



21 世纪高职高专规划教材
计算机应用系列

ASP.NET 动态网站设计

王淑敏 主编

马世霞 刘 丹 白艳玲 副主编



清华大学出版社

北 京

内 容 简 介

本书通过讲解一个完整的项目“在线考试系统”的设计与实现,详细介绍了 ASP.NET 项目开发的过程,并将 ASP.NET 所有的知识和技能穿插其中。本书共分 15 个任务进行讲解,内容包括 ASP.NET 概述及运行环境的构建,C# 语言基础与实体类创建,为在线考试系统制作导航系统,三层架构与系统框架,ADO.NET 数据库操作与数据访问层类的创建,验证控件与用户登录,ASP.NET 内置对象与登录页面完善,GridView 控件、DetailsView 控件与考生信息显示,主题与母版页,DataList 控件与试题信息管理,第三方控件与试题的信息添加,Repeater 控件与前台试题显示,用户控件与网站版权,文件及文件夹操作,网站部署与定制。

本书适用于各类高等职业院校计算机技术专业,也可作为计算机培训班的教材,还可供从事网页设计的技术人员学习参考。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

ASP.NET 动态网站设计/王淑敏主编. —北京:清华大学出版社,2010.7

(21 世纪高职高专规划教材. 计算机应用系列)

ISBN 978-7-302-22873-8

I. ①A… II. ①王… III. ①主页制作—程序设计—高等学校:技术学校—教材

IV. ①TP393.092

中国版本图书馆 CIP 数据核字(2010)第 099195 号

责任编辑:张龙卿(sdzlq123@163.com)

责任校对:袁 芳

责任印制:杨 艳

出版发行:清华大学出版社

<http://www.tup.com.cn>

社 总 机:010-62770175

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

地 址:北京清华大学学研大厦 A 座

邮 编:100084

邮 购:010-62786544

印 刷 者:北京密云胶印厂

装 订 者:三河市兴旺装订有限公司

经 销:全国新华书店

开 本:185×260

印 张:18.5

字 数:445 千字

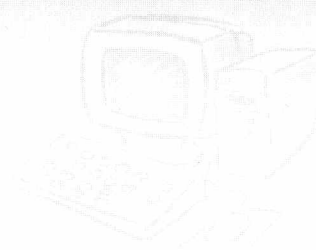
版 次:2010 年 7 月第 1 版

印 次:2010 年 7 月第 1 次印刷

印 数:1~3000

定 价:29.50 元

产品编号:034787-01



前言

由于微软软件开发工具产品的易学易用,ASP.NET 在当今动态网页与网站设计技术中占有主导地位。根据教育部有关教育教学改革精神,结合当前高职高专教育的特点,编者在总结多年教学经验的基础上,编写了本书。

本书以一个完整的项目“在线考试系统”为导向,以任务为基本单元,将知识和技能穿插其中,讲解了 ASP.NET 技术。在确定开发项目时一方面考虑实用性;另一方面考虑到理论知识的系统性,并采用大家感兴趣、熟悉且可以操作的实际项目。

本书具有以下特点。

(1) 采用项目驱动式教材编写体系。本书将一个大的项目案例的实现划分为若干任务,分别讲解,在完成每个具体任务时,会将相关的知识点进行透彻讲解。

(2) 内容新颖,注重应用。尽管现在市场上 ASP.NET 方面的书已经有不少,但能满足教学需要的却很难找。针对这种状况,我们在编写时特别注意以下三点:一是注意知识点和工具应用的紧密结合;二是注意知识体系的更新,整个项目采用目前流行的三层架构;三是注意实用性,不但采用真实项目,代码编写也比较规范,便于学生将来快速适应工作环境。

(3) 理论知识系统性比较强。本书大部分任务都包括知识准备和任务实施,任务实施注重动手能力的培养,并逐步完成在线考试系统的制作;知识准备部分为当前任务的实施奠定了理论基础,并保证学习过程中理论知识讲解的系统性。

本书由王淑敏任主编,马世霞、刘丹、白艳玲任副主编。具体编写工作分工如下:任务 1、任务 12 由王淑敏编写,任务 3、任务 4、任务 14 由马世霞编写,任务 6、任务 8、任务 15 由刘丹编写,任务 9、任务 10 由白艳玲编写,任务 5、任务 7 由邵伟民编写,任务 11、任务 13 由胡海鹤编写,绪论部分的项目介绍、任务 2 由白林如编写。在本书的编写过程中,得到了清华大学出版社的大力支持,在此表示衷心的感谢。

虽然我们在编写过程中倾注了大量心血,但书中难免有疏漏之处,恳请广大读者批评指正。

编 者

2010 年 5 月

目 录

绪论 案例说明——在线考试系统介绍	1
0.1 系统分析和开发环境	1
0.2 系统功能模块设计	1
0.3 系统运行界面	4
0.4 Web.config 文件和数据库操作公共类	6
0.4.1 Web.config 文件	6
0.4.2 数据库操作的公共类	6
0.5 主要模块设计	8
0.5.1 考生登录页面	8
0.5.2 考生考试页面	9
0.5.3 考试成绩显示页面	10
0.5.4 考试系统后台管理登录页面	11
0.5.5 考试系统后台学生信息管理页面	12
任务1 ASP.NET 概述及运行环境的构建	14
1.1 知识准备	14
1.1.1 实现动态站点的关键技术	14
1.1.2 ASP.NET 介绍	16
1.2 任务实施	17
1.2.1 安装 Visual Studio.NET 集成开发环境	17
1.2.2 第一个 ASP.NET 程序	21
1.2.3 搭建 ASP.NET 的运行环境	25
练习	30
实训	30
任务2 C# 语言基础与实体类创建	32
2.1 知识准备	32
2.1.1 C# 语言基础	32
2.1.2 C# 语言中的变量和运算符	35
2.1.3 C# 语言中的 Console 类	39



2.1.4	C# 语言中的控制语句	40
2.1.5	数组	46
2.1.6	C# 语言面向对象程序设计	48
2.1.7	常用的类和函数	54
2.2	任务实施	58
2.2.1	为在线考试系统创建实体类	58
2.2.2	在线考试系统倒计时的实现	59
练习		61
实训		62
任务 3	为在线考试系统制作导航系统	64
3.1	知识准备	64
3.1.1	XML 文件	64
3.1.2	TreeView 控件	66
3.1.3	站点地图	66
3.1.4	SiteMapPath 控件	67
3.2	任务实施	67
3.2.1	为在线考试系统制作树型目录	67
3.2.2	为在线考试系统制作站点导航	72
3.3	知识和技能扩展——Menu 控件与网站菜单	73
3.3.1	网站菜单	73
3.3.2	Menu 控件	74
练习		76
实训		76
任务 4	三层架构与系统框架	79
4.1	知识准备	79
4.1.1	三层架构介绍	79
4.1.2	在线考试系统的系统结构	80
4.2	任务实施	81
4.2.1	用三层架构搭建“在线考试系统”系统框架	81
4.2.2	为在线考试系统创建模型层	83
练习		85
实训		86
任务 5	ADO.NET 数据库操作与数据访问层类的创建	89
5.1	知识准备	89
5.1.1	ADO.NET 简介	89
5.1.2	Connection 对象	91
5.1.3	Command 对象	94



5.1.4	DataReader 对象	95
5.1.5	DataSet 对象	97
5.1.6	DataAdapter 对象	98
5.2	任务实施	101
5.2.1	为在线考试系统数据访问层创建 DBHelper 类	101
5.2.2	创建 StudentService.cs 类	104
练习		109
实训		110
任务 6	验证控件与用户登录	112
6.1	知识准备	112
6.1.1	ASP.NET 控件基础	112
6.1.2	HTML 控件	113
6.1.3	标准服务器控件	120
6.1.4	验证控件基础	131
6.2	任务实施	133
6.2.1	创建管理员用户登录页面	133
6.2.2	用户登录的实现	135
6.2.3	使用验证控件完善管理员登录功能	137
练习		138
实训		138
任务 7	ASP.NET 内置对象与登录页面完善	140
7.1	知识准备	140
7.1.1	ASP.NET 内置对象概述	140
7.1.2	Page 对象	141
7.1.3	Request 对象	142
7.1.4	Response 对象	143
7.1.5	Application 对象	144
7.1.6	Session 对象	146
7.1.7	Server 对象	149
7.1.8	Cookie 对象	151
7.2	任务实施	153
7.2.1	登录页面的完善	153
7.2.2	网页浏览计数器	155
练习		156
实训		157
任务 8	GridView 控件、DetailsView 控件与考生信息显示	159
8.1	知识准备	159



8.1.1	数据源控件	159
8.1.2	数据绑定控件	160
8.1.3	数据绑定方法	167
8.2	任务实施	167
8.2.1	在后台管理中显示考生信息	167
8.2.2	考生详细信息显示	172
	练习	177
	实训	178
任务 9	主题与母版页	179
9.1	知识准备	179
9.1.1	主题概述	179
9.1.2	母版页概述	181
9.1.3	母版页的制作	182
9.2	任务实施	183
9.2.1	添加在线考试系统主题	183
9.2.2	母版页的套用	186
9.2.3	为在线考试系统后台管理页面制作母版	187
	练习	191
	实训	191
任务 10	DataList 控件与试题信息管理	192
10.1	知识准备	192
10.1.1	DataList 控件	192
10.1.2	DataList 控件的分页和排序	194
10.2	任务实施	197
10.2.1	使用 DataList 控件分页显示试题信息	197
10.2.2	删除试题信息	201
10.2.3	对试题信息排序	203
	练习	204
	实训	205
任务 11	第三方控件与试题的信息添加	209
11.1	知识准备	209
11.1.1	第三方控件介绍	209
11.1.2	FreeTextBox 控件	209
11.2	任务实施	211
	练习	221
	实训	222



任务 12 Repeater 控件与前台试题显示	224
12.1 知识准备	224
12.1.1 Repeater 控件	224
12.1.2 Repeater 控件使用举例	225
12.2 任务实施	228
12.2.1 考试之前的准备工作	228
12.2.2 使用 Repeater 控件显示试题信息	231
12.2.3 保存考生试题信息	236
12.2.4 考生成绩统计	240
练习	241
实训	241
任务 13 用户控件与网站版权	244
13.1 知识准备	244
13.2 任务实施	246
13.2.1 使用用户控件创建网站版权信息	246
13.2.2 在模板页中使用用户控件	247
练习	248
实训	248
任务 14 文件及文件夹操作	250
14.1 文件	250
14.1.1 文件概述	250
14.1.2 文件的上传和下载	250
14.2 文件和目录操作	255
14.2.1 文件操作	255
14.2.2 目录操作	259
14.3 XML 文件的操作	261
14.3.1 XML 文件的写入	261
14.3.2 XML 文件的读取	262
14.3.3 XML 文件的显示和验证	264
练习	264
实训	265
任务 15 网站部署与定制	266
15.1 知识准备	266
15.1.1 ASP.NET 中的配置文件概述	266
15.1.2 ASP.NET 中的配置文件类型	267
15.1.3 ASP.NET 中的配置文件结构	267



15.1.4	网站管理工具.....	271
15.2	任务实施.....	276
15.2.1	数据库连接配置.....	276
15.2.2	身份验证配置.....	278
15.2.3	自定义错误.....	279
15.2.4	sessionState 配置	280
练习		283
实训		283
参考文献		284

绪论 案例说明——在线考试系统介绍

本部分首先介绍一下本书中后面内容使用到的案例——在线考试系统,以便大家对案例的整体结构有所了解。

0.1 系统分析和开发环境

在线考试系统是学校针对学生的考试系统。学校可以通过该考试系统建立自己的网上考场,使学生直接进行网上考试,既减少了成本,又避免了笔试的烦琐过程,使考试过程变得轻松、方便;同时还可以有效地控制考试的作弊现象,确保考试的公开、公平、公正性。

该在线考试系统采用三层设计模式,分别为用户界面表示层、业务逻辑层、数据访问层,另外还有一个模型层,如图 0-1 所示。

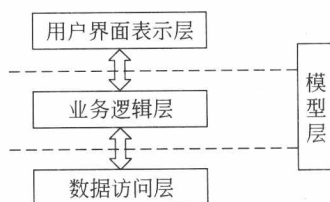


图 0-1 三层开发模式及模型层

本系统开发工具: Visual Studio 2005 和 SQL Server 2005。

0.2 系统功能模块设计

系统开发的总体任务是实现信息关系的系统化、规范化和自动化。

系统的功能模块如图 0-2 所示。

系统采用 SQL Server 2005 数据库,包含系统管理员表、题库表、科目表、试卷表、考生试卷表、试题类型表、班级表、学生表、成绩表等十余个表,如图 0-3~图 0-12 所示。

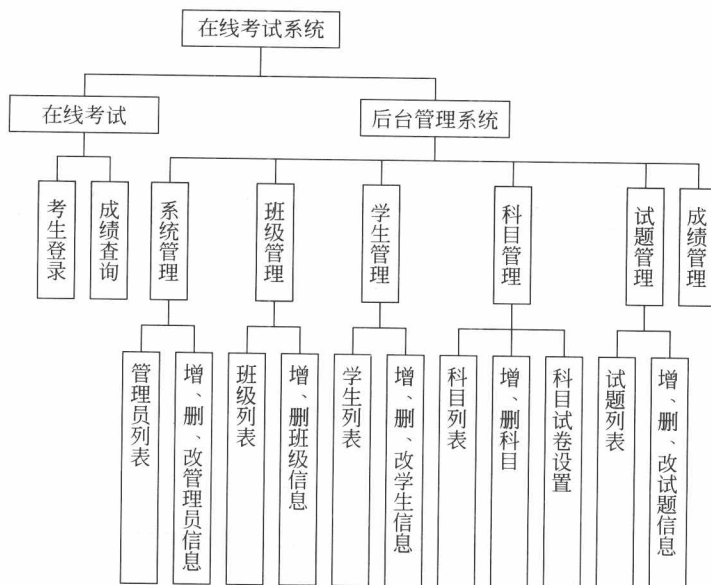


图 0-2 系统功能模块

表 - dbo.exam_admin		
列名	数据类型	允许空
admin_id	int	☐
admin_name	varchar(20)	☐
admin_pass	varchar(20)	☐
mgquestions	int	☒
mgStudents	int	☒
mgSystem	int	☒

图 0-3 系统管理员表

表 - dbo.exam_allQuestions		
列名	数据类型	允许空
question_id	int	☐
subject_id	int	☐
question_type	int	☐
question_subject	varchar(200)	☐
question_keys	varchar(10)	☐
a	varchar(100)	☒
b	varchar(100)	☒
c	varchar(100)	☒
d	varchar(100)	☒

图 0-4 题库表

表 - dbo.exam_allsubject		
列名	数据类型	允许空
subject_id	int	☐
subject_name	varchar(50)	☐

图 0-5 科目表



表 - dbo.exam_canTry		
列名	数据类型	允许空
stu_id	varchar(20)	☐
paperdb_name	varchar(50)	☒

图 0-6 试卷表(一)

表 - dbo.exam_papermg		
列名	数据类型	允许空
paper_id	int	☐
subject_id	int	☐
paper_name	varchar(50)	☐
test_time	int	☐
type1_num	int	☐
type2_num	int	☐
type3_num	int	☐
score1	int	☒
score2	int	☒
score3	int	☒

图 0-7 试卷表(二)

表 - dbo.exam_papers		
列名	数据类型	允许空
question_id	int	☐
subject_id	int	☐
question_type	int	☐
question_subject	varchar(200)	☐
question_key	varchar(10)	☐
a	varchar(100)	☒
b	varchar(100)	☒
c	varchar(100)	☒
d	varchar(100)	☒
user_key	varchar(50)	☒

图 0-8 考生试卷表

表 - dbo.exam_questionType		
列名	数据类型	允许空
type_id	int	☐
type_name	char(10)	☐

图 0-9 试题类型表

表 - dbo.exam_stuClass		
列名	数据类型	允许空
class_id	int	☐
class_name	varchar(50)	☐

图 0-10 班级表

表 - dbo.exam_students		
列名	数据类型	允许空
stu_id	varchar(10)	☐
stu_name	varchar(20)	☐
class_id	int	☐
isLogin	int	☐
isSubmit	int	☐
stu_pwd	varchar(20)	☐

图 0-11 学生表



表 - dbo.exam_stuScore		
列名	数据类型	允许空
score_id	int	☐
stu_id	varchar(10)	☐
subject_id	int	☐
paperdb_name	varchar(50)	☐
try_date	varchar(50)	☒
stu_score	int	☐

图 0-12 成绩表

0.3 系统运行界面

在线考试系统的操作流程如下：

首先,进入在线考试系统首页面,如图 0-13 所示。管理员可以选择“进入后台管理”选项对系统进行维护操作,如图 0-14 和图 0-15 所示。考生选择“开始考试”选项后,进入考试登录页面,如图 0-16 所示。考生选择考试科目并输入学号和密码,单击“开始考试”按钮,浏览器转到考试页面,考生可在线答题,如图 0-17 所示。

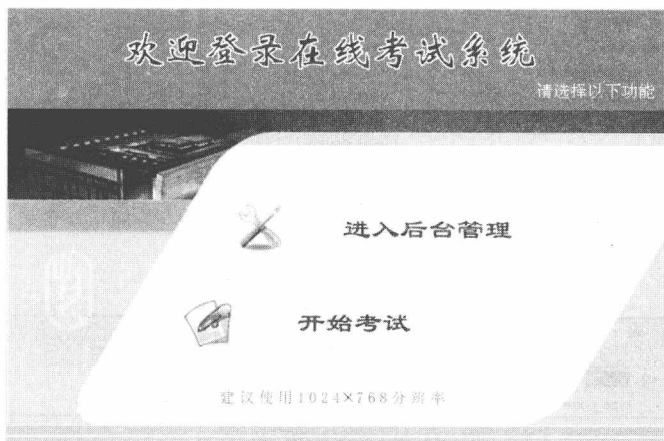


图 0-13 在线考试系统首页

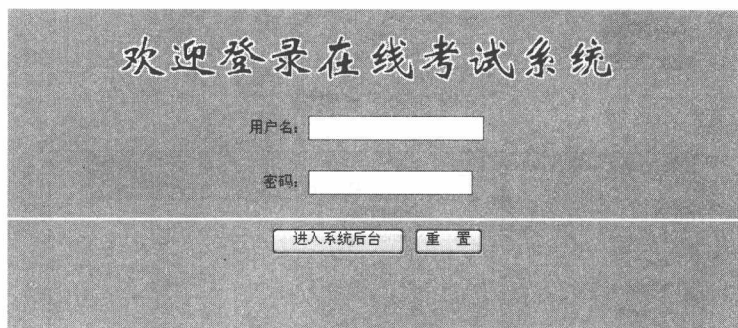


图 0-14 后台管理登录页面



后台管理

管理员控制面板

管理员信息

管理员列表

添加管理员

班级管理

班级列表

添加班级

学生管理

学生列表

添加学生

科目管理

试题管理

成绩管理

退出

您的位置: 首页 > 管理员后台 > 学生管理 > 学生列表

学号	学生姓名	班级	密码	详细	删除
0000001	白嘉辉	07级1班	0	详细	删除
0000002	武超帅	07级1班	0	详细	删除
0000003	杨光宇	07级1班	0	详细	删除
0000004	张西良	07级2班	3	详细	删除
0000005	朱彬彬	07级1班	0	详细	删除
0000006	朱志伟	07级1班	0	详细	删除
0000007	赵磊	07级1班	0	详细	删除
0000008	戴亮	07级1班	0	详细	删除
0000009	张鹏飞	07级2班	0	详细	删除
0000010	邢鑫	07级1班	0	详细	删除

1 2 3 4

图 0-15 后台管理

欢迎登录在线考试系统

请选择考试科目: <-- 请选择 -->

学号:

密码:

图 0-16 学生登录

《计算机基础》考试试卷

你是第8位登录的考生

一、单项选择题

1. 微型计算机(PC机)的文件系统采用的是_____结构。

☐ 环型

☐ 星型

☐ 树型

☐ 网型

2. 为解决某一特定问题而设计的指令序列称为_____。

☐ 文档

☐ 语言

☐ 程序

☐ 系统

3. 计算机病毒是一种_____。

☐ 特殊的计算机部件

图 0-17 考试页面

答题完毕后,单击“提交”按钮,系统计算考试成绩后,浏览器将被定向到成绩显示页面,如图 0-18 所示。在图 0-16 所示页面中,输入学号和密码后单击“查看已考科目成绩”按钮,



也可进入成绩显示页面(图 0-18),显示该考生的已考科目成绩。

图 0-18 成绩显示

0.4 Web.config 文件和数据库操作公共类

0.4.1 Web.config 文件

为了提高程序的灵活性和可维护性,我们把数据库连接信息存放在 Web.config 文件中。在用户界面表示层的根目录下打开 Web.config 文件,在<configuration>配置代码块中添加<connectionStrings>代码块内容,如下所示。

```
<configuration>
  <appSettings/>
  <connectionStrings>
    <add name="onLineExam" connectionString="Data Source=.;Initial Catalog=
      onLineExam1;User ID=sa;password=sa"/>
  </connectionStrings>
  <system.Web>
    ...
    <authentication mode="Forms">
      <forms name="Login" loginUrl="~/admin/AdminLogin.aspx" timeout="60"/>
    </authentication>
  </system.Web>
</configuration>
```

0.4.2 数据库操作的公共类

从软件工程的角度考虑,代码的编写要尽可能地实现重用,或者说,同样的代码要避免写两次以上。为了减少数据库操作代码的重复,将对数据库操作的共同部分提取出来,封装到一个类中(ConnDBHelper.cs)。可以通过调用类中的方法,轻松地实现数据库的操作,并把编程的精力集中在实现应用的逻辑上。该类文件存放在三层结构中的数据访问层,namespace OnLineDAL 语句把该类的命名空间定义为 OnLineDAL,与数据访问层的命名空间一致,该层中每个使用该类的文件都可引用。

类 ConnDBHelper.cs 的相关代码如下:



```
using System;
using System.Collections.Generic;
using System.Text;
using System.Data;
using System.Data.SqlClient;
namespace OnLineDAL
{
    public class ConnDBHelper
    {
        private static SqlConnection connection;
        public static SqlConnection Connection
        {
            get
            {
                //直接读数据库连接信息
                //string connectionString="Data Source=.;Initial Catalog=onLineExam1;
                //User ID=sa;password=sa";
                //从 Web.config 配置文件中读入数据库连接信息
                string connectionString=ConfigurationManager.ConnectionStrings
                ["onLineExam"].ConnectionString;           //配置文件连接方式
                if (connection==null)
                {
                    connection=new SqlConnection(connectionString);
                    connection.Open();
                }
                else if (connection.State==System.Data.ConnectionState.Closed)
                {
                    connection.Open();
                }
                else if (connection.State==System.Data.ConnectionState.Broken)
                {
                    connection.Close();
                    connection.Open();
                }
                return connection;
            }
        }
        public static int ExecuteCommand(string safeSql)
        {
            SqlCommand cmd=new SqlCommand(safeSql, Connection);
            int result=cmd.ExecuteNonQuery();
            return result;
        }

        public static int ExecuteCommand(string sql, params SqlParameter[] values)
        {
            SqlCommand cmd=new SqlCommand(sql, Connection);
            cmd.Parameters.AddRange(values);
            return cmd.ExecuteNonQuery();
        }
    }
}
```