河北师大软件学院 *@Software College*

前端开发与HTML5
程序设计基础

王岩

2.13 MVC与ThinkPHP框架

（一）

框架

- 框架（Framework）：
  - （1）建筑工程中，由梁、柱等联结而成的结构
  - （2）比喻事物的组织、结构

1 第三章 ThinkPHP
第1节 ThinkPHP基础
2 本节内容
3 重点/难点
4 本节内容
5 什么是框架？
●框架（Framework）：
➢（1）建筑工程中，由梁、柱等联结而
➢（2）比喻事物的组织、结构
6 软件中的框架
●从软件的复用谈起：
➢无论是什么类型的软件，都具有相同基
模式实现程序的基础流程。
➢程序普遍具有一些通用性的操作，如数
●软件框架
➢框架是可被应用开发者定制的应用骨架
一些通用操作的API。框架提供了程序
人员实现具体的商业逻辑。

软件框架

- 从软件的复用谈起：
  - 无论是什么类型的应用，都具有类似的基础流程。通常应用MVC模式。
  - 程序普遍具有一些通用性的操作，如数据库操作。

- 软件框架
  - 框架是可被应用开发者定制的应用骨架。实现MVC模式并提供一些通用操作的API。框架提供了程序底层通用部件，由开发人员实现具体的业务逻辑。

使用框架的优势

- 快速完成项目开发
  - 框架提供了程序基础结构
  - 框架提供通用API，方便调用
- 轻松实现大中型项目
- 利于团队合作
  - 业务逻辑人员、前台开发人员等分别编辑不同页面，分工明确，不会产生混乱。
- 利于项目后期维护

常见主流PHP框架

- Yii
- laravel
- CodeIgniter
- Zend Framework

ThinkPHP框架的特点

- 快速、简单的面向对象轻量级框架
- 丰富的API
- 封装数据库CURD操作简单易用
- 易于扩展
- 国内团队开发、文档教程齐全

构建第一个ThinkPHP程序

下载

- 下载地址：
  - http://www.thinkphp.cn/down.html
- 版本选择：
  - 版本：3.2.* （完整版）

框架目录结构

```
www  WEB部署目录（或者子目录）
├─index.php          入口文件
├─README.md          README文件
├─Application        应用目录
├─Public             资源文件目录
└─ThinkPHP           框架目录
```

```
├─ThinkPHP 框架系统目录（可以部署在非web目录下面）
│    ├─Common          核心公共函数目录
│    ├─Conf            核心配置目录
│    ├─Lang            核心语言包目录
│    ├─Library         框架类库目录
│    │    ├─Think      核心Think类库包目录
│    │    ├─Behavior   行为类库目录
│    │    ├─Org        Org类库包目录
│    │    ├─Vendor     第三方类库目录
│    │    ├─ ...       更多类库目录
│    ├─Mode            框架应用模式目录
│    ├─Tpl             系统模板目录
│    ├─LICENSE.txt     框架授权协议文件
│    ├─logo.png        框架LOGO文件
│    ├─README.txt      框架README文件
│    └─ThinkPHP.php    框架入口文件
```

框架初始化

- 将ThinkPHP目录复制到网站目录中，并访问首页进行初始化。
  - 第一次访问时，创建公共模块Common、默认Home模块、运行时Runtime目录

自动生成目录

```
Application
├Common              应用公共模块
|   ├Common          应用公共函数目录
|   └Conf            应用公共配置文件目录
├Home                默认生成的Home模块
|   ├Conf            模块配置文件目录
|   ├Common          模块函数公共目录
|   ├Controller      模块控制器目录
|   ├Model           模块模型目录
|   └View            模块视图文件目录
├Runtime             运行时目录
|   ├Cache           模版缓存目录
|   ├Data            数据目录
|   ├Logs            日志目录
|   └Temp            缓存目录
```

初识控制器（Controller）

- 修改Application/Home/Controller/IndexController.class.php中的index方法，查看页面效果。

理解MVC

传统典型应用开发

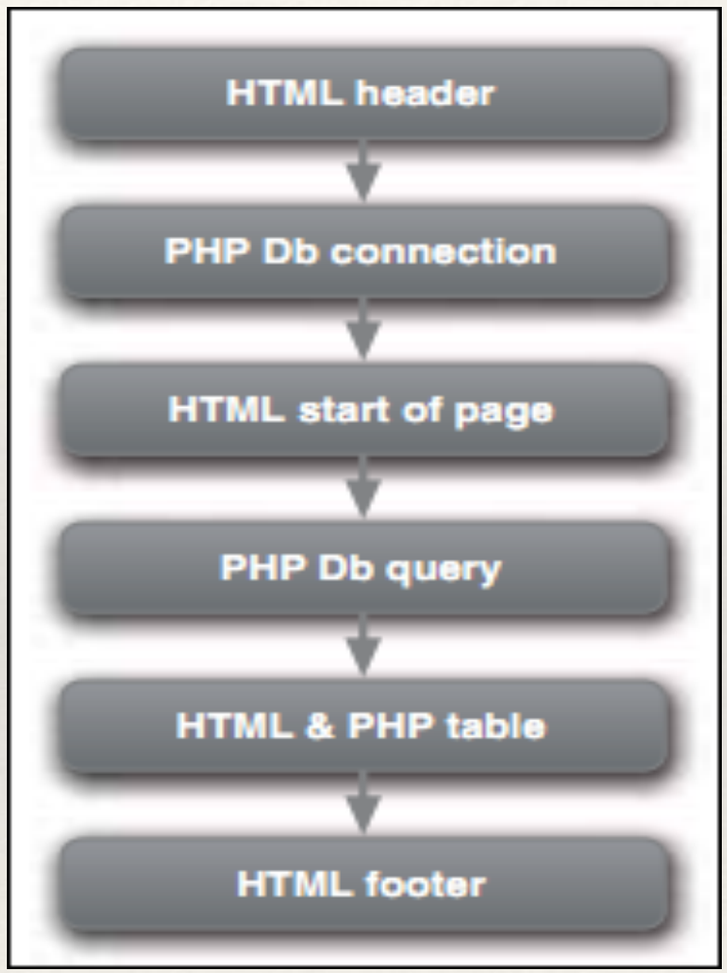
- 引入页头文件（PHP代码引入HTML代码文件）
- 当前页面的标题等信息（HTML代码）
- 数据库操作（PHP操作数据库代码）
- 输出帖子列表（PHP、HTML、数据库）
- 分页码（PHP代码、HTML代码）
- 关闭数据库（PHP操作数据库代码）
- 引入页脚文件（PHP代码引入HTML代码文件）

代码

```php
<?php
    //编辑自己的帖子
    $id = $_GET['id'];

    //引入页头文件
    include ('header.inc.php');

    //查询主贴信息
    $sql = "select message_title, message_content from message where message_id = $id;";
    $msgs = mysql_query($sql);
    $msg = mysql_fetch_array($msgs, MYSQL_ASSOC);
?>
        <!-- 面包屑(即"留言板->登录"之类的内容) -->
        <div class="navbar clearfix">
            <div class="breadcrumb">
                <ul>
                    <li class="first"><a href="index.php">简单留言板</a></li>
                    <li> <span class="arrow sep">&#x25BA;</span> 修改帖子</li>
                </ul>
            </div>
        </div>
```



优缺点总结

- PHP代码、HTML代码、操作数据库代码混合在一起
- 优点
  - 方便编写程序，易于理解程序中各部分内容的作用
  - 易于学习，开发成本很低
- 缺点
  - 不利于代码的重复利用（或者频繁使用include语句，或者代码不能重复利用，如数据库操作等）
  - 不利于较大项目的团队合作（开发人员不需要理解CSS、JavaScript等技术；前台人员不需要了解数据库、PHP等技术）
  - 不利于代码的后期扩展（如增加一个功能，需要修改绝大部分代码）

逻辑分析

- 一般而言，Web应用程序有三部分内容组成
  - 模型：数据库操作部分，例如：获得数据库的数据、向数据库中插入、删除、更新数据、……
  - 视图：HTML前台代码部分，例如：用户看到的HTML代码结构、CSS样式、JavaScript前台页面交互效果、……
  - 控制器：连接模型和视图的操作逻辑，例如：把数据库查询结果输出到HTML中、获得用户输入的表单数据、获得用户HTTP请求信息、反馈HTTP响应信息给用户、……

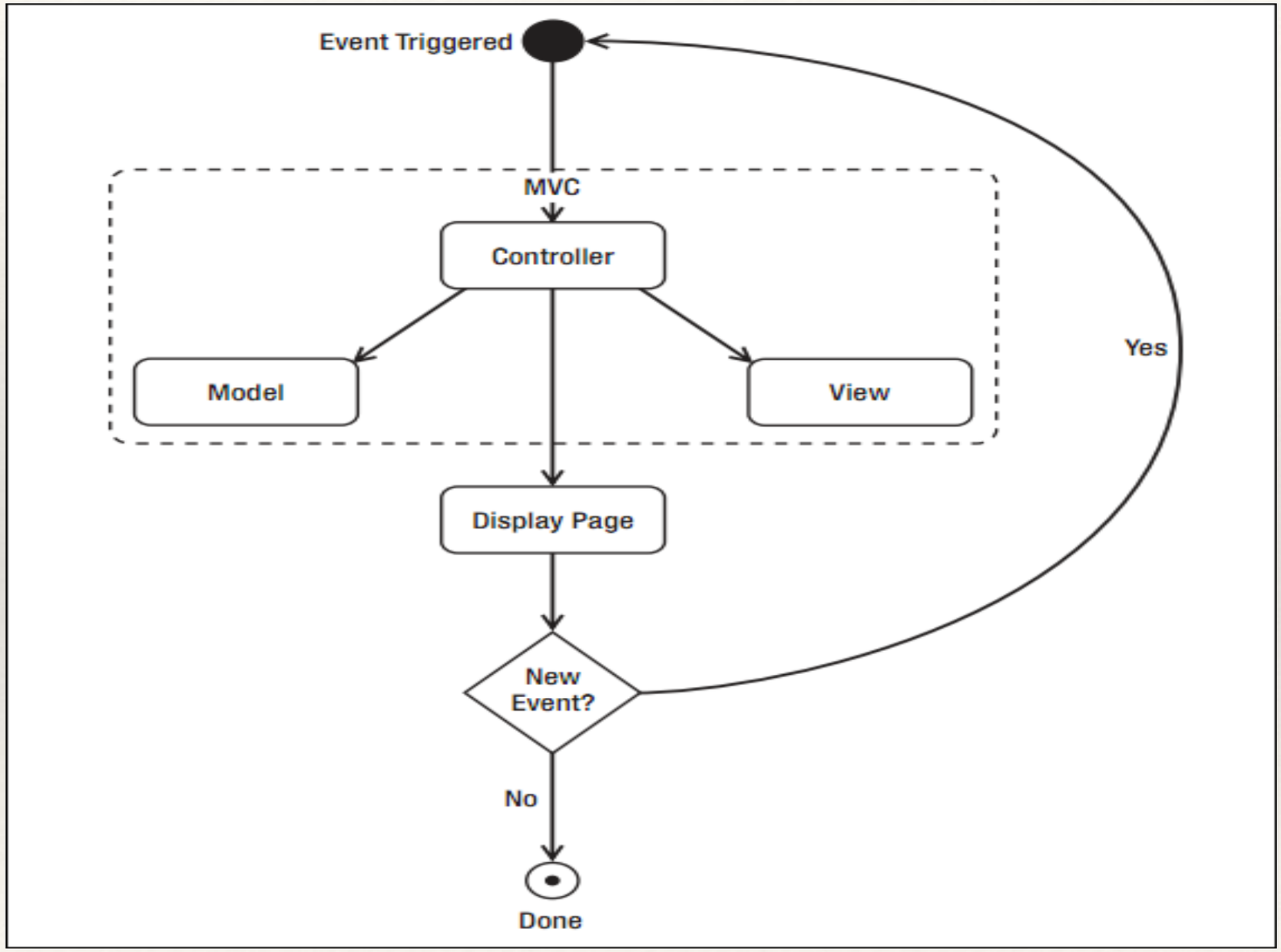
MVC

- MVC是一种软件框架模式
- 它强制的使应用程序的输入、处理、输出分开。使用MVC的应用程序被分为三个核心部件：模型（Model）、视图（View）、控制器（Controller）

MVC

- 模型（Model）：数据库处理、session管理、数据校验等所有与数据相关的任务，是MVC中工作量最大的一部分
- 视图（View）：生成HTML代码（可能含有少量PHP代码）、生成RSS阅读等所有与Web页面结构、样式相关的任务
- 控制器（Controller）：接收客户请求、搜集所需资源、返回适当等与用户请求、响应交互的沟通Model和View的核心动作模块

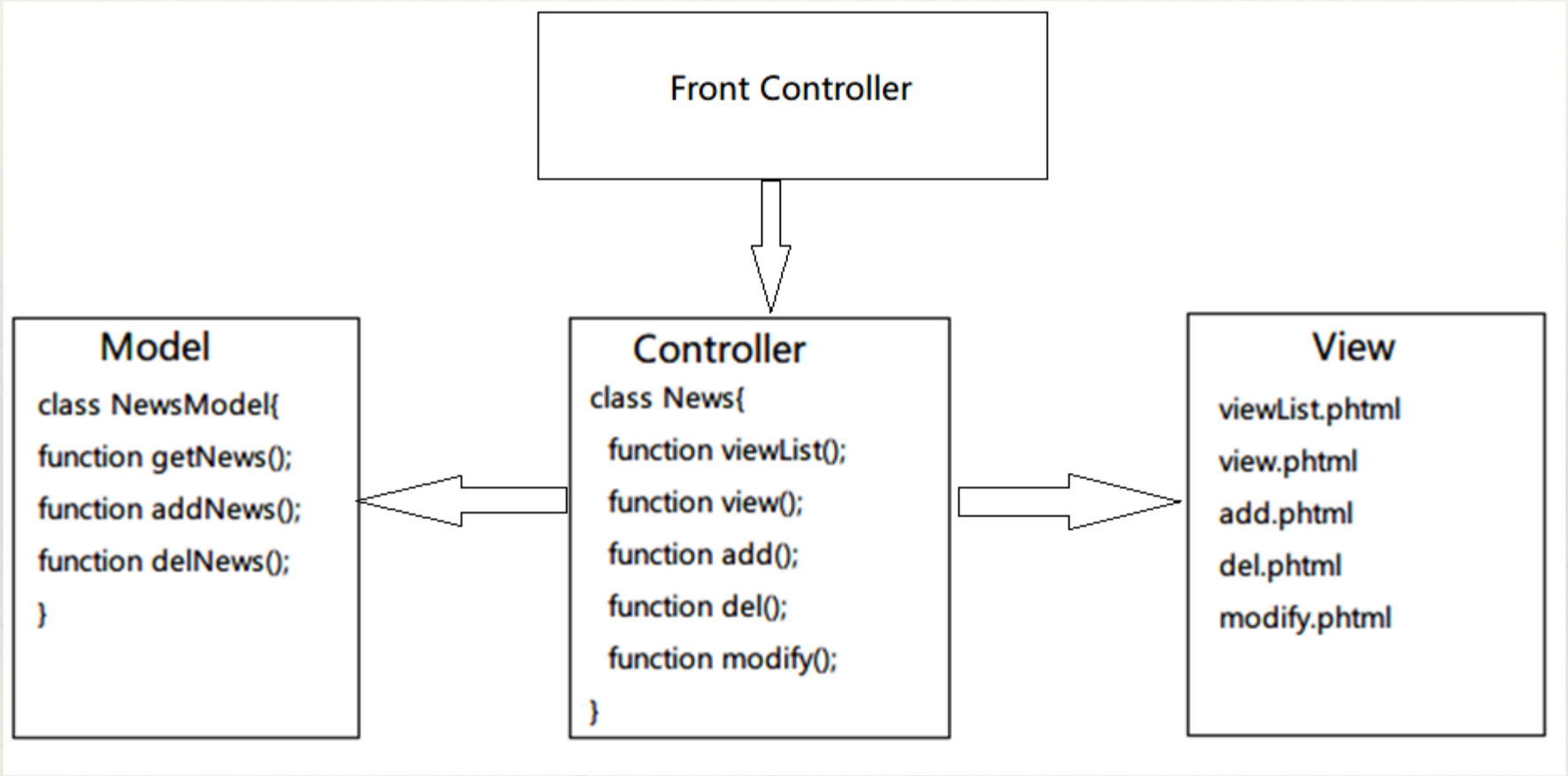
MVC执行过程


Event Triggered
MVC
Controller
Model
View
Display Page
New Event?
Yes
No
Done

MVC设计示例

- 对于某程序中新闻模块的设计
- 实现功能：新闻的增删改查
- 传统形式：
- 传统形式需要的程序文件：
  - viewNewsList.php
  - viewNews.php
  - editNews.php
  - addNews.php
  - delNews.php
- 缺点：1. 程序文件多而繁杂 2. 代码混乱 3. 不利于沟通协作 4. 不利于维护（更新、扩展和修改）5. …

MVC设计示例


Front Controller
Model
class NewsModel{
function getNews();
function addNews();
function delNews();
}
Controller
class News{
function viewList();
function view();
function add();
function del();
function modify();
}
View
viewList.phtml
view.phtml
add.phtml
del.phtml
modify.phtml

前端控制器

- 前端控制器（Front Controller）负责：
  - 接收HTTP请求
  - 根据请求决定执行哪一个控制器（Controller）中的哪一个动作（Action）
  - 控制内容输出

单一入口

- 所有的请求都跳转到统一的一个程序入口文件，叫做引导文件。
  - 在ThinkPHP中，项目目录下index.php即为引导文件。
  - 该文件进行程序的初始化，生成并调用前端控制器实例。
- 如何指定访问不同页面？（通过url参数）
  - http://localhost/app/index.php/module/controller/action
    - 访问控制器module模块中controller类中的action方法
  - http://localhost/app/index.php/module/controller/action/var/value/
    - 访问控制器module类中的action方法并传递参数var，值为value

请求URL与控制器对应关系

localhost/app/	Home模块下的Index控制器index方法
localhost/app/index.php/Home/Index/lists	Home模块下的Index控制器lists方法
localhost/app/index.php/Home/News/add	Home模块下的News控制器add方法
localhost/app/index.php/Home/News/edit/id/2	Home模块下的News控制器edit方法，附带参数id=2
localhost/app/index.php/Admin	Admin模块下的Index控制器index方法

MVC框架

- MVC结构由框架实现，利用框架编程时，只需实现具体业务逻辑

- 程序设计
  - 将传统的功能页面进行控制器、动作的划分
  - 拆分出数据库操作部分，形成Model类
  - 页面显示拆分在View中

ThinkPHP中的MVC

- 控制器（Controller）的写法
  - 控制器文件写在 Application/模块名/Controller目录中
  - 控制器文件命名为：控制器名称+Controller.class.php，控制器名称首字母大写
  - 如：UserController.class.php
  - 控制器类名为：控制器名称+Controller
  - 控制器继承Controller类

ThinkPHP中的MVC

- 视图（View）的写法
- 视图文件的位置和命名：
  - 在Application/模块名/View 目录中
    - 以控制器为名建立目录，即一个控制器对应一个视图目录
    - 以操作名为视图文件名建立视图文件，操作名+.html
- 视图中主要书写HTML、CSS代码
- 在视图中的某些位置如要使用程序中的某些变量的值，使用{变量名}的方式，如{$name}
- 在控制器中，使用$this->display();方法调用视图显示。

ThinkPHP中的MVC

- 模型（Model）的写法
  - 模型文件写在Application/模块名/Model/目录中
  - 模型文件名称：模型名+Model.class.php。模型名跟其对应的表名相同，采用驼峰命名法
  - 模型类名跟模型文件名称相同
  - 模型类继承Model类
  - 在控制器类中，使用全局函数D实例化模型类后，即可调用模型类中的方法

谢谢！