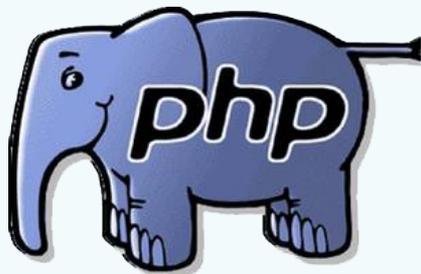
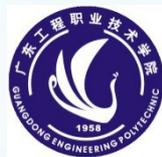


# 项目 13 ThinkPHP 框架 专业列表页面制作





## 任务一

创建专业列表

## 任务二

添加分页功能



# 案例展示

## 专业信息列表

The screenshot displays a web browser window with the URL `st/student/index.php/Admin/Major/showList.html`. The page title is "学生管理系统" (Student Management System) and it shows a user logged in as "admin". The breadcrumb path is "院系专业 > 专业列表". A sidebar on the left contains navigation menus for "院系专业", "学生管理", "系统设置", and "教学系统". The main content area shows a table of majors.

专业	班级	操作
应用物理学	20170101	编辑 删除
	20170102	编辑 删除

学生管理系统 本项目仅供学习使用

0.3134s

# 1、实例化模型

- 开发手册：【模型】 - 【模型实例化】
- ThinkPHP 中实例化模型有三种方式：

方法	示例
D 方法	<code>\$model=D('User');</code>
M 方法	<code>\$model=M('User');</code>
直接实例化	<code>\$model=new \Home\Model\UserModel();</code>



## ( 1 ) D 方法实例化

- D 方法的作用就是实例化一个模型类对象，该方法只有一个参数，参数值就是模型的名称。
- D 方法也可以不带参数直接使用，当不传递任何参数进行实例化时，得到的是 ThinkPHP 提供的基础模型类 `\Think\Model` 的实例。
- 当传递了模型名，而该模型类又存在的时候，实例化得到的就是这个模型类的实例。



## ( 2 ) M 方法实例化

- M 方法与 D 方法用法一样，所不同的是，M 方法不论是否有参数，实例化的都是 ThinkPHP 框架提供的基础模型类 `\Think\Model` 的实例，实际上 D 方法在没有找到定义模型类时，也会自动调用 M 方法。
- 因此在不涉及到自定义模型操作的时候，建议使用 M 方法而不使用 D 方法。



### ( 3 ) 直接实例化

- 顾名思义，直接实例化就是和实例化其他类库文件一样实例化模型类，例如：

```
// 实例化 Home 模块下的 Goods 模型类
$Goods = new \Home\Model\GoodsModel();

// 实例化 Admin 模块下的 User 模型类
$User = new \Admin\Model\UserModel();
```

- 这样就  
表。

象操作指定的数据



## 2、读取数据

- 读取数据表数据是项目中最常用的数据操作，在 ThinkPHP 中，find、select 以及 getField 操作以用于读取数据。
  - ( 1 ) find 操作
- find 操作会在 SQL 语句最后添加一个限定条件 “LIMIT 1”，表示仅取出一条数据，并且这条数据以**一维数组**的形式返回。



## 2、读取数据

### ( 2 ) select 操作

- **select** 操作与 **find** 操作的区别就在于 **select** 操作生成的 SQL 语句中没有 **LIMIT** 语句，并且数据是以**二维数组**的形式返回，因此 **select** 操作能够获取多条数据。

### ( 3 ) getField 操作

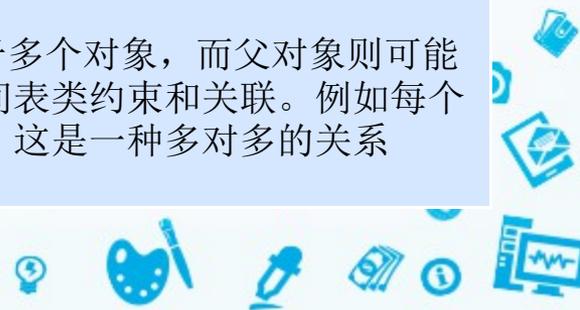
- **getField** 操作是从数据表中读取指定字段，并以字符串的形式返回。读取字段值其实就是获取数据表中的某个列的多个或者单个数据。



### 3、关联模型

- 开发手册：【模型】 - 【关联模型】
- 关联模型可以很轻松的完成数据表的关联 **CURD** 操作，目前支持的关联关系有四种，如下表所示：

关联关系	说明
HAS_ONE	HAS_ONE 关联表示当前模型拥有一个子对象，例如每个员工都有一个人事档案，这是一种一对一的关系
BELONGS_TO	BELONGS_TO 关联表示当前模型从属于另外一个父对象，例如每个用户都属于一个部门，这也是一种一对一的关系
HAS_MANY	HAS_MANY 关联表示当前模型拥有多个子对象，例如每个用户有多篇文章，这是一种一对多的关系
MANY_TO_MANY	MANY_TO_MANY 关联表示当前模型可以属于多个对象，而父对象则可能包含有多个子对象，通常两者之间需要一个中间表类约束和关联。例如每个用户可以属于多个组，每个组可以有多个用户，这是一种多对多的关系



### 3、关联模型

- 由于性能问题，新版取消了自动关联查询机制，而统一使用 **relation** 方法进行关联操作，**relation** 方法不但可以启用关联还可以控制局部关联操作。



## 4、乱码解决

- 在入口文件添加以下代码：
- `header('Content-type:text/html;charset=utf-8');`



## 5、ThinkPHP 模板标签

- 开发手册：【模板】 - 【内置标签】

### ( 1 ) <notempty>

- **notempty** 标签用来判断模板变量是否为空值，只有当变量非空时，才执行 **<notempty>** 中的代码。相当于 PHP 中的 “**!empty()**”。**notempty** 标签格式如下：

```
<notempty name="username">
    username 不为空
<else />
    username 为空
</notempty>
```

号。与



## 5、ThinkPHP 模板标签

### ( 2 ) <foreach>

- **foreach** 标签通常用于查询数据集（**select** 方法）的结果输出，通常模型的 **select** 方法返回的结果是一个二维数组，可以直接使用 **foreach** 标签进行输出。

name表示数据源 item表示循环变量。

可以输出索引，如下：

```
1. <foreach name="list" item="vo" >
2.   {$key}|{$vo.id}:{$vo.name}
3. </foreach>
```

也可以定义索引的变量名

```
1. <foreach name="list" item="vo" key="k" >
2.   {$k}|{$vo.id}:{$vo.name}
3. </foreach>
```



## 5、ThinkPHP 模板标签

### (3) <if>

- if 标签用来在视图文件中替代 PHP 中的 if 判断语句，语法格式如下：

```
<if condition="$num eq 100">
{$name} 等于 100
<else />
```

- {\$name} 不等于 100
- ```
</if>
```

法，因为会混淆模板解析。



生的表  
不支持

```
if($name == 100){
    echo "{$name} 等于 100";
}else{
    echo "{$name} 不等于 100";
}
```



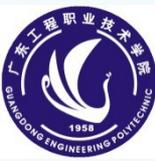
## 6、模板中使用函数

- 开发手册：【模板】 - 【使用函数】
- 函数参数只有一个变量
  - `{$data.name|md5} = <?php echo (md5($data['name'])); ?>`
- 函数参数有多个，且变量不是第一个传参，需要用 `###` 标识变量位置
  - `{$create_time|date="y-m-d",###} = <?php echo (date("y-m-d",$create_time)); ?>`
- 如果前面输出的变量在后面定义的函数的第一个参数，则可以直接使用
  - `{$data.name|substr=0,3} = <?php echo (substr($data['name'],0,3)); ?>`



# MajorModel

工于建构 成于创造

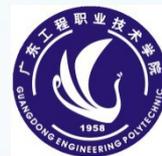


```
<?php
namespace Admin\Model;
use Think\Model\RelationModel;
class MajorModel extends RelationModel{
    protected $_link = array(
        'Class' => self::HAS_MANY,|
    );
}
```



# MajorController

工于建构 成于创造

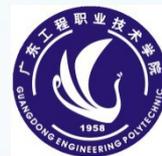


```
<?php
namespace Admin\Controller;
use Think\Controller;
class MajorController extends Controller{
    public function showList(){
        $major_info=D('Major')->relation(true)->select();
        // var_dump($major_info);
        // dump($major_info[0]['Class']);die;
        $this -> assign('major_info',$major_info);
        $this -> display();
    }
}
```



# showList.html

工于建构 成于创造

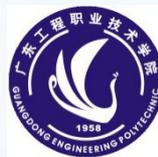


```
1 <include file='Index/header' title='专业列表' />
2
3 <h2 class="main-right-nav">院系专业 &gt; 专业列表</h2>
4 <div class="main-right-titbox">
5     <ul><li><a href="#">专业列表</a></li></ul>
6 </div>
7 <table class="table">
8     <tr><th>专业</th><th>班级</th><th>操作</th></tr>
9     <notempty name='major_info'>
10        <foreach name='major_info' item='v'>
11            <foreach name='v.Class' item='vv'>
12                <tr align="center">
13                    <if condition='$key eq 0'>
14                        <td rowspan="{ $v.Class|count}" class="table-major">{$v.major_name}</td>
15                    </if>
16                    <td width="40%"><a href="#">{$vv.class_name}</a></td>
17                    <td><div align="center">编辑 删除</div></td>
18                </tr>
19            </foreach>
20        </foreach>
21    <else />
22        <tr><td colspan="3">您所查询的记录不在</td>
23    </notempty>
24 </table>
25 <include file='Index/foote' />
```



## 7、使用公共类，防止不经过登录 直接浏览系统内页

工于建构 成于创造



### • **CommonController 类（初始化操作）**

```
namespace Admin\Controller;
use Think\Controller;
class CommonController extends Controller{
    public function _initialize(){
        if(!session('?admin_name')){
            $this -> error('非法用户，请登录',U('Index/login'));
            return;
        }
    }
}
```

### • **MajorController 类**

```
namespace Admin\Controller;
class MajorController extends CommonController{
    public function showList(){
        $major_info=D('Major')->relation(true)->select();
        $this -> assign('major_info',$major_info);
        $this -> display();
    }
}
```



## 8、空操作

- 空操作是指系统在找不到请求的操作方法的时候，会定位到空操作（`_empty`）方法来执行，利用这个机制，我们可以实现错误页面和一些URL的优化。
- 例如：“添加专业”功能页面在开发完成前，可采取以下代码替代：

```
public function _empty(){  
    $this -> assign('waitSecond',1);//设置等待时间  
    $this -> error('开发中...');  
    return;  
}
```



# 案例展示

## 专业信息列表

院系专业 > 专业列表

专业列表

专业	班级	操作
应用物理学	20170101	编辑 删除
	20170102	编辑 删除

12 下一页

专业列表

专业	班级	操作
电子信息科学与技术	20180101	编辑 删除
	20180102	编辑 删除

上一页 12

# 1、数据分页步骤

## 开发手册：【专题】 - 【数据分页】

### 第一种：利用Page类和limit方法

```
1. $User = M('User'); // 实例化User对象
2. $count      = $User->where('status=1')->count();// 查询满足要求的总记录数
3. $Page       = new \Think\Page($count,25);// 实例化分页类 传入总记录数和每页显示的记录数(25)
4. $show       = $Page->show();// 分页显示输出
5. // 进行分页数据查询 注意limit方法的参数要使用Page类的属性
6. $list = $User->where('status=1')->order('create_time')->limit($Page->firstRow.','.$Page->listRows)->select();
7. $this->assign('list',$list);// 赋值数据集
8. $this->assign('page',$show);// 赋值分页输出
9. $this->display(); // 输出模板
```



## 2、将数据分页功能放置在模型中

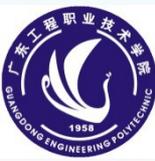
```
<?php
namespace Admin\Model;
use Think\Model\RelationModel;
class MajorModel extends RelationModel{
    protected $_link=array(
        'Class' => self::HAS_MANY,//定义Major表与Class表之间是一对多关系
    );

    public function getList(){
        $count=$this->relation(true)->count();// 查询满足要求的总记录数
        $Page = new \Think\Page($count,1);// 实例化分页类 传入总记录数和每页显示的记录数(1)
        $Page -> setConfig('prev','上一页');//设置分页显示格式
        $Page -> setConfig('next','下一页');//设置分页显示格式
        $show = $Page -> show();// 分页显示输出
        // 进行分页数据查询 注意limit方法的参数要使用Page类的属性
        $data = $this->relation(true)->limit($Page->firstRow.','.$Page->listRows)->select();
        //返回数组
        return array(
            'show' => $show,
            'data' => $data,
        );
    }
}
```



### 3、控制器调用相关模型生成数据并选择模板输出

工于建构 成于创造



```
<?php
namespace Admin\Controller;
class MajorController extends CommonController{
    public function showList(){
        $model = D('Major');//实例化Major模型
        $major_info = $model -> getList();//调用Major模型类的getList方法，获取分页显示与数据集
        $this -> assign('page',$major_info['show']);
        $this -> assign('major_info',$major_info['data']);
        $this -> display();
    }

    public function _empty(){
        $this -> assign('waitSecond',1);//设置等待时间
        $this -> error('开发中...');
        return;
    }
}
```



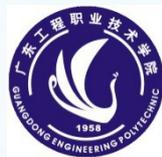
## 4、思考

- `$Page = new \Think\Page($count,1);`
- 以上代码中的 1 可通过配置文件方式设置，再用 `C()` 获取



# 【项目总结】

工于建构 成于创造



1. 实例化模型有 3 种方法，重点掌握 D 方法与 M 方法的区分，建议大家认真体会，仔细做题。
2. 连贯操作读取数据库的 3 种方式也是各有千秋，每个命令有它自己使用的场合，一定要区分使用的场合呢！
3. 使用关联模型时，可以很轻松完成数据表的关联 CURD 操作，这是我们的重点内容，它可支持的关联关系有 4 种，大家可以通过开发手册进行练习。另外提醒大家熟能生巧，多多练习才能提高代码编写效率哦！

