



Web程序设计
—ASP.NET



© 主编 杨 玥



Web 程序设计

——ASP. NET

主 编 杨 玥

副主编 吴 瑕 李 莹 刘 洋

张 岩 武兴睿 张 巍

杨 岚 姜婉宁 赵玲玲

于希江



北京理工大学出版社

BEIJING INSTITUTE OF TECHNOLOGY PRESS

内 容 简 介

本书是 CDIO 项目驱动型规划教材，以任务为中心，以职业岗位能力为目标，按照企业网站开发的基本流程组织教材内容。通过精心构造的项目，从需求分析、系统设计、系统开发、系统测试到系统部署，循序渐进地向读者展现知识结构，让读者在做项目的过程中轻松掌握 ASP.NET 网站开发技术。

版权专有 侵权必究

图书在版编目 (CIP) 数据

Web 程序设计: ASP.NET / 杨玥主编. —北京: 北京理工大学出版社, 2020.9
ISBN 978 - 7 - 5682 - 9055 - 5

I. ①W… II. ①杨… III. ①网页制作工具 - 程序设计 - 高等学校 - 教材
IV. ①TP393.092

中国版本图书馆 CIP 数据核字 (2020) 第 176522 号

出版发行 / 北京理工大学出版社有限责任公司

社 址 / 北京市海淀区中关村南大街 5 号

邮 编 / 100081

电 话 / (010) 68914775 (总编室)

(010) 82562903 (教材售后服务热线)

(010) 68948351 (其他图书服务热线)

网 址 / <http://www.bitpress.com.cn>

经 销 / 全国各地新华书店

印 刷 / 三河市天利华印刷装订有限公司

开 本 / 787 毫米 × 1092 毫米 1/16

印 张 / 10

字 数 / 241 千字

版 次 / 2020 年 9 月第 1 版 2020 年 9 月第 1 次印刷

定 价 / 30.00 元

责任编辑 / 王玲玲

文案编辑 / 王玲玲

责任校对 / 刘亚男

责任印制 / 李志强

图书出现印装质量问题，请拨打售后服务热线，本社负责调换

前 言

随着网络技术的快速发展，网络程序设计和 Web 应用技术得到了广泛的应用。ASP.NET 技术是 Microsoft 公司推出的基于 Microsoft.NET 框架的新一代网络程序设计和 Web 应用开发工具，是 Web 应用开发的主流技术之一。

本书内容包括项目导入、11 个子项目和项目总结。11 个子项目分别为信息发布网站需求概述、信息发布网站的三层架构、信息发布网站的创建、信息发布网站的页面布局和设计、网站用户控件的创建、网站信息验证功能、网站管理员登录功能、网站导航控件的应用、网站的数据访问应用、网站测试、网站的发布实现，这些子项目从整体上形成了信息发布网站的开发过程。

本书以 Visual Studio 2005 和 SQL Server 2005 为开发平台，使用 C# 开发语言，提供大量源于作者多年教学积累和项目开发经验的实例。在学习本书中的项目前，读者需要掌握网页制作、数据库程序设计和软件工程等知识。

为了激发读者的学习兴趣，让读者快速掌握 ASP.NET 网站开发技术，本书以信息发布网站的开发过程为线索，从企业网站开发的角度出发逐步展开。以项目为驱动，使学生从一开始就带着项目开发任务进入学习，在做项目的过程中逐渐掌握完成任务所需的知识和技能；一步一步地解决问题，向成功靠近，每一个单项工作任务（子项目）的完成都会带来小小的成功喜悦，增加一点点自信，引发继续向上的动力。

本书概念清晰，逻辑性强，讲解循序渐进，语言通俗易懂，适合作为高等学校计算机相关专业的 Web 应用程序设计、Web 数据库应用等课程的教材，也适合从事 Web 应用程序开发的初级、中级人员学习参考。

由于本书涉及的范围比较广泛，加之项目教学在我国又是新生事物，开展的时间还不长，书中不足之处在所难免，敬请读者批评指正。

CONTENTS

目录

信息发布网站的项目导入	1
子项目 1 信息发布网站需求概述	2
1.1 信息发布网站的需求分析	2
1.2 信息发布网站的功能模块设计	2
1.3 信息发布网站的数据库设计	3
子项目 2 信息发布网站的三层架构	5
2.1 三层架构的概念	5
2.2 新闻信息显示与检索模块	6
2.3 新闻评论模块	19
2.4 后台用户管理模块	22
子项目 3 信息发布网站的创建	27
3.1 Visual Studio 2010 介绍	27
3.2 新建信息发布网站项目	32
子项目 4 信息发布网站的页面布局和设计	35
4.1 表格的设计	35
4.2 DIV + CSS 样式设计	36
4.3 母版页的设计	39
4.4 主题和皮肤的设计	41
4.5 信息发布网站母版页设计	42
4.6 信息发布网站主题应用	44
子项目 5 网站用户控件的创建	47
5.1 用户控件概述	48
5.2 UserControl 类	48
5.3 用户控件的属性和事件	48
5.4 在页面上使用用户控件	50
5.5 用户登录和显示模块	50
5.6 新闻统计模块	51
5.7 热门新闻和新闻搜索模块	52
5.8 后台功能控件	53
子项目 6 网站信息验证功能	54
6.1 验证控件的过程	55
6.2 验证对象的模型	55



6.3	ASP.NET 的验证类型	56
6.4	验证控件的对象模型	57
6.5	错误信息的布局与显示	57
6.6	使用验证控件	58
6.7	验证组	61
6.8	用户注册模块	62
6.9	用户修改信息模块	63
子项目 7	网站管理员登录功能	65
7.1	Web 应用的认证	65
7.2	Web 应用的授权	68
7.3	ASP.NET 登录控件	70
7.4	使用 Role 实现 Web 应用的授权	75
7.5	管理员登录模块	78
子项目 8	网站导航控件的应用	79
8.1	站点地图	79
8.2	导航控件	80
8.3	站点地图的嵌套使用	84
8.4	站点地图文件的实施	85
8.5	导航控件的应用	85
子项目 9	网站的数据访问应用	87
9.1	数据处理控件	88
9.2	数据源控件	91
9.3	数据绑定控件	92
9.4	ADO.NET 的基本概念	104
9.5	利用 SQL 语句的数据库访问操作	115
9.6	状态管理	116
9.7	网站数据访问项目实施	122
子项目 10	网站测试	133
10.1	系统测试的目的	133
10.2	系统测试的基本过程	134
10.3	网站测试实施	135
子项目 11	网站的发布实现	137
11.1	发布网站	137
11.2	Web 项目安装包	145
11.3	复制网站	146
11.4	信息发布网站的发布实施	148
11.5	网站的安装包	149
信息发布网站的项目总结		152
参考文献		153

信息发布网站的项目导入

当今社会是信息竞争的社会，企业信息化建设是提高企业管理效率的必要途径，在这样一个信息化建设中，企业的新闻发布网站是企业对外快速传播信息的门户。这个“门户”让拥有它的企业能够及时发布最新信息，让用户第一时间获取信息，以占有市场先机。谁拥有互联网，谁就拥有了信息；谁拥有了信息，谁就能占据有利竞争地位，这已经成为一条新的市场竞争规则。对于每一个企业来说，信息发布网站的存在是非常有必要的。

因此，采用信息发布网站进行 Web 程序开发，将 ASP.NET 中的所有概念和技术应用到信息发布网站的开发当中，按照软件工程的思想进行网站开发。本项目可以实现用户管理、新闻类别管理、新闻信息管理等功能，可以满足对用户、新闻类别和新闻信息的增删改查的操作要求。本项目的开发通过信息发布网站需求概述、信息发布网站的三层架构、信息发布网站的创建、信息发布网站的页面布局和设计、网站用户控件的创建、网站信息验证功能、网站管理员登录功能、网站导航控件的应用、网站的数据访问应用、网站测试、网站的发布实现等几个子项目来完成。

首先对信息发布网站进行需求分析。

子项目 1

信息发布网站需求概述

本项目任务

1. 信息发布网站的需求分析。
2. 信息发布网站的功能模块设计。
3. 信息发布网站的数据库设计。

任务指标

1. 设计信息发布网站的功能模块图和功能模块内容。
2. 设计信息发布网站的数据库 news2008。

1.1 信息发布网站的需求分析

来自全国各地的用户可以在 Internet 上匿名浏览新闻信息，但只有经过注册，具有会员资格的用户才能发布新闻。用户注册后，可以修改个人信息；按新闻栏目浏览新闻信息；搜索满足一定条件的新闻；对新闻进行评论；浏览到点击率最高的新闻，同时可以知道每条新闻评论的数目及每个新闻栏目新闻的数量。系统注册用户分为普通用户和管理员用户，普通用户具有上述权限；管理员用户可以对普通用户的资料进行添加与删除、创建与维护新闻内容及新闻评论等。

1.2 信息发布网站的功能模块设计

信息发布网站包括前台和后台两部分功能。前台功能主要包括用户注册、修改已注册用户信息、注册用户发布新闻、新闻搜索功能、新闻数量的统计、新闻评论、热点新闻统计及



浏览、按类别浏览新闻；后台功能主要包括现有新闻管理、发布新的新闻、新闻审核、新闻评论管理、新闻栏目管理、系统用户管理。新闻发布系统功能模块如图 1.1 所示。

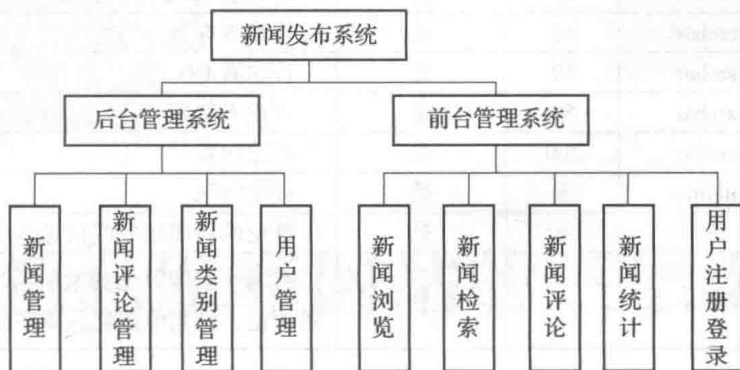


图 1.1 信息发布网站功能模块

1.3 信息发布网站的数据库设计

企业信息发布网站系统目前采用 SQL Server 2008 数据库系统。创建数据库名为 news2008，有 4 个表，分别是新闻信息表 (News)、新闻评论表 (Comments)、新闻类别表 (BigClass) 和用户信息表 (User)。

1. 新闻信息表 (News)

新闻信息表主要用来保存新闻的基本信息，如新闻名、发布时间等。新闻信息表的结构见表 1.1。

表 1.1 新闻信息表 (News)

字段名称	类型	大小/B	是否为空	描述
N_id	int	4	否	新闻 ID (自增主键)
Title	varchar	50	是	新闻标题
Info	text	16	是	新闻内容
Bigclassid	varchar	50	是	新闻分类 ID
Username	varchar	50	是	新闻编辑人姓名
Infotime	datetime	8	是	上传新闻时间
Hit	int	4	是	新闻点击率 (默认值: 0)
Flag	varchar	50	是	是否通过审核 (默认值: 未审核)
Cindex	int	4	是	该新闻的索引

2. 新闻评论表 (Comments)

新闻评论表主要用来保存新闻评论的相关信息。新闻评论表的结构见表 1.2。



表 1.2 新闻评论表 (Comments)

字段名称	类型	大小/B	是否为空	描述
C_id	int	4	否	评论者 ID (自增主键)
C_user	varchar	50	是	评论者姓名
C_qq	varchar	50	是	评论者 QQ
C_email	varchar	50	是	评论者邮箱
C_word	varchar	200	是	评论内容
C_time	datetime	8	是	评论时间
Newsid	int	4	是	评论的新闻在新闻信息表中的 N_id 值
Cindex	int	4	是	对同一新闻评论的索引值 (如同一新闻有两条评论, 则其值按先后顺序为 1、2)

3. 新闻类别表 (BigClass)

新闻类别表主要用来保存新闻类别的相关信息。新闻类别表的结构见表 1.3。

表 1.3 新闻类别表 (BigClass)

字段名称	类型	大小/B	是否为空	描述
B_id	int	4	否	新闻类别 ID (自增主键)
Name	varchar	50	是	新闻类别名称
Flag	char	10	是	是否显示分类标记 (默认值: 显示)
Cindex	int	4	是	新闻类别索引
Newscount	int	4	是	每类新闻对应的新闻总数 (默认值: 0)

4. 用户信息表 (User)

用户信息表主要用来保存用户的相关信息。用户信息表的结构见表 1.4。

表 1.4 用户信息表 (User)

字段名称	类型	大小/B	是否为空	描述
U_id	int	4	否	用户 ID (自增主键)
UserName	varchar	50	否	用户姓名
Password	varchar	50	否	用户密码
Email	varchar	50	是	用户邮箱
Lever	varchar	50	否	用户级别 (分为普通用户和管理员)

对信息发布网站的需求进行分析之后, 已经确定了该系统中所有实现的功能、数据库的设计内容, 并且已经明确系统将采用三层架构来实现。

子项目 2

信息发布网站的三层架构

本项目任务



1. 新闻信息显示与检索模块。
2. 新闻评论模块。
3. 后台用户管理模块。

任务指标



1. 创建 MODEL 类库、DAL 类库和 BLL 类库。
2. 在 MODEL 类库中创建 UserInfo.cs 类、CommentsInfo.cs 类、NewsInfo.cs 类和 BigClassInfo.cs 类。
3. 在 DAL 类库中创建 DBbase.cs 类、NewsAccess.cs 类、BigClassAccess.cs 类、CommentsAccess.cs 类、UserAccess.cs 类和 FormatString.cs 类。
4. 在 BLL 类库中创建 UserLogic.cs 类、NewsLogic.cs 类、BigClassLogic.cs 类和 CommentsLogic.cs 类。

对信息发布网站的需求进行分析之后，已经确定了所要实现的功能模块内容，可以直接在表现层编写代码，直接访问数据库。但是这种方式对程序的安全性和后续程序的扩展都是非常不利的。因此，采用三层架构的模式来开发信息发布网站。

2.1 三层架构的概念

三层架构是对程序设计结构的一个笼统的概念，可以从三层到七层不等。七层分别是 SqlServerDAL 数据访问层、MODEL 实体层、DAL 数据访问接口层、DALFactory 数据访问工厂层、DBUtility 数据访问抽象层、BLL 业务逻辑层、UI 界面层。

①SqlServerDAL 数据访问层是数据库或者数据源。在 .NET 中，它通常是一个 SQL Server



或 Access 数据库，但不限于这两种形式，还可能是 Oracle、MySQL，甚至是 XML。

②MODEL 层对数据字段进行操作。在 MODEL 层中，数据从 WEB UI 层写入 MODEL 层中，而 DAL 层从 MODEL 中读取。

③DAL 数据访问接口层对数据访问层方法进行规范。如果有几个不同的数据库，规定不同数据访问层的方法名相同，则利于调用。接口层对方法进行规范，它规范数据访问层的方法，即不同的数据库调用采用相同的方法。接口层可以通过接口的实例访问数据访问层，对不同的数据库操作起来比较方便。

④DALFactory 数据访问工厂层对数据库模型进行操作，运用反射，可以返回数据库实例，也就是定义要调用的是哪个数据库。工厂就是利用反射的方法将某个数据访问层返回，因此，在业务逻辑层中，接口的实例化的数据是根据工厂类返回的数据。工厂的作用，就是定义一个方法，这个方法定义返回的是哪个数据库的实例。

⑤DBUtility 数据访问抽象层从数据访问层中分离出来，让数据访问层定义 SQL 方法，而 DBUtility 层定义执行 SQL 语句的类，一般运用微软自带的类 SQLHelper.cs。

⑥BLL 业务逻辑层利用接口对象调用数据访问层的方法，在 ASP.NET 中，该层包括使用 SqlConnection 或 OleDb 从 SQL Server 或 Access 数据库获取数据、更新数据及删除数据，并把取得的数据放到 DataReader 或 DataSet 中返回给表现层。返回的数据也许只有一个整型数字，比如一个表的行记录数目，但这也要用数据层的数据进行计算。

⑦UI 界面层，也就是表现层，用于用户接口的展示，以及用业务层的类和对象来“驱动”这些接口。在 ASP.NET 中，该层包括 aspx 页面、用户控件、服务器控件及某些与安全相关的类和对象。

2.2 新闻信息显示与检索模块

新闻信息显示与检索是本系统的重要功能之一。本模块中包括新闻内容的显示与检索和新闻栏目的显示与检索两部分。

在本模块中，MODEL 类库中的类 NewsInfo、BigClassInfo 主要完成对数据库中的新闻内容表 News 和新闻栏目表 BigClass 中字段的定义。DAL 类库中的类 NewsAccess、BigClassAccess 主要是对新闻内容及类别操作的各种功能的具体实现。BLL 类库中的类 NewsLogic、BigClassLogic 则是对 DAL 类库中类的逻辑调用。可见，要完成新闻信息显示与检索，就要对此任务功能进行分析，完成各个类库中类的实现。

在本模块中，采用标准的三层架构，这三层架构是完成系统前后台功能的基础。

新闻内容的显示与检索需要完成实体层 MODEL 类中的 NewsInfo 类、数据访问接口层 DAL 类库中的 NewsAccess 类、业务逻辑层 BLL 类库中的 NewsLogic 类。其中，NewsInfo 类中定义的属性对应新闻内容表 News 中的字段；NewsAccess 类用于实现新闻内容显示与索引的基本方法；NewsLogic 类则用于完成对 NewsAccess 类的逻辑调用，从而实现新闻内容显示与检索的功能。

新闻栏目的显示与检索要完成实体层 MODEL 类库中的 BigClassInfo 类、数据访问接口层 DAL 类库中的 BigClassAccess 类、逻辑层 BLL 类库中的 BigClassLogic 类。其中，BigClassInfo



类中定义的属性对应新闻栏目表 BigClass 中的字段；BigClassAccess 类用于实现新闻栏目显示与索引的基本方法；BigClassLogic 类则用于完成对 BigClassAccess 类的逻辑调用，从而实现新闻类别显示与检索的功能。

在本模块的功能实现上，需要对数据库进行操作，因此，DAL 层的 DBbase 类是完成这些功能的必要前提；在系统中需要对字符串的长度进行控制，DAL 层的 FormatString 类正是用于完成此功能。

依次在解决方案下右击，选择“添加”→“新建项目”，在弹出的窗口中选择类库，并分别命名为 MODEL、DAL、BLL 类库。

在 DAL 类库名上单击鼠标右键，选择“添加引用”，在弹出的窗口中单击“项目”选项卡，然后选择项目名称“MODEL”，单击“确定”按钮。

在建立 BLL 类库时，需要添加对 MODEL 类库和 DAL 类库的引用。

1. MODEL 类库

在新闻信息显示与检索模块中，MODEL 类库中包含 NewsInfo.cs 类和 BigClassInfo.cs 类。

(1) NewsInfo.cs 类

NewsInfo.cs 类的访问修饰符应该设为“public”，这样才可以被其他层的类访问。定义 NewsInfo.cs 类的形式如下所示：

```
public class NewsInfo
{
}
```

NewsInfo 类中主要进行属性的设置。各个属性对应数据库 News 表中的相应字段。

主要程序代码包含 9 个内部变量：

设置新闻 ID 的内部变量为整型的 _n_id；新闻标题的内部变量为字符串类型的 _title；新闻内容的内部变量为字符串类型的 _info；栏目 ID 的内部变量为整型的 _BigClassID；发布者名称的内部变量为字符串类型的 _username；发布时间的内部变量为字符串类型的 _infotime；点击率的内部变量为整型的 _hit；判断审核是否通过的内部变量为整型的 _flag，其值为“1”和“0”，分别表示通过和没通过；最大索引的内部变量为整型的 _cindex，用来保存数值，通过这个最大索引可以获得一个整数，这个整数表示新闻总数。

定义 9 个公共属性，分别是 int 类型的 N_id、string 类型的 Title、string 类型的 Info、int 类型的 BigClassID、string 类型的 UserName、string 类型的 Infotime、int 类型的 Hit、int 类型的 Flag 和 int 类型的 Cindex。使用 get 访问器来返回所对应的内部变量的值，使用 set 访问器来设置所对应的内部变量的值。

(2) BigClassInfo.cs 类

BigClassInfo.cs 类的访问修饰符设为“public”，这样才可以被其他层的类访问。定义 BigClassInfo.cs 类的形式如下所示：

```
public class BigClassInfo
{
}
```



BigClassInfo.cs 类中主要进行属性的设置,各个属性对应数据库 BigClass 表中的相应字段。

BigClassInfo.cs 类包含 5 个内部变量:

设置新闻类别 ID 的内部变量为整型的 `_b_id`; 新闻栏目名称的内部变量为字符串类型的 `_name`; 显示分类标记的内部变量为字符串类型的 `_flag`; 最大索引的内部变量为整型的 `_cindex`, 是一个整数, 最大索引代表新闻类别的总数; 新闻总数的内部变量为整型的 `_newscount`, 表示每类新闻类别中所包含的新闻总数。

定义 5 个公共属性, 分别是 `int` 类型的 `B_id`、`string` 类型的 `Name`、`string` 类型的 `Flag`、`int` 类型的 `Cindex` 和 `int` 类型的 `Newscount`。使用 `get` 访问器来返回所对应的内部变量的值, 使用 `set` 访问器来设置所对应的内部变量的值。

2. DAL 层

在新闻信息显示与检索模块中, DAL 类库中包含 `DBbase.cs` 类、`NewsAccess.cs` 类和 `BigClassAccess.cs` 类。

(1) DBbase.cs 类

首先在 `Web.config` 文件中添加如下内容:

```
<appSettings >
  <add key = "conStr" value = "data source = .;integrated security =
true;database = news2008" />
</appSettings >
```

设置数据库连接的字符串为 `data source = .;integrated security = true;database = news2008`, 表示连接本地的 `news2008` 数据库, 并且访问的方式为 Windows 集成的访问方式。数据库连接的关键字为 `conStr`。

`DBbase.cs` 类的访问修饰符设为 “`public`”, 这样才可以被其他层的类访问。但是在定义 `DBbase.cs` 类之前, 需要引用命名空间, 如下所示:

```
using System.Data;
using System.Data.SqlClient;
```

`DBbase.cs` 类主要是实现数据库连接及对 SQL 命令的执行。定义 `DBbase.cs` 类的形式如下所示:

```
public class DBbase
{
}
```

首先在 `DBbase.cs` 类中定义连接字符串 `strCon`, 通过系统配置中的 `AppSettings["conStr"]` 来获取定义在 `Web.config` 文件中的 `conStr` 关键字的值。通过 `SqlConnection` 类创建 `con` 对象, 实例化连接对象。其代码如下所示:



```
public static string strCon = System.Configuration.Configuration-  
Settings.AppSettings["conStr"].ToString();  
SqlConnection con = new SqlConnection(strCon);
```

①自定义检测连接的方法 `CheckConnection()`，判断 `strCon` 连接是否已经打开，若连接是关闭的，则打开 `SqlConnection` 连接，通过调用 `Open()` 方法实现。

②自定义方法 `ReturnDataSet()`，方法的返回值类型为数据集 `DataSet`，并且方法在调用过程中包含一个 `string` 类型的参数 `strSQL`。首先调用自定义方法 `CheckConnection()` 打开数据库的连接，然后通过 `SqlDataAdapter` 类型的对象 `sda` 执行 SQL 语句，并且将获取的内容填充到数据集 `ds` 中，返回数据集 `ds`。如果在执行过程中出现错误，那么抛出异常语句，返回 `ex.Message` 的错误消息。执行之后将数据库连接关闭。

③自定义方法 `GetDataRow()`，方法的返回值类型为数据行 `DataRow`，并且方法在调用过程中包含一个 `string` 类型的参数 `strSQL`。首先调用自定义方法 `CheckConnection()` 打开数据库的连接，然后通过 `SqlDataAdapter` 类型的对象 `sda` 执行 SQL 语句，并且将获取的内容填充到数据集 `ds` 中，返回 `ds.Tables[0].Rows [0]` 数据行。如果在执行过程中出现错误，那么抛出异常语句，返回 `ex.Message` 的错误消息。执行之后将数据库连接关闭。

④自定义方法 `ExecuteNonQuery()`，方法的返回值类型为布尔类型 `bool`，并且方法在调用过程中包含两个参数：一个是 `bool` 类型的参数 `IsPro`，一个是 `string` 类型的参数 `strSQL`。首先调用自定义方法 `CheckConnection()` 打开数据库的连接，然后通过 `SqlCommand` 类型的对象 `com` 执行 SQL 语句，接着进行判断，如果参数 `IsPro` 为 `True`，那么 `com` 对象的命令类型为存储过程 `StoredProcedure`，否则，命令类型为文本 `Text`。将 SQL 语句赋值给 `CommandText`，通过调用对象 `com` 的 `ExecuteNonQuery()` 方法来执行 SQL 语句。执行之后将数据库连接关闭。如果执行成功，返回 `true`，否则，返回 `false`。

⑤自定义方法 `ExecuteNonQuery()`，方法的返回值类型为空，表示只在这个自定义方法中执行操作。方法在调用过程中只包含一个参数，是 `string` 类型的参数 `strSQL`。首先调用自定义方法 `CheckConnection()` 打开数据库的连接，然后通过 `SqlCommand` 类型的对象 `com` 执行 SQL 语句，通过调用对象 `com` 的 `ExecuteNonQuery()` 方法来执行 SQL 语句。如果在执行过程中出现错误，那么抛出异常语句，返回 `ex.Message` 的错误消息。执行之后将数据库连接关闭。

⑥自定义方法 `ReturnTable()`，方法的返回值类型为数据表 `DataTable`，并且方法在调用过程中包含一个 `string` 类型的参数 `strSQL`。首先调用自定义方法 `CheckConnection()` 打开数据库的连接，然后通过 `SqlDataAdapter` 类型的对象 `sda` 执行 SQL 语句，并且将获取的内容填充到数据集 `ds` 中，返回 `ds.Tables[0]` 数据表。如果在执行过程中出现错误，那么抛出异常语句，返回 `ex.Message` 的错误消息。执行之后将数据库连接关闭。

⑦自定义方法 `ReturnDataReader()`，方法的返回值类型为对象 `SqlDataReader`，方法在调用过程中只包含一个参数，是 `string` 类型的参数 `strSQL`。首先调用自定义方法 `CheckConnection()` 打开数据库的连接，然后通过 `SqlCommand` 类型的对象 `com` 执行 SQL 语句，创建 `SqlDataReader` 类型的对象 `myReader`，通过调用 `com` 对象的 `ExecuteReader()` 方法读取结果并赋值给 `myReader`，返回其值。如果在执行过程中出现错误，那么抛出异常语句，返回 `ex.Message` 的错误消息。



⑧自定义方法 `ReturnRowCount()`，方法的返回值类型为整型 `int`，并且方法在调用过程中包含一个 `string` 类型的参数 `strSQL`。首先调用自定义方法 `CheckConnection()` 打开数据库的连接，然后通过 `SqlDataAdapter` 类型的对象 `da` 执行 SQL 语句，并且将获取的内容填充到数据集 `ds` 中，返回 `ds.Tables[0].Rows.Count`，返回该 `strSQL` 语句查询出的数据行的总数。如果在执行过程中出现错误，那么返回的值为 0。执行之后将数据库连接关闭。

(2) NewsAccess.cs 类

类的访问修饰符设为“`public`”，这样才可以被其他层的类访问。但是在定义 `NewsAccess.cs` 类之前，需要引用命名空间，如下所示：

```
using System.Data;
using System.Data.SqlClient;
```

`NewsAccess.cs` 类主要通过构造 SQL 语句及调用 `DBbase` 类中的方法，实现关于新闻信息的显示与检索。定义 `NewsAccess.cs` 类的形式如下所示：

```
public class NewsAccess
{
}
```

在定义 `NewsAccess` 类中的方法之前，首先要实例化 `DBbase` 类的对象，用于调用其内部方法来执行不同的操作。其代码如下所示：

```
DBbase db = new DBbase();
```

①自定义方法 `GetNewID()`，返回的类型为数据集 `DataSet`。在方法中，定义 SQL 语句，从 `News` 数据表中查询 `Flag` 字段值为已审核的 `N_id` 字段，并且将这个 SQL 语句作为参数传递给 `DBbase.cs` 类中的 `ReturnDataSet()` 方法，获取现有新闻 ID。

②自定义方法 `GetBigClassIDByNewsID()`，返回的类型为数据集 `DataSet`。在方法中，定义 SQL 语句，从 `News` 数据表中查询 `N_id` 字段值为现有新闻的 `BigClassID` 字段，并且将这个 SQL 语句作为参数传递给 `DBbase.cs` 类中的 `ReturnDataSet()` 方法，获取现有新闻 ID 的类型 ID。

③自定义方法 `GetData_news()`，返回的类型为数据集 `DataSet`。在方法中，定义 SQL 语句，从 `News` 数据表和 `BigClass` 数据表中查询 `News.Flag` 字段值为已审核，并且 `News.BigClassID` 字段值为 `BigClass.B_id` 字段值同时满足时的 `News` 表中的所有字段值和 `BigClass` 表中的 `Name` 字段值。将这个 SQL 语句作为参数传递给 `DBbase.cs` 类中的 `ReturnDataSet()` 方法，获取全部新闻内容。

④自定义方法 `GetData_news()`，返回的类型为数据集 `DataSet`。方法在调用过程中有一个参数是整型的 `BigClassID`。在方法中，定义 SQL 语句，从 `News` 数据表和 `BigClass` 数据表中查询 `News.Flag` 字段值为已审核，`News.BigClassID` 字段值为参数 `BigClassID` 值，并且 `News.BigClassID` 字段值为 `BigClass.B_id` 字段值同时满足时的 `News` 表中的所有字段值和 `BigClass` 表中的 `Name` 字段值。将这个 SQL 语句作为参数传递给 `DBbase.cs` 类中的 `ReturnDataSet()` 方法，获取全部新闻内容。

⑤自定义方法 `GetDataByBigClass()`，返回的类型为数据集 `DataSet`。方法在调用过程中



有一个参数是整型的 `BigClassID`。在方法中，定义 SQL 语句，从 `News` 数据表中查询 `Flag` 字段值为已审核，并且 `BigClassID` 字段值为参数 `BigClassID` 值。满足这两个条件的 `News` 表中的所有字段值按照 `Cindex` 索引值进行降序排序。将这个 SQL 语句作为参数传递给 `DBbase.cs` 类中的 `ReturnDataSet()` 方法，分栏目获取新闻列表。

⑥自定义方法 `GetDataByBigClassTopN()`，返回的类型为数据集 `DataSet`。方法在调用过程中有两个参数：一个是整型的 `BigClassID`，另一个是整型的变量 `n`。在方法中，定义 SQL 语句，从 `News` 数据表中查询 `BigClassID` 字段值为参数 `BigClassID` 值，并且 `Flag` 字段值为已审核。满足这两个条件的 `News` 表中前 `n` 条记录的所有字段值按照 `Cindex` 索引值进行降序排序。将这个 SQL 语句作为参数传递给 `DBbase.cs` 类中的 `ReturnDataSet()` 方法，获取每个分栏目的前 `n` 条新闻。

⑦自定义方法 `GetDataNewN()`，返回的类型为数据集 `DataSet`。方法在调用过程中有一个参数是整型的变量 `n`。在方法中，定义 SQL 语句，从 `News` 数据表中查询 `Flag` 字段值为已审核，满足这个条件的 `News` 表中前 `n` 条记录的 `N_id`、`Title`、`InfoTime` 和 `Hit` 字段值按照 `Cindex` 索引值进行降序排序。将这个 SQL 语句作为参数传递给 `DBbase.cs` 类中的 `ReturnDataSet()` 方法，获取最新 `n` 条新闻。

⑧自定义方法 `GetDataTopNHits()`，返回的类型为数据集 `DataSet`。方法在调用过程中有一个参数是整型的变量 `n`。在方法中，定义 SQL 语句，从 `News` 数据表中查询 `Flag` 字段值为已审核，满足这个条件的 `News` 表中前 `n` 条记录的 `N_id`、`Title` 和 `Hit` 字段值按照 `Hit` 点击率进行降序排序。将这个 SQL 语句作为参数传递给 `DBbase.cs` 类中的 `ReturnDataSet()` 方法，获取点击率最高的 `n` 条新闻标题。

⑨自定义方法 `GetMaxCindex()`，返回的类型为整型。在该方法中，定义一个 SQL 语句，查询 `News` 表中的所有字段值，并将这个 SQL 语句作为参数传递给 `DBbase.cs` 类中的 `ReturnDataSet()` 方法，该方法返回的是一个 `DataSet` 数据集，获取这个数据集的数据表的行的个数。如果大于 0，那么定义另一个 SQL 语句。从 `News` 数据表中查询 `Cindex` 最大的字段 `cindex`，将这个 SQL 语句作为参数传递给 `DBbase.cs` 类中的 `ReturnDataSet()` 方法，返回的是一个数据集。获取这个数据集的数据表的第一行第一列的字段值，并转换成整型变量，返回这个整型变量值。

⑩自定义方法 `AddNews()`，方法的返回值类型为布尔类型 `bool`。方法中有一个参数，是 `MODEL` 实体层的 `NewsInfo` 类的对象。在该方法中，首先获取系统当前的时间，并且添加一条新闻时，要调用 `GetMaxCindex()` 方法获取新闻总数的索引值，并执行加 1 的操作。使用 SQL 语句的 `insert` 语法，添加 `M_news.Title`、`M_news.Info`、`M_news.BigClassID`、`M_news.UserName`、`times` 和 `cindex` 变量到 `News` 表中。将 SQL 语句作为参数传递给 `DBbase.cs` 类的 `ExecuteNonQuery()` 方法，添加成功，返回 `true`，否则，返回 `false`。

⑪自定义方法 `UpdateNews()`，方法的返回值类型为布尔类型 `bool`。方法中有一个参数，是 `MODEL` 实体层的 `NewsInfo` 类的对象。在该方法中，使用 SQL 语句的 `update` 语法，修改 `News` 表中字段值分别为 `M_news.Title`、`M_news.BigClassID`、`M_news.Info` 和 `M_news.UserName`，条件是 `News` 表中的 `N_id` 字段的值为 `M_news.N_id`。将 `false` 和 SQL 语句作为参数传递给 `DBbase.cs` 类的 `ExecuteNonQuery()` 方法，修改指定的新闻，修改成功，返回 `true`，否则，返回 `false`。