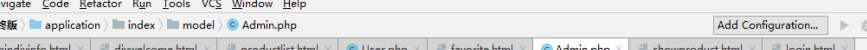


项目改进：管理员模块之补充代码

编写 index 模块下的模型类 Admin 的代码文件 Admin.php

☺ application\index\model\Admin.php



The screenshot shows the PhpStorm IDE interface. The title bar indicates the project is 'thinkphp5pet' and the file being edited is 'application/index/model/Admin.php'. The menu bar includes File, Edit, View, Navigate, Code, Refactor, Run, Tools, VCS, Window, and Help. The breadcrumb navigation shows the path: application > index > model > Admin.php. The code editor displays the following PHP code:

```

1 <?php
2 namespace app\index\model;
3 use think\Model;
4 class Admin extends Model
5 {
6     protected static $table = 'admin';
7 }

```

The code is syntactically highlighted, and the IDE interface includes a sidebar on the left for the Project view and a toolbar on the right for navigation and search.

修改 index 模块下的“用户登录”模板文件 login.html

☺ 主要是增加了一个复选框

☺ application\index\view\user\login.html

```
30 <div class="category">
31 <a href="{:url('index/index?cid=...
32 </div>
33 {/volist}
34 </div>
35 <div id="content">
36 <form action="{:url('index/user/dologin')}" method="post">
37 <table>
38 <tr>
39 <td class="ttd"><label for="username">用户名</label></td>
40 <td><input type="text" name="username" id="username" required></td>
41 </tr>
42 <tr>
43 <td class="ttd"><label for="password">密码</label></td>
44 <td><input type="password" name="password" id="password" required></td>
45 </tr>
46 <tr>
47 <td class="ttd">用户类型</td>
48 <td>
49 <input type="checkbox" name="admin" value="admin" checked="checked"/>管理员
50 </td>
51 </tr>
52 <tr>
53 <td colspan="2" align="center">
54 <button>登录</button>
55 <button type="reset">重置</button>
56 </td>
57 </tr>
58 </table>
59 </form>
60 <p class="pl">
61 <a href="{:url('index/user/register')}">没有账号? 点击注册</a>
62 </p>
63 </div>
64 </div>
65 </body>
66 </html>
```

编写 admin 模块下的 Index 控制器文件 Index.php

☺ application\admin\controller\Index.php

```
thinkphp5pet最终版 [D:\ZHANGLI\改进\thinkphp5pet最终版] - ...\application\admin\controller\Index.php [thinkphp5] - PhpStorm (Administrator)
File Edit View Navigate Code Refactor Run Tools VCS Window Help
thinkphp5pet最终版 application admin controller Index.php
Add Configuration...
Project
thinkphp5pet最终版 (thinkphp5)
  application
    admin
      controller
        Index.php
      model
      view
        index
          addproduct.html
          adduser.html
          admininfo.html
          editproduct.html
          edituser.html
          prodlist.html
          userlist.html
    extra
    index
      controller
        Index.php
        User.php
      model
        Admin.php
        User.php
      view
        index
          cart.html
          checkout.html
          divwelcome.html
          favorite.html
          productlist.html
          showproduct.html
      user
        editinfo.html
        login.html
        register.html
    .htaccess
    command.php
Structure
2 Favorites
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
<?php
namespace app\admin\controller;
use app\index\model\User as UserModel;
use think\Controller;
use think\Db;

class Index extends Controller
{
    private function checkLogin()
    {
        if (!session( name: 'user.userId' ))
        {
            $this->error( msg: '请登录', url: 'index/User/login' );
        }
        if (session( name: 'user.usertype' ) != "admin")
        {
            $this->error( msg: '请用管理员方式登录', url: 'index/User/login' );
        }
    }

    public function index()
    {
        $count = Db::table( table: 'account' )->count();
        $this->assign( name: 'count', $count );
        $list = Db::table( table: 'account' )
            ->order( field: 'username' )
            ->paginate( listRows: 5 );
        $this->assign( name: 'list', $list );
        // return $this->fetch( "userlist" );
        return $this->fetch( template: "index" );
    }
}
```

```
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
public function addproduct()
{
    $this->checkLogin();
    $catlist = Db::table( table: 'category' )->select();
    $this->assign( name: 'catlist', $catlist );
    return $this->fetch();
}

public function doaddproduct()
{
    $prname = input( key: 'post.proname' );
    $category = input( key: 'post.category' );
    $price = input( key: 'post.price' );
    $des = input( key: 'post.des' );
    $file = request()->file( name: 'proimage' );
    $info = $file->validate( ['ext'=>'jpg,png,gif'] )
        ->move( path: ROOT_PATH . 'public' . DS . 'static' . DS . 'proimages', savename: '' );
    if ($info) {
        // echo $info->getExtension();
        // echo $info->getSaveName();
        // echo $info->getFilename();
        $imagefile=$info->getFilename();
    }
    else{
        // echo $file->getError();
        $imagefile="";
    }
    // INSERT product(categoryid,productname,descn,image,unitprice)
    // VALUES(1,'烤鸭','动静皆宜','proimages/birds/icon-pelican.gif','86.00');
    $addsql=' INSERT product(categoryid,productname,descn,image,unitprice) ' ;
    $addsql.=' VALUES(:categoryid,:productname,:descn,:image,:unitprice)';
}
```

```

60     $res=Db::execute($addsSql,
61         ['categoryid' => $category,
62          'productname' => $prname,
63          'descn' => $des,
64          'image' => "prodimages/".$imagefile,
65          'unitprice' => $price]);
66     if ($res==1)
67     {
68         $this->success( msg: "新增宠物成功!", url: 'Index/prodlist');
69     }
70     else
71     {
72         $this->error( msg: "新增失败!");
73     }
74     public function prodlist()
75     {
76         // $catlist = Db::table("category")->select();
77         // $this->assign('catlist', $catlist);
78         $this->checkLogin();
79         $count = Db::table( table: 'product' )->count();
80         $this->assign( name: 'count', $count);
81         $list = Db::view( join: 'category', field: 'categoryid,name' )
82             ->view( join: 'product', ['productname','image','unitprice','productid'],
83                 on: 'category.categoryid = product.categoryid' )
84             ->order( field: 'productid' )
85             ->paginate( listRows: 5);
86         $this->assign( name: 'list',$list);
87         return $this->fetch( template: "prodlist");
88     }
89     public function delproduct($pid)
90     {
91         $this->checkLogin();
92         Db::execute( sql: 'DELETE FROM product WHERE productid=:productid',
93             [ 'productid' => $pid]);
94         return $this->prodlist();
95     }
96     public function editproduct($pid)
97     {
98         $this->checkLogin();
99         $catlist = Db::table( table: "category")->select();
100         $this->assign( name: 'catlist', $catlist);
101         $prod = Db::table( table: 'product' )
102             ->where( field: 'productid', $pid)
103             ->find();
104         $this->assign( name: 'proditem', $prod);
105         return $this->fetch();
106     }
107     public function doeditproduct()
108     {
109         $pid = input( key: 'post.pid');
110         $prname = input( key: 'post.prname');
111         $category = input( key: 'post.category');
112         $price = input( key: 'post.price');
113         $des = input( key: 'post.des');
114         $file = request()->file( name: 'proimage');
115         if (!empty($file))
116         {
117             $info = $file->validate(['ext'=>'jpg,png,gif'])
118                 ->move( path: ROOT_PATH . 'public' . DS . 'static' . DS . 'prodimages', savename: '');
119             if($info)
120             {
121                 $imagefile=$info->getFilename();
122             }
123             else
124             {
125

```

```

119         $imagefile="";
120     }
121     else
122     {
123         $imagefile="";
124     }
125     //
126     //
127     $editsql="UPDATE product SET categoryid=:categoryid,productname=:productname,";
128     if ($imagefile!="")
129     {
130         $editsql.="descn=:descn,unitprice=:unitprice ";
131     }
132     else
133     {
134         $editsql.="descn=:descn,image=:image,unitprice=:unitprice ";
135     }
136     $editsql.="WHERE productid=:productid ";
137
138     if ($imagefile!="")
139     {
140         $res=Db::execute($editsql,
141             ['categoryid' => $category,
142              'productname' => $proname,
143              'descn' => $des,
144              'productid' => $pid,
145              'unitprice' => $price]);
146     }
147     else
148     {
149         $res=Db::execute($editsql,
150             ['categoryid' => $category,
151              'productname' => $proname,
152              'descn' => $des,
153              'image' => "prodimages/".$imagefile,
154              'productid' => $pid,
155              'unitprice' => $price]);
156     }
157     if ($res==1)
158     {
159         $this->success( msg: "修改宠物成功!", url: 'prodlist');
160     }
161     else
162     {
163         $this->error( msg: "修改失败!");
164     }
165 }
166
167 public function userlist()
168 {
169     $this->checkLogin();
170     $count = Db::table( table: 'account')->count();
171     $this->assign( name: 'count', $count);
172     $list = Db::table( table: 'account')
173         ->order( field: 'username')
174         ->paginate( listRows: 5);
175     $this->assign( name: 'list',$list);
176     return $this->fetch( template: "userlist");
177 }
178
179 public function deluser($uid)
180 {
181     $this->checkLogin();
182     Db::execute( sql: 'DELETE FROM account WHERE username=:username',
183         [ 'username' => $uid]);
184     return $this->userlist();
185 }
186
187 public function adduser()
188 {
189     $this->checkLogin();
190     return $this->fetch();
191 }
192
193 public function doadduser()
194 {
195

```

```
179 $username = input( key: 'post.username');
180 $truename = input( key: 'post.truename');
181 $sex = input( key: 'post.sex');
182 if ($sex=="male")
183     $sex="男";
184 else
185     $sex="女";
186 $email = "用户未填";
187 $address = "用户未填";
188 $phone = "用户未填";
189 if (empty($username))
190 {
191     $this->error( msg: '用户名不能为空');
192 }
193 //检测用户是否已注册
194 $user = UserModel::getByUsername($username);
195 if (!empty($user))
196 {
197     $this->error( msg: '用户名已存在');
198 }
199 $data = array(
200     'username' => $username,
201     'password' => md5($username),
202     'truename' => $truename,
203     'sex' => $sex,
204     'email' => $email,
205     'address' => $address,
206     'phone' => $phone
207 );
```

```
208 if ($result = UserModel::create($data)) {
209     $this->success( msg: '新增用户成功, 密码等于用户名', url: 'userlist');
210 }
211 else {
212     $this->error( msg: '新增用户失败');
213 }
214 }
215 }
216 public function edituser($uid)
217 {
218     $this->checkLogin();
219     $user = Db::table( table: 'account')
220         ->where( field: 'username', $uid)
221         ->find();
222     $this->assign( name: 'useritem', $user);
223     return $this->fetch();
224 }
225 public function doedituser()
226 {
227     $username = input( key: 'post.username');
228     $truename = input( key: 'post.truename');
229     $sex = input( key: 'post.sex');
230     if ($sex=="male")
231         $sex="男";
232     else
233         $sex="女";
234     $email = input( key: 'post.email');
235     $address = input( key: 'post.address');
236     $phone = input( key: 'post.phone');
237 }
```

```
238 $data = array(
239     'username' => $username,
240     'truename' => $truename,
241     'sex' => $sex,
242     'email' => $email,
243     'address' => $address,
244     'phone' => $phone
245 );
246 $editsql='UPDATE account SET truename=:truename,sex=:sex,';
247 $editsql.='email=:email,address=:address,phone=:phone ' ;
248 $editsql.='WHERE username=:username ' ;
249 $res=Db::execute($editsql,$data);
250 // if ($res==1){
251 $this->success( msg: '修改用户成功', url: 'userlist');
252 }
253 // else {
254 //     $this->error('修改用户失败');
255 // }
256 }
257 public function resetpass($uid)
258 {
259     $editsql='UPDATE account SET password=:pass';
260     $editsql.=' WHERE username=:username ' ;
261     $res=Db::execute($editsql,['username' => $uid,'pass' => $uid]);
262     $this->success( msg: '密码已经重置成功, 新密码等于用户名!', url: 'userlist');
263 }
264 }
```