

第 30 章 新闻发布系统实现

该章节将实现新闻模板。第二十八章对新闻发布系统进行了总体功能和数据设计，新闻发布系统总体分成用户浏览、新闻图片模块和新闻发布模块三个部分。其中新闻图片管理模块已经在第二十九章中实现；本章将具体介绍新闻发布模块的创建过程。新闻发布模块实现对一般新闻和图片新闻的发布操作，以及对各项新闻的统一修改和管理操作。

本章要点包括以下内容：

- ☐ 新闻模块需要创建的文件及逻辑关系分析
- ☐ 实现新闻分类的添加和管理功能
- ☐ 实现一般新闻和图片新闻的发布功能
- ☐ 实现新闻的统一管理功能
- ☐ 创建相应的 `JavaBean` 类文件

30.1 需要创建的文件

30.1.1 添加新闻分类版块JSP页面

需要创建的文件包括如下两个部分：

- ☐ `addnewsboard.jsp` 页面：对各个新闻进行分类，该页面就是为新闻添加一个新的新闻分类版块，编辑完版块信息之后，单击“添加”按钮进行提交。
- ☐ `addNewsBoardok.jsp` 页面：进行相应的新闻版块操作，即调用 `Newsboardfactory` 类中的 `Add(NewsBoard newsboard)`方法在数据库表 `newsboard` 中插入一个新闻版块信息。

30.1.2 新闻分类的管理JSP页面

需要创建的新闻分类管理 JSP 页面如下：

- ☐ `editnewsboard.jsp` 页面：修改某个新闻分类版块。
- ☐ `editnewsboardok.jsp` 页面：对新闻版块修改进行实质性的操作，即调用 `NewsFactory` 类中的 `Edit(NewsBoard newsboard)`方法进行数据库表 `newsboard` 中的版块信息进行更改。
- ☐ `delnewsboard.jsp` 页面：对某个指定的新闻分类版块进行删除操作。

30.1.3 新闻发布JSP页面

需要创建的新闻分布 JSP 页面文件如下：

- ☐ `publicnews.jsp` 页面：管理员通过该页面进行新闻的编辑。
- ☐ `publishnewsok.jsp` 页面：该页面程序进行新闻的发布操作，调用 `ReadTemplates.java` 类来读取新闻模板文件，然后将相应的新闻信息插入，再调用 `WriteHtml.java` 类程序把新的内容写入到指

定的静态页面文件中。

- ❑ editpaper.jsp 页面：该页面程序提供了新闻文本的编辑功能，例如 Word 中的复制、粘贴以及剪切等编辑功能。
- ❑ editpaper.js 程序：该 JavaScript 程序中实现了各种编辑功能。需要在 editpaper.jsp 页面中使用 `<jsp:include>` 标签来引入该 JavaScript 文件。
- ❑ publishphotonews.jsp 页面：与 publishnews.jsp 页面程序相类似，也是用来进行新闻发布的，只是该页面专门用于图片新闻的信息发布，即在前台页面中提供的是以图片为链接的新闻。
- ❑ publishphotonewsok.jsp 页面：进行图片新闻发布的操作。

30.1.4 新闻管理JSP页面

需要创建的新闻管理 JSP 页面文件如下：

- ❑ newsmanagement.jsp 页面：该页面对所有新闻信息进行统一管理，管理员可以查看所有新闻的基本信息以及进行删除、移动、重发和新闻头条、热点信息的修改等操作。
- ❑ Changetophot.jsp 页面：对某新闻的头条和热点信息进行实质性的修改。
- ❑ Movenews.jsp 页面：在该页面中选择需要修改的新闻类型和所在的新闻版块，然后单击“修改”按钮进行修改信息的提交。
- ❑ movenewsok.jsp 页面：该页面对某条新闻的类型和所在版块信息进行实质性的修改。
- ❑ Republishnews.jsp 页面：对某条新闻进行重发操作，这是通过修改该新闻的发布日期实现的，即把以前发布日期修改为当前日期。
- ❑ delnews.jsp 页面：删除某条新闻信息。

30.1.5 新闻模板文件

创建的新闻模板文件名为 `qqnews.template`。对发布的新闻需要定制一个模板，然后在这模板文件中插入不同的新闻内容来生成不同的新闻信息。在模板文件中使用特定字符串来表示不同参数的插入位置，例如 `<$title$>` 表示该处需要插入新闻标题 `title` 参数值。整个新闻的发布过程：首先调用 `ReadTemplates.java` 类中的 `readTemplateContent()` 方法来读取新闻模板文件 `qqnews.template`，然后调用 `ReplaceAll.java` 类中的 `replace()` 方法将相应参数值插入到模板文件的相应位置，再调用 `WriteHtml` 类中的 `save()` 方法将这生成的新闻内容写入到一个静态文件（该静态文件的后缀为 `.shtml`）中。

30.1.6 JavaBean类文件

需要创建的相应 JavaBean 类文件如下：

- ❑ NewsBoard.java 接口类：该类定义新闻版块属性相对应的 `set` 和 `get` 方法。
- ❑ SqlNewsBoard.java 类：该类定义了与新闻版块相关的属性变量，并实现了 `NewsBoard` 接口类中所定义的所有 `set` 和 `get` 方法。
- ❑ NewsFactory.java 抽象类：该类定义了多个与新闻版块相关的数据操作类。
- ❑ SqlNewsFactory.java 类：该类继承抽象类 `NewsFactory`，并实现抽象类中已经定义的所有方法。
- ❑ News.java 接口类：定义新闻属性相对应的 `set` 和 `get` 方法。
- ❑ SqlNews.java 类：该类定义了与新闻相关的属性变量，并实现了 `News` 接口类中所定义的所有

set 和 get 方法。

- ❑ NewsFactory.java 抽象类：该类定义了所有与新闻相关的数据库操作方法。
- ❑ SqlNewsFactory.java 类：继承 NewsFactory 抽象类，并实现抽象类中定义的所有方法。
- ❑ ReadTemplates.java 类：读取文件的类，专门用于读取新闻模板文件中的内容，并把其转换为字符串类型返回。
- ❑ ReplaceAll.java 类：该类用于将新闻各参数值插入到新闻模板文件的相应位置。
- ❑ WriteHtml.java 类：该类专门将新插入参数的模板文件内容重新写入到一个静态页面文件中。

以上简单地介绍了需要创建的各文件。下面通过图来显示这些创建文件之间的功能流程以及相互之间的关系。逻辑流程如下面的图 30.1 所示。

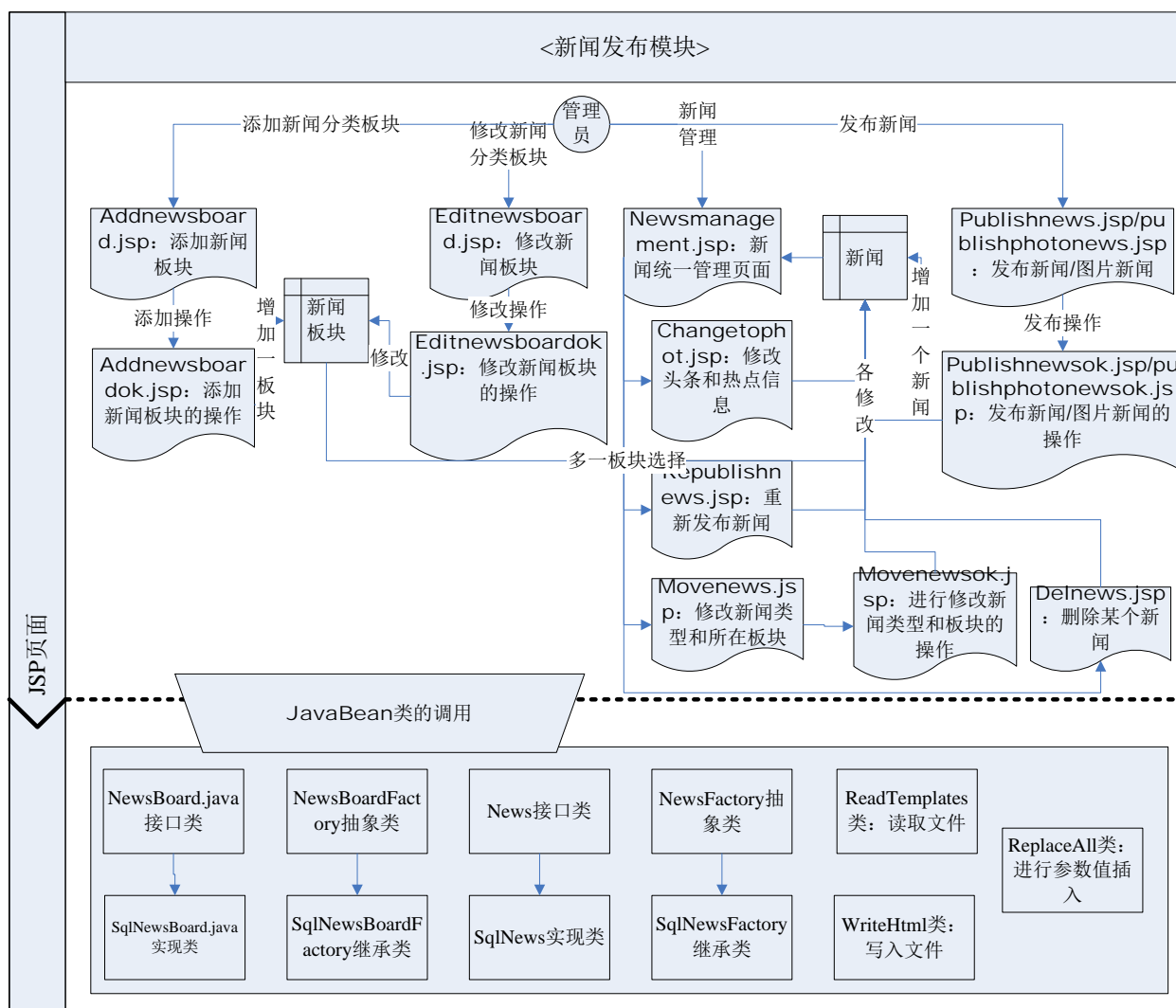


图 30.1 页面逻辑流程和关系图

30.2 实现新闻分类的添加功能

创建添加新闻分类的 JSP 页面。

30.2.1 添加新闻分类的页面addnewsboard.jsp

该页面用于添加新闻分类版块，详细代码如下：

```
<%@ page import="cn.com.zzb.news.*"%>
<%@ page import="java.util.Iterator"%>
<%@ page import="cn.com.zzb.util.*"%>
<%@ page import="java.text.*"%>
<html>
<head><title>添加新闻版块</title></head>
<link rel="stylesheet" href="admin.css" type="text/css">
<style type="text/css">
<!--
body {
background-color:#6699CC;
vertical-align:top
}
td {
color:#FFFFFF;
line-height:20px;
letter-spacing:3px;
font-family:"宋体";
}
.TopHeader{
background-color:#FFCC33;
text-align:center
}
-->
</style>
<script language="javascript">
<!-- //初步检验输入的版块信息
function dosubmit()
{
    if(document.form1.BoardName.value==""){
        alert("请填写版块名称");
        document.form1.BoardName.focus();
    }
    else if(document.form1.Introduce.value==""){
        alert("还是说上几句吧!");
        document.form1.Introduce.focus();
    }
    else{
        document.all.form1.submit();
    }
}
-->
</script>
<body>
<div align="center" style="font-weight:bold; font-size:18px; color:#0000A0">添加新闻版块</div>
<hr width="95%" color="#CCCCCC" size="1px">
```

程序说明:

(1) 在输入框中填写新闻分类版块信息之后, 单击“添加”进行表单的提交, 首先会由 JavaScript 中的 `dosubmit()` 方法来进行数据的初步检验, 如果检验正确则正式提交给 `addnewsboardok.jsp` 页面进行相应处理。

(2) 该页面还迭代输出了所有的新闻分类版块信息，管理员可以查看所有信息，并进行“修改”和“删除”操作。

30.2.2 处理新闻分类添加的页面addNewsBoardok.jsp

处理新闻分类版块的提交操作，把添加的新闻分类版块信息插入到数据库表 newsboard 中，该页面的代码如下：

```
<%@ page import="cn.com.zzb.news.*"%>
<%@ page import="java.util.Iterator"%>
<%@ page import="cn.com.zzb.util.*"%>
<%@ page import="java.text.*"%>
<html>
<head><title>添加新闻版块</title></head>
<link rel="stylesheet" href="admin.css" type="text/css">
<%
String BoardName = ParameterUtils.getString(request,"BoardName");           //取得传递的 BoardName 参数
String Introduce = ParameterUtils.getString(request,"Introduce");           //取得传递的 Introduce 参数
%>
<body>
<table width="100%" border="0" cellpadding="0" cellspacing="0" align="center">
  <tr>
    <td>
<%
Factory factory = Factory.getInstance();
NewsBoard newsboard = factory.getNewsBoard();                               //初始化 NewsBoard 实例对象
NewsBoardFactory newsboardfactory = factory.getNewsBoardFactory();           //初始化 NewsBoardFactory 实例
//进行数据的封装
newsboard.setName(BoardName);
newsboard.setIntroduce(Introduce);
if(newsboardfactory.Add(newsboard)==1)                                       //如果添加新的版块成功
  out.println("<div align='center'><font color='red' size=4>添加成功</font></div>");
else                                                                           //否则添加失败
  out.println("<div align='center'><font color='red' size=4>添加失败</font></div>");
%>
    </td>
  </tr>
  <tr>
    <td align="center" height="40"><a href="AddNewsBoard.jsp">继续添加</a></td>
  </tr>
</table>
</body>
</html>
```

程序说明：首先把上页面传递过来的 BoardName 和 Introduce 参数封装到 newsboard 对象中，然后调用 NewsBoardFactory 类中的 Add (NewsBoard newsboard) 方法把新闻版块添加到数据库中。

30.3 实现新闻分类的管理功能

对新闻分类的管理包括修改和删除新闻分类。

30.3.1 修改新闻分类的页面editnewsboard.jsp

指定一个新闻版块，进行该版块信息修改，该页面代码如下：

```
<%@ page import="cn.com.zzb.news.*"%>
<%@ page import="java.util.Iterator"%>
<%@ page import="cn.com.zzb.util.*"%>
<%@ page import="java.text.*"%>
<html>
<head>
<link rel="stylesheet" href="admin.css" type="text/css">
<title>修改新闻版块</title>
</head>
<script language="javascript">
<!-- //初步检查输入框中的数据是否合理
function dosubmit()
{
    if(document.form1.BoardName.value=="")                //如果输入的新闻版块为空
    { alert("请填写版块名称");
      document.form1.BoardName.focus();
    }
    else if(document.form1.Introduce.value=="")            //如果输入的版块介绍为空
    {
        alert("还是说上几句吧!");
        document.form1.Introduce.focus();
    }
    Else                                                    //否则，初步检验合理
    {document.all.form1.submit();}
}
-->
</script>
<%
Factory factory = Factory.getInstance();
int BoardID = ParameterUtils.getInt(request,"ID");        //获取传递的 BoardID 参数
NewsBoard newsboard = factory.getNewsBoard();              //初始化 NewsBoard 实例对象
NewsBoardFactory newsboardfactory = factory.getNewsBoardFactory(); //初始化 NewsFactory 实例对象
newsboard = newsboardfactory.ListByID(BoardID);            //通过 BoardID，取出该新闻版块对象
//进行数据的拆分
String BoardName=newsboard.getName();
String Introduce = newsboard.getIntroduce();
%>
<body>
<div align="center" style="font-weight:bold; font-size:18px; color:#0000A0">修改新闻版块</div>
<hr width="95%" color="#CCCCCC" size="1px">
<br>
```

```

<form name="form1" method="post" action="EditNewsBoardok.jsp?BoardID=<%=BoardID%>">
<table width="60%" border="1" cellspacing="0" cellpadding="0" align="center" bordercolor="#CCCCCC">
  <tr>
    <td width="30%">新闻版块名称: </td>
    <td>&nbsp;<input type="text" name="BoardName" size=40 value=<%=BoardName%></td>
  </tr>
  <tr>
    <td>简介: </td>
    <td>&nbsp;<textarea name="Introduce" rows="15" cols="45"><%=Introduce%></textarea></td>
  </tr>
  <tr>
    <td colspan="2" align="center">
      <input type="button" value="修改" onClick="javascript:dosubmit()"> |
      <input type="reset" value="重置">
    </td>
  </tr>
</table>
</form>
</body>
</html>

```

程序说明：该页面程序和添加新闻版块的页面相类似，修改完版块信息之后，单击“修改”进行提交，并交付给 editnewsboardok.jsp 页面进行相应处理。

30.3.2 处理新闻分类修改的页面editnewsboardok.jsp

调用该页面程序，进行新闻分类版块的信息更新，代码如下：

```

<%@ page import="cn.com.zzb.news.*"%>
<%@ page import="java.util.Iterator"%>
<%@ page import="cn.com.zzb.util.*"%>
<%@ page import="java.text.*"%>
<html>
<head>
<link rel="stylesheet" href="admin.css" type="text/css">
<title>修改新闻版块</title>
</head>
<%
String BoardName = ParameterUtils.getString(request,"BoardName");           //取得 BoardName 参数值
String Introduce = ParameterUtils.getString(request,"Introduce");           //取得 Introduce 参数值
int BoardID = ParameterUtils.getInt(request,"BoardID");                     //取得传递的 BoardID 参数
%>
<body>
<br>
<hr width="95%" color="#CCCCCC" size="1px">
<table width="100%" border="0" cellspacing="0" cellpadding="0" align="center">
  <tr>
    <td>
<%
Factory factory = Factory.getInstance();                                     //初始化 Factory 实例
NewsBoard newsboard = factory.getNewsBoard();                             //初始化 NewsBoard 实例对象

```



```
NewsBoardFactory newsboardfactory = factory.getNewsBoardFactory();           //初始化 NewsBoardFactory 实例
//新闻版块的数据封装
newsboard.setID(BoardID);
newsboard.setName(BoardName);
newsboard.setIntroduce(Introduce);
if(newsboardfactory.Edit(newsboard)==1)                                     //如果新闻版块的信息修改成功
    out.println("<div align='center'><font color='red' size=4>修改成功</font></div>");
else                                                                        //否则，修改不成功
    out.println("<div align='center'><font color='red' size=4>修改失败</font></div>");
%>
</td>
</tr>
<tr>
    <td align="center" height="40"><a href="AddNewsBoard.jsp">返回新闻版块添加首页</a></td>
</tr>
</table>
</body>
</html>
```

程序说明：首先把上页面传递的修改数据封装到 newsboard 对象中，然后调用 NewsBoardFactory 类中的 Edit(NewsBoard newsboard)方法来进行新闻分类版块的信息更新。

30.3.3 删除新闻分类的页面delnewsboard.jsp

在 `addnewsboard.jsp` 页面也提供了新闻分类板块的“删除”链接，单击“删除”链接将跳转到该页面进行删除操作，代码如下：

```
<%@ page import="cn.com.zzb.news.*"%>
<%@ page import="java.util.Iterator"%>
<%@ page import="cn.com.zzb.util.*"%>
<%@ page import="java.text.*"%>
<html>
<head>
<link rel="stylesheet" href="admin.css" type="text/css">
<title>删除新闻版块</title>
</head>
<%
int BoardID = ParameterUtils.getInt(request,"ID");
%>
<body>
<br>
<hr width="95%" color="#CCCCCC" size="1px">
<table width="100%" border="0" cellspacing="0" cellpadding="0" align="center">
  <tr>
    <td>
<%
Factory factory = Factory.getInstance(); //初始化 Factory 类
NewsBoard newsboard = factory.getNewsBoard(); //初始化 NewsBoard 实例对象
NewsBoardFactory newsboardfactory = factory.getNewsBoardFactory(); //初始化 NewsBoardFactory 实例对象
if(newsboardfactory.Del(BoardID)==1)
{
//进行数据库删除操作，如果删除成功
```

```

        out.println("<div align='center'><font color='red' size=4>删除成功</font></div>");
        response.sendRedirect("AddNewsBoard.jsp");
    }
    Else //否则，删除失败
        out.println("<div align='center'><font color='red' size=4>删除失败</font></div>");
    %>
    </td>
</tr>
</table>
</body>
</html>

```

程序说明：如果新闻分类版块删除成功，页面将重定向到 addnewsBoard.jsp 页面。

30.4 实现新闻发布功能

发布的新闻包括一般新闻和图片新闻，下面首先介绍如何实现一般新闻的发布。

30.4.1 新闻发布的页面publicnews.jsp

该页面用于一般新闻的发布，一般新闻在前台页面显示的是文字链接，该页面程序详细讲解如下：

```

<%@ page import="cn.com.zzb.news.*"%>
<%@ page import="java.util.Iterator"%>
<%@ page import="cn.com.zzb.util.*"%>
<%@ page import="java.text.*"%>
<html>
<head><title>发布新闻</title></head>
<link rel="stylesheet" href="admin.css" type="text/css">
<style type="text/css">
<!--
body {
background-color:#6699CC;
vertical-align:top
}
td {
color:#FFFFFF;
line-height:20px;
letter-spacing:3px;
font-family:"宋体";
}
.TopHeader{
background-color:#FFCC33;
text-align:center
}
-->
</style>
<script language="javascript">
<!--

```

```

function pic_preview() {                                     //打开预览的图片页面
    var picName=document.form2.pic.value;                   //获取到需要浏览的图片名称
    if(picName==null || picName=="")
        window.alert("没有选中图片，无法完成预览。");
    else
        window.open("picPreview.jsp?picName="+picName);     //给出地址，打开新的页面浏览该图片
}
function pic_insert() {                                     //把图片插入到文本编辑框中
    window.frames (IFrameObject.name).focus();             //将光标聚焦在编辑框中
                                                            //获取到编辑框上下文对象
    MyRangeObject = window.frames(IFrameObject.name).document.selection.createRange();
    MyRangeObject.pasteHTML(document.form2.pic.value);       //把图片复制到编辑框对象中
}
-->
</script>
<%
    Factory factory = Factory.getInstance();
    NewsBoard newsboard = factory.getNewsBoard();             //初始化 NewsBoard 实例对象
    NewsBoardFactory newsboardfactory = factory.getNewsBoardFactory(); //初始化 NewsBoardFactory 实例对象
    Pic pic = factory.getPic();                               //初始化 Pic 实例对象
    PicFactory picfactory = factory.getPicFactory();          //初始化 PicFactory 实例对象
%>
<body>
<br>
<div align="center" style="font-weight:bold; font-size:18px; color:#0000A0">发布新闻</div>
<hr width="95%" color="#CCCCCC" size="1px">
<br>
<table width="100%" border="0" cellspacing="0" cellpadding="0" align="center">
<tr>
<td>
        <jsp:include page="editpaper.jsp"/>                 <!--将带有文本编辑器代码的页面引入 -->
        <form name="form2" method="post" action="PublishNewsOk.jsp">
        <center>
            <table width="100%">
            <tr>
                <td width="10%" align="center" valign="middle">板 块</td>
                <td>
                    <select name="BoardID">
                    <%
                        Iterator iterator = newsboardfactory.List();           //取出所有的新闻分类版块的信息
                        while(iterator.hasNext())
                        {
                            newsboard = (NewsBoard)iterator.next();
                            out.println("<optionvalue="+newsboard.getID()+">" +newsboard.getName()+"</option>");
                        }
                    %>
                    </select></td>
            </tr>
            <tr>
                <td width="10%" align="center" valign="middle">标 题</td>

```

[illegible]

```

link=pic.getLink(),
align=pic.getAlign(),
valign=pic.getValign(),
fileName=pic.getFileName(),
//获得该图片文件的后缀类型
fileNameExt=fileName.substring(fileName.lastIndexOf(".")+1,fileName.length());
//判断该文件是否为一般的图片文件
if(!(fileNameExt.equals("jpg") || fileNameExt.equals("gif") || fileNameExt.equals("png") ||
        fileNameExt.equals("bmp")))

{
    picInfo.append("<a href='../pic/'+"+fileName+" target='_blank'>
    <img src='../pic/unknow.gif' width="+width+" height="+height+" border="+border+"></a>");
}

else
{
    if(type == 3 && !link.equals("") && link!=null)
        picInfo.append("<a href="+link+" target=_blank>");
    picInfo.append("<img src='../pic/'+"+fileName+"");
    if(width!=0)
        picInfo.append(" width="+width);
    if(height!=0)
        picInfo.append(" height="+height);
    picInfo.append(" border="+border);
    if(!align.equals(""))
        picInfo.append(" align="+align);
    picInfo.append(" >");
    if(type == 3 && !link.equals("") && link!=null)
        picInfo.append("</a>");
}
out.println("<option
value='"+picInfo+"\">"+type+"="+boardID+"="+pic.getPicName()+"("+dateString+")</option>");
}

%>
</select>
<input type=button name=preview value="预览" onClick="pic_preview()">
<input type=button name=insert value="插入" onClick="pic_insert()">
</td>
</tr>
</table>
<table width="100%">
<tr><td width="10%" align="center" valign="middle">内容</td>
<td><IFRAME width="98%" name=contentframe height="270" onfocus="SetEdit(this)"
marginWidth=1 marginHeight=1 scrolling=yes></IFRAME>
</td>
</tr>
</table>
<textarea name="Content" style="display:'none'" rows="1" cols="20"> </textarea>
<center>

```

[illegible]

程序说明:

(1) 在下拉框“pic”中已经列出了所有的图片选项。管理员可以选择其中一个图片，然后单击“插入”按钮即可调用 JavaScript 中 `pic_insert()` 方法把选择的图片插入到编辑框中。

(2) 在下拉框中选择一个图片选项，管理员也可以单击“浏览”按钮调用 JavaScript 中的 `pic_preview()`方法来查看该图片的效果。

(3) 该程序中使用了`<jsp:include>`动作元素来引入 `editpaper.jsp` 外在文件，该外部文件提供了文本编辑功能，详细见下面该文件的介绍。

(4) 在程序中,单击“Source”按钮将调用 JavaScript 中的 EditConCode()方法,使得编辑器中的文本内容以源代码格式显示,这个 JavaScript 的方法是定义在 editpaper.js 文件中,由于引入的 editpaper.jsp 文件中定义该 JavaScript 文件,所以在该页面程序中就不需要重复定义了。

(5) 管理员将新闻信息填写无误之后, 单击“提交”进行新闻的发布。

30.4.2 处理新闻发布的页面publishnewsok.jsp

该页面对发布的新闻信息进行处理，首先读取新闻模块文件 `qqnews.template`，然后把上页面传递过来的各个参数值插入到模块文件的指定位置，从而生成新闻的静态页面 `shtml`，并保存在一定的路径下，该页面代码详细如下：

```
<%@ page import="cn.com.zzb.news.*"%>
<%@ page import="java.util.Iterator"%>
<%@ page import="cn.com.zzb.util.*"%>
<%@ page import="java.text.*"%>

<html>
<head>
<link rel="stylesheet" href="admin.css" type="text/css">
<title>发布信息</title>
</head>
<%
Factory factory = Factory.getInstance();
int BoardID = ParameterUtils.getInt(request,"BoardID");           //获得传递版块号参数 BoardID
String Title = ParameterUtils.getString(request,"Title");         //获得传递标题参数 Title
String Keyword = ParameterUtils.getString(request,"Keyword");     //获得关键字参数 Keyword
String FirstFrom = ParameterUtils.getString(request,"FirstFrom"); //获取文章出处参数 FirstFrom
String Author = ParameterUtils.getString(request,"Author");       //获得作者参数 Author
String Editor = ParameterUtils.getString(request,"Editor");       //获得编者参数值 Editor

```

```

String Department = ParameterUtils.getString(request,"Department");    //获得编辑者所在部分参数 Department
String Content = ParameterUtils.getString(request,"Content");          //获取文章的内容参数 Content
String attributes[] =request.getParameterValues("attributes");          //关于文章是热点新闻还是头条新闻
%>
<body>
<br>
<hr width="95%" color="#CCCCCC" size="1px">
<table width="100%" border="0" cellspacing="0" cellpadding="0" align="center">
    <tr>
        <td>
            <%
String filePath = application.getRealPath("../template/qqnews.template"); //取得新闻发布的模板文件
String[] flag = {"<$title$>","<$date$>","<$author$>","<$from$>","<$content$>","<$editer$>","<$list1$>","
                "<$list2$>","<$list3$>","<$belongto$>"};

String templateContent;
try
{
/**
*读取操作，调用 ReadTemplates 类中的 getTlpContent（String path）方法来读模板文件内容
*然后把读取出来的文件内容保存在 templateContent 变量中
*/
    templateContent = ReadTemplates.getTlpContent(filePath);
}
catch(ReadTemplateException e){
    throw new Exception(e.getMessage());
}
//下面在模板文件中用 flag 数组变量所标识的位置处插入指定值
templateContent = ReplaceAll.replace(templateContent,flag[0],Title);          //插入文章标题
templateContent = ReplaceAll.replace(templateContent,flag[1],getDate.getStringDate()); //插入日期
templateContent = ReplaceAll.replace(templateContent,flag[2],Author);          //插入文章作者
templateContent = ReplaceAll.replace(templateContent,flag[3],FirstFrom);        //插入文章出处
templateContent = ReplaceAll.replace(templateContent,flag[4],Content);          //插入文章内容
templateContent = ReplaceAll.replace(templateContent,flag[5],Editor);          //插入编辑者=
templateContent = ReplaceAll.replace(templateContent,flag[6],"等待添加购物车"); //
templateContent = ReplaceAll.replace(templateContent,flag[7],"等待添加分类"); //
templateContent = ReplaceAll.replace(templateContent,flag[8],"等待添加时尚新闻"); //
templateContent = ReplaceAll.replace(templateContent,flag[9],Department);      //插入编辑者所在部门
// 根据时间生成静态 Shtml 的文件名与路径名
Calendar calendar = Calendar.getInstance();
String fileName = String.valueOf(calendar.getTimeInMillis()) + ".shtml";      //生成静态文件名
// 生成路径名，如果是 linux 系统把 "\\" 改为 "/"
String pathName = application.getRealPath("../news")+ "/" + calendar.get(Calendar.YEAR) +
                "/" + (calendar.get(Calendar.MONTH)+1) + "/" + calendar.get(Calendar.DAY_OF_MONTH)+"/";
/**
* 写文件操作，把 templateContent 内容写入到一个生成的静态 Shtml 文件中，然后保存在特定路径下。
*/
try {
    WriteHtml.save(templateContent,pathName,fileName);
}
catch(WriteFileException e){

```

```

        throw new Exception("新闻发布失败。可能是目录不具备 IO 操作权限。请与管理员联系。");
    }

    News news = factory.getNews();                //初始化 News 实例对象
    NewsFactory newsFactory = factory.getNewsFactory();    //初始化 NewsFactory 实例对象
                                                    //把新闻信息风转在 News 对象中

    news.setBoardID(BoardID);
    news.setEditor(Editor);
    news.setTitle(Title);
    news.setKeyword(Keyword);
    news.setAuthor(Author);
    news.setFirstFrom(FirstFrom);
    news.setDepartment(Department);
    if(attributes!=null){
        for(int i=0;i<attributes.length;i++){
            if(attributes[i].equals("1")){
                news.setIfTop(1);
            }
            if(attributes[i].equals("2")){
                news.setIfHot(1);
            }
        }
    }
}

                                                    //下面封装该新闻文件的路径信息
news.setNewsFilePath(calendar.get(Calendar.YEAR)+"-"+(calendar.get(Calendar.MONTH)+1)+"-"+
calendar.get(Calendar.DAY_OF_MONTH)+"-"+fileName);
news.setPublishDate(calendar.getTimeInMillis());
news.setContent("ddd");
news.setType(1);                                //这里发布的是一般新闻，所以设置为 1
news.setPicID(0);
int state=newsFactory.AddNews(news);             //把新闻相关的信息保存在数据库中
if(state==1)                                     //如果添加数据库成功
    out.println("<div align='center'><font size=3 color='red'>发布成功</font></div>");
else                                             //否则添加失败
    out.println("<div align='center'><font size=3 color='red'>失败</font></div>");
    %>
    </td>
</tr>
</table>
<br>
<center><br><br>
    <a href="PublishNews.jsp">继续发布</a><br>
</center>
</body>
</html>

```

程序说明：

(1) 新闻发布中使用了一个新闻模板文件 `qqnews.template`（见下面对该文件的详细介绍），首先调用 `ReadTemplates` 类（该类后面有讲解）中的 `getTlpContent(String filePath)` 方法（其中 `filePath` 为新闻模块文件的路径）来读取模板文件中的内容，并把读取出来的所有内容保存在 `templateContent` 变量中，

以方便后面的操作。

(2) 调用 `ParameterUtils` 类中的相应方法来获取到上页面传递过来的各新闻发布的参数，再调用 `ReplaceAll` 类中的 `replace()` 方法把相应的参数值插入到模板文件的指定位置，是通过 `flag[]` 数组来标识模块文件中的各个插入位置。

(3) 使用 `WriteHtml` 类中的 `save()` 方法把插入了新闻信息的 `templateContent` 内容写入到新生成的静态文件（后缀为 `.shtml`）中。在前台页面，就是通过调用该生成的静态页面文件来浏览该新闻内容的。

(4) 把发布的新闻各信息保存在数据库中，其中包括生成的静态页面文件的路径地址。

30.5 实现文本编辑器功能

使用过 word 编辑工具的读者应该对这一概念比较好理解。该实现的文本编辑器将实现基本的文本格式调整，包括字体大小和颜色的调整。

30.5.1 文本编辑页面 `editpaper.jsp`

下面文件提供了用于编辑文本的操作界面，类似 word 中的斜体、加粗和下划线等操作。该页面代码如下：

```
<script language=JavaScript src="editpaper.js"></script>
<!--编辑排版-->
<FIELDSET style="PADDING-RIGHT: 0px; BACKGROUND: buttonface; PADDING-LEFT: 0px;
MARGIN-BOTTOM: 3px; PADDING-BOTTOM: 0px; PADDING-TOP: 0px">
<table width="100%" height=30 border="0" cellpadding="0" cellspacing="0" bgColor=buttonface>
  <tr>
    <td width="4%">
      <div align="center">
        <IMG onmouseup="return But_onmouseup(this)" onmousedown="return But_onmousedown(this)"
onmouseover="return But_onmouseover(this)" onclick=EditFont(20)
onmouseout ="return But_onmouseout(this)" alt=打印
src="../../images/print.gif" width="20" height="20" >
      </div>
    </td>
    <td width="4%">
      <div align="center">
        <IMG onmouseup="return But_onmouseup(this)" onmousedown="return But_onmousedown(this)"
onmouseover="return But_onmouseover(this)" onclick=EditFont(16)
onmouseout ="return But_onmouseout(this)" alt=撤销
src="../../images/undo.gif" width="20" height="20" >
      </div>
    </td>
    <td width="4%">
      <div align="center">
        <IMG onmouseup="return But_onmouseup(this)" onmousedown="return But_onmousedown(this)"
onmouseover="return But_onmouseover(this)" onclick=EditFont(17)
onmouseout ="return But_onmouseout(this)" alt=恢复
src="../../images/redo.gif" width="20" height="20" >
      </div>
    </td>
  </tr>
</table>
</FIELDSET>
```

```

</div>
</td>
<td width="64%">
    <div align="center">
        <select name=SelFontType size=1  onchange="return SelFontType_onchange()"
            style="FONT-SIZE: 12px">
            <option value="" selected>字体</option>
            <option value="宋体">宋体</option>
            <option value="黑体">黑体</option>
            <option value="楷体_GB2312">楷体</option>
            <option value="仿宋_GB2312">仿宋</option>
            <option value="MS Song">MS 宋体</option>
            <option value="MS Hei">MS 黑体</option>
            <option value="MingLiU">MingLiU</option>
            <option value="Arial">Arial</option>
            <option value="Arial Black">Arial Black</option>
            <option value="Arial Narrow">Arial Narrow</option>
            <option value="Bradley Hand ITC">Bradley Hand ITC</option>
            <option value="Brush Script MT">Brush Script MT</option>
            <option value="Century Gothic">Century Gothic</option>
            <option value="Comic Sans MS">Comic Sans MS</option>
            <option value="Courier">Courier</option>
            <option value="Courier New">Courier New</option>
            <option value="MS Sans Serif">MS Sans Serif</option>
            <option value="Script">Script</option>
            <option value="System">System</option>
            <option value="Times New Roman">Times New Roman</option>
            <option value="Viner Hand ITC">Viner Hand ITC</option>
            <option value="Verdana">Verdana</option>
            <option value="Wide Latin">Wide Latin</option>
            <option value="Wingdings">Wingdings</option>
        </select>
        <!-- 提供文本 “字体选择” “操作” -->
        <select name=SelFontSize size=1
            onchange="return SelFontSize_onchange()" style="FONT-SIZE: 12px">
            <option value="2">缺省</option>
            <option value="1">一号</option>
            <option value="2">二号</option>
            <option value="3">三号</option>
            <option value="4">四号</option>
            <option value="5">五号</option>
            <option value="6">六号</option>
            <option value="7">七号</option>
            <option value="2" selected>字号</option>
        </select>
        <!-- 提供文本 “字体大小” “操作” -->
        <select name=SelFontColor size=1
            onchange="return SelFontColor_onchange()" style="FONT-SIZE: 12px">
            <option value="">缺省</option>
            <option value="red" style="BACKGROUND-COLOR: red">红色</option>

```

```

<option value="blue" style="BACKGROUND-COLOR: blue">蓝色</option>
<option value="green" style="BACKGROUND-COLOR: green">绿色</option>
<option value="yellow" style="BACKGROUND-COLOR: yellow">黄色</option>
<option value="pink" style="BACKGROUND-COLOR: pink" >粉红</option>
<option value="cyan" style="BACKGROUND-COLOR: cyan">天青</option>
<option value="white" style="BACKGROUND-COLOR: white">白色</option>
<option value="darkred" style="BACKGROUND-COLOR: darkred">深红</option>
<option value="darkblue" style="BACKGROUND-COLOR: darkblue">深蓝</option>
<option value="darkgreen" style="BACKGROUND-COLOR: darkgreen">深绿</option>
<option value="Goldenrod" style="BACKGROUND-COLOR: Goldenrod">土黄</option>
<option value="darkcyan" style="BACKGROUND-COLOR: darkcyan">蓝绿</option>
<option value="gray" style="BACKGROUND-COLOR: gray">浅灰</option>
<option value="black" style="BACKGROUND-COLOR: black">黑色</option>
<option value="" selected>字色</option>
</select>

<!-- 提供文本 “字体背景色”操作 -->

<select name=SelBackColor size=1
      onchange="return SelBackColor_onchange()" style="FONT-SIZE: 12px" >
  <option value="">缺省</option>
  <option value="red" style="BACKGROUND-COLOR: red">红色</option>
  <option value="blue" style="BACKGROUND-COLOR: blue">蓝色</option>
  <option value="green" style="BACKGROUND-COLOR: green">绿色</option>
  <option value="yellow" style="BACKGROUND-COLOR: yellow">黄色</option>
  <option value="pink" style="BACKGROUND-COLOR: pink" >粉红</option>
  <option value="cyan" style="BACKGROUND-COLOR: cyan">天青</option>
  <option value="white" style="BACKGROUND-COLOR: white">白色</option>
  <option value="darkred" style="BACKGROUND-COLOR: darkred">深红</option>
  <option value="darkblue" style="BACKGROUND-COLOR: darkblue">深蓝</option>
  <option value="darkgreen" style="BACKGROUND-COLOR: darkgreen">深绿</option>
  <option value="Goldenrod" style="BACKGROUND-COLOR: Goldenrod">土黄</option>
  <option value="darkcyan" style="BACKGROUND-COLOR: darkcyan">蓝绿</option>
  <option value="gray" style="BACKGROUND-COLOR: gray">浅灰</option>
  <option value="black" style="BACKGROUND-COLOR: black">黑色</option>
  <option value="" selected>字体背景色彩</option>
</select>
</div></td>

<td width="4%">
  <!-- 提供字体 “加粗”操作 -->
  <div align="center">
    <IMG onmouseup="return But_onmouseup(this)" onmousedown="return But_onmousedown(this)"
      onmouseover="return But_onmouseover(this)" onclick=EditFont(1)
      onmouseout ="return But_onmouseout(this)" alt=加粗
      src=" ../images/bold.gif" width="20" height="20" >
    </div>
  </td>

<td width="4%">
  <!-- 提供 “斜体”操作 -->
  <div align="center">
    <IMG onmouseup="return But_onmouseup(this)" onmousedown="return But_onmousedown(this)"
      onmouseover="return But_onmouseover(this)" onclick=EditFont(2)
      onmouseout ="return But_onmouseout(this)" alt=斜体
      src=" ../images/italic.gif" width="16" height="20" >

```

```

        </div>
    </td>
    <td width="4%">
        <!-- 提供 “下划线” 操作 -->
        <div align="center">
            <IMG onmouseup="return But_onmouseup(this)" onmousedown="return But_onmousedown(this)"
                onmouseover="return But_onmouseover(this)" onclick=EditFont(3)
                onmouseout ="return But_onmouseout(this)" alt=下划线
                src=" ../images/under.gif" width="20" height="20" >
        </div>
    </td>
    <td width="4%">&nbsp;   </td>
</tr>
</table>
<hr height=1 width=100%>
<table width="100%" height=30 border="0" cellpadding="0" cellspacing="0" bgColor=buttonface>
    <tr>
        <td width="4%">
            <!-- 提供 “插入表格” 操作 -->
            <div align="center">
                <IMG onmouseup="return But_onmouseup(this)" onmousedown="return But_onmousedown(this)"
                    onmouseover="return But_onmouseover(this)" onclick=InsertTable()
                    onmouseout ="return But_onmouseout(this)" alt=插入表格
                    src=" ../images/table.gif" width="20" height="20" >
            </div>
        </td>
        <td width="4%">
            <!-- 提供 “剪切” 操作 -->
            <div align="center">
                <IMG onmouseup="return But_onmouseup(this)" onmousedown="return But_onmousedown(this)"
                    onmouseover="return But_onmouseover(this)" onclick=EditFont(4)
                    onmouseout ="return But_onmouseout(this)" alt=剪切
                    src=" ../images/cut.gif" width="20" height="20" >
            </div>
        </td>
        <td width="4%">
            <!-- 提供 “复制” 操作 -->
            <div align="center">
                <IMG onmouseup="return But_onmouseup(this)" onmousedown="return But_onmousedown(this)"
                    onmouseover="return But_onmouseover(this)" onclick=EditFont(5)
                    onmouseout ="return But_onmouseout(this)" alt=复制
                    src=" ../images/copy.gif" width="20" height="20" >
            </div>
        </td>
        <td width="4%">
            <!-- 提供 “粘贴” 操作 -->
            <div align="center">
                <IMG onmouseup="return But_onmouseup(this)" onmousedown="return But_onmousedown(this)"
                    onmouseover="return But_onmouseover(this)" onclick=EditFont(6)
                    onmouseout ="return But_onmouseout(this)" alt=粘贴
                    src=" ../images/paste.gif" width="20" height="20" >
            </div>
        </td>
        <td width="4%">
            <!-- 提供 “左对齐” 操作 -->
            <div align="center">

```

```

<IMG onmouseup="return But_onmouseup(this)" onmousedown="return But_onmousedown(this)"
  onmouseover="return But_onmouseover(this)" onclick=EditFont(7)
  onmouseout ="return But_onmouseout(this)" alt=左对齐
  src="../images/aleft.gif" width="20" height="20" >
</div>
</td>
<td width="4%">                                <!-- 提供“居中”操作 -->
  <div align="center">
    <IMG onmouseup="return But_onmouseup(this)" onmousedown="return But_onmousedown(this)"
      onmouseover="return But_onmouseover(this)" onclick=EditFont(8)
      onmouseout ="return But_onmouseout(this)" alt=居中
      src="../images/center.gif" width="20" height="20" >
    </div>
  </td>
<td width="4%">                                <!-- 提供“右对齐”操作 -->
  <div align="center">
    <IMG onmouseup="return But_onmouseup(this)" onmousedown="return But_onmousedown(this)"
      onmouseover="return But_onmouseover(this)" onclick=EditFont(9)
      onmouseout ="return But_onmouseout(this)" alt=右对齐
      src="../images/aright.gif" width="20" height="20" >
    </div>
  </td>
<td width="4%">                                <!-- 提供“编号”操作 -->
  <div align="center">
    <IMG onmouseup="return But_onmouseup(this)" onmousedown="return But_onmousedown(this)"
      onmouseover="return But_onmouseover(this)" onclick=EditFont(10)
      onmouseout ="return But_onmouseout(this)" alt=编号
      src="../images/nlist.gif" width="20" height="20" >
    </div>
  </td>
<td width="4%">                                <!-- 提供“项目编号”操作 -->
  <div align="center">
    <IMG onmouseup="return But_onmouseup(this)" onmousedown="return But_onmousedown(this)"
      onmouseover="return But_onmouseover(this)" onclick=EditFont(11)
      onmouseout ="return But_onmouseout(this)" alt=项目符号
      src="../images/blist.gif" width="20" height="20" >
    </div>
  </td>
<td width="4%">                                <!-- 提供“减少缩进量”操作 -->
  <div align="center">
    <IMG onmouseup="return But_onmouseup(this)" onmousedown="return But_onmousedown(this)"
      onmouseover="return But_onmouseover(this)" onclick=EditFont(12)
      onmouseout ="return But_onmouseout(this)" alt=减少缩进量
      src="../images/ileft.gif" width="20" height="20" >
    </div>
  </td>
<td width="4%">                                <!-- 提供“增加缩进量”操作 -->
  <div align="center">
    <IMG onmouseup="return But_onmouseup(this)" onmousedown="return But_onmousedown(this)"
      onmouseover="return But_onmouseover(this)" onclick=EditFont(13)

```

```

onmouseout="return But_onmouseout(this)" alt=增加缩进量
src="../../images/iright.gif" width="20" height="20" >
</div>
</td>
<td width="4%">
<div align="center">
<IMG onmouseup="return But_onmouseup(this)" onmousedown="return But_onmousedown(this)"
onmouseover="return But_onmouseover(this)" onclick=EditFont(14)
onmouseout="return But_onmouseout(this)" alt=下标
src="../../images/subscript.gif" width="20" height="20" >
</div>
</td>
<td width="4%">
<div align="center">
<IMG onmouseup="return But_onmouseup(this)" onmousedown="return But_onmousedown(this)"
onmouseover="return But_onmouseover(this)" onclick=EditFont(15)
onmouseout="return But_onmouseout(this)" alt=上标
src="../../images/superscript.gif" width="20" height="20" >
</div>
</td>
<td width="4%">
<div align="center">
<IMG onmouseup="return But_onmouseup(this)" onmousedown="return But_onmousedown(this)"
onmouseover="return But_onmouseover(this)" onclick=circle("left")
onmouseout="return But_onmouseout(this)" alt=左环绕
src="../../images/alignleft.gif" width="20" height="20" >
</div>
</td>
<td width="4%">
<div align="center">
<IMG onmouseup="return But_onmouseup(this)" onmousedown="return But_onmousedown(this)"
onmouseover="return But_onmouseover(this)" onclick=circle("texttop")
onmouseout="return But_onmouseout(this)" alt=居中环绕
src="../../images/aligncenter.gif" width="20" height="20" >
</div>
</td>
<td width="4%">
<div align="center">
<MG onmouseup="return But_onmouseup(this)" onmousedown="return But_onmousedown(this)"
onmouseover="return But_onmouseover(this)" onclick=circle("right")
onmouseout="return But_onmouseout(this)" alt=右环绕
src="../../images/alignright.gif" width="20" height="20" >
</div>
</td>
<td width="4%">&nbsp;</td>
<td width="4%">&nbsp;</td>
<td width="6%">&nbsp;</td>
</tr>
</table>
</FIELDSET>

```

<!--编辑排版-->

程序说明：该页面提供了编辑文本的操作，效果如图 30.2 所示。提供编辑操作功能还需要使用到 JavaScript 文件 editpaper.js，在页面程序已经使用<script language=JavaScript src="editpaper.js"></script>进行了定义。

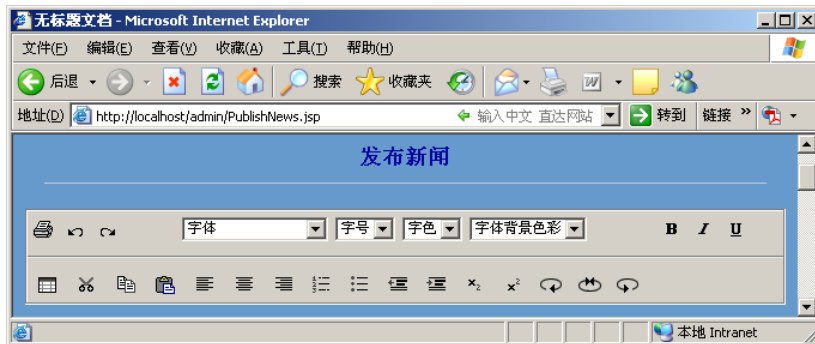


图 30.2 文本编辑界面

30.5.2 文本编辑操作的JavaScript文件Editpaper.js

该文件为 JavaScript 文件，提供了各种文本编辑功能的具体方法，在 editpaper.jsp 文件中得到调用，该文件的详细代码如下：

```
var IFrameObject,TempWin;
function window_onload() {           //该方法进行变量的初始化
    var CurObjTime,YY,MM,DD,CurrentDateStr;
    CurObjTime = new Date();
    YY = CurObjTime.getFullYear ();
    MM = CurObjTime.getMonth () + 1;
    DD = CurObjTime.getDate ();
    CurrentDateStr = YY + "年" + MM + "月" + DD + "日";
}
function SetEdit(myobject){
    var ClipVal = "";
    if (window.frames (myobject.name).document.designMode!="On"){
        window.frames (myobject.name).document.open ();
        window.frames (myobject.name).document.write ("<body style='font-size:16px'>");
        window.frames (myobject.name).document.close ();
        window.frames (myobject.name).document.designMode="On";
    }
    IFrameObject = myobject;
}
function But_onmouseover(MyButObject) {           //鼠标悬浮时触发的事件
                                                    //设置对象的边框颜色和边框宽度

    MyButObject.style.borderLeftColor = "buttonhighlight";
    MyButObject.style.borderTopColor = "buttonhighlight";
    MyButObject.style.borderBottomColor = "buttonshadow";
    MyButObject.style.borderRightColor = "buttonshadow";
    MyButObject.border = 1;
}
function But_onmouseout(MyButObject) {           //鼠标移开时触发的事件
```

```

    MyButObject.border = 0;
}
function But_onmousedown(MyButObject) {           //鼠标按下时触发的事件
    MyButObject.style.borderLeftColor = "black";
    MyButObject.style.borderTopColor = "black";
    MyButObject.style.borderBottomColor = "white";
    MyButObject.style.borderRightColor = "white";
    MyButObject.border = 1;
}
function But_onmouseup(MyButObject) {             //鼠标按起时触发的事件
    But_onmouseover(MyButObject);
}
function SelFontType_onchange() {                 //设置字体类型的方法
    var SelNum,TypeVal;
    SelNum = SelFontType.selectedIndex;
    TypeVal = SelFontType.options [SelNum].value;
    window.frames (IFrameObject.name).document.execCommand ("FontName","",TypeVal);
}
function SelFontSize_onchange() {                 //设置字体大小的方法
    var SelNum,SizeVal;
    SelNum = SelFontSize.selectedIndex;
    SizeVal = SelFontSize.options [SelNum].value;
    window.frames (IFrameObject.name).document.execCommand ("fontsize","",SizeVal);
}
function SelFontColor_onchange() {               //设置字体颜色的方法
    var SelNum,ColorVal;
    SelNum = SelFontColor.selectedIndex;
    ColorVal = SelFontColor.options [SelNum].value;
    window.frames (IFrameObject.name).document.execCommand ("ForeColor","",ColorVal);
}
function SelBackColor_onchange() {              //设置背景颜色的方法
    var SelNum,BgColorVal;
    SelNum = SelBackColor.selectedIndex;
    BgColorVal = SelBackColor.options [SelNum].value;
    window.frames (IFrameObject.name).document.execCommand ("BackColor","",BgColorVal);
}
function EditFont(myVar){                        //通过 myVar 传递的参数进行各种编辑
    var CommandString;
    switch (myVar)
    {
        case 1: CommandString = "Bold";break;           //加粗
        case 2: CommandString = "Italic";break;         //斜体
        case 3: CommandString = "Underline";break;      //下划线
        case 4: CommandString = "Cut";break;             //剪切
        case 5: CommandString = "Copy";break;           //复制
        case 6: CommandString = "Paste";break;          //粘贴
        case 7: CommandString = "JustifyLeft";break;    //左对齐
        case 8: CommandString = "JustifyCenter";break;  //居中
        case 9: CommandString = "JustifyRight";break;   //右对齐
        case 10: CommandString = "InsertOrderedList";break; //编号
    }
}

```



```

        case 11: CommandString = "InsertUnorderedList";break;           //项目编号
        case 12: CommandString = "Outdent";break;                     //减少缩进量
        case 13: CommandString = "Indent";break;                     //增加缩进量
        case 14: CommandString = "Subscript";break;                  //下标
        case 15: CommandString = "Superscript";break;                //上标
        case 16: CommandString = "Undo";break;                       //撤销
        case 17: CommandString = "Redo";break;                       //恢复
        case 20: CommandString = "Print";break;                      //打印
    }

    //如果是编号、项目编号、减少缩进量以及增加缩进量等操作时
    if (CommandString=="InsertOrderedList" || CommandString=="InsertUnorderedList" ||
        CommandString=="Outdent" || CommandString=="Indent"){
        window.frames (IFrameObject.name).document.execCommand (CommandString);
        window.frames (IFrameObject.name).focus();
    }
    else{ //否则进行其他操作
        window.frames (IFrameObject.name).document.execCommand (CommandString);
    }
}

function DelUrl(){ //删除链接
    var MyRangeObject;
    window.frames (IFrameObject.name).focus();
    MyRangeObject = window.frames(IFrameObject.name).document.selection.createRange ();
    if (window.frames(IFrameObject.name).document.selection.type=="Text"){
        window.frames (IFrameObject.name).document.execCommand ("Unlink");
    }
    else if (window.frames(IFrameObject.name).document.selection.type=="Control"){
        ImgObj = MyRangeObject.item(0);
        window.frames (IFrameObject.name).document.execCommand ("Unlink");
        ImgObj.outerHTML = ImgObj.outerHTML;
    }
}

function FinishUrl(){ //设置链接
    var UrlVal,MyRangeObject,NewPath,ImgObj;
    UrlVal = document.form1.TranContral.value;
    if (UrlVal!="") {
        window.frames (IFrameObject.name).focus();
        MyRangeObject = window.frames(IFrameObject.name).document.selection.createRange ();
        window.frames (IFrameObject.name).document.execCommand ("Unlink");
        if (window.frames(IFrameObject.name).document.selection.type=="Text"){
            if (MyRangeObject.htmlText!="") {
                NewPath = "<a href='" + UrlVal + "' target='_blank'>" + MyRangeObject.htmlText + "</a>";
                MyRangeObject.pasteHTML(NewPath);
            }
        }
        else if (window.frames(IFrameObject.name).document.selection.type=="Control"){
            ImgObj = MyRangeObject.item(0);
            NewPath = "<a href='" + UrlVal + "' target='_blank'>" + ImgObj.outerHTML + "</a>";
            ImgObj.outerHTML = NewPath;
        }
    }
}

```

```

    }
    else{
        alert ("编辑 ! ! ! ");
    }
}
function circle(AlignType){ //进行环绕操作, type 指定左环绕、居中环绕或者右环绕
    var MyRangeObject,PicHtmlString,NewPath,ImgObj;
    if (AlignType!="") {
        window.frames (IFrameObject.name).focus();
        MyRangeObject = window.frames(IFrameObject.name).document.selection.createRange ();
        if (window.frames(IFrameObject.name).document.selection.type=="Control"){
            ImgObj = MyRangeObject.item(0);
            if (ImgObj.align!=AlignType)
                ImgObj.align = AlignType;
            else
                ImgObj.align = ""
        }
        else{
            //alert ("编辑 ! ! ! ");
        }
    }
    else{
        //alert ("编辑 ! ! ! ");
    }
}
function InsertTable(){ //插入表格操作
    var TableVal,MyRangeObject;
    var row,col,widths,borders,bordercolor,TabArray,OutString,i,j;
    TableVal = window.showModalDialog("addtable/InsertTable.htm", "", "dialogHeight: 180px; dialogWidth: 280px; dialogTop: 200px; dialogLeft: 200px; help: no; resizable: 1; status: 0");
    if (TableVal != null) {
        TabArray = TableVal.split("|")
        row = TabArray[0];
        col = TabArray[1];
        widths = TabArray[2];
        borders = TabArray[3];
        bordercolor = TabArray[4];
        OutString = "<table cellSpacing=0 cellPadding=0 width=" + widths + "% border=" + borders + " border=" + bordercolor + ">";

        for (i=1;i<=row;i++){
            OutString = OutString + "<tr>";
            for (j=1;j<=col;j++){
                OutString = OutString + "<td width=" + Math.floor(100/col) + "% valign=top></td>";
            }
            OutString = OutString + "</tr>";
        }
        OutString = OutString + "</table>";
        window.frames (IFrameObject.name).focus();
        MyRangeObject = window.frames(IFrameObject.name).document.selection.createRange ();
        MyRangeObject.pasteHTML(OutString);
    }
}

```

```

    }
    else{
        //alert ("编辑！ ！ ！");
    }
}
function Form2Submit(){           //进行表单提交
    document.form2.Content.value = window.frames("contentframe").document.body.innerHTML;
    document.form2.submit ();
}
function EditConCode(){           //将文本转为 HTML 源代码格式
    window.frames (IFrameObject.name).focus();
    window.frames (IFrameObject.name).document.body.innerText =
        window.frames(IFrameObject.name).document.body.innerHTML;
    form2.visualedit.style.backgroundColor = "white";
    form2.viewcode.style.backgroundColor = "#C8D2EC";
    form2.visualedit.disabled = false;
    form2.viewcode.disabled = true;
    window.frames (IFrameObject.name).focus();
}
function EditConVis(){           //重新把文本转为设计格式
    window.frames (IFrameObject.name).focus();
    window.frames(IFrameObject.name).document.body.innerHTML =
        window.frames(IFrameObject.name).document.body.innerText;
    form2.visualedit.style.backgroundColor = "#C8D2EC";
    form2.viewcode.style.backgroundColor = "white";
    form2.visualedit.disabled = true;
    form2.viewcode.disabled = false;
    window.frames (IFrameObject.name).focus();
}
function RemoveFormat(){         //撤销所有的格式设置
    var ClearContent;
    window.frames (IFrameObject.name).focus();
    ClearContent = window.frames(IFrameObject.name).document.body.innerHTML;
    window.frames(IFrameObject.name).document.body.innerText = RemoveHtmlTag(ClearContent);
    window.frames (IFrameObject.name).focus();
}

```

该程序代码：该 JavaScript 代码编写多个方法来实现页面文本编辑操作的功能。

30.5.3 Publishphotonews.jsp 图片新闻的发布页面

和 publishnews.jsp 相类似，不同就是该页面专门用于图片新闻的发布，即在前台页面中提供图片链接的新闻。该页面的详细代码如下：

```

<%@ page import="cn.com.zzb.news.*"%>
<%@ page import="java.util.Iterator"%>
<%@ page import="cn.com.zzb.util.*"%>
<%@ page import="java.text.*"%>
<html>
<head><title>发布图片新闻</title></head>
<link rel="stylesheet" href="admin.css" type="text/css">

```

```

</style>
<script language="javascript">
<!--
    function pic_preview() {                                //预览图片
        var picName=document.form2.pic.value;
        if(picName==null || picName=="")
            window.alert("没有选中图片，无法完成预览。");
        else
            window.open("picPreview.jsp?picName="+picName);
    }
    function pic_insert() {                                //将选择的图片插入到文本框中
        window.frames (IFrameObject.name).focus();
        MyRangeObject = window.frames(IFrameObject.name).document.selection.createRange();
        MyRangeObject.pasteHTML(document.form2.pic.value);
    }
    function set_frist(){
        document.form2.first_pic.disabled=!document.form2.first_pic.disabled;
        document.form2.pic_select.disabled=!document.form2.pic_select.disabled;
        document.form2.first_pic.value="";
        document.form2.picid.value="";
    }
    function select_pic(){                                //打开 selpic.jsp 页面进行图片的选择
        var w=window.open("selpic.jsp?type=1");
    }
    function doForm2Submit(){                              //进行表单的提交
        document.form2.Content.value = window.frames("contentframe").document.body.innerHTML;
        document.form2.submit ();
    }
-->
</script>
<%
    Factory factory = Factory.getInstance();
    NewsBoard newsboard = factory.getNewsBoard();           //初始化 NewsBoard 实例对象
    NewsBoardFactory newsboardfactory = factory.getNewsBoardFactory(); //初始化 NewsBoardFactory 实例
    Pic pic = factory.getPic();                             //初始化 Pic 实例对象
    PicFactory picfactory = factory.getPicFactory();         //初始化 PicFactory 实例对象
%>
<body>
<br>
<div align="center" style="font-weight:bold; font-size:18px; color:#0000A0">发布图片新闻</div>
<hr width="95%" color="#CCCCCC" size="1px">
<br>
<table width="100%" border="0" cellspacing="0" cellpadding="0" align="center">
    <tr>
        <td>
            <jsp:include page="editpaper.jsp"/>
            <form name="form2" method="post" action="PublishPhotoNewsOk.jsp">
                <center>
                    <table width="100%">
                        <tr>

```

[illegible]

[illegible]

30.6.1 处理图片新闻发布的页面publishphotonewsok.jsp

执行图片新闻发布的操作，部分代码和 publishnewsok.jsp 程序相类似，只是多了对关联图片的处理，详细代码如下：

```
<%@ page import="cn.com.zzb.news.*"%>
<%@ page import="java.util.Iterator"%>
<%@ page import="cn.com.zzb.util.*"%>
<%@ page import="java.text.*"%>
<html>
<head>
<link rel="stylesheet" href="admin.css" type="text/css">
<title>无标题文档</title>
</head>
<%
Factory factory = Factory.getInstance();
int BoardID = ParameterUtils.getInt(request,"BoardID");           //获取新闻版块的 ID
int PicID = ParameterUtils.getInt(request,"picid");               //获取关联图片的 ID
int Type = ParameterUtils.getInt(request,"Type");                 //获取到新闻类型参数 Type
String Title = ParameterUtils.getString(request,"Title");         //获取到新闻标题参数 Title
String Keyword = ParameterUtils.getString(request,"Keyword");     //获取到新闻关键字参数 Keyword
String FirstFrom = ParameterUtils.getString(request,"FirstFrom"); //获取到文章出处的参数 FirstFrom
String Author = ParameterUtils.getString(request,"Author");       //获取到文章作者参数 Author
String Editor = ParameterUtils.getString(request,"Editor");       //获取到新闻编辑者参数 Editor
String Department = ParameterUtils.getString(request,"Department"); //获取到编辑者所在部门的参数
String Content = ParameterUtils.getString(request,"Content");     //文章内容的参数
String attributes[] =request.getParameterValues("attributes");    //获取该新闻是否设置为头条或者热点
%>
<body>
<br>
<hr width="95%" color="#CCCCCC" size="1px">
<table width="100%" border="0" cellspacing="0" cellpadding="0" align="center">
<tr>
<td>
<%
String filePath = "";
filePath = application.getRealPath("/template/qnews.template"); //取出新闻模块文件的路径
//设置新闻模板文件中需要插入参数的位置
String[] flag = {"<$title$>","<$date$>","<$author$>","<$from$>","<$content$>","<$editor$>","<$list1$>","
"<$list2$>","<$list3$>","<$belongto$>"};

String templateContent;
try{
templateContent = ReadTemplates.getTlpContent(filePath); //读取新闻模板文件内容
}
catch(ReadTemplateException e){
throw new Exception(e.getMessage());
}
//在新闻模板文件的各个位置插入相应的参数值，从而生成一个新闻文件
templateContent = ReplaceAll.replace(templateContent,flag[0],Title);
templateContent = ReplaceAll.replace(templateContent,flag[1],GetDate.getStringDate());
```



```

templateContent = ReplaceAll.replace(templateContent,flag[2],Author);
templateContent = ReplaceAll.replace(templateContent,flag[3],FirstFrom);
templateContent = ReplaceAll.replace(templateContent,flag[4],Content);
templateContent = ReplaceAll.replace(templateContent,flag[5],Editor);
templateContent = ReplaceAll.replace(templateContent,flag[6],"等待添加购物车");
templateContent = ReplaceAll.replace(templateContent,flag[7],"等待添加分类");
templateContent = ReplaceAll.replace(templateContent,flag[8],"等待添加时尚新闻");
templateContent = ReplaceAll.replace(templateContent,flag[9],Department);
// 根据时间得文件名与路径名
Calendar calendar = Calendar.getInstance();
String fileName = String.valueOf(calendar.getTimeInMillis()) + ".shtml";
// 如果是 linux 系统把 "\\" 改为 "/"
String pathName = application.getRealPath("../news")+"/"+calendar.get(Calendar.YEAR) +
    "/" + (calendar.get(Calendar.MONTH)+1) + "/" + calendar.get(Calendar.DAY_OF_MONTH)+"/";
/**
 * 生成一个 shtml 静态页面文件，然后把编辑好的 templateContent 内容写入到 shtml 文件中。
 */
try {
    WriteHtml.save(templateContent,pathName,fileName);
}
catch(WriteFileException e){
    throw new Exception("新闻发布失败。可能是目录不具备 IO 操作权限。请与管理员联系。");
}

News news = factory.getNews();
NewsFactory newsFactory = factory.getNewsFactory();
//进行数据的封装
news.setBoardID(BoardID);
news.setEditor(Editor);
news.setTitle(Title);
news.setKeyword(Keyword);
news.setAuthor(Author);
news.setFirstFrom(FirstFrom);
news.setDepartment(Department);
if(attributes!=null){
    for(int i=0;i<attributes.length;i++){
        if(attributes[i].equals("1")){
            news.setIfTop(1);
        }
        if(attributes[i].equals("2")){
            news.setIfHot(1);
        }
    }
}
}
news.setNewsFilePath(calendar.get(Calendar.YEAR)+"/"+(calendar.get(Calendar.MONTH)+1)+"/"+
    calendar.get(Calendar.DAY_OF_MONTH)+"/"+fileName);
news.setPublishDate(calendar.getTimeInMillis());
news.setContent("ddd");
news.setType(Type); //Type 为图片类型
news.setPicID(PicID);

```

```

int state=newsFactory.AddNews(news);           //将发布的新闻信息保存在数据库表中
if(state ==1)                                  //如果添加成功
    out.println("<div align='center'><font size=3 color='red'>发布成功</font></div>");
else                                           //否则 添加失败
    out.println("<div align='center'><font size=3 color='red'>失败</font></div>");
    %>
    </td>
</tr>
</table>
<br>
<center><br><br>
    <a href="PublishPhotoNews.jsp">继续发布</a><br>
</center>
</body>
</html>

```

程序说明：在数据库表 news 中，使用 Type 字段来表示发布新闻的类型。Type=1 表示一般新闻，图片又分成很多种；Type=2 表示在新闻栏中显示的一般图片新闻；Type=3 表示在页面 Top 栏上显示的图片新闻；Type=4 表示在页面两侧栏上显示的图片新闻；Type=5 表示在页面中间栏上显示的图片新闻；Type=6 表示的是浮动图片新闻。读者可以根据需要再进行细分。

30.7 实现新闻管理功能

对新闻的管理包括分类查看所有信息、删除新闻、设置某新闻为头条或者热点、新闻重发以及新闻版块之间的移动。

30.7.1 新闻管理的主页面Newsmanagement.jsp

对所有新闻进行统一管理的页面，详细代码如下：

```

<%@ page import="cn.com.zzb.news.*"%>
<%@ page import="java.util.Iterator"%>
<%@ page import="cn.com.zzb.util.*"%>
<%@ page import="java.text.*"%>
<html>
<head><title>新闻管理</title></head>
<link rel="stylesheet" href="admin.css" type="text/css">
<%
//定义数组存放所有图片新闻的类型
String TypeName[] = {"一般文字新闻","一般图片新闻","页面 Top 栏上[250*60]","页面两侧栏上[150*50]","页面中间
栏上[450*80]","浮动图片新闻"};
Factory factory = Factory.getInstance();
News news = factory.getNews();                               //初始化 News 实例对象
NewsFactory newsfactory = factory.getNewsFactory();          //初始化 NewsFactory 实例对象
NewsBoard newsboard = factory.getNewsBoard();                //初始化 NewsBoard 实例对象
NewsBoardFactory newsboardfactory = factory.getNewsBoardFactory(); //初始化 NewsBoardFactory 实例对象
Iterator iterator;
int j=0;

```

```

%>
<body>
<br>
<div align="center" style="font-weight:bold; font-size:18px; color:#0000A0">新闻管理</div>
<hr width="95%" color="#CCCCCC" size="1px">
<br>
<%
for(int i=0;i<6;i++)                                //循环输出六类新闻
{
%>
<table width="100%" border="1"bordercolordark="#FFFFFF" bordercolorlight="#333333">
  <tr bgcolor="#0099FF">
    <td colspan="7" align="center" style="font-weight:bold ">
      <font color="#990033">新闻管理</font>
    </td>
  </tr>
  <tr>
    <td align="center" width="4%">序号</td>
    <td width="38%" align="center">标题</td>
    <td align="center" width="10%">所在版块</td>
    <td align="center" width="17%">更改</td>
    <td align="center" width="13%">操作</td>
    <td align="center" width="10%">日期</td>
    <td align="center" width="8%">编辑者</td>
  </tr>
<%
  iterator = newsfactory.ListTypeNews(i+1);          //取出 Type 为 i+1 的所有新闻
  j=0;
  while(iterator.hasNext())                          //迭代输出这类新闻
  {
    news = (News)iterator.next();
    newsboard = newsboardfactory.ListByID(news.getBoardID()); //取出该新闻的版块信息
    j++;
%>
  <tr>
    <td align="center"><%=j%></td>
    <td>&middot;
    <a href="../news/<%=news.getNewsFilePath()%>" target="_blank" style="text-decoration:underline ">
      <%=news.getTitle()%></a>
    </td>
    <td align="center"><%=newsboard.getName()%></td>
    <form method="post" action="changeTopHot.jsp">
      <td align="center">
        <input type="hidden" name="NewsID" value=<%=news.getID()%>>
        <input type="checkbox" name="attributes" value="1" <%if(news.getIfTop()==1){%> checked<%;}%> >置顶
        <input type="checkbox" name="attributes" value="2" <%if(news.getIfHot()==1){%>checked<%;}%> >热点
        <input type="submit" name="submit" value="更改">
      </td>
    </form>
    <td align="center">

```

```
<a href="DelNews.jsp?NewsID=<%=news.getID()%>"
  onclick="if(confirm('是否删除此条新闻? ')) {return true;}return false;">删除</a>|
<a href="MoveNews.jsp?NewsID=<%=news.getID()%>">移动</a>|
<a href="RePublishNews.jsp?NewsID=<%=news.getID()%>">重发</a></td>
<td align="center"><%=GetDate.getStringDate(news.getPublishDate())%></td>
<td align="center"><%=news.getEditor()%></td>
</tr>
<%
}
%>
</table>
<br>
<%
}
%>
</body>
</html>
```

程序说明：该程序的运行效果如图 30.3 所示，提供了多个新闻管理的操作。

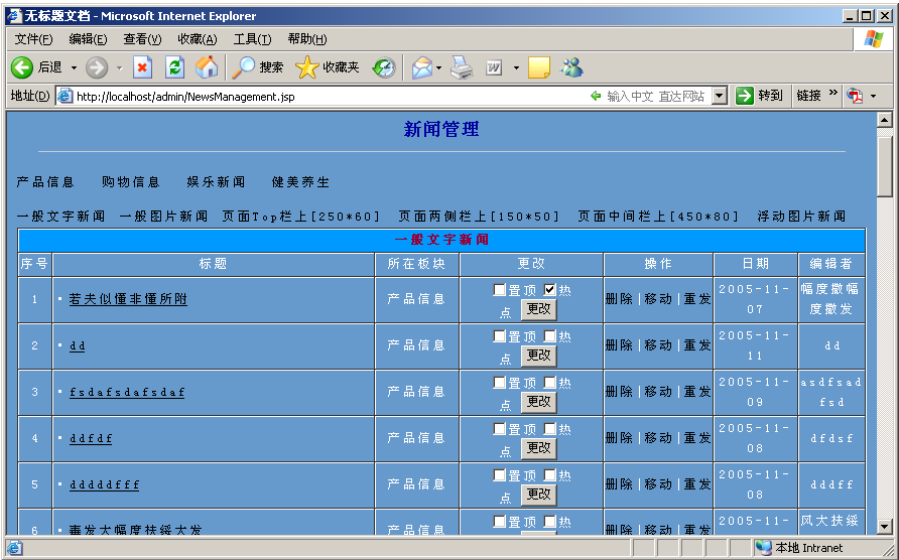


图 30.3 新闻管理页面

30.7.2 修改新闻头条和热点的页面Changetophot.jsp

Newsmanagement.jsp 页面提供了对新闻设置头条和热点的管理，单击“更改”按钮将执行 Changetophot.jsp 页面进行相应的操作。该页面的程序代码如下：

```
<%@ page import="cn.com.zzb.news.*"%>
<%@ page import="java.util.Iterator"%>
<%@ page import="cn.com.zzb.util.*"%>
<%@ page import="java.text.*"%>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312">
<link rel="stylesheet" href="admin.css" type="text/css">
```

```

<title>设置新闻头条和热点</title>
</head>
<%
int NewsID = ParameterUtils.getInt(request,"NewsID");           //取得需要修改的新闻 ID
String attributes[] = request.getParameterValues("attributes"); //使用数组取得头条和热点信息
int IfTop = 0;
int IfHot = 0;
News news = Factory.getInstance().getNews();                    //初始化 News 实例对象
NewsFactory newsfactory = Factory.getInstance().getNewsFactory();//初始化 NewsFactory 实例对象
String previewHttp = request.getHeader("REFERER");              //获取到上页面的地址
%>
<body>
<br>
<hr width="95%" color="#CCCCCC" size="1px">
<br>
<table width="100%" border="0" cellspacing="0" cellpadding="0" align="center">
  <tr>
    <td>
      <%
if(attributes != null){
    for(int i=0;i<attributes.length;i++){
        if(attributes[i].equals("1")){
            IfTop = 1;
        }
        if(attributes[i].equals("2")){
            IfHot = 1;
        }
    }
}

if( newsfactory.EditTopHot(IfTop,IfHot,NewsID) == 1)           //修改新闻头条和热点成功
    out.println("<div align='center'><font color='red' size=4>==修改成功==</font></div>");
else                                                            //否则，修改失败
    out.println("<div align='center'><font color='red' size=4>==修改失败==</font></div>");
%>
    </td>
  </tr>
  <tr><td height="40" align="center"><a href=<%=previewHttp%>>返回上一页</a></td></tr>
</table>
</body>
</html>

```

程序说明：调用 NewsFactory 类中的 EditTopHot(int IfTop,int IfHot,int NewsID)来进行新闻的头条和热点信息进行修改，详细修改方法可以参考 NewsFactory 类代码。

30.7.3 新闻移动的页面Movenews.jsp

当需要在 newsmanagement.jsp 页面上把一个新闻从一个版块中移到另一个版块中时，将需要调用该 JSP 程序。该程序代码如下：

```
<%@ page import="cn.com.zzb.news.*"%>
```

```

<%@ page import="java.util.Iterator"%>
<%@ page import="cn.com.zzb.util.*"%>
<%@ page import="java.text.*"%>
<html>
<head><title>移动新闻</title></head>
<link rel="stylesheet" href="admin.css" type="text/css">
<%
String TypeName[] = {"一般文字新闻","一般图片新闻","页面 Top 栏上[250*60]","页面两侧栏上[150*50]","页面中间
栏上[450*80]","浮动图片新闻"};
int NewsID = ParameterUtils.getInt(request,"NewsID");           //获取到需要移动的新闻 ID
News news = Factory.getInstance().getNews();                     //初始化 News 实例对象
NewsFactory newsfactory = Factory.getInstance().getNewsFactory(); //初始化 NewsFactory 实例对象
NewsBoard newsboard = Factory.getInstance().getNewsBoard();      //初始化 NewsBoard 实例
                                                                    //初始化 NewsBoardFactory 实例
NewsBoardFactory newsboardfactory = Factory.getInstance().getNewsBoardFactory();
String previewHttp = request.getHeader("REFERER");               //获取到上页面的地址
news = newsfactory.ListByID(NewsID);                             //返回需要移动的新闻对象
int ntype = 0;
int nboard = 0;
if(news!=null){                                                  //如果返回的新闻对象不为空
    ntype= news.getType();
    nboard= news.getBoardID();
}
else{                                                            //否则, 返回新闻管理页面
    response.sendRedirect("NewsManagement.jsp");
}
%>
<body>
<br>
<hr width="95%" color="#CCCCCC" size="1px">
<br>
<table width="100%" border="1" cellspacing="0" cellpadding="0">
    <tr bgcolor="#0099FF">
        <td colspan="6" align="center" style="font-weight:bold ">
            <font color="#990033">所移动的新闻信息如下</font>
        </td>
    </tr>
    <tr>
        <td width="53%" align="center">标题</td>
        <td align="center" width="17%">更改</td>
        <td align="center" width="8%">操作</td>
        <td align="center" width="10%">日期</td>
        <td align="center" width="8%">编辑者</td>
    </tr>
    <tr>
        <td>&middot;<%=news.getTitle()%></td>
        <form method="post" action="changeTopHot.jsp">
            <td align="center">
                <input type="hidden" name="NewsID" value=<%=news.getID()%>>
                <input type="checkbox" name="attributes" value="1"

```

```

        <%if(news.getIfTop()==1) out.print("checked");%> >置顶
    <input type="checkbox" name="attributes" value="2"
        <%if(news.getIfHot()==1) out.print("checked");%> >热点
    <input type="submit" name="submit" value="更改">
</td>
</form>
<td align="center">
    <a href="DelNews.jsp?NewsID=<%=news.getID()%>"
        onclick="if(confirm('是否删除此条新闻? ')) {return true;}{return false;}">删除</a>|
    <a href="MoveNews.jsp?NewsID=<%=news.getID()%>">移动</a></td>
<td align="center"><%=GetDate.getStringDate(news.getPublishDate())%></td>
<td align="center"><%=news.getEditor()%></td>
</tr>
</table>
<form method="post"
    action="MoveNewsOk.jsp?previewHttp=<%=java.net.URLEncoder.encode(previewHttp)%>">
<input type="hidden" value=<%=NewsID%> name="NewsID">
<table width="40%" border="0" cellspacing="0" cellpadding="0" align="center">
<tr>
<td width="25%" height="40">&nbsp;&nbsp;&nbsp;选择新闻类型</td>
<td width="75%">&nbsp;&nbsp;&nbsp;
    <select name = "Type">
    <%
        for(int i=0;i<TypeName.length;i++){
    <%
        <option value=<%=i+1%> <%if(nType==(i+1)) { %>selected<%}%>><%=TypeName[i]%></option>
    <%
        }
    <%
        }
    </select>
</td>
</tr>
<tr>
<td height="40">&nbsp;&nbsp;&nbsp;选择所在版块</td>
<td>&nbsp;&nbsp;&nbsp;
    <select name="Board">
    <%
        Iterator iterator = newsboardfactory.List();
        while(iterator.hasNext()){
            newsboard = (NewsBoard)iterator.next();
    <%
        <option value=<%=newsboard.getID()%> <%if(nboard == newsboard.getID()) {%>selected<%}%>>
            <%=newsboard.getName()%></option>
    <%
        }
    <%
        }
    </select>
</td>
</tr>
</tr>
</tr>

```

```

        <td colspan="2" align="center">
            <input type="submit" value="移动"
                onclick="if(confirm('请核实选择的信息，确定移动嘛？')) {return true;}{return false;}">
        </td>
    </tr>
</table>
</form>
<table width="100%" border="0" cellspacing="0" cellpadding="0" align="center">
    <tr>
        <td>
            </td>
        </td>
    </tr>
    <tr>
        <td height="40" align="center"><a href=<%=previewHttp%>>返回上一页</a></td></tr>
    </tr>
</table>
</body>
</html>

```

程序说明：给指定的新闻重新选择类型和所在的版块信息，然后单击“移动”，提交给 `movenewsok.jsp` 页面程序进行相应的操作处理。

30.7.4 处理新闻移动的页面 `Movenewsok.jsp`

该页面进行相应的新闻移动操作，主要对数据库表 `news` 中的版块信息进行修改，详细代码如下：

```

<%@ page import="cn.com.zzb.news.*"%>
<%@ page import="java.util.Iterator"%>
<%@ page import="cn.com.zzb.util.*"%>
<%@ page import="java.text.*"%>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312">
<link rel="stylesheet" href="admin.css" type="text/css">
<title>无标题文档</title>
</head>
<%
int NewsID = ParameterUtils.getInt(request,"NewsID");           //获取得传递的新闻编号参数 NewsID
int BoardID = ParameterUtils.getInt(request,"Board");           //获取得传递的版块编号参数 BoardID
int Type = ParameterUtils.getInt(request,"Type");
String previewHttp = ParameterUtils.getString(request,"previewHttp");
News news = Factory.getInstance().getNews();
NewsFactory newsfactory = Factory.getInstance().getNewsFactory();
%>
<body>
<br>
<hr width="95%" color="#CCCCCC" size="1px">
<br>
<table width="100%" border="0" cellspacing="0" cellpadding="0" align="center">
    <tr>
        <td>
            <%
if( newsfactory.ChangeBoardType(BoardID,Type,NewsID) == 1)

```



```

        out.println("<div align='center'><font color='red' size=4>==移动成功==</font></div>");
    else
        out.println("<div align='center'><font color='red' size=4>==移动失败==</font></div>");
    %>
    </td>
</tr>
<tr><td height="40" align="center"><a href=<%=previewHttp%>>返回上一页</a></td></tr>
</table>
</body>
</html>

```

程序说明：该页面程序进行相应新闻移动操作，调用 NewsFactory 类中的 ChangeBoardType(int BoardID,int Type,int NewsID)方法来修改新闻的类型和版块数据。

30.7.5 实现新闻重发的页面Republishnews.jsp

有时需要对已经发布的新闻进行重发操作，实质上就是修改新闻的发布日期，将其修改为当前日期，从而能够在按日期排序显示新闻时能够重新排在前面，该页面的详细代码如下：

```

<%@ page import="cn.com.zzb.news.*"%>
<%@ page import="java.util.Iterator"%>
<%@ page import="cn.com.zzb.util.*"%>
<%@ page import="java.text.*"%>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312">
<link rel="stylesheet" href="admin.css" type="text/css">
<title>新闻重发</title>
</head>
<%
int NewsID = ParameterUtils.getInt(request,"NewsID");           //获取传递的新闻编号 NewsID
News news = Factory.getInstance().getNews();                    //初始化 News 实例对象
NewsFactory newsfactory = Factory.getInstance().getNewsFactory(); //初始化 NewsFactory 实例对象
String previewHttp = request.getHeader("REFERER");              //获取到前页面的地址
%>
<body>
<br>
<hr width="95%" color="#CCCCCC" size="1px">
<br>
<table width="100%" border="0" cellspacing="0" cellpadding="0" align="center">
    <tr>
        <td>
            <%
//如果将新闻日期修改成当前日期成功
if( newsfactory.RePublishNews(Calendar.getInstance().getTimeInMillis(),NewsID) == 1)
    out.println("<div align='center'><font color='red' size=4>==重新发布成功==</font></div>");
else
    out.println("<div align='center'><font color='red' size=4>==重新发布失败==</font></div>");
%>
        </td>
    </tr>
</table>

```

```

        <tr><td height="40" align="center"><a href=<%=previewHttp%>>返回上一页</a></td></tr>
    </table>
</body>
</html>

```

程序说明：该页面程序调用 NewsFactory 类中的 RePublishNews (long currentDate, int NewsID) 方法来修改新闻的发布日期，这里日期是长整数类型，调用 Calendar.getInstance().getTimeInMillis()方法获得当前的日期。

30.7.6 删除新闻的页面 Delnews.jsp

调用该页面程序进行新闻的删除操作，详细代码如下：

```

<%@ page import="cn.com.zzb.news.*"%>
<%@ page import="java.util.Iterator"%>
<%@ page import="cn.com.zzb.util.*"%>
<%@ page import="java.text.*"%>
<html>
<head>
<link rel="stylesheet" href="admin.css" type="text/css">
<title>删除新闻操作</title></head>
<%
int NewsID = ParameterUtils.getInt(request,"NewsID");           //获取得新闻编号 NewsID
News news = Factory.getInstance().getNews();                   //初始化 New 实例对象
NewsFactory newsfactory = Factory.getInstance().getNewsFactory(); //初始化 NewsFactory 实例对象
String previewHttp = request.getHeader("REFERER");             //获得上页面的地址
%>
<body>
<br>
<hr width="95%" color="#CCCCCC" size="1px">
<table width="100%" border="0" cellspacing="0" cellpadding="0" align="center">
    <tr>
        <td>
<%
if( newsfactory.DelNews(NewsID) == 1)                             //如果该新闻删除成功
    out.println("<div align='center'><font color='red' size=4>== 删除成功 ==</font></div>");
else
    out.println("<div align='center'><font color='red' size=4>== 删除失败 ==</font></div>");
%>
        </td>
    </tr>
    <tr><td height="40" align="center"><a href=<%=previewHttp%>>返回上一页</a></td></tr>
</table>
</body>
</html>

```

程序说明：调用 NewsFactory 类中的 DelNews(int NewsID)方法进行删除操作。

30.8 创建新闻模板文件News.template

该文件是新闻发布的模板文件，即把需要发布的新闻内容插入到该模板文件中，然后生成一个特定新闻相对应的静态 shtml 页面文件，保存在指定目录下，该模块文件的详细代码如下：

```
<%@ page import="cn.com.zzb.news.*"%>
<%@ page import="java.util.Iterator"%>
<%@ page import="cn.com.zzb.util.*"%>
<%@ page import="java.text.*"%>
<html>
<head>
<title>『 新闻系统 』--<$title$></title>                <!-- <$title$>标记需要插入新闻标题的位置 -->
</head>
<style type="text/css">
<!--
body {
    margin-left: 0px;
    margin-top: 0px;
    margin-right: 0px;
    margin-bottom: 0px;
    background-color: #CCCCCC;
}
-->
</style>
<style type="text/css">
<!--
.style1 {color: #FFFFFF}
-->
</style>
<body background="../../images/bg1.gif" topmargin="8">
<table border="0" cellpadding="0" cellspacing="0" width="778" align="center">
    <tr>
        <td width="150" rowspan="3"></td>
        <td height="20" align="right">
            <a href="../../buyproduct.jsp?P_ID=0">购物车</a>
            
            <a href="../../zhuce.jsp">新会员注册</a><a href="../../login.jsp">登陆</a>
            <a href="../../helf.jsp">服务中心</a>
        </td>
    </tr>
</tr>
<tr>
        <td height="25">
            <table border="0" cellpadding="0" cellspacing="0" width="100%">
                <tr>
                    <td width="6"></td>
                    <td width="35" bgcolor="#0066ff" align="center"><a href="index.jsp" class="whitebold">首页</a></td>
                    <td width="6"></td>
                    <td width="3"></td>
                    <td width="6"></td>
                </tr>
            </table>
        </td>
    </tr>
</table>
```

```

<td bgcolor="#0066ff">
<table border="0" cellpadding="0" cellspacing="0" width="100%">
  <tr>
    <td width="80" align="center"><a href="#" class="whitebold">商行简介</a></td>
    <td width="80" align="center"><a href="#" class="whitebold">个人信息</a></td>
    <td width="80" align="center"><a href="#" class="whitebold">联系我们</a></td>
    <td width="80" align="center"><a href="#" class="whitebold">购物车</a></td>
    <td width="80" align="center"><a href="#" class="whitebold">商业论坛</a></td>
    <td>&nbsp;</td>
  </tr>
</table>
</td>
</tr>
</table>
</td>
</tr>
<tr> <td height="25" bgcolor="#CCCCFF" ></td></tr>
</table>
<table width="778" border="0" cellspacing="0" cellpadding="0" align="center">
  <tr><td height="1"></td></tr>
  <tr><td height="1" bgcolor="#FFCCFF"></td></tr>
  <tr><td height="30"></td></tr>
</table>
<table width="778" border="0" cellspacing="0" cellpadding="0" align="center">
  <tr><td height="2"></td></tr>
  <tr>
    <td height="700" valign="top">
      <!--主体部分-->
      <table border="0" width="778" cellpadding="0" cellspacing="0">
        <tr>
          <td width="160" valign="top">
            <table width="100%" border="0" cellpadding="0" cellspacing="0">
              <tr>
                <td width="155">
                  <table border="0"><tr><td height="3"></td></tr></table>
                  <table border="0" cellpadding="0" cellspacing="0" width="155" class="blackborder">
                    <tr><td></td></tr>
                    <tr><td height="100" valign="top"> <$list1$></td></tr>
                  </table>
                  <table border="0"><tr><td height="3"></td></tr></table>
                  <table border="0" cellpadding="0" cellspacing="0" width="155" class="blackborder">
                    <tr><td></td></tr>
                    <tr><td height="400" valign="top"> <$list2$></td></tr>
                  </table>
                  <table border="0"><tr><td height="3"></td></tr></table>
                  <table border="0" cellpadding="0" cellspacing="0" width="155" class="blackborder">
                    <tr><td></td></tr>
                    <tr><td height="150" valign="top"> <$list3$></td></tr>
                  </table>
                <td></td>
              </tr>
            </table>
          </td>
          <td></td>
        </tr>
      </table>
    </td>
  </tr>
</table>

```

```

<table border="0" cellpadding="0" cellspacing="0" width="155" class="blackborder">
  <tr><td></td></tr>
  <tr>
    <td height="100" align="center" style="font-weight:bold; color:#0000FF; font-size:15px"></td>
  </tr>
</table>
<table border="0"><tr><td height="3"></td></tr></table>
</td>
</tr>
</table>
</td>
<td valign="top">
<table width="600" border="0" cellspacing="0" cellpadding="0" align="center">
  <tr><td height="35" align="center" valign="bottom" class="title"> <$title$> </td></tr> <!--插入新闻标题-->
  <tr><td><hr size="1" noshade></td></tr>
  <tr>
    <td height="30" align="right" valign="top">
      <b>发布人: </b><$editor$> <!-- 插入新闻编辑者-->
      <b>发布时间: </b><$date$> <!-- 插入新闻发布日期-->
      <b>作者: </b><$author$> <!-- 插入新闻文章作者-->
      <b>单位: </b><$belongto$> <!-- 插入该文章的首次出处-->
    </td>
  </tr>
  <tr><td><span class="newstext"> </span><$content$></td></tr> <!-- 插入新闻内容-->
  <tr><td align="right" valign="bottom"> <$from$></td></tr> <!-- 插入编辑者所在部门-->
</table>
</td>
</tr>
</table>
</table>
<!--主体部分-->
</td>
</tr>
</table>
</body>
</html>

```

程序说明：程序中使用<\$name\$>格式来标记需要插入参数值的位置，name 为参数名称，例如 title、editor 或者 author 等参数。

30.9 创建相应的JavaBean类文件

下面创建页面中调用的所有 JavaBean 文件。

30.9.1 新闻分类的接口类NewsBoard.java

创建新闻分类的接口类 NewsBoard.java，该类定义了新闻版块相关属性的 set 和 get 方法，代码如下：

```

package cn.com.zzb.news;
public interface NewsBoard {

```

```
public abstract void setID(int ID);
public abstract int getID();
public abstract void setName(String Name);
public abstract String getName();
public abstract void setIntroduce(String Introduce);
public abstract String getIntroduce();
}
```

30.9.2 新闻分类的实现类SqlNewsBoard.java

该类实现上面接口类所定义的所有方法，并定义新闻版块相关的所有属性变量，用于新闻版块有关的数据封装，详细实现代码如下：

```
package cn.com.zzb.news;
public class SqlNewsBoard implements NewsBoard {
    private int ID;           //定义的新闻版块 ID
    private String Name;      //定义的新闻版块名称
    private String Introduce; //定义新闻版块简介的变量
    public void setID(int ID) {
        this.ID = ID;
    }
    public int getID() {
        return ID;
    }
    public void setName(String Name) {
        this.Name = Name;
    }
    public String getName() {
        return Name;
    }
    public void setIntroduce(String Introduce) {
        this.Introduce = Introduce;
    }
    public String getIntroduce() {
        return Introduce;
    }
}
```

30.9.3 新闻分类的操作抽象类NewsBoardFactory.java

新闻版块的操作类，定义了用于版块数据操作的基本方法，代码如下：

```
package cn.com.zzb.news;
import java.util.*;
public abstract class NewsBoardFactory {
    public abstract int Add(NewsBoard newsboard); //添加新闻版块
    public abstract int Del(int ID);              //删除新闻版块
    public abstract Iterator List();              //列举出所有的新闻版块信息
    public abstract NewsBoard ListByID(int ID);   //列举出其中某一个新闻版块的信息
    public abstract int Edit(NewsBoard newsboard); //编辑某一个新闻版块
}
```

}

30.9.4 新闻分类的操作继承类SqlNewsBoardFactory.java

该类为 NewsBoardFactory 抽象类的继承类，实现抽象类中定义的所有方法，该类用于有关新闻版块的数据库操作，详细代码如下：

```
package cn.com.zzb.news;
import java.util.*;
import cn.com.zzb.sql.DBConnect;
public class SqlNewsBoardFactory extends NewsBoardFactory {
    public int Add(NewsBoard newsboard) //添加一个新闻版块
    {
        int state = 0 ;
        String str = "insert into NewsBoard(Name,Introduce) //进行新闻版块插入的 SQL 语句
            values("+newsboard.getName()+",""+newsboard.getIntroduce()+")";
        try{
            DBConnect dbconnect = new DBConnect(str); //数据库连接初始化
            dbconnect.excuteUpdate(); //数据库操作的执行
            state = 1;
            dbconnect.close(); //释放数据库连接资源
        }
        catch(Exception e) {
            e.printStackTrace();
            state = 0;
        }
        return state;
    }
    public int Del(int ID) { //删除一个新闻版块
        int state = 0 ;
        String str = "delete from NewsBoard where ID = "+ID;
        try{
            DBConnect dbconnect = new DBConnect(str);
            dbconnect.excuteUpdate();
            state = 1;
            dbconnect.close();
        }
        catch(Exception e){
            e.printStackTrace();
            state = 0;
        }
        return state;
    }
    public Iterator List(){ //列举出所有的新闻版块信息
        ArrayList arraylist = new ArrayList();
        String str = "select * from NewsBoard order by ID";
        try{
            DBConnect dbconnect = new DBConnect(str);
            dbconnect.excuteQuery();
            NewsBoard newsboard;
```

```

        while(dbconnect.next()){
            //迭代处理每个新闻版块信息
            //进行新闻版块的数据封装
            newsboard = Factory.getInstance().getNewsBoard();
            newsboard.setID(dbconnect.getInt(1));
            newsboard.setName(dbconnect.getString(2));
            newsboard.setIntroduce(dbconnect.getString(3));
            arraylist.add(newsboard);
        }
        dbconnect.close();
    }
    catch(Exception e) {
        e.printStackTrace();
    }
    return arraylist.iterator();
}

public NewsBoard ListByID(int ID) {
    //列举出某一个新闻版块信息

    NewsBoard newsboard = null;
    String str = "select * from NewsBoard where ID = "+ID;
    try{
        DBConnect dbconnect = new DBConnect(str);
        dbconnect.excuteQuery();
        if(dbconnect.next()){
            newsboard = Factory.getInstance().getNewsBoard();
            newsboard.setID(dbconnect.getInt(1));
            newsboard.setName(dbconnect.getString(2));
            newsboard.setIntroduce(dbconnect.getString(3));
        }
        dbconnect.close();
    }
    catch(Exception e) {
        e.printStackTrace();
    }
    return newsboard;
}

public int Edit(NewsBoard newsboard) {
    int state = 0;
    String str = "update NewsBoard set Name='"+newsboard.getName()+"",
        Introduce='"+newsboard.getIntroduce()+"' where ID="+newsboard.getID();
    try{
        DBConnect dbconnect = new DBConnect(str);
        dbconnect.excuteUpdate();
        state = 1;
        dbconnect.close();
    }
    catch(Exception e) {
        e.printStackTrace();
        state = 0;
    }
    return state;
}

```



```

    }
}

```

30.9.5 新闻的接口类News.java

有关新闻信息的数据封装接口类，代码如下：

```

package cn.com.zzb.news;

public interface News {

    public abstract void setID(int i);           //新闻编号
    public abstract int getID();
    public abstract void setBoardID(int i);      //新闻所在板块的编号
    public abstract int getBoardID();
    public abstract void setTitle(String s);     //新闻标题
    public abstract String getTitle();
    public abstract void setEditor(String s);   //编辑者
    public abstract String getEditor();
    public abstract void setKeyword(String s);  //新闻关键字
    public abstract String getKeyword();
    public abstract void setNewsFilePath(String s); //新闻生成静态网页所存地址
    public abstract String getNewsFilePath();
    public abstract void setPublishDate(long l); //发布日期
    public abstract long getPublishDate();
    public abstract void setFirstFrom(String s); //出自何处
    public abstract String getFirstFrom();
    public abstract void setIfHot(int i);        //设置此新闻是否为热点
    public abstract int getIfHot();
    public abstract void setIfTop(int i);        //设置此新闻是否放在顶部
    public abstract int getIfTop();
    public abstract void setDCount(int i);       //点击计数
    public abstract int getDCount();
    public abstract void setContent(String s);   //新闻内容
    public abstract String getContent();
    public abstract void setAuthor(String s);    //原作者
    public abstract String getAuthor();
    public abstract void setDepartment(String s); //发布部门
    public abstract String getDepartment();
    public abstract void setType(int i);        //设置新闻类型
    public abstract int getType();
    public abstract void setPicID(int PicID);    //设置使用的图片 ID
    public abstract int getPicID();
}

```

程序说明：该类和下面的实现类 `SqlNews` 一起用作对新闻相关数据的封装，数据库表 `news` 中共有 16 属性字段，对应各个字段，该接口类定义了相应的 `set` 和 `get` 方法。

30.9.6 新闻的实现类SqlNews.java

该实现类定义了数据库表 `news` 中各字段对应的属性变量，并实现了接口类所定义的各属性变量的 `set` 和 `get` 方法，详细代码如下：

```

package cn.com.zzb.news;
public class SqlNews implements News {
    private int ID;                //新闻编号
    private int BoardID;           //新闻所在板块的编号
    private String Title;          //新闻标题
    private String Editor;         //编辑者
    private String FirstFrom;      //出自何处
    private String Author;         //原作者
    private String Department;     //发布部门设置
    private String Keyword;        //新闻关键字
    private String NewsFilePath;   //生成静态网页所存地址
    private String Content;        //新闻内容
    private long PublishDate;      //发布日期
    private int DCount;            //点击计数
    private int IfHot;             //设置此新闻是否为热点新闻
    private int IfTop;             //此新闻是否放在顶部
    private int Type;              //设置新闻类型
    private int PicID;             //设置使用的图片 ID
    .....
                                //各属性变量对象的 set 和 get 方法
    .....
}

```

程序说明：News 接口类和 SqlNews 实现类一起对新闻相关的数据进行封装操作。

30.9.7 新闻操作的抽象类NewsFactory.java

与新闻相关的数据操作类，该类为抽象类，定义了新闻的各数据库操作方法，代码如下：

```

package cn.com.zzb.news;
import java.util.Iterator;
public abstract class NewsFactory {
    public abstract int AddNews(News news);                //添加一个新闻信息
    public abstract int DelNews(int ID);                   //删除一个新闻信息
    public abstract int DelNewsByBoardID(int BoardID);     //删除一个版块下的所有新闻信息
    public abstract Iterator List(int BoardID,int pages,int e); //分页列举出一个版块下的所有新闻信息
    public abstract Iterator ListTopNews(int e,int Type,int BoardID); //列举某类型，版块下的前 e 条头条新闻
    public abstract Iterator ListHotNews(int e,int Type,int BoardID); //列举某类型，版块下的前 e 条热点新闻
                                                                //分页列举出某类型某版块下的所有新闻
    public abstract Iterator ListHotNewsByPages(int e,int Type,int pages,int BoardID);
    public abstract int CountListHotNews(int Type,int BoardID); //计算出某类型某版块下的热点新闻的数量
    public abstract News ListByID(int ID);                 //列举出某一个新闻信息
    public abstract int getIDByNewsFilePath(String NewsFilePath); //通过新闻静态页面的路径查找到 ID
    public abstract Iterator ListTypeNews(int Type);       //列举出某个类型下的所有新闻信息
    public abstract int EditTopHot(int IfTop,int IfHot,int ID); //更新某新闻的头条和热点信息
    public abstract int ChangeBoardType(int BoardID,int Type,int ID); //更改某新闻的类型和所在版块
    public abstract int RePublishNews(long NowDate,int ID); //重新发布某新闻
    public abstract int CountNewsNumByBoardID(int BoardID); //计算某版块下的所有新闻信息
}

```

30.9.8 新闻操作的继承类SqlNewsFactory.java

下面类将继承以上的抽象类，并实现抽象类中所定义的所有方法，该类的具体实现代码如下：

```
package cn.com.zzb.news;
import cn.com.zzb.sql.DBConnect;
import java.util.*;
public class SqlNewsFactory extends NewsFactory {           //添加一个新闻信息
    public int AddNews(News news) {
        int ID = 0;
        String str = "insert into News(BoardID,Title,Editor,FirstFrom,Author,Department,Keyword,
                        NewsFilePath,Content,PublishDate,DCount,IfHot,IfTop,Type,PicID)
                        values("+news.getBoardID()+",""+news.getTitle()+",""+news.getEditor()+",""+
                        news.getFirstFrom()+",""+news.getAuthor()+",""+news.getDepartment()+",""+
                        news.getKeyword()+",""+news.getNewsFilePath()+",""+news.getContent()+",""+
                        news.getPublishDate()+",0,""+news.getIfHot()+",""+news.getIfTop()+",""+
                        news.getType()+",""+news.getPicID()+")";

        try{
            DBConnect dbconnect = new DBConnect(str);
            dbconnect.excuteUpdate();
            dbconnect.close();
            ID = 1;
        }
        catch(Exception e) {
            e.printStackTrace();
            ID = 0;
        }
        return ID;
    }
    public int getIDByNewsFilePath(String NewsFilePath) {           //通过新闻静态页面的路径查找到 ID
        int ID = 0;
        String str = "select ID from News where NewsFilePath='"+NewsFilePath+"'";
        try{
            DBConnect dbconnect = new DBConnect(str);
            dbconnect.excuteQuery();
            if(dbconnect.next()){
                ID = dbconnect.getInt(1);
            }
            dbconnect.close();
        }
        catch(Exception e) {
            e.printStackTrace();
        }
        return ID;
    }
    public int DelNews(int ID) {           //删除一个新闻信息
        int state = 0;
        String str = "delete from News where ID = "+ID;
        try{
            DBConnect dbconnect = new DBConnect(str);
```

```

        dbconnect.excuteUpdate();
        state = 1;
        dbconnect.close();
    }
    catch(Exception e) {
        e.printStackTrace();
        state = 0;
    }
    return state;
}

public int DelNewsByBoardID(int BoardID) { //删除一个版块下的所有新闻信息
    int state = 0;
    String str = "delete from News where BoardID = "+BoardID;
    try{
        DBConnect dbconnect = new DBConnect(str);
        dbconnect.excuteUpdate();
        state = 1;
        dbconnect.close();
    }
    catch(Exception e) {
        e.printStackTrace();
        state = 0 ;
    }
    return state ;
}

public Iterator List(int BoardID,int pages,int e){ //分页列举出一个版块下的所有新闻信息
    ArrayList arraylist = new ArrayList();
    int num = (pages - 1)*e;
    String str = "select top "+e+" * from News where ID not in(select top "+num+" ID from News
        where BoardID="+BoardID+" order by IfTop desc,IfHot desc,PublishDate desc) and
        BoardID="+BoardID+" order by IfTop desc,IfHot desc,PublishDate desc";
    if(e == 0)
        str = "select * from News where BoardID="+BoardID+"
            order by IfTop desc,IfHot desc,PublishDate desc";
    try{
        DBConnect dbconnect = new DBConnect(str);
        dbconnect.excuteQuery();
        News news;
        while(dbconnect.next()){ //迭代处理每个取出的新闻信息
            news = Factory.getInstance().getNews(); //将属性信息封装到 News 实例对象中

            news.setID(dbconnect.getInt(1));
            news.setBoardID(dbconnect.getInt(2));
            news.setTitle(dbconnect.getString(3));
            news.setEditor(dbconnect.getString(4));
            news.setFirstFrom(dbconnect.getString(5));
            news.setAuthor(dbconnect.getString(6));
            news.setDepartment(dbconnect.getString(7));
            news.setKeyword(dbconnect.getString(8));
            news.setNewsFilePath(dbconnect.getString(9));

```

```

        news.setContent(dbconnect.getString(10));
        news.setPublishDate(dbconnect.getLong(11));
        news.setDCount(dbconnect.getInt(12));
        news.setIfHot(dbconnect.getInt(13));
        news.setIfTop(dbconnect.getInt(14));
        news.setType(dbconnect.getInt(15));
        news.setPicID(dbconnect.getInt(16));
        arraylist.add(news);
    }
    dbconnect.close();
}
catch(Exception ee) {
    ee.printStackTrace();
}
return arraylist.iterator();
}
public Iterator ListTopNews(int e,int Type,int BoardID) {           //列举出某类型和版块下前 e 个头条新闻
    ArrayList arraylist = new ArrayList();
    String str = "";
    if(BoardID == 0) {                                           //如果参数 BoardID=0 表示所有版块
        if(e == 0)                                              //如果 e 参数为 0 表示取出所有的新闻
            str = "select * from News where Type="+Type+"
                    order by IfTop desc,IfHot desc,PublishDate desc";
        else                                                    //否则，取出前 e 个新闻信息
            str = "select top "+e+" * from News where Type="+Type+"
                    order by IfTop desc,IfHot desc,PublishDate desc";
    }
    else{                                                        //否则，取出某版块下的所有新闻信息
        if(e == 0)
            str = "select * from News where Type="+Type+" and BoardID="+BoardID+"
                    order by IfTop desc,IfHot desc,PublishDate desc";
        else
            str = "select top "+e+" * from News where Type="+Type+" and BoardID="+BoardID+"
                    order by IfTop desc,IfHot desc,PublishDate desc";
    }
    try{
        DBConnect dbconnect = new DBConnect(str);
        dbconnect.excuteQuery();
        News news;
        while(dbconnect.next()){                                //迭代处理每一个取出的新闻信息
            news = Factory.getInstance().getNews();
            news.setID(dbconnect.getInt(1));
            news.setBoardID(dbconnect.getInt(2));
            news.setTitle(dbconnect.getString(3));
            news.setEditor(dbconnect.getString(4));
            news.setFirstFrom(dbconnect.getString(5));
            news.setAuthor(dbconnect.getString(6));
            news.setDepartment(dbconnect.getString(7));
            news.setKeyword(dbconnect.getString(8));
            news.setNewsFilePath(dbconnect.getString(9));

```

```

        news.setContent(dbconnect.getString(10));
        news.setPublishDate(dbconnect.getLong(11));
        news.setDCount(dbconnect.getInt(12));
        news.setIfHot(dbconnect.getInt(13));
        news.setIfTop(dbconnect.getInt(14));
        news.setType(dbconnect.getInt(15));
        news.setPicID(dbconnect.getInt(16));
        arraylist.add(news);
    }
    dbconnect.close();
}
catch(Exception ee) {
    ee.printStackTrace();
}
return arraylist.iterator();
}
public Iterator ListHotNews(int e,int Type,int BoardID) {           //列举某类型和版块下前 e 个热点新闻
    ArrayList arraylist = new ArrayList();
    String str = "";
    if(BoardID == 0){                                               //BoardID 为 0 表示取出所有版块下的信息
        if(e == 0)                                                 //e 为 0 表示所有新闻信息
            str = "select * from News where Type="+Type+" and BoardID!=1 and BoardID!=2
                    order by IfHot desc,PublishDate desc";
        else                                                         //否则，只取出前 e 个新闻信息
            str = "select top "+e+" * from News where Type="+Type+" and BoardID!=1 and BoardID!=2
                    order by IfHot desc,PublishDate desc";
    }
    else{                                                            //否则，只取出某版块下的新闻信息
        if(e == 0)
            str = "select * from News where Type="+Type+" and BoardID="+BoardID+"
                    order by IfHot desc,PublishDate desc";
        else
            str = "select top "+e+" * from News where Type="+Type+" and BoardID="+BoardID+"
                    order by IfHot desc,PublishDate desc";
    }
    try{
        DBConnect dbconnect = new DBConnect(str);
        dbconnect.excuteQuery();
        News news;
        while(dbconnect.next()){                                     //迭代处理某个取出的新闻信息
            news = Factory.getInstance().getNews();
            news.setID(dbconnect.getInt(1));
            news.setBoardID(dbconnect.getInt(2));
            news.setTitle(dbconnect.getString(3));
            news.setEditor(dbconnect.getString(4));
            news.setFirstFrom(dbconnect.getString(5));
            news.setAuthor(dbconnect.getString(6));
            news.setDepartment(dbconnect.getString(7));
            news.setKeyword(dbconnect.getString(8));
            news.setNewsFilePath(dbconnect.getString(9));

```

```

        news.setContent(dbconnect.getString(10));
        news.setPublishDate(dbconnect.getLong(11));
        news.setDCount(dbconnect.getInt(12));
        news.setIfHot(dbconnect.getInt(13));
        news.setIfTop(dbconnect.getInt(14));
        news.setType(dbconnect.getInt(15));
        news.setPicID(dbconnect.getInt(16));
        arraylist.add(news);
    }
    dbconnect.close();
}
catch(Exception ee)
{
    ee.printStackTrace();
}
return arraylist.iterator();
}
public Iterator ListHotNewsByPages(int e,int Type,int pages,int BoardID)
{
    //分页列举某类型和版块下所有的热点新闻
    ArrayList arraylist = new ArrayList t();
    int num = (pages-1)*e;
    String str = "";
    if(BoardID == 0)
    {
        if(e == 0)
            str = "select * from News where Type="+Type+" and BoardID!=1 and BoardID!=2
                order by IfTop desc,IfHot desc,PublishDate desc";
        else
            str = "select top "+e+" * from News where Type="+Type+" and BoardID!=1 and BoardID!=2
                and ID not in(select top "+num+" ID from News where Type="+Type+" and BoardID!=1
                and BoardID!=2 order by IfTop desc,IfHot desc,PublishDate desc)
                order by IfTop desc,IfHot desc,PublishDate desc";
    }
    else{
        if(e == 0)
            str = "select * from News where Type="+Type+" and BoardID="+BoardID+"
                order by IfTop desc,IfHot desc,PublishDate desc";
        else
            str = "select top "+e+" * from News where Type="+Type+" and BoardID="+BoardID+"
                and ID not in(select top "+num+" ID from News where Type="+Type+"
                and BoardID="+BoardID+" order by IfTop desc,IfHot desc,PublishDate desc)
                order by IfTop desc,IfHot desc,PublishDate desc";
    }
}
try{
    DBConnect dbconnect = new DBConnect(str);
    dbconnect.excuteQuery();
    News news;
    while(dbconnect.next()){
        news = Factory.getInstance().getNews();
    }
}

```

```

        news.setID(dbconnect.getInt(1));
        news.setBoardID(dbconnect.getInt(2));
        news.setTitle(dbconnect.getString(3));
        news.setEditor(dbconnect.getString(4));
        news.setFirstFrom(dbconnect.getString(5));
        news.setAuthor(dbconnect.getString(6));
        news.setDepartment(dbconnect.getString(7));
        news.setKeyword(dbconnect.getString(8));
        news.setNewsFilePath(dbconnect.getString(9));
        news.setContent(dbconnect.getString(10));
        news.setPublishDate(dbconnect.getLong(11));
        news.setDCount(dbconnect.getInt(12));
        news.setIfHot(dbconnect.getInt(13));
        news.setIfTop(dbconnect.getInt(14));
        news.setType(dbconnect.getInt(15));
        news.setPicID(dbconnect.getInt(16));
        arraylist.add(news);
    }
    dbconnect.close();
}
catch(Exception ee) {
    ee.printStackTrace();
}
return arraylist.iterator();
}
public int CountListHotNews(int Type,int BoardID) {           //计算某类型和版块下的所有新闻数量
    int count = 0;
    String str = "";
    if(BoardID == 0) {
        str = "select count(*) from News where Type="+Type+" and BoardID!=1 and BoardID!=2";
    }
    else{
        str = "select count(*) from News where Type="+Type+" and BoardID="+BoardID;
    }
    try{
        DBConnect dbconnect = new DBConnect(str);
        dbconnect.excuteQuery();
        while(dbconnect.next()){
            count = dbconnect.getInt(1);
        }
        dbconnect.close();
    }
    catch(Exception ee) {
        ee.printStackTrace();
    }
    return count;
}
public Iterator ListTypeNews(int Type) {
    ArrayList arraylist = new ArrayList();
    String str = "select * from News where Type="+Type+"

```



```
        order by BoardID,IfTop desc,IfHot desc,PublishDate desc";
    try{
        DBConnect dbconnect = new DBConnect(str);
        dbconnect.excuteQuery();
        News news;
        while(dbconnect.next()){
            news = Factory.getInstance().getNews();
            news.setID(dbconnect.getInt(1));
            news.setBoardID(dbconnect.getInt(2));
            news.setTitle(dbconnect.getString(3));
            news.setEditor(dbconnect.getString(4));
            news.setFirstFrom(dbconnect.getString(5));
            news.setAuthor(dbconnect.getString(6));
            news.setDepartment(dbconnect.getString(7));
            news.setKeyword(dbconnect.getString(8));
            news.setNewsFilePath(dbconnect.getString(9));
            news.setContent(dbconnect.getString(10));
            news.setPublishDate(dbconnect.getLong(11));
            news.setDCount(dbconnect.getInt(12));
            news.setIfHot(dbconnect.getInt(13));
            news.setIfTop(dbconnect.getInt(14));
            news.setType(dbconnect.getInt(15));
            news.setPicID(dbconnect.getInt(16));
            arraylist.add(news);
        }
        dbconnect.close();
    }
    catch(Exception e){
        e.printStackTrace();
    }
    return arraylist.iterator();
}

public News ListByID(int ID){
    String str = "select * from News where ID = "+ID;
    News news = null;
    try{
        DBConnect dbconnect = new DBConnect(str);
        dbconnect.excuteQuery();
        if(dbconnect.next()){
            news = Factory.getInstance().getNews();
            news.setID(dbconnect.getInt("ID"));
            news.setBoardID(dbconnect.getInt("BoardID"));
            news.setTitle(dbconnect.getString("Title"));
            news.setEditor(dbconnect.getString("Editor"));
            news.setFirstFrom(dbconnect.getString("FirstFrom"));
            news.setAuthor(dbconnect.getString("Author"));
            news.setDepartment(dbconnect.getString("Department"));
            news.setKeyword(dbconnect.getString("Keyword"));
            news.setNewsFilePath(dbconnect.getString("NewsFilePath"));
            news.setContent(dbconnect.getString("Content"));
```

```
        news.setPublishDate(dbconnect.getLong("PublishDate"));
        news.setDCount(dbconnect.getInt("DCount"));
        news.setIfHot(dbconnect.getInt("IfHot"));
        news.setIfTop(dbconnect.getInt("IfTop"));
        news.setType(dbconnect.getInt("Type"));
        news.setPicID(dbconnect.getInt("PicID"));
    }
    dbconnect.close();
}
catch(Exception e){
    e.printStackTrace();
}
return news;
}

public int EditTopHot(int IfTop,int IfHot,int ID){
    int state = 0;
    String str = "update News set IfTop="+IfTop+",IfHot="+IfHot+" where ID="+ID;
    try{
        DBConnect dbconnect = new DBConnect(str);
        dbconnect.excuteUpdate();
        state = 1;
        dbconnect.close();
    }
    catch(Exception e){
        e.printStackTrace();
        state = 0;
    }
    return state;
}

public int ChangeBoardType(int BoardID,int Type,int ID){
    int state = 0;
    String str = "update News set BoardID="+BoardID+",Type="+Type+" where ID="+ID;
    try{
        DBConnect dbconnect = new DBConnect(str);
        dbconnect.excuteUpdate();
        state = 1;
        dbconnect.close();
    }
    catch(Exception e){
        e.printStackTrace();
        state = 0;
    }
    return state;
}

public int RePublishNews(long NowDate,int ID){
    int state = 1;
    String str = "update News set PublishDate="+NowDate+" where ID="+ID;
    try{
        DBConnect dbconnect = new DBConnect(str);
        dbconnect.excuteUpdate();
```

```

        state = 1;
        dbconnect.close();
    }
    catch(Exception e){
        e.printStackTrace();
    }
    return state ;
}
public int CountNewsNumByBoardID(int BoardID){
    int count = 0;
    String str = "select count(*) from News where BoardID="+BoardID;
    try{
        DBConnect dbconnect = new DBConnect(str);
        dbconnect.excuteQuery();
        if(dbconnect.next()){
            count = dbconnect.getInt(1);
        }
        dbconnect.close();
    }
    catch(Exception e){
        e.printStackTrace();
    }
    return count;
}
}

```

程序说明：该类新闻数据的操作类，实现了 NewsFactory 抽象类中所定义的所有方法。

21.7.9 文件读取操作类ReadTemplates.java

发布新闻的模版文件 qqnews.template 已经在前面介绍了，这里创建类文件就是用来读取上面的模版文件，详细代码如下：

```

package cn.com.zzb.util;
import java.io.FileInputStream;           //文件读取流类
import java.io.IOException;              //IO 流的异常处理类
public class ReadTemplates {
    private static String tlpContent = null; //该字符串变量用来保存读取的文件内容
    public static String getTlpContent(String s) //返回读取的文件内容，其中 s 是文件存放路径
        throws ReadTemplateException
    {
        if(tlpContent == null) //如果字符串变量 tlpContent 为空
        {
            try
            {
                tlpContent = readTemplateContent(s); //调用下面实现的 readTemplateContent()方法
                if(tlpContent == null) //如果读取得内容为空将跑出异常
                    throw new ReadTemplateException("tlpContent is null !");
            }
            catch(ReadTemplateException readtemplateexception)
            {
                throw new ReadTemplateException("read template error !");
            }
        }
    }
}

```

```

    }
    return tlpContent;
}
private static synchronized String readTemplateContent(String s)           //读取文件的方法，s 为文件路径
    throws ReadTemplateException
{
    String s1 = null;
    try
    {
        FileInputStream fileinputstream = new FileInputStream(s);           //初始化该文件的输入（读取）流
        int i = fileinputstream.available();                               //返回文件内容的总字节数
        byte abyte0[] = new byte[i];                                     //初始化一个字节数据
        fileinputstream.read(abyte0);                                     //把文件读取的内容存放在数组中
        fileinputstream.close();
        s1 = new String(abyte0);                                           //强转换成字符串
    }
    catch(IOException ioexception)
    {
        throw new ReadTemplateException("io error !");
    }
    return s1;
}
}

```

程序说明：文件读取使用的类为 `import java.io.FileInputStream`，并且文件读取返回的是字节类型，所以这里需要定义一个字节数组来存放读取的内容，最后强硬转换成字符串返回。

30.9.10 标签替换操作类 ReplaceAll.java

在新闻模板文件中，标记多个用于插入相应新闻参数值的位置，该类就是用于实现插入操作，详细的代码内容如下：

```

package cn.com.zzb.util;
public class ReplaceAll {
    public static String replace(String s, String s1, String s2)
    {
        //s 为模板文件的读取内容；s1 为插入位置；s2 为插入参数值
        if(s == null)
            return null;
        StringBuffer stringbuffer = new StringBuffer();
        int i = s.length();           //模板文件内容的字节长度
        int j = s1.length();         //插入位置的标记名长度
        int k;
        int l;
        for(k = 0; (l = s.indexOf(s1, k)) >= 0; k = l + j)
        {
            stringbuffer.append(s.substring(k, l));
            stringbuffer.append(s2); //插入参数值
        }
        if(k < i)
            stringbuffer.append(s.substring(k));
        return stringbuffer.toString();
    }
}

```

```
}  
}
```

下面将要介绍的是文件写入操作：

30.9.11 写文件操作类WriteHtml.java

将插入相应新闻信息的模板文件内容再重新写入到一个生成的静态页面文件（后缀为.shtml）中，该新闻发布之后，用户即可在前台页面中通过存放在数据库表 **news** 中的路径来浏览该新闻信息，该写入文件类的代码如下：

```
package cn.com.zzb.util;  
import java.io.*;  
public class WriteHtml {  
    public static void save(String s, String s1, String s2)    //s1 为写入文件目录，s2 为文件名，s 为新闻内容  
        throws WriteFileException  
    {  
        try{  
            getFile(s1);                                //查看 s1 目录是否存在，不存在将创建  
                                                        //初始化一个写入操作流  
            FileOutputStream fileoutputstream = new FileOutputStream(s1 + s2);  
            byte abyte0[] = s.getBytes();              //将新闻内容硬转化字节数组类型存放  
            fileoutputstream.write(abyte0);             //将新闻内容写入  
            fileoutputstream.close();  
        }  
        catch(IOException ioexception) {  
            throw new WriteFileException("write file error !");  
        }  
    }  
    private static void getFile(String s)  
        throws WriteFileException  
    {  
        try  
        {  
            File file = new File(s);                    //初始化 s 目录  
            if(!file.exists())                          //查看此目录是否存在，如果不存在  
                file.mkdirs();                          //则创建 s 目录  
        }  
        catch(Exception exception)  
        {  
            throw new WriteFileException("wirte File error !");  
        }  
    }  
}
```

程序说明：程序中的 `getFile(String s)` 方法用来查看 `s` 目录路径是否存在，如果不存在则创建该目录；其中的 `save(String s, String s1, String s2)` 方法将插入相应参数的新闻内容写入到指定的文件中，其中 `s` 为插入相应参数值的新闻信息；`s1` 为写入文件的目录名；`s2` 即为写入文件的名称。

30.10 本章小结

本章介绍了新闻发布和管理模块，其中新闻可以分成一般新闻与图片新闻两种，对图片进行操作和管理的功能已经在上一章介绍。

本章介绍的新闻发布模块涉及到的重点是使用了新闻模板概念。通过在新闻模板中动态插入新闻信息来生产静态的新闻页面。然后用户可以在新闻浏览页面中点击文字链接或者图片链接来查看相应的新闻信息。其中涉及到文件读写操作。