

.NET



.NET软件工程师就业培训系列教程

C# Web 2.0 应用程序设计

■ 杨晓光 编著

.NET SOFTWARE ENGINEER

定位明确，强调程序设计技能的重要性

内容完备，实例丰富，紧密联系实战

强调程序外的项目开发方法和软件工程概念



清华大学出版社



北京交通大学出版社



TP312/3630

2010

.NET 软件工程师就业培训系列教程

C# Web 2.0 应用程序设计

杨晓光 编著

清华大学出版社
北京交通大学出版社

· 北京 ·

内 容 简 介

本书系统介绍了利用 Visual Studio. NET 2008 和 ASP. NET 3.5 开发 Web 应用程序的基本知识,并对通讯录实例给出两种不同的开发方法——快速开发方法和基于三层架构的开发方法,使读者对网站开发有深入理解。

全书共 14 章,主要涵盖了 Web 2.0 和 ASP. NET 基础知识,包括利用常用控件构建 Web 页面、利用主题和母版页定制网页样式、利用站点地图和导航控件构建网站导航系统、利用 Profile 和 Web 部件构建个性化站点、利用 ASP. NET AJAX 给用户以更好体验,以及如何操作和展示数据库中的数据。

本书讲解突出重点,结合实战,精选实用例程,摒弃了控件说明书模式。本书可作为高等院校教材,亦适合培训机构作为教学参考。

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 13501256678 13801310933

图书在版编目(CIP)数据

C# Web 2.0 应用程序设计教程/杨晓光编著. —北京:北京交通大学出版社,2009.12
(.NET 软件工程师就业培训系列教程)

ISBN 978-7-81123-897-6

I. ① C… II. ① 杨… III. ① C 语言-程序设计-教材 IV. ① TP312

中国版本图书馆 CIP 数据核字(2009)第 202995 号

责任编辑:谭文芳 特邀编辑:李晓敏

出版发行:清华大学出版社 邮编:100084 电话:010-62776969

北京交通大学出版社 邮编:100044 电话:010-51686414

印刷者:北京东光印刷厂

经 销:全国新华书店

开 本:185×260 印张:19.5 字数:499 千字

版 次:2010 年 2 月第 1 版 2010 年 2 月第 1 次印刷

书 号:ISBN 978-7-81123-897-6/TP·544

印 数:1~4 000 册 定价:31.00 元

本书如有质量问题,请向北京交通大学出版社质监组反映。对您的意见和批评,我们表示欢迎和感谢。

投诉电话:010-51686043, 51686008; 传真:010-62225406; E-mail: press@bjtu.edu.cn。

前 言

ASP.NET 引入了大量新特性、新功能,利用它可以轻松地开发出功能强大的 Web 应用程序。如微软网站 (www.microsoft.com)、戴尔网站 (www.dell.com)、当当网 (www.dangdang.com) 等大型网站都是采用 ASP.NET 开发的。ASP.NET 的宗旨就是化繁为简,降低 Web 开发的难度,减少代码量,提高开发效率。然而这一切是以其自身的复杂性为代价的,本书力求以通俗易懂的语言,简明而实用的例程帮助读者化解这种复杂性,降低学习难度。

全书共 14 章。第 1 章概要介绍了 Web 2.0 和 ASP.NET 基础知识,并以一个例子介绍如何利用 Visual Studio.NET 2008 开发 ASP.NET Web 应用程序;第 2 章介绍 ASP.NET 的工作原理,主要包括页面模型和事件处理,也给出了一些常用控件的用法;第 3 章介绍如何在页面中加入各种验证功能;第 4 章介绍如何应用样式和主题设计网页外观,以及利用母版页定制整个网站的页面结构;第 5 章介绍利用站点地图和导航控件构建网站导航系统;第 6 章给出状态维护的各种方法;第 7 章针对数据操作,介绍如何利用程序和控件操作及展示数据库中的数据;第 8 章介绍网站安全和用户管理的知识;第 9 章介绍如何开发用户控件和自定义服务器控件;第 10 章介绍如何用 Profile 存储用户个性化信息;第 11 章介绍利用 Web 部件构建 Web 页面,使用户可以自定义页面内容;第 12 章介绍 ASP.NET Ajax 方面内容,展示如何利用 Ajax 给用户以更好的体验;第 13 章和第 14 章给出通讯录综合实例的两种开发方法,分别介绍如何用快速开发理念和三层架构思想开发 Web 应用程序,使读者真正领会各种开发方法的魅力。

为了帮助读者理解和领会 ASP.NET 的知识点,每章都提供许多由短小精简代码组成的例程。同时,为了便于大家练习,这些例程均按页面外观和编程逻辑组织而成,这也符合 ASP.NET 的设计理念。这些精选的例程又与实际紧密相连,稍加修改即可用于实际项目。在一些重要章节,又给出了一些综合实例,以帮助读者综合运用本章知识点、乃至前几章知识点解决实际问题。最后,利用两章的篇幅给出通讯录实例的两种开发方式,详述快速开发和基于三层架构的开发步骤。这些例程的源代码可通过“<http://press.bjtu.edu.cn>”网站下载或与出版社编辑联系 (cbstwf@jg.bjtu.edu.cn)。

本书主要由杨晓光编写,郑志荣、郭文平、马延宏、傅岚岚、杨晓君、杨亚红等也参与编写和源代码调试,在此一并表示感谢。本书在编写过程中,作者力求精益求精,但难免出现一些疏漏,希望读者批评指正。

本书可作为高等院校教材,亦适合培训机构作为教学参考。

编 者
2009 年 12 月

目 录

第 1 章 C# Web 2.0 技术概述	1
1.1 Web 2.0 技术概述	1
1.2 Web 开发技术概述	2
1.2.1 ASP 技术	2
1.2.2 JSP 技术	2
1.2.3 PHP 技术	3
1.2.4 ASP.NET 技术	3
1.3 ASP.NET 技术概述	3
1.3.1 .NET 框架	3
1.3.2 公共语言运行库	4
1.3.3 框架类库	4
1.4 利用 ASP.NET 开发 Web 应用程序	5
1.4.1 Visual Studio.NET 集成开发环境	5
1.4.2 创建第一个 ASP.NET Web 应用程序	9
1.5 课外练习	12
第 2 章 ASP.NET Web Form 设计	13
2.1 Web Form 基本结构	13
2.2 ASP.NET 页面模型	16
2.2.1 单文件页面模型	17
2.2.2 代码隐藏模型	17
2.3 Web Form 的事件处理	18
2.3.1 事件模型	18
2.3.2 回发	21
2.3.3 跨页发送	22
2.3.4 Web Form 页面生命周期	22
2.4 ASP.NET 基本控件	23
2.4.1 控件概述	23
2.4.2 Label 控件	25
2.4.3 按钮控件	25
2.4.4 TextBox 控件	31
2.4.5 CheckBox 控件	33
2.4.6 RadioButton 控件	35
2.4.7 显示图像	35

2.4.8 列表显示	37
2.4.9 超级链接	45
2.5 课外练习	46
第3章 网页输入合法性验证	47
3.1 非空验证	47
3.2 比较验证	50
3.3 范围验证	53
3.4 正则表达式验证	55
3.4.1 正则表达式概述	55
3.4.2 实现正则表达式验证	56
3.5 自定义验证	58
3.5.1 服务器端自定义验证	58
3.5.2 客户端自定义验证	60
3.6 合成验证	61
3.7 课外练习	63
第4章 页面主题与母版页	65
4.1 页面主题	65
4.1.1 主题的基本概念	65
4.1.2 应用 CSS 样式	66
4.1.3 应用外观	67
4.1.4 应用主题	70
4.1.5 通过 Web. config 设置页面主题	75
4.2 母版页	76
4.2.1 母版页的基本概念	76
4.2.2 母版页的设计与应用	76
4.2.3 建立内容页	78
4.2.4 母版页的高级特性	80
4.3 课外练习	83
第5章 站点地图与导航	84
5.1 站点地图	84
5.1.1 站点地图的基本模型	84
5.1.2 获取站点地图数据	85
5.1.3 编程访问站点地图	86
5.2 显示导航路径	88
5.2.1 创建导航路径	89
5.2.2 定制导航路径	89
5.3 显示导航菜单	91
5.3.1 Menu 控件概述	91
5.3.2 创建菜单	92

5.3.3 菜单的核心属性	95
5.3.4 定制菜单	96
5.4 显示树状结构的菜单	99
5.4.1 TreeView 控件概述	99
5.4.2 创建树状菜单	100
5.4.3 核心属性和事件	101
5.4.4 定制树状菜单	101
5.5 Music 音乐站的导航系统	104
5.6 课外练习	106
第6章 ASP.NET 状态维护	107
6.1 视图状态与控件状态	107
6.1.1 视图状态	107
6.1.2 控件状态	110
6.2 Cookie、Session 与 Application	110
6.2.1 Cookie	110
6.2.2 Application	113
6.2.3 Session	114
6.3 其他状态维护方法	116
6.3.1 隐藏域	116
6.3.2 查询字符串	117
6.4 基于 Session 的购物车	118
6.5 课外练习	121
第7章 数据库数据操作	122
7.1 数据库 Web 程序基本模型	122
7.2 使用 ADO.NET 对象访问数据库	123
7.2.1 连接数据库	123
7.2.2 执行 SQL 语句和存储过程	127
7.2.3 获取数据集	131
7.3 使用 SqlDataSource 控件访问数据库	134
7.3.1 操作数据库	135
7.3.2 使用参数操作数据库	135
7.4 使用 ObjectDataSource 控件访问数据库	138
7.5 GridView 控件展现表数据	140
7.5.1 显示数据	140
7.5.2 实现分页	141
7.5.3 编辑数据	143
7.5.4 定制 GridView	145
7.6 DetailsView 控件分行展现表数据	147
7.6.1 导航记录	148

7.6.2 编辑记录	148
7.7 DataList 控件.....	149
7.8 课外练习	152
第8章 网站安全与用户管理	153
8.1 ASP.NET 网站安全模型	153
8.1.1 IIS 安全验证	154
8.1.2 ASP.NET 安全验证.....	154
8.2 登录控件的使用	156
8.2.1 登录界面的设计.....	156
8.2.2 登录界面处理方法	157
8.3 用户注册控件的使用	159
8.3.1 用户注册控件界面设计	159
8.3.2 用户注册控件的扩展	160
8.4 用户状态控件与密码更改控件	163
8.4.1 LoginName 控件	163
8.4.2 LoginStatus 控件	163
8.4.3 ChangePassword 控件	165
8.4.4 PasswordRecovery 控件.....	166
8.5 用户管理	167
8.5.1 用户管理的前期准备工作	168
8.5.2 使用 Membership 对成员进行管理	169
8.5.3 使用 MembershipUser 获取用户信息	170
8.6 角色管理	171
8.7 课外练习	173
第9章 用户控件及自定义服务器控件设计	174
9.1 自定义服务器控件概述	174
9.1.1 用户控件基本知识	174
9.1.2 自定义服务器控件基本知识	175
9.2 用户控件设计	175
9.2.1 界面设计	175
9.2.2 属性设计	177
9.2.3 事件设计	178
9.3 自定义服务器控件基本模型	181
9.4 自定义服务器控件的属性设计及状态维护	184
9.5 自定义服务器控件的事件设计	186
9.6 课外练习	189
第10章 Profile 与应用程序个性化	190
10.1 Profile 的基本模型	190
10.2 Profile 数据操作	191

10.2.1	创建 Profile	191
10.2.2	使用 Profile	192
10.2.3	Profile 高级用法	194
10.3	匿名用户的 Profile 数据操作	200
10.3.1	开启匿名用户的 Profile	200
10.3.2	删除匿名用户的 Profile	200
10.4	基于 Profile 的购物车设计	201
10.5	课外练习	205
第 11 章	Web 部件	206
11.1	Web 部件概述	206
11.2	创建和使用 Web 部件	208
11.3	Web 部件管理器	211
11.3.1	切换显示模式	211
11.3.2	以编程方式控制 Web 部件	215
11.4	创建 Web 部件区域	217
11.5	在运行时添加 Web 部件	222
11.5.1	从服务器控件列表中添加 Web 部件	223
11.5.2	从被关闭的 Web 部件列表中添加 Web 部件	224
11.5.3	从导入的 Web 部件列表中添加 Web 部件	224
11.6	自定义 Web 部件区域	228
11.6.1	定制 Web 部件的外观	229
11.6.2	定制 Web 部件的行为	229
11.6.3	定制 Web 部件的布局	230
11.6.4	定制 Web 部件的自定义属性	231
11.7	课外练习	234
第 12 章	ASP. NET Ajax	235
12.1	ASP. NET Ajax 概述	235
12.1.1	什么是 Ajax?	235
12.1.2	ASP. NET Ajax 框架	236
12.2	使用 ASP. NET Ajax 服务器控件	237
12.2.1	ScriptManager 控件	237
12.2.2	UpdatePanel 控件	240
12.2.3	UpdateProgress 控件	249
12.2.4	Timer 控件	252
12.3	使用客户端脚本库	256
12.3.1	类型系统	256
12.3.2	客户端生命周期事件	257
12.3.3	DOM 元素引发的事件	258
12.3.4	使用 Web 服务	260

12.4	调试和跟踪 Ajax	263
12.5	基于 ASP.NET Ajax 的聊天室	264
12.6	课外练习	267
第 13 章	基于快速开发方式的通讯录设计	268
13.1	系统概述	268
13.1.1	系统总体结构设计	268
13.1.2	数据库设计	269
13.2	功能实现	269
13.2.1	主页	269
13.2.2	联系人管理	272
13.2.3	组管理	276
第 14 章	基于三层架构的通讯录设计	279
14.1	三层架构概述	279
14.2	系统概述	280
14.2.1	系统总体结构设计	280
14.2.2	数据库设计	280
14.2.3	系统的目录结构	280
14.3	创建实体类	281
14.4	实现数据访问层	283
14.5	实现业务逻辑层	287
14.6	实现表示层	289
14.6.1	母版页	289
14.6.2	导航系统	290
14.6.3	主题与外观	291
14.6.4	主页	291
14.6.5	联系人管理	293
14.6.6	编辑联系人	295
14.6.7	添加联系人	296
14.6.8	组管理	298
14.6.9	编辑组	300
14.6.10	添加组	301

第 1 章 C# Web 2.0 技术概述

从 Web 1.0 发展到 Web 2.0,是互联网发展的又一次革命性变革,它强调用户参与、与用户互动、用户个性化等。而实现这些特性,仅靠传统的 Web 开发技术显然不行,但近几年兴起的动态 Web 设计技术却是一个不错的选择。在这些技术中,ASP.NET 又是一个佼佼者。本章主要介绍 Web 2.0 相关内容,以及一些动态 Web 开发技术,并概要介绍 ASP.NET 技术。

主要内容:

-
- ☒ Web 2.0 相关内容
 - ☒ 动态 Web 开发技术
 - ☒ ASP.NET 技术
-

1.1 Web 2.0 技术概述

Web 2.0 不是一种新技术,而是一个概念,是各种技术的统称。它使得互联网从少数人发布信息、多数人浏览信息方式演变为大家共同参与、共同主导方式,使得用户从被动接受信息转变为主动参与,从而极大地调动了用户的积极性与创造性。

Web 2.0 的核心是参与、分享与合作,所有的网络行为,都可用“参与、分享、合作”来做诠释。

对 Web 2.0 来说,目前比较成熟的技术有 Blog、Ajax、RSS、Wiki、Tag 和 SNS。

Blog 中文名称为博客,英文全称为 Web log,缩写为 Blog。它是个人或群体以时间顺序所作的一种记录,且不断更新。其本质就是一种日记形式的个人网页。

Ajax 全称为 Asynchronous JavaScript and XML(异步 JavaScript 和 XML),它不是一项技术,而是几项技术以全新方式的结合体,这些技术包括 XHTML、CSS、DOM、XML、XSLT 和 JavaScript。通过它的异步交互和动态更新能力,使得基于 Web 的应用程序有了基于 PC 的应用程序的体验。

RSS 全称为 Really Simple Syndication 或 Rich Site Summary(真正简单的聚合或丰富站点摘要),它是一种与其他站点共享内容的机制。利用 RSS 可以订阅站点上感兴趣的内容,当该内容发生变化时,会得到通知。

Wiki 来源于夏威夷语的 wee kee wee kee,是一种多人协作的写作工具。Wiki 站点由多人维护,每个人(包括任何访问者)都可以发表意见,都可以对某个主题的内容进行浏览、创建和更改。

Tag 中文名称为标签,它是一种新的组织和管理在线信息的方式。通过给帖子、图片等添

加 Tag, 可以看到站点上具有相同 Tag 的内容。

SNS 全称为 Social Network Service(社会性网络软件), 依据六度理论, 以认识朋友的朋友为基础, 扩展自己的人脉, 并在需要的时候, 获得人脉的支持。

1.2 Web 开发技术概述

随着互联网的发展, 特别是 Web 2.0 的出现, 单纯的静态网页已经远远不能满足需要, 动态网页技术应运而生。在众多的动态网页设计技术中, 比较优秀的有 ASP、JSP、PHP 和 ASP.NET 技术。

1.2.1 ASP 技术

ASP(Active Server Pages, 动态服务器页面) 是一种服务器端开发技术, 可以创建动态网页, 以及交互式 Web 应用程序。它在 HTML 中嵌入脚本, 从而完成各种程序逻辑, 以及数据库访问任务。同时, 它还能够使用 COM 组件, 从而极大地扩展了其使用范围。与静态网页相比, ASP 技术具有以下特点。

- (1) 与 JavaScript 或 VBScript 脚本结合, 实现动态网页技术。
- (2) ASP 程序简单易懂, 并且无须编译, 直接在服务器端运行。
- (3) 利用 ASP 提供的内置对象可以很容易地编写出交互式 Web 应用程序。
- (4) 使用各种 COM 组件可以无限扩充 Web 应用程序的功能。
- (5) 客户端仅需使用浏览器即可浏览 ASP 页面内容。

1.2.2 JSP 技术

JSP(Java Server Pages, 中文名称) 是由 Sun 公司主导建立的一种动态网页技术标准。它在 HTML 中嵌入 Java 程序段和 JSP 标记, 从而产生动态网页的处理逻辑。

JSP 是在服务器端执行的, 当 Web 服务器接收到 JSP 网页的请求时, JSP 引擎执行其中的程序段, 然后将执行结果以 HTML 形式返回给客户端, 并在客户端浏览器中显示, 因此客户端只需要浏览器。

JSP 最大的特性是跨平台, 既能在 Windows 平台下运行, 也能在 Linux 和 Unix 平台下运行, 从而可以实现一次编写而在各处运行。除此之外, 它还具有以下特性。

(1) 将内容的生成和显示进行分离。其页面外观的显示由 HTML 或者 XML 构成, 其动态内容由 JSP 标识或者小脚本来生成。生成内容的逻辑被封装在标识和 JavaBeans 组件中, 并且捆绑在脚本中, 因此在编辑和使用 JSP 页面时, 不影响内容的生成。

(2) 支持可重用的基于组件的设计。在 JSP 页面中, 可以使用可重用的、跨平台的组件(JavaBeans 或者 Enterprise JavaBeans 组件)来完成各种复杂的任务。

(3) 采用标识简化开发。标准的 JSP 标识能够访问和实例化 JavaBeans 组件, 设置或者检索组件属性, 下载 Applet 等任务。同时 JSP 是可以扩展的, 开发人员可以建立自己的标识库来完成一些特定任务。

(4) 支持复杂的 Web 应用。由于 JSP 的内置语言为 Java, 因此它具有 Java 的所有特性, 可以开发复杂的 Web 应用。

1.2.3 PHP 技术

PHP(Hypertext Preprocessor,超文本预处理器)是一种开放源代码的脚本编程语言。它与 ASP 类似,是一种服务器端技术,通过在 HTML 中嵌入脚本实现动态页面逻辑。它借用了 C、Java 和 Perl 语言的大量语法,并加入自己独特的语法,使得网站编程人员可以快速上手和使用。PHP 具有非常强大的功能,能实现 CGI 或者 JavaScript 的所有功能,而且支持几乎所有流行的数据库及操作系统。PHP 技术具有以下特性。

- (1) 支持多种平台。PHP 支持 Windows、Unix 和 Linux 平台,还支持 IIS、Apache 等 Web 服务器。
- (2) 开放源代码。PHP 可以免费使用。
- (3) 面向对象。高版本的 PHP 支持面向对象编程。
- (4) 效率高。PHP 消耗相当少的系统资源。

1.2.4 ASP.NET 技术

ASP.NET 不是 ASP 的下一个版本,而是 .NET Framework 的有机组成部分,它是一个统一的 Web 开发模型,包括使用尽可能少的代码生成企业级 Web 应用程序所必需的各种服务,可以使用 Visual Basic.NET、C#.NET、JScript.NET 和 J#.NET 编程语言,具有开发利用公共语言运行库、类型安全、继承等方面的优点的 ASP.NET Web 应用程序。它还具有以下优点。

(1) 简单易用,开发效率高。许多工作都可以用可视化的方式完成,极大降低了代码的编写量,提高了开发效率。

(2) 功能强大。ASP.NET 提供了许多预置功能,如站点导航、用户管理、母版页、主题、Web 部件、Ajax 等,利用它们可以开发出强大的 Web 应用。

(3) 跨语言集成。.NET 框架利用一个被称为通用类型系统(CTS)的规范,使得利用各种 .NET 编程语言编写的程序能够集成在一起。有了跨语言集成,就不必强求项目小组的人员使用同一种编程语言,从而减轻了开发人员学习各种开发语言的负担。

(4) 易于管理和部署。ASP.NET 提供了两个网站管理工具(ASP.NET MMC 管理单元和 ASP.NET 网站管理工具),简化了网站管理和配置工作。至于部署就更简单了,大多数情况下,只需要拷贝、粘贴文件,而不必考虑讨厌的注册表问题。另外,对于大型网站,它还提供了复制网站工具及“Web 安装”项目,来解决部署问题。

1.3 ASP.NET 技术概述

ASP.NET 是 .NET 框架(.NET Framework)的一部分,ASP.NET 的许多特性都由 .NET 框架提供。因此,有必要了解 .NET 框架。

1.3.1 .NET 框架

.NET 框架是支持生成和运行下一代应用程序(Web 应用程序和 Windows 应用程序)和 Web 服务(XML Web Services)的一个 Windows 组件。它提供创建、部署和运行 .NET 应用程序和 Web 服务的一个环境。

.NET 框架主要由公共语言运行库和框架类库构成,如图 1-1 所示。

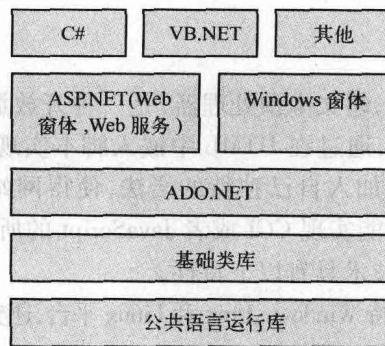


图 1-1 NET 框架构成

1.3.2 公共语言运行库

公共语言运行库(Common Language Runtime, CLR)为用 .NET 编程语言编写的代码(称为托管代码)提供运行环境。它提供内存管理、线程管理、代码执行、代码安全验证、编译及其他系统服务。它是一个类似于虚拟机的纯软件平台,它屏蔽了底层硬件和各种操作系统的差异,使得我们编写的 .NET 应用程序可运行于各种平台(当然这是一个长远目标,现阶段还只能运行于 Windows 平台)之上。 .NET 应用程序的运行步骤如下。

- (1) 用一种 .NET 编程语言(如 C#)编写 .NET 应用程序。
- (2) 使用编译器(如 C#编译器)将源代码编译为 Microsoft 中间语言(MSIL)。
- (3) 在执行时,公共语言运行库的实时编译器(JIT)将 MSIL 编译为本机代码。
- (4) 运行代码。

从以上执行过程可以看到,编译过程分为两个阶段。第一阶段是把源代码编译成 MSIL;第二阶段把 MSIL 编译为本机代码。这样做的好处是使 .NET 应用程序能够跨平台运行,因为不同平台的差异体现在第二阶段,而这个阶段是由公共语言运行库提供的,只要这个平台能够提供公共语言运行库,就可以在这个平台上运行 .NET 应用程序。

1.3.3 框架类库

框架类库提供了一套庞大的面向对象的可重用类型集合,它提供了对系统功能的访问,是建立 .NET 应用程序、组件和控件的基础。利用框架类库可以高效开发多种应用程序,包括 Web 应用程序、Windows 应用程序和 Web 服务。

框架类库用命名空间进行逻辑分组,以方便搜索和引用它们。表 1-1 列出了一些常用的命名空间。

表 1-1 常用的命名空间

名 称	说 明
System	包含用于定义常用值和引用数据类型、事件和事件处理程序、接口、属性和处理异常的基础类和基类
System. Collections	包含定义各种对象集合(如列表、队列、位数组、哈希表和字典)的接口和类
System. Collections. Generic	包含定义泛型集合的接口和类

续表

名 称	说 明
System. Data	包含利用 ADO. NET 访问和处理数据的类
System. Linq	包含支持使用语言集成查询 (LINQ) 的类和接口
System. Web	提供支持浏览器/服务器通信的类和接口
System. Web. UI	包含以可视化形式出现在 Web 应用程序中的控件和页类
System. Web. UI. WebControls	包含创建 Web 服务器控件的类
System. Web. Services	包含创建 Web 服务的类
System. Xml	提供对 XML 数据进行访问和处理的类

1.4 利用 ASP. NET 开发 Web 应用程序

开发 ASP. NET Web 应用程序有以下两种方式。

- (1) 利用 Visual Studio. NET 集成开发环境。
- (2) 利用记事本(或其他纯文本编辑器软件)。

利用记事本开发 ASP. NET Web 应用程序,要求开发人员对 ASP. NET 有深入了解,且开发效率低下,因此利用 Visual Studio. NET 集成开发环境开发 ASP. NET Web 应用程序是一种不错的选择。下面先熟悉一下 Visual Studio. NET 集成开发环境。

1.4.1 Visual Studio. NET 集成开发环境

1. 初识集成开发环境

Visual Studio. NET 集成开发环境由菜单栏、工具栏及停靠或自动隐藏在编辑器侧边的各种工具窗口组成,如图 1-2 所示。这些菜单、工具栏和工具窗口会随着集成开发环境所处理的项目或文件类型不同而发生变化。

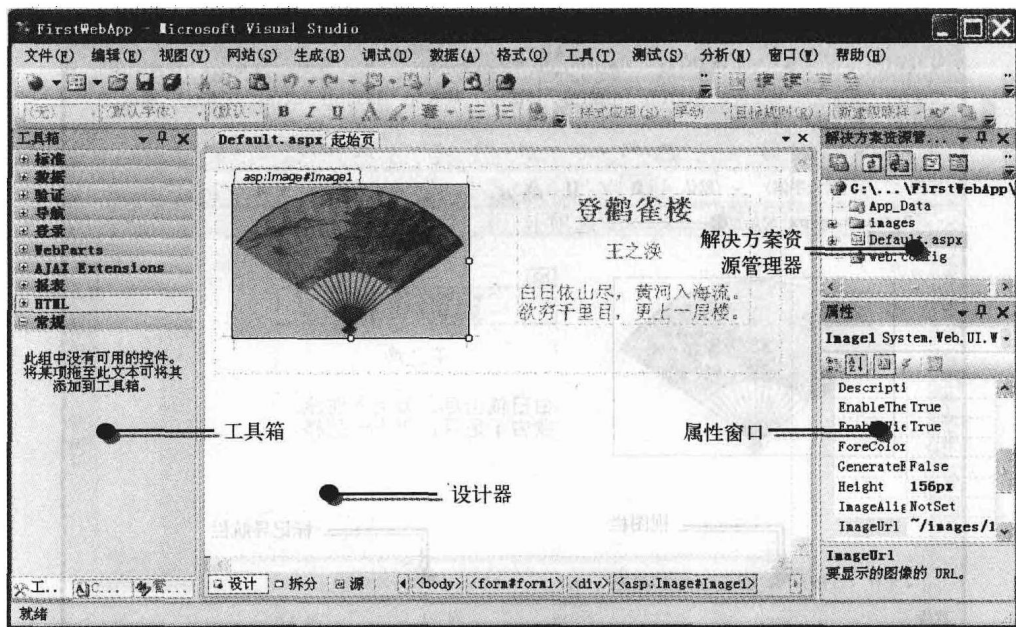


图 1-2 集成开发环境

2. 解决方案资源管理器窗口

解决方案就是一个容器,它可以包含多个项目。这些项目可以是网站、Windows 应用程序、类库等。设立解决方案的目的就是把多个项目组织在一起形成一个工作单元,这样就允许在一个 Visual Studio .NET 实例中分别操作这些项目,而不必运行多个 Visual Studio .NET 实例,这对于大型项目的开发非常有用。同时也可以把同一配置应用到不同的项目上,从而简化操作。

项目是应用程序的容器,可以容纳和管理程序文件、文件夹、引用、资源文件、配置文件等。

解决方案资源管理器(如图 1-3 所示)是对解决方案所包含的项目和各种文件进行管理的工具。通过解决方案资源管理器可以向解决方案中添加新项目、编辑已有项目,也可以向项目中添加新项(Web 窗体、类文件、配置文件等)或编辑已有项。通过双击解决方案资源管理器中的项,可以打开与该项关联的窗口,用于显示和编辑该内容。



图 1-3 解决方案资源管理器

3. 设计器窗口

设计器用来设计 ASP .NET Web 页面的外观。可以把设计器看作绘画的画布,通过用鼠标拖曳的方式,在 Web 页面上添加新项及调整已有项的位置和大小。设计器窗口如图 1-4 所示。



图 1-4 设计器窗口

1) 页面选项卡

在设计器窗口中,每一个打开的 ASP.NET Web 页面都以一个选项卡形式展现出来。如果打开了多个 ASP.NET Web 页面,就会出现多个选项卡。通过单击相应的选项卡,就可以从一个 ASP.NET Web 页面快速切换到另一个 ASP.NET Web 页面。

2) 视图

在设计器窗口中,提供了三种视图,它们是设计视图、拆分视图和源视图。在设计器窗口的下部有一个视图栏,上面有三个按钮,分别是“设计”、“拆分”和“源”。利用它们可以在三种视图之间切换。在设计视图中,可以用所见即所得的方式设计 ASP.NET Web 页面外观。可以从工具箱中拖曳控件,并将其置于 ASP.NET Web 页面上;在源视图中,可以用编辑 HTML 标记的形式设计 ASP.NET Web 页面外观;拆分视图是设计视图和源视图的结合,上部显示源视图的内容,下部显示设计视图的内容。这样在源视图中编辑 HTML 标记后,就可以在设计视图中直接看到修改后的结果,而不必在两个视图间来回切换。

3) 标记导航

在设计器窗口的下部有一个标记导航栏,用于在 HTML 标记间快速导航。源视图中的所有 HTML 标记在标记导航栏上都有一个按钮与之对应,单击按钮就会在源视图中选中该标记所对应的内容。利用它可以快速找到相应标记,对于大型和复杂页面,这一功能非常有用。

4) 智能标记

当把控件从工具箱拖曳到 ASP.NET Web 页面上时,许多控件会展现智能标记,如图 1-5 所示。通过单击智能标记会打开一个常见任务列表。单击或选择这些任务,就可以完成对控件的一些常用操作。

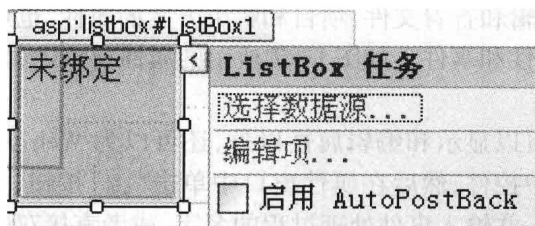


图 1-5 智能标记

4. 代码编辑器窗口

代码编辑器是编写应用程序源代码的工具。打开代码编辑器窗口的常用方法如下。

(1) 在解决方案资源管理器窗口中,双击源代码文件打开代码编辑器窗口。

(2) 在设计器中,右键单击,在快捷菜单中选择“查看代码”,打开代码编辑器窗口。

代码编辑器窗口如图 1-6 所示。在其顶部有两个下拉列表框,分别是“类型”下拉列表框和“成员”下拉列表框。“类型”下拉列表框中列出当前文件中所有类名,可以使用它快速定位到这些类声明处;“成员”下拉列表框列出类中所定义的各个成员,利用它可快速定位到这些成员声明处。

在代码编辑器中,源代码以各种不同的颜色显示,以区分不同的代码结构,从而帮助用户发现由于粗心或因拼写而造成的错误。默认情况下,代码编辑器中的常规代码为黑色,关键字