**《PHP项目开发》实验报告**

|  |  |  |  |
| --- | --- | --- | --- |
| 实验名称 | 留言板系统——MVC中基础框架的设计 | | |
| 学院 | 计信学院 | 班级 | 计技172 |
| 姓名 | 邢晨浩 | 学号 | 2017107228 |
| 任课老师 | 云善明 | 成绩 |  |

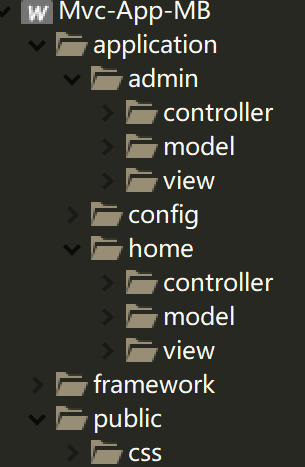
**一、实验目的和要求**

1、掌握MVC设计模式的目录划分与文件组织结构。

2、掌握MVC设计模式的基础模型类、基础框架类的定义。

**二、实验内容**

1、MVC中目录划分与文件组织结构



2、基础模型类的定义

/\*\*

\* 基础模型类

\*/

class model {

protected $db; //保存数据库对象

public function \_\_construct(){

$this->initDB(); // 初始化数据库

}

private function initDB() {

//实例化数据库操作类

$this->db = MySQLPDO::getInstance($GLOBALS['config']['db']);

}

/\*\*

\* 输入过滤

\* @param $arr array 需要处理的字段

\* @param $func callback 用于处理的函数

\*/

protected function filter($arr,$func){

foreach($arr as $v){

//指定默认值

if(!isset($\_POST[$v])){

$\_POST[$v] = '';

}

//调用函数处理

$\_POST[$v] = $func($\_POST[$v]);

}

}

}

3、基础框架类的定义

/\*\*

\* 框架基础类

\*/

class framework{

public function runApp(){

$this->loadConfig(); //加载配置

$this->registerAutoLoad(); //注册自动加载方法

$this->getRequestParams(); //获得请求参数

$this->dispatch(); //请求分发

}

/\*\*

\* 注册自动加载方法

\*/

private function registerAutoLoad(){

spl\_autoload\_register(array($this,'user\_autoload'));

}

/\*\*

\* 自动加载方法

\* $param $class\_name string 类名

\*/

public function user\_autoload($class\_name){

//定义基础类列表

$base\_classes = array(

//类名 => 所在位置

'model' => './framework/model.class.php',

'MySQLPDO' => './framework/MySQLPDO.class.php',

'page' => './framework/page.class.php',

'captcha' => './framework/captcha.class.php',//新增此行代码

);

//依次判断 基础类、模型类、控制器类

if (isset($base\_classes[$class\_name])){

require $base\_classes[$class\_name];

}elseif (substr($class\_name,-5) == 'Model'){ //substr:截取字符串

require './application/'.PLATFORM."/model/{$class\_name}.class.php";

}elseif (substr($class\_name, -10) == 'Controller'){

require './application/'.PLATFORM."/controller/{$class\_name}.class.php";

}

}

/\*\*

\* 载入配置文件

\*/

private function loadConfig(){

//使用全局变量保存配置

$GLOBALS['config'] = require './application/config/app.conf.php';

}

/\*\*

\* 获取请求参数，p=平台 c=控制器 a=方法

\*/

private function getRequestParams(){

//当前平台

define('PLATFORM', isset($\_GET['p']) ? $\_GET['p'] : $GLOBALS['config']['app']['default\_platform']);

//得到当前控制器名

define('CONTROLLER', isset($\_GET['c']) ? $\_GET['c'] : $GLOBALS['config'][PLATFORM]['default\_controller']);

//当前方法名

define('ACTION', isset($\_GET['a']) ? $\_GET['a'] : $GLOBALS['config'][PLATFORM]['default\_action']);

}

/\*\*

\* 请求分发

\*/

private function dispatch(){

//实例化控制器

$controller\_name = CONTROLLER.'Controller';

$controller = new $controller\_name;

//调用当前方法

$action\_name = ACTION . 'Action';

$controller->$action\_name();

}

}

**三、实验总结**

1、完成了实验要求内容

2、掌握MVC设计模式的目录划分与文件组织结构。

3、掌握了MVC设计模式的基础模型类、基础框架类的定义