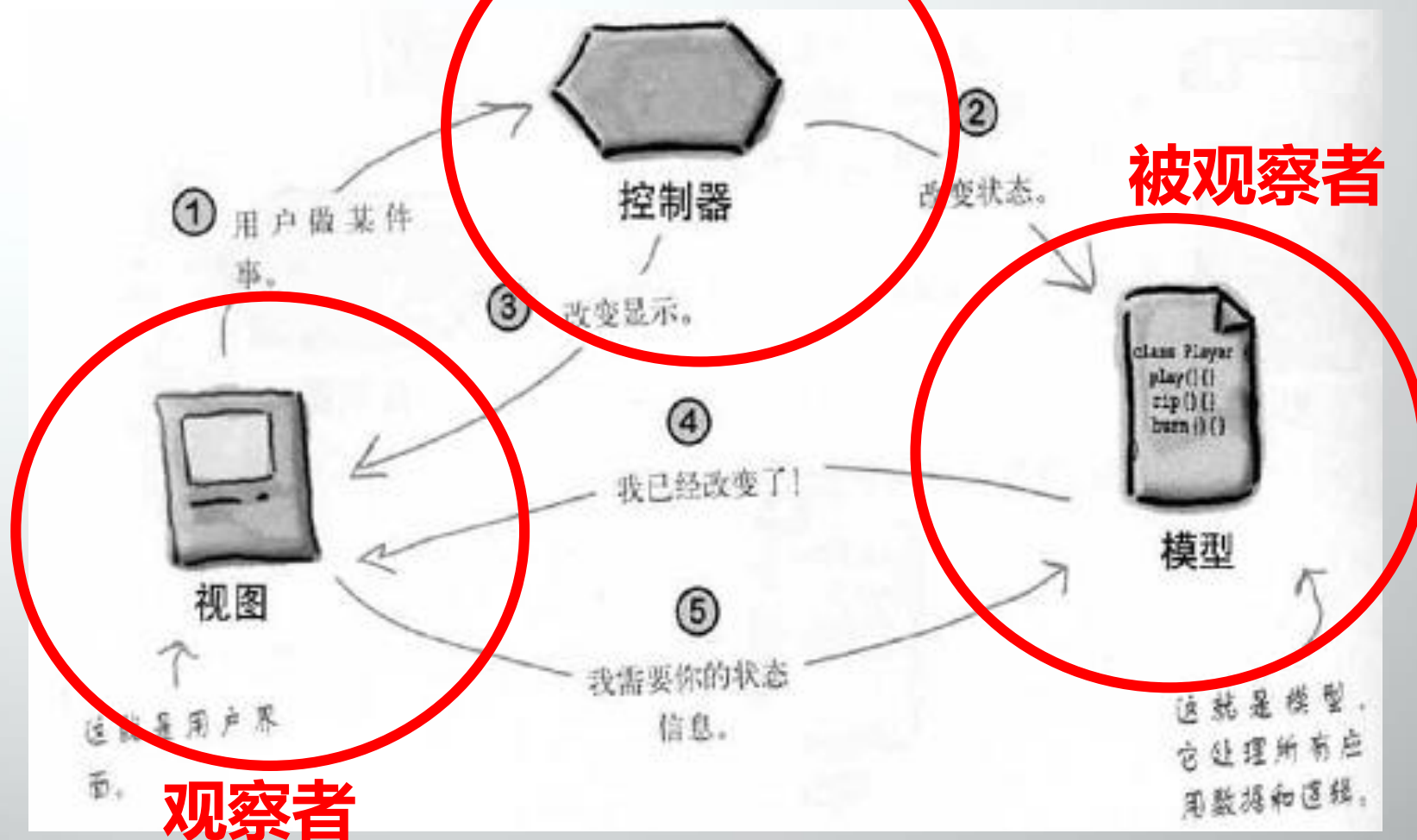
# ■ MVC——复合模式之王

- 观察者模式
- 策略模式
- 组合模式
- .....

## ■ MVC观察者运作细节

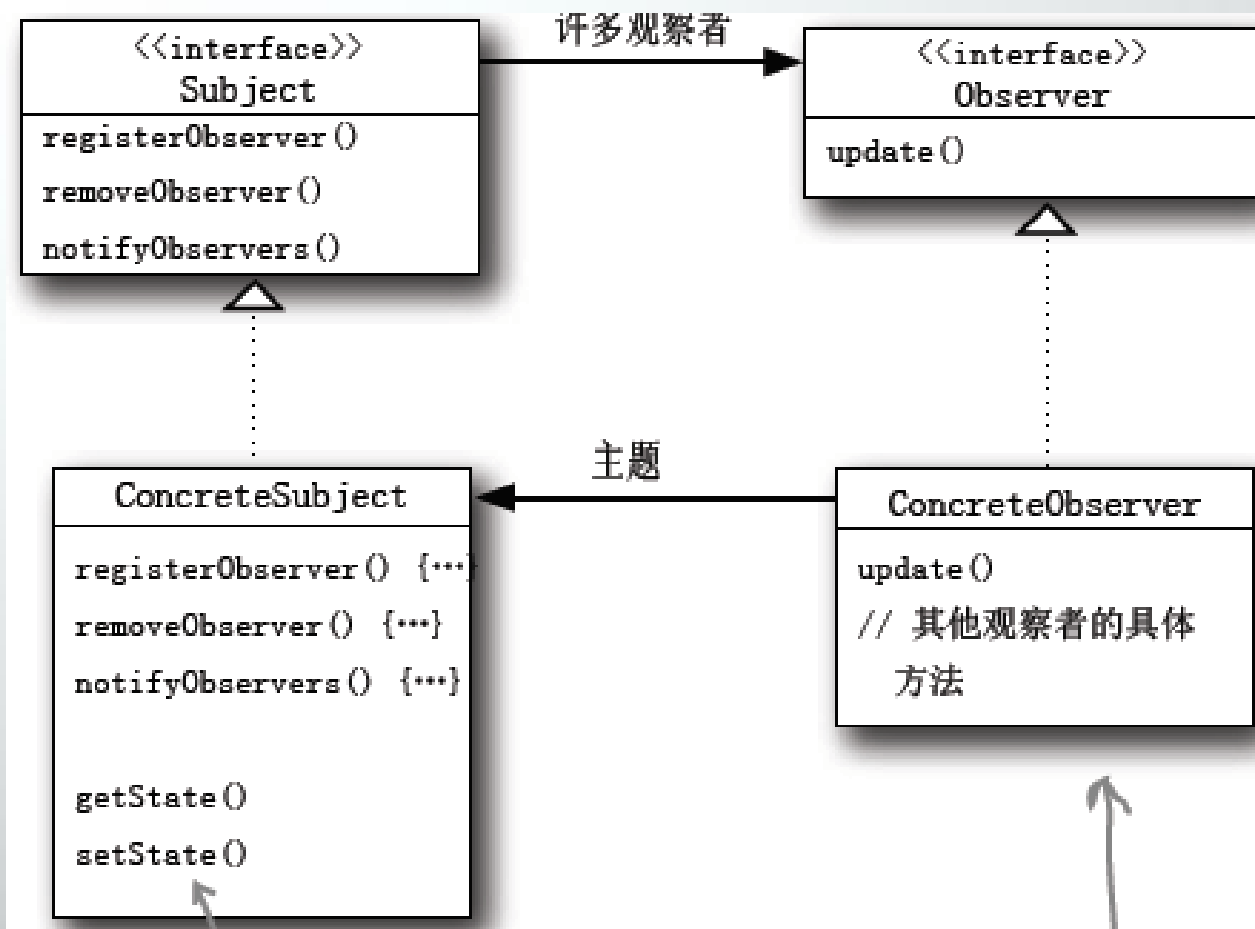


## ■ MVC观察者运作细节

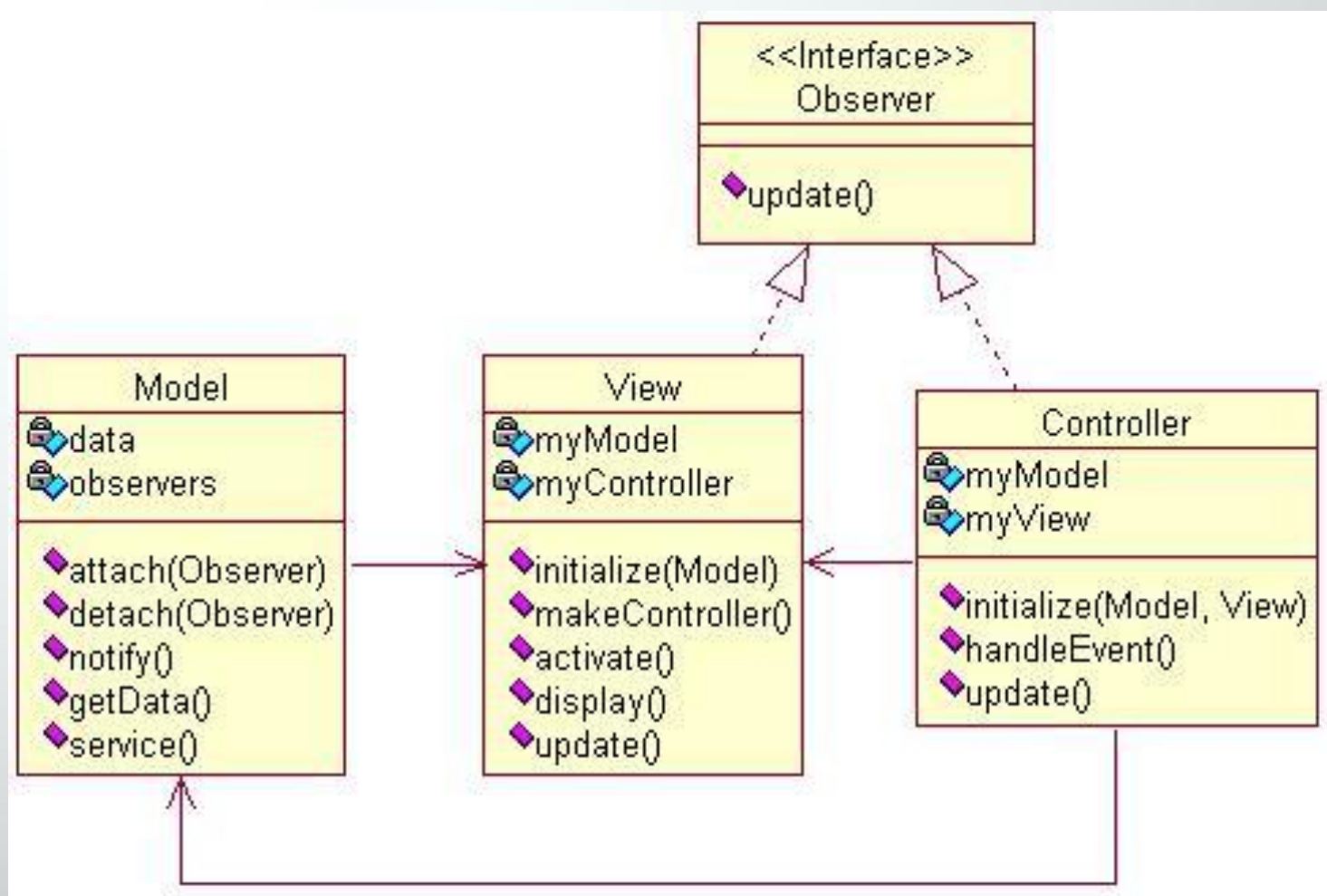


## ■ 观察者模式回顾

■ 让对象能够在状态改变时被通知



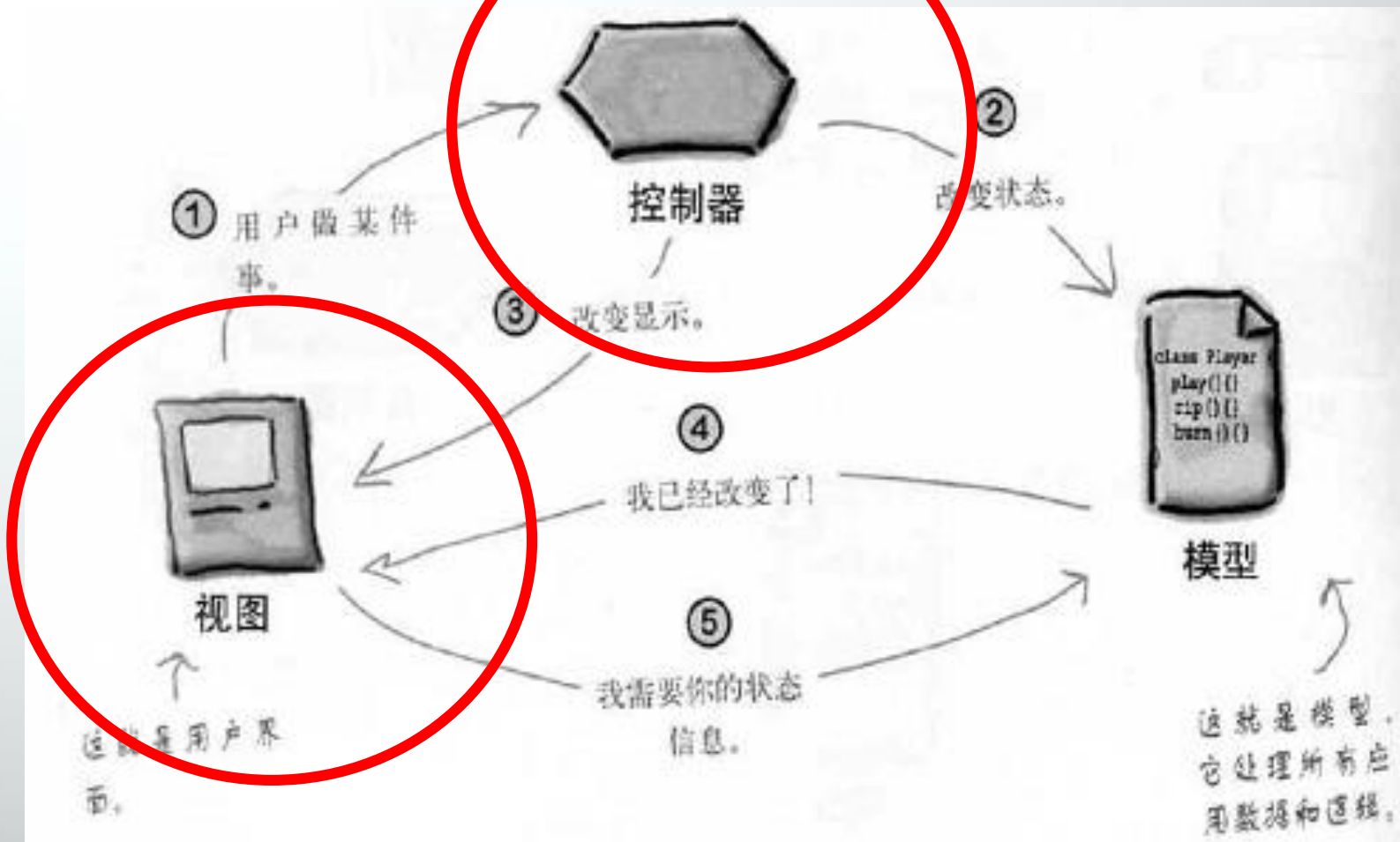
## ■ MVC的设计类图





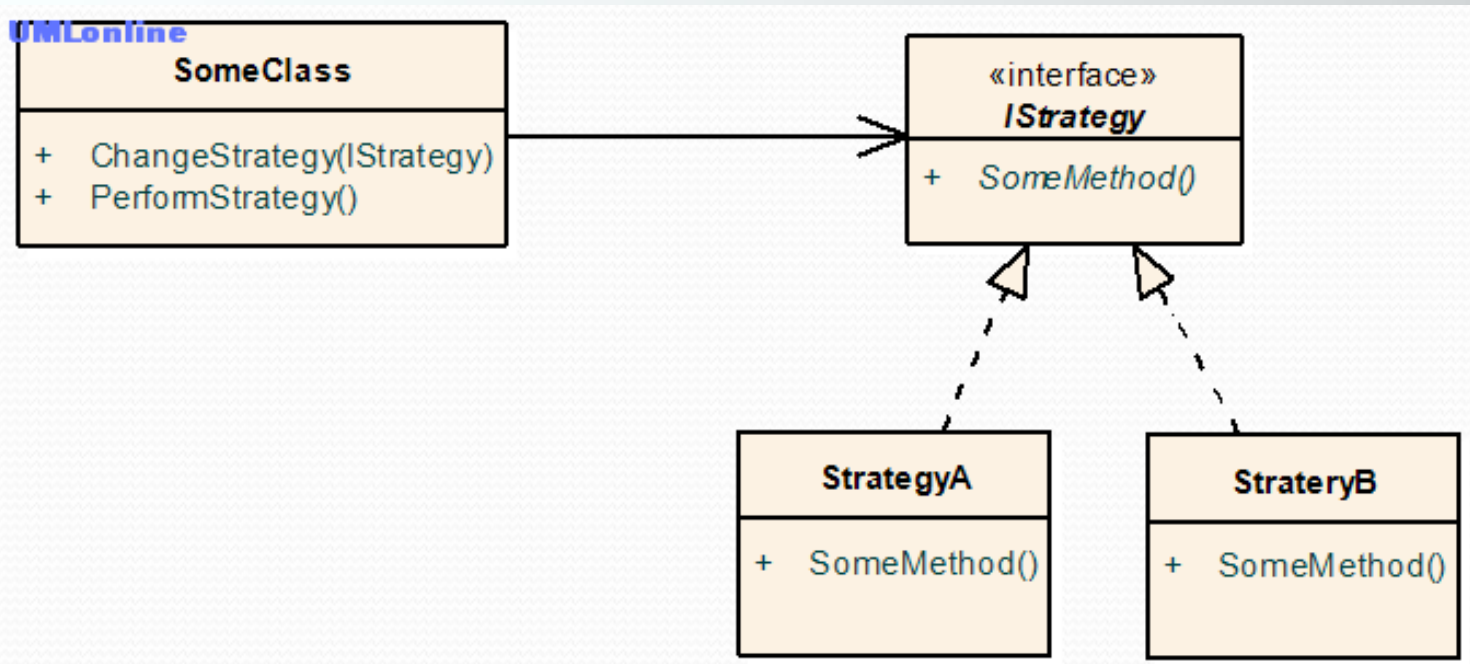
# 策略模式

## ■ MVC策略运作细节

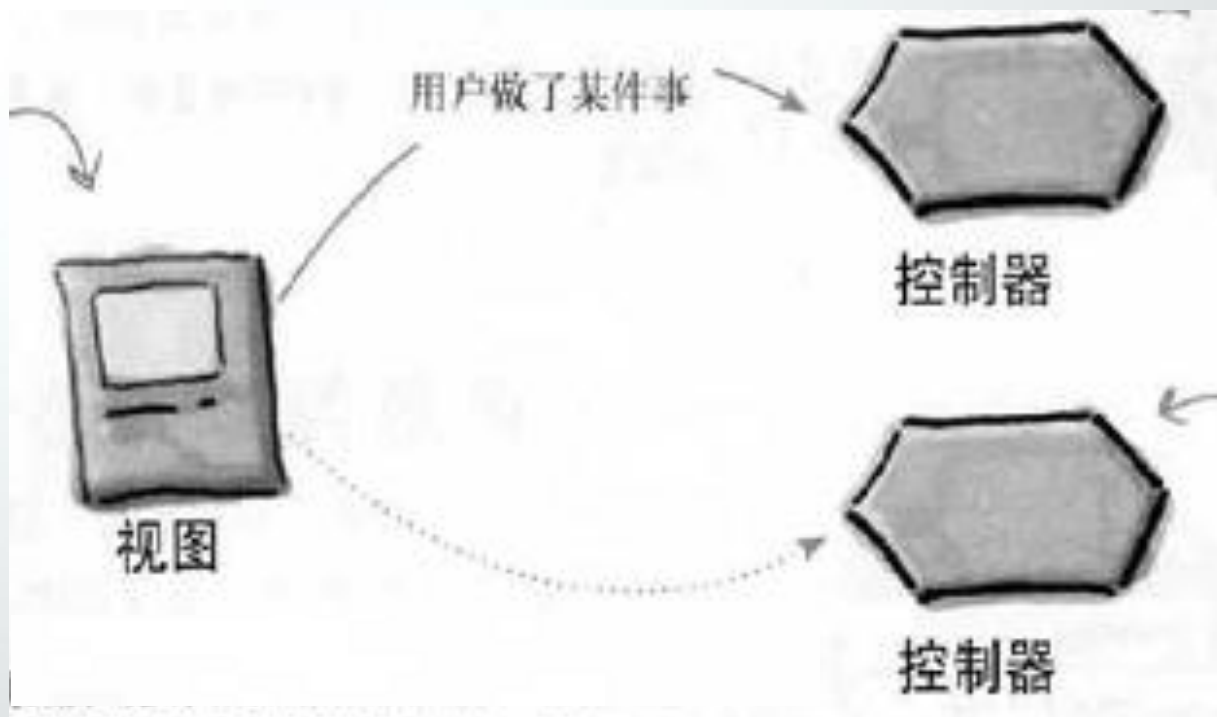


## 策略模式

- 策略模式定义了一系列的算法，并将每一个算法封装起来，而且使它们还可以相互替换。策略模式让算法独立于使用它的客户而独立变化



## 策略模式在MVC中的应用



## 策略模式在MVC中的应用

昵图网nipic.com/xiaofanfan66  
CHINA MOBILE  
移动信息专家

邮件系统

lintong@chinamobile.com [帮助, 退出]

收信

写信

文件夹

未读邮件(15)

收件箱

草稿箱

已发送

已删除邮件

垃圾邮件

病毒文件夹

邮箱服务

发送彩信

个人通讯录

企业通讯录

邮箱选项

查找

帮助

退出

收件箱 (共 265 封, 其中未读邮件 209 封)

删除

永久删除

转移

标记

首页

上一页

下一页

末页

页数: 1 / 6

<input type="checkbox"/>	发件人	主题	时间		大小
<input type="checkbox"/>	Webmaster	来自Webmaster的邮件	2009-02-25		103.18M
<input type="checkbox"/>	林同	主题审核	2009-02-21		103.18M
<input type="checkbox"/>	冯霜	参加生日会	2009-02-25		28.63K
<input type="checkbox"/>	Webmaster	2008年度年度大会的通知, 请大家准时参加	2009-02-21		69.67K
<input type="checkbox"/>	林同	主题审核	2009-02-25		16.83K
<input type="checkbox"/>	冯霜	参加生日会	2009-02-21		10.8K
<input type="checkbox"/>	Webmaster	2008年度年度大会	2009-02-25		103.18M
<input type="checkbox"/>	林同	主题审核	2009-02-21		103.18M
<input type="checkbox"/>	冯霜	参加生日会	2009-02-21		28.63K

选择: 全部 - 未读 - 已读 - 反选

首页

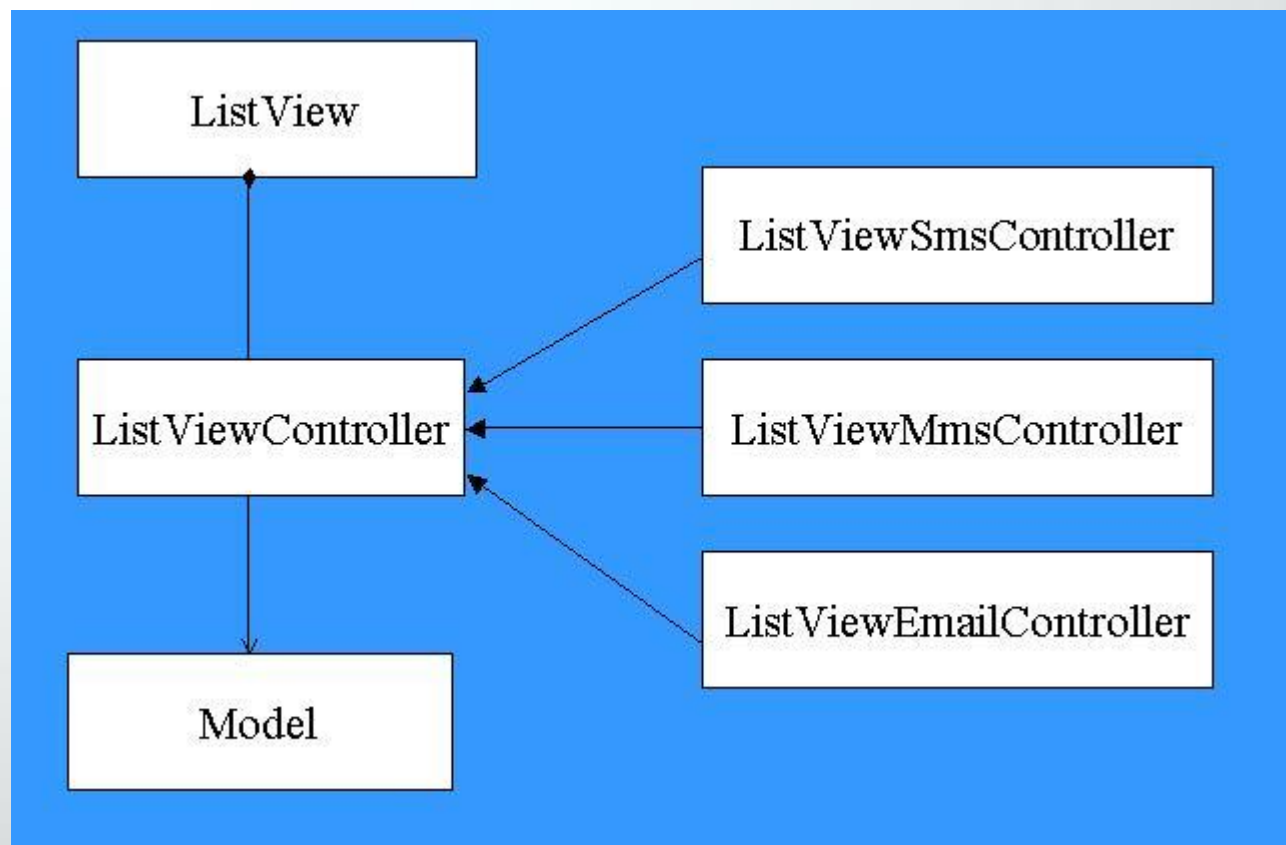
上一页

下一页

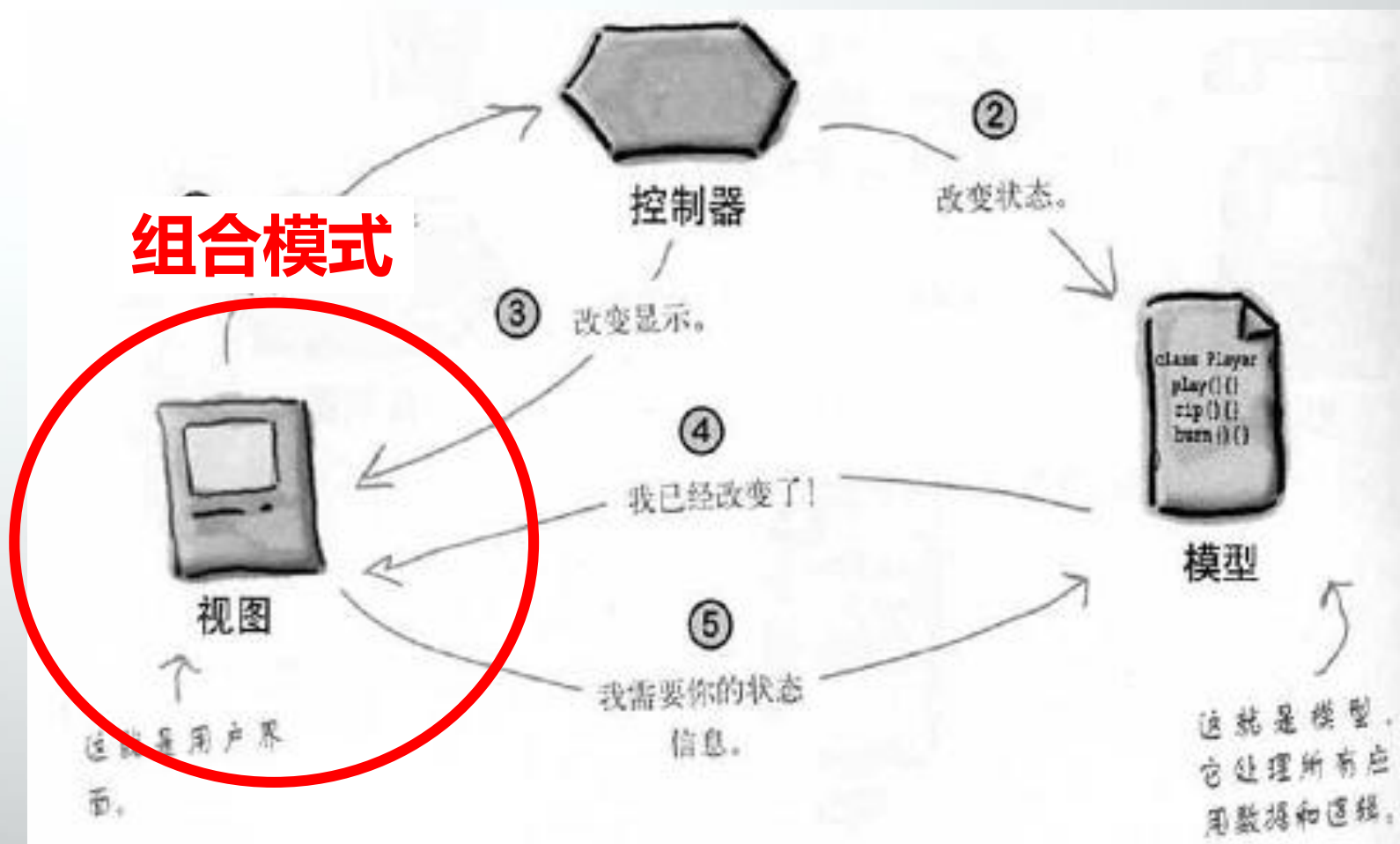
末页

页数: 1 / 6

## 策略模式在MVC中的应用



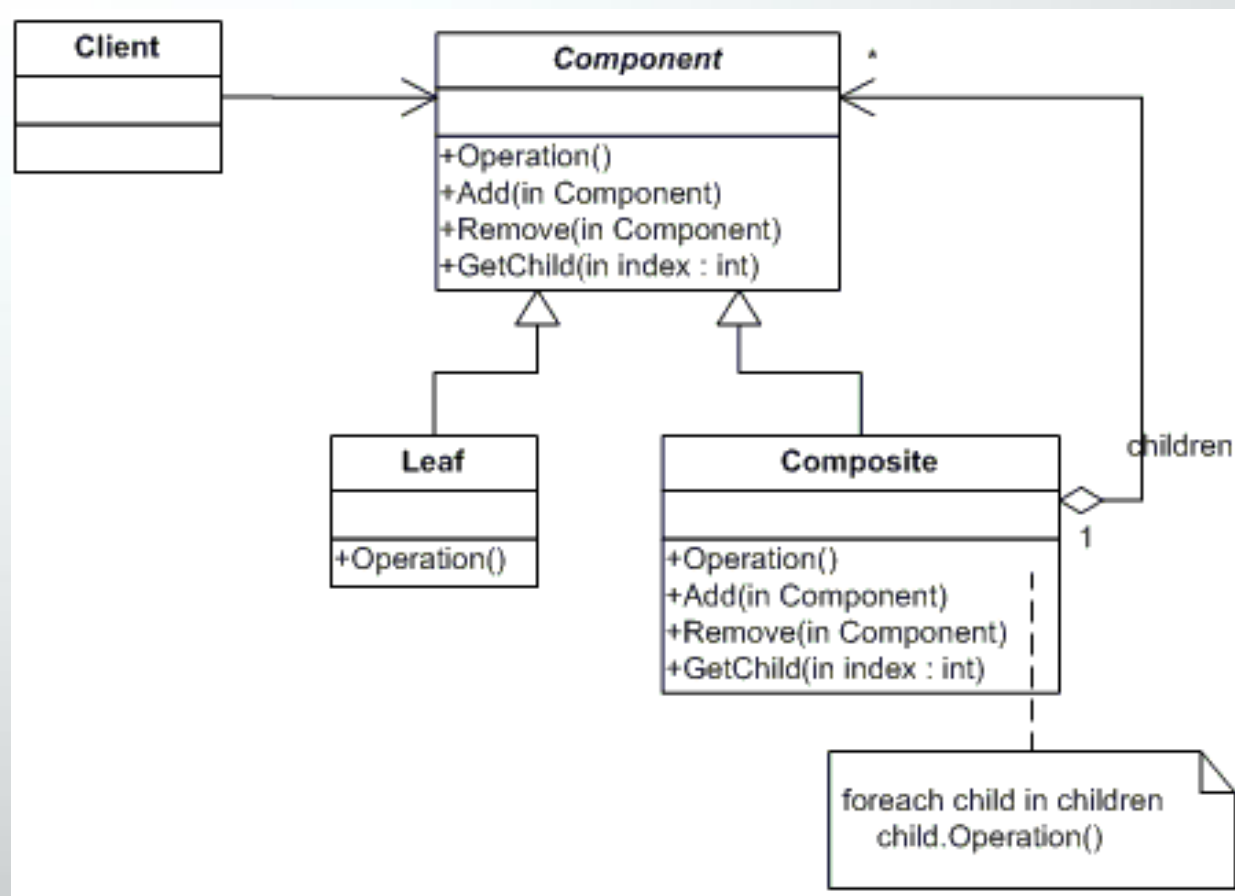
## ■ MVC运作细节





## 组合模式回顾

- 客户用一致的方式处理对象集合和单个对象。

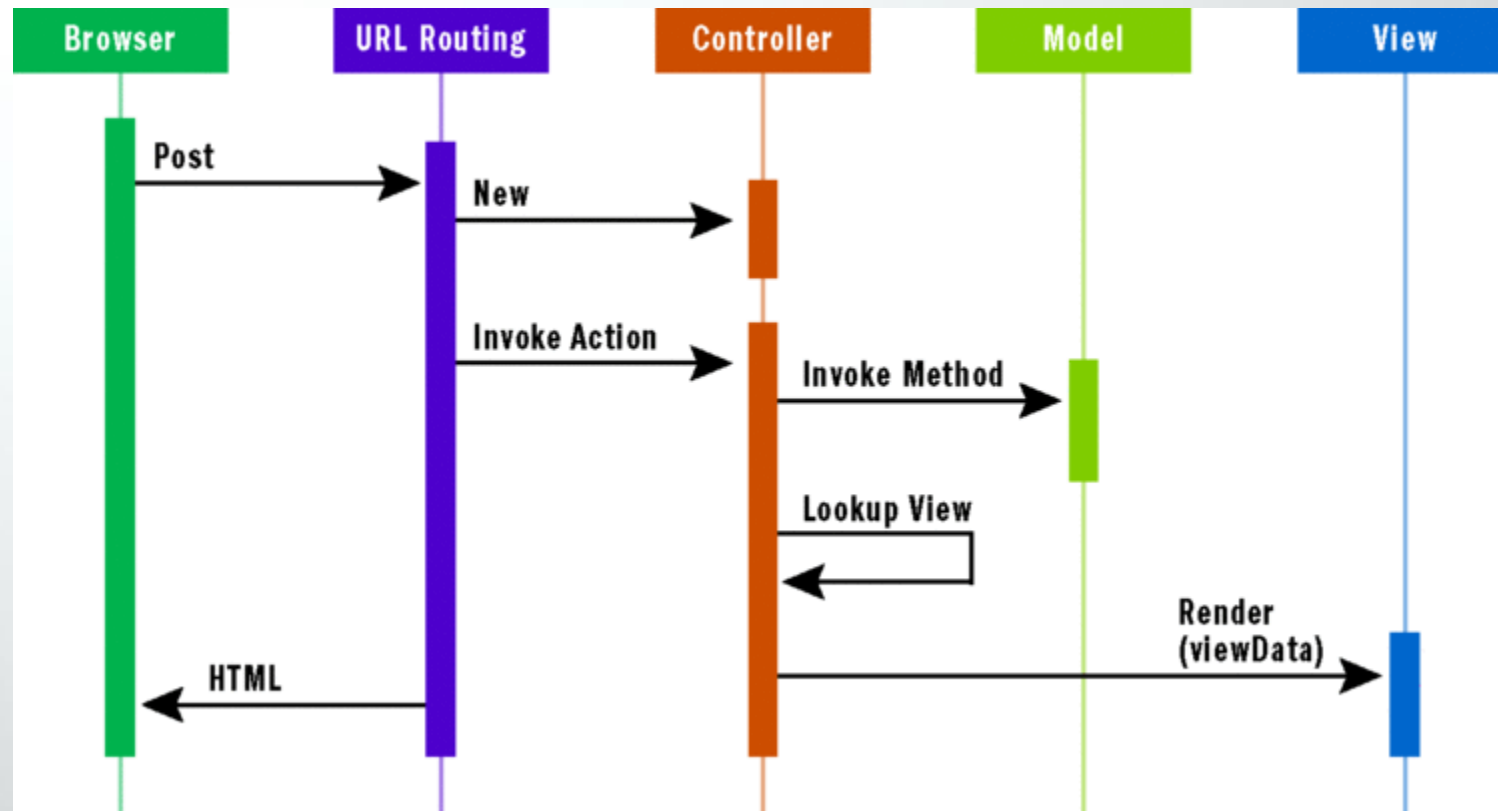


## 组合模式在MVC中的应用





## ■ MVC模式变体——Model2



## ■ MVC不足

- View是可以直接访问Model的
- View是依赖于Model的
- 有一些业务逻辑在View里实现

## ■ 课程内容

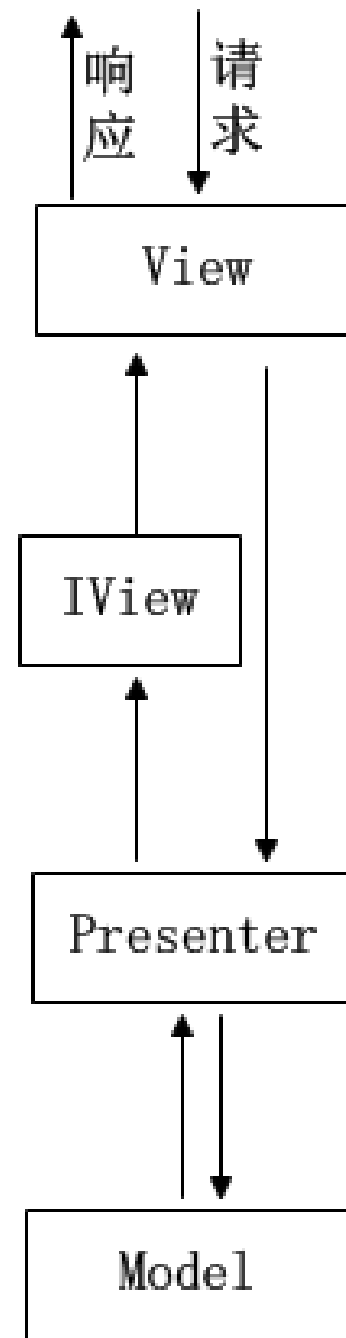
- MVC概述
- MVC实现细节
- MVC之设计模式
- 表现层演化MVP

## 表现层的演化——MVP

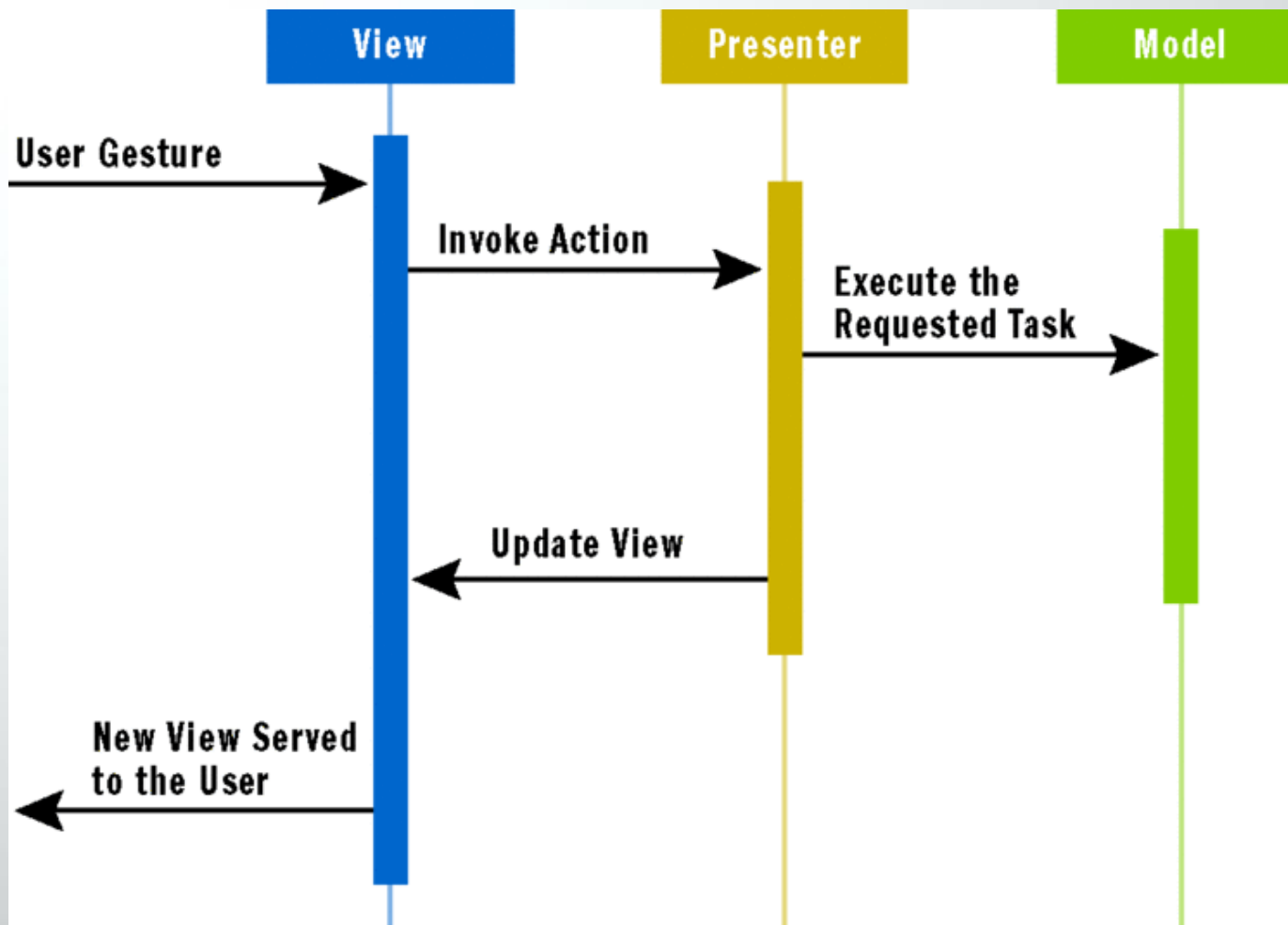
### ■ 模型-视图-展示器

### ■ Model View Presenter

- 模型与视图完全分离
- 可以更高效地使用模型，因为所有的交互都发生在一个地方——Presenter内部
- 我们可以将一个Presenter用于多个视图，而不需要改变Presenter的逻辑。这个特性非常的有用，因为视图的变化总是比模型的变化频繁
- 如果我们把逻辑放在Presenter中，那么我们就可以脱离用户接口来测试这些逻辑（单元测试）

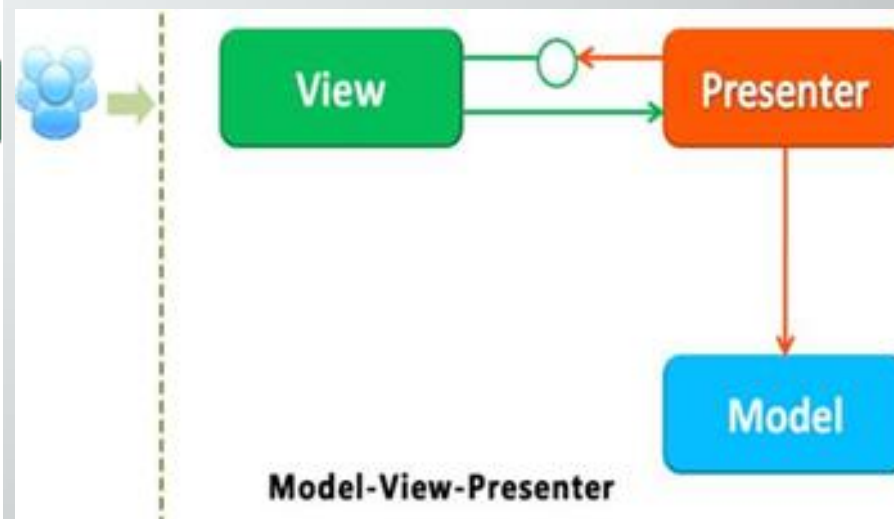
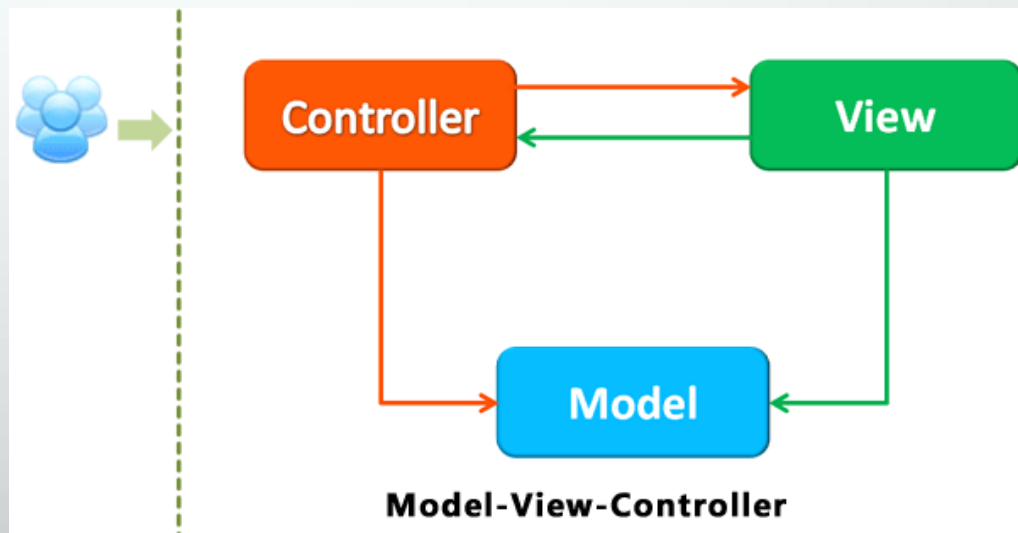


## ■ MVP交互过程



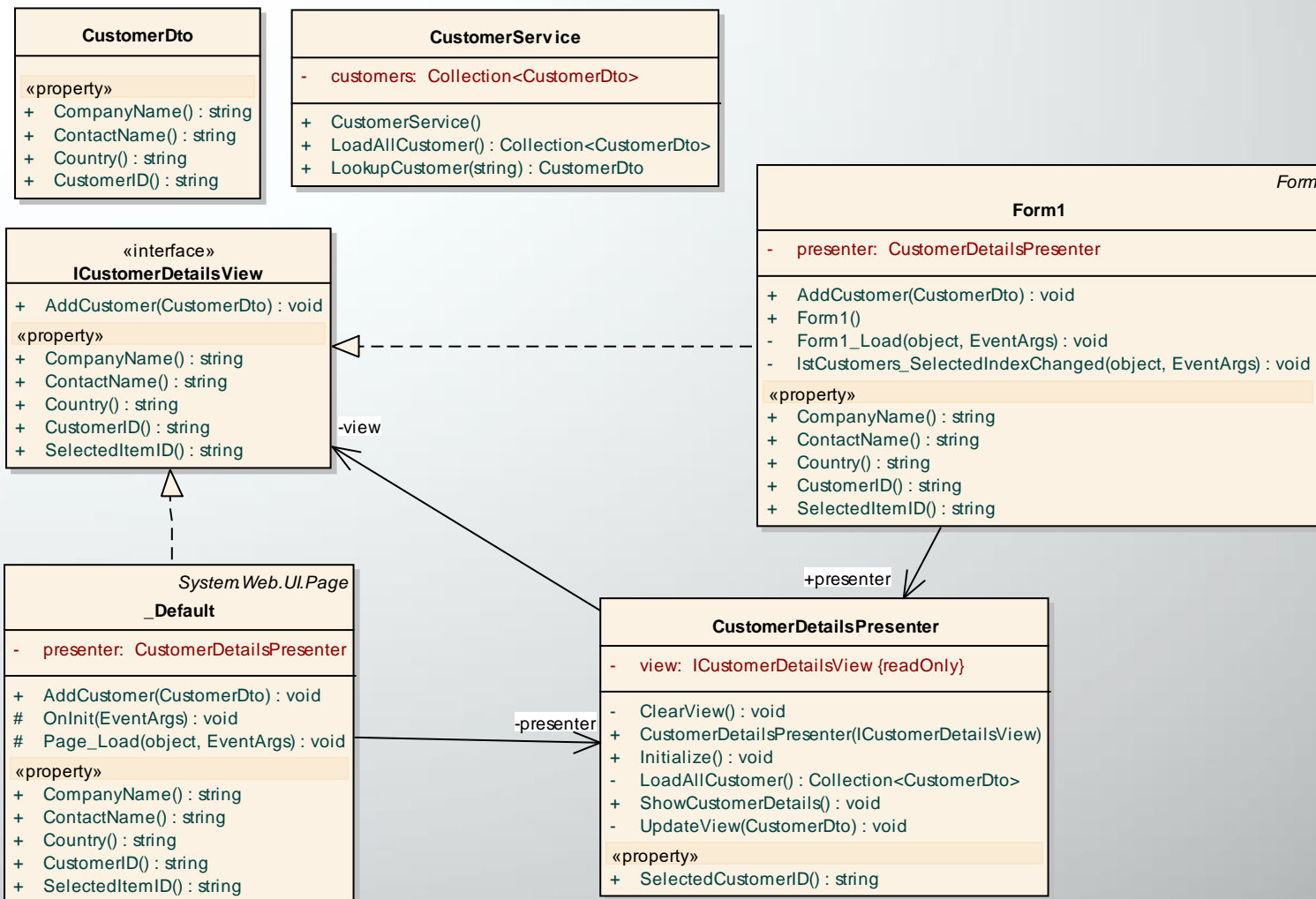
## ■ MVP与MVC对比

- 视图并不了解模型
- 展示器将忽略视图中使用的具体UI技术
- 视图可以被模拟，以便测试



# MVP案例实例

class Use Case Model





## ■ ■ 小结



**Thank You , 谢谢 !**