

21世纪高等院校计算机教材系列

ASP.NET

程序设计教程(C#版)

● 常永英 主编
● 崔 淼 马润成 梁云杰 等编著



购书可获得增值回报
提供教学用电子教案



机械工业出版社
CHINA MACHINE PRESS



TP393.092/829

2007

21 世纪高等院校计算机教材系列

ASP.NET 程序设计教程 (C#版)

常永英 主编

崔 淼 马润成 梁云杰 等编著



机械工业出版社

本书以 Visual Studio 2005 为开发平台,以 C#为开发语言,面向无程序设计基础的读者,采用案例方式,全面细致地介绍 ASP.NET 2.0 网站开发技术。全书共分为 10 章,主要内容包括 ASP.NET 2.0 和 Visual Studio 2005 开发环境简介、C# 2005 编程基础、结构化程序设计方法、常用 Web 服务器控件、ASP.NET 2.0 内置对象、数据库应用程序开发、Web Service 简介和发布 ASP.NET 2.0 网站等方面的内容。本书各章均配有一定数量的习题,以方便学生练习。

为配合教学,本书的配套教材《ASP.NET 程序设计教程(C#版)上机指导与习题解答》对教材中的习题做了详细解答,同时每章还增加了实训实验环节,以方便读者对本章内容进行综合性练习。

本书适合作为高等院校计算机专业教材,也可作为各类 ASP.NET (C#) 程序设计培训班的教学用书,还可作为广大计算机爱好者的自学参考书。

图书在版编目(CIP)数据

ASP.NET 程序设计教程: C#版 / 常永英主编. —北京: 机械工业出版社, 2007.8

(21 世纪高等院校计算机教材系列)

ISBN 978-7-111-22262-0

I. A… II. 常… III. ① 主页制作—程序设计—高等学校—教材 ② C 语言—程序设计—高等学校—教材 IV. TP393.092 TP312

中国版本图书馆 CIP 数据核字 (2007) 第 136239 号

机械工业出版社 (北京市百万庄大街 22 号 邮政编码 100037)

策 划: 胡毓坚

责任编辑: 赵 慧

责任印制: 洪汉军

北京双青印刷厂印刷

2007 年 9 月第 1 版·第 1 次印刷

184mm×260mm·19.5 印张·482 千字

0001—5000 册

标准书号: ISBN 978-7-111-22262-0

定价: 29.00 元

凡购本书,如有缺页、倒页、脱页,由本社发行部调换

销售服务热线电话: (010) 68326294

购书热线电话: (010) 88379639 88379641 88379643

编辑热线电话: (010) 88379739

封面防伪标均为盗版

出版说明

信息技术是当今世界发展最快、渗透性最强、应用最广的关键技术，是推动经济增长和知识传播的重要引擎。在我国，随着国家信息化发展战略的贯彻实施，信息化建设已进入了全方位、多层次推进应用的新阶段。现在，掌握计算机技术已成为 21 世纪人才应具备的基本素质之一。

为了进一步推动计算机技术的发展，满足计算机学科教育的需求，机械工业出版社聘请了全国多所高等院校的一线教师，进行了充分的调研和讨论，针对计算机相关课程的特点，总结教学中的实践经验，组织出版了这套“21 世纪高等院校计算机教材系列”。

本套教材具有以下特点：

- (1) 反映计算机技术领域的新发展和新应用。
- (2) 注重立体化教材的建设，多数教材配有电子教案、习题与上机指导或多媒体光盘等。
- (3) 针对多数学生的学习特点，采用通俗易懂的方法讲解知识，逻辑性强、层次分明、叙述准确而精炼、图文并茂，使学生可以快速掌握，学以致用。
- (4) 符合高等院校各专业人才的培养目标及课程体系的设置，注重培养学生的应用能力，强调知识、能力与素质的综合训练。
- (5) 适合各类高等院校、高等职业学校及相关院校的教学，也可作为各类培训班和自学用书。

机械工业出版社

前 言

目前, 计算机网络应用的广度和深度正以极快的速度向各个领域渗透, 在软件开发和程序设计技术方面, 基于计算机网络的 B/S (浏览器/服务器) 架构 Web 应用程序逐步占据了主流地位, 使传统的 C/S (客户端/服务器) 架构应用程序面临着巨大的挑战。

ASP.NET 2.0 是美国微软公司推出的新一代企业级 B/S 架构 Web 应用程序的开发平台, 它建立在 Microsoft .NET 框架的通用语言运行环境 (Common Language Runtime, CLR) 之上, 可用于在服务器端生成功能强大的 Web 应用程序, 建立分布式、多层架构的应用环境。与以前的 Web 开发模型 (ASP、JSP、PHP 等) 相比, ASP.NET 2.0 具有开发效率高、使用简单、支持多种语言等特点。

C#是从 C 和 C++发展而来的程序设计语言, 也是为 ASP.NET 量身订做的程序设计语言, 它能最大限度地发挥 .NET 平台的优势。

本书以 Microsoft Visual Studio 2005 为开发平台, 以 C#为开发语言, 面向毫无程序设计基础的读者, 采用案例方式, 从零开始, 全面细致地介绍了 ASP.NET 2.0 的基础知识、特点和具体的应用。特别是在数据库应用、ASP.NET 2.0 内置控件及对象、程序设计技巧等方面通过大量相对完整的实例进行详细的介绍, 避免了大量空洞理论的堆砌。

本教材在例题讲解的处理上采用“任务驱动”方式。即先给出设计目标, 然后介绍为实现该目标而采取的设计方法。为初学者考虑, 程序设计中的操作以详尽的表述结合图例来说明, 以求读者对每一步操作清清楚楚。在代码设计中尽可能多地给出注释, 以求读者对每一行代码的意义及其前后联系明明白白。针对初学者的特点, 本书在编排上注意做到简明扼要、由浅入深和循序渐进, 力求通俗易懂、简捷实用。本书概念清晰、逻辑性强、层次分明、例题丰富, 符合教师教学和学生学习的习惯。本教材在编写的主导思想上突出一个“用”字, 避免繁琐的、长篇大论的理论阐述, 紧紧抓住培养学生基本编程技能这个纲, 以求达到学以致用的目的。

全书共分为 10 章, 主要内容包括: ASP.NET 2.0 和 Visual Studio 2005 开发环境简介, C# 2005 编程基础, 结构化程序设计方法, 常用 Web 服务器控件, ASP.NET 2.0 内置对象, 数据库应用程序开发, Web Service 简介和发布 ASP.NET 2.0 网站等方面的内容。本书各章均配有一定数量的习题, 以方便学生课后练习。

对于初学程序设计的学生, 往往会出现“上课听得懂, 课后不会做”的现象。为方便学生上机练习和教师组织编程训练, 我们编写了本书的配套教材《ASP.NET 程序设计教程 (C#版) 上机指导与习题解答》对教材中的习题做了详细解答, 同时每章还增加了一个实训实验环节, 以达到对本章内容进行综合性练习的目的。

为配合本书的教学, 方便教师讲课, 我们制作了教学课件, 课件内容浓缩了本书的教学要点, 可作为教师的板书来演示。教学课件可在机械工业出版社网站 www.cmpedu.com 上下载。为方便教学和实验, 我们将书中的例题、习题的源代码文件打包压缩, 该压缩文件也可从上面的网站下载。

本书由常永英主编, 编写分工为: 崔淼编写第 1 章, 孙利娟、贾燕玲、雷成茂编写第 2

章，姜桦、王勇、刘婕编写第 3 章，马润成编写第 4、9 章，姚芬、马亚平、孙涛编写第 5 章，常永英编写第 6 章，梁云杰编写第 7 章，李培红、李冠军、臧顺娟、孙洪玲编写第 8 章，葛立峰、李晶、刘瑞广、李瑛编写第 10 章。全书由常永英统稿。

由于编者水平有限，本书难免会出现一些错误或不当之处，恳请批评指正。

作者

目 录

出版说明

前言

第 1 章 ASP.NET 2.0 概述	1
1.1 程序设计方法的发展历程	1
1.1.1 程序设计方法的发展	1
1.1.2 程序设计语言的发展	2
1.1.3 C/S 和 B/S 架构体系	3
1.1.4 多层开发架构	5
1.2 .NET 框架简介	6
1.3 ASP.NET 2.0 的体系结构	7
1.3.1 ASP.NET 2.0 的逻辑架构	8
1.3.2 ASP.NET 2.0 网站的组成文件	8
1.4 习题	11
第 2 章 Visual Studio 2005 集成开发环境	12
2.1 管理项目	13
2.1.1 新建项目	13
2.1.2 集成开发环境中的主要子窗口	15
2.1.3 管理项目文件	19
2.2 使用 Visual Studio 2005 的帮助系统	21
2.2.1 动态帮助	21
2.2.2 智能感知	21
2.2.3 MSDN Library 帮助系统	23
2.2.4 联机帮助	24
2.3 创建一个简单的 ASP.NET 网站	25
2.3.1 设计要求及设计方法分析	26
2.3.2 创建 ASP.NET 网站	26
2.3.3 设计 Web 页面	26
2.3.4 设置对象属性	30
2.3.5 编写事件代码	31
2.3.6 运行及调试程序	32
2.4 习题	33
第 3 章 C# 2005 程序设计基础	34
3.1 C#程序的代码结构	34
3.1.1 C#代码的组成要素及书写规则	34
3.1.2 C#程序的格式	36

3.2	数据类型	37
3.2.1	数值类型	38
3.2.2	字符类型	38
3.2.3	布尔类型和对象类型	39
3.3	变量与常量	39
3.3.1	变量及其作用域	39
3.3.2	常量	43
3.3.3	类型转换	45
3.4	运算符与表达式	48
3.4.1	运算符与表达式类型	48
3.4.2	运算符的优先级与结合性	53
3.5	C# 2005 的常用方法与属性	55
3.5.1	日期时间类常用方法与属性	55
3.5.2	常用数学方法与字段	55
3.5.3	字符串的方法与属性	56
3.5.4	类型转换方法	56
3.5.5	随机方法	57
3.6	习题	57
第 4 章	结构化程序设计方法	59
4.1	基本控件和顺序结构程序设计	59
4.1.1	标签控件	59
4.1.2	按钮控件	62
4.1.3	文本框控件	63
4.1.4	顺序结构程序设计	64
4.2	选择结构程序设计	67
4.2.1	选择结构程序设计的概念	67
4.2.2	if...else 结构	68
4.2.3	if...else if 结构	70
4.2.4	多分支选择结构	73
4.3	循环结构程序设计	79
4.3.1	循环结构程序设计的概念	79
4.3.2	for 循环	79
4.3.3	while 循环	83
4.3.4	do...while 循环	87
4.3.5	转向语句和循环的嵌套	90
4.4	习题	92
第 5 章	数组、枚举与结构	95
5.1	数组概述	95
5.1.1	数组与数组元素	95

5.1.2	数组的类型	96
5.2	声明与访问数组	96
5.2.1	声明和访问一维数组	96
5.2.2	声明和访问多维数组	101
5.3	控件数组	105
5.3.1	创建控件数组	105
5.3.2	使用控件数组	107
5.3.3	多控件共享单一事件	110
5.4	数组的属性与方法	112
5.4.1	数组的常用属性	112
5.4.2	数组的常用方法	113
5.5	自定义数据类型	114
5.5.1	结构类型	114
5.5.2	枚举类型	118
5.6	习题	119
第6章	ASP.NET 2.0 常用控件	123
6.1	ASP.NET 控件概述	123
6.1.1	HTML 控件	123
6.1.2	服务器端控件	123
6.2	标准服务器端控件	125
6.2.1	ImageButton 和 HyperLink 控件	125
6.2.2	Image 和 ImageMap 控件	126
6.2.3	RadioButton 和 RadioButtonList 控件	131
6.2.4	CheckBox 和 CheckList 控件	133
6.2.5	ListBox 和 DropDownList 控件	134
6.2.6	FileUpload 控件	137
6.2.7	Table 控件	141
6.2.8	容器控件	144
6.2.9	Calendar 控件	147
6.2.10	AdRotator 控件	153
6.3	数据验证控件	155
6.3.1	数据验证控件概述	156
6.3.2	必须项验证控件	157
6.3.3	比较验证控件	159
6.3.4	范围验证控件	161
6.3.5	正则表达式验证控件	162
6.3.6	自定义验证控件	163
6.3.7	验证摘要控件	166
6.4	用户自定义控件	168

6.4.1	用户自定义控件概述	168
6.4.2	创建用户控件	168
6.4.3	在设计视图中添加用户控件	169
6.4.4	程序运行时动态添加用户控件	170
6.4.5	公开构成控件的属性	171
6.5	习题	173
第 7 章	ASP.NET 内置对象	176
7.1	类与对象概述	176
7.1.1	类和对象的概念	176
7.1.2	类与对象的本质	176
7.1.3	类成员	177
7.2	类的字段与属性	177
7.2.1	定义一个类	178
7.2.2	声明与使用对象	179
7.3	ASP.NET 对象概述	180
7.4	Page 对象	181
7.4.1	Page 对象的属性	181
7.4.2	Page 对象的常用方法	182
7.4.3	Page 对象的常用事件	183
7.5	Response 对象	185
7.5.1	Response 对象的常用属性	185
7.5.2	Response 对象的常用方法	185
7.5.3	使用 Response 对象	186
7.6	Request 对象	189
7.6.1	Request 对象的常用属性	189
7.6.2	Request 对象的常用方法	189
7.6.3	使用 Request 对象	190
7.7	Server 对象	196
7.7.1	Server 对象的常用属性	196
7.7.2	Server 对象的常用方法	196
7.8	Session 对象	202
7.8.1	Session 的工作原理	202
7.8.2	Session 对象的常用属性及方法	202
7.8.3	向 Session 对象中存入数据	203
7.8.4	从 Session 对象中取出数据	203
7.9	Application 对象	206
7.9.1	Application 对象和 Session 对象的区别	206
7.9.2	Application 对象的常用属性、方法和事件	206
7.9.3	Application 对象中的数据存取	207

7.10 习题	209
第8章 设计数据库应用网站	212
8.1 数据库基础知识	212
8.1.1 有关数据库的概念	212
8.1.2 关系型数据库	212
8.2 使用数据库系统	214
8.2.1 Microsoft Access	214
8.2.2 Microsoft SQL Server	215
8.3 SQL 查询基础	219
8.3.1 查询语句 Select	219
8.3.2 插入记录语句 Insert	221
8.3.3 修改记录语句 Update	221
8.3.4 删除记录语句 Delete	221
8.4 ADO.NET 简介	222
8.4.1 数据库通用接口	222
8.4.2 ADO.NET 的数据模型	223
8.4.3 数据源控件简介	224
8.5 使用数据控件访问数据库	229
8.5.1 使用 GridView 数据控件	229
8.5.2 使用 FormView 数据控件	232
8.5.3 使用 Repeater 控件	233
8.5.4 使用 DetailsView 控件	237
8.6 使用 ADO.NET 对象	242
8.6.1 ADO.NET 对象概述	243
8.6.2 数据集	243
8.6.3 数据表	244
8.6.4 数据行	244
8.6.5 数据列	244
8.6.6 数据关系	245
8.6.7 Connection	245
8.6.8 Command	247
8.6.9 DataAdapter	253
8.6.10 DataReader	267
8.6.11 DataView	270
8.7 习题	274
第9章 创建和使用 Web Service	278
9.1 Web Service 简介	278
9.1.1 了解 Web Service	278
9.1.2 Web Service 的相关标准和规范	279

9.2	创建 Web Service	281
9.2.1	创建 Web Service 项目	281
9.2.2	编写实现 Web Service 功能的代码	281
9.2.3	测试 Web Service	282
9.3	在 ASP.NET 应用程序中调用 Web Service	283
9.3.1	添加 Web 引用	283
9.3.2	编写调用代码	284
9.4	在 Windows 应用程序中调用 Web Service	284
9.4.1	创建能接收参数的 Web Service	284
9.4.2	创建调用 Web Service 的 Windows 程序	286
9.5	习题	288
第 10 章	部署与发布 ASP.NET 网站	289
10.1	在 Web 服务器中配置 IIS	289
10.1.1	为站点准备一个 IP 地址	289
10.1.2	创建站点主目录	289
10.1.3	配置 IIS	290
10.2	创建 FTP 远程管理账号	292
10.3	发布 ASP.NET 网站	293
10.3.1	使用 FTP 方式发布网站	293
10.3.2	发布网站到本地文件夹	295
10.4	使用 Web 安装项目部署应用程序	295
10.4.1	创建与生成 Web 安装程序	295
10.4.2	部署安装项目到 Web 服务器	298
10.5	习题	300

第 1 章 ASP.NET 2.0 概述

从早期的静态 HTML，到动态的 CGI、ASP、JSP 和 PHP，Internet 应用的快速发展推动了网页制作及网站设计方法的不断进步。虽然 ASP 易学好用，有相当的普及度，但由于在语言编写环境、代码重用及安全性、可靠性等方面仍然存在着明显的不足，严重地限制了 ASP 的发展势头。为了更好地适应网络环境的应用，Microsoft 公司推出了全新的 ASP.NET、ASP.NET 2.0。

ASP.NET 2.0 (Active Server Pages, 动态服务器页面) 是美国微软公司在 2005 年推出的新一代 Web 应用程序开发技术，主要用于开发基于 .NET 环境的、B/S (浏览器/服务器) 结构的网站应用程序。ASP.NET 2.0 支持与 .NET 兼容的 C#、Visual Basic .NET 等程序设计语言。用户可使用安装在本地计算机中的 Internet 浏览器，方便地访问存放在 IIS (Internet Information Services) 服务器中的应用程序，这与传统的 C/S (客户端/服务器) 结构的、基于操作系统的应用程序相比在使用方法上有很大的不同。

1.1 程序设计方法的发展历程

程序是一系列的计算机指令，用来控制计算机自动完成某一任务。编写这些指令的工作称为“程序设计”，也称为“编程”。按照某种规则编写出来的指令集合称为“程序”。指令书写时必须遵循的书写格式和规则的集合称为程序语言，是人和计算机交互的工具。

1.1.1 程序设计方法的发展

从 1946 年第一台计算机诞生起，在这短短的 50 多年，计算机技术发展迅速。程序设计方法经历了机器语言、汇编语言、高级语言到面向对象的程序设计语言的多个阶段，设计方法也从传统的程序设计方法、结构化程序设计方法、可视化程序设计方法，逐步过渡到了今天的面向对象程序设计方法。

1. 传统的编程方法

在计算机出现的早期，由于它的价格昂贵、内存小、速度慢，程序员为了在较小的内存中进行大量的科学计算问题，并尽量节省 CPU 的开销，不得不使用各种技巧，手工编写各种高效能的应用程序。这些程序常有可读性差、可维护性差、通用性差的通病。

随着计算机技术及电子工业的迅速发展，计算机的应用开始向各个领域渗透，高级语言蓬勃兴起，程序员需要解决的问题更复杂，程序规模更大。特别是 Windows 操作系统的出现及其迅速的发展，使传统的程序设计方法几乎走到了绝境。

2. 结构化程序设计

结构化程序设计是 20 世纪 70 年代出现的一种程序设计思想，结构化程序设计方法要求把程序的结构规定为顺序、选择和循环 3 种基本结构。限制使用语句间的跳转 (goto 语句)，并提出了自顶向下、逐步求精、模块化程序设计等设计原则。它以模块分割方法作为对大型

系统进行分析的手段，使其最终转化为上述 3 种基本结构。其目的是为了解决团队开发大型软件时，如何实现高效率、高可靠性的问题。程序的可读性好、可维护性好已成为评价程序质量的首要标准。

结构化程序设计方法虽然得到了广泛的使用，但仍存在一些问题。首先是，模块分割仍然是主要针对控制流的，很难自然、准确地反映客观世界。再有，结构化程序设计中只突出了实现功能的操作方法（程序模块），而被操作的数据（变量）则处于实现具体功能的从属地位，程序模块和数据结构是松散地耦合在一起的。因此当程序比较复杂时就容易出错，不易维护。

3. 面向对象的程序设计

面向对象的程序设计思想是 20 世纪 80 年代初提出的，起源于 Smalltalk 语言。为了在程序设计中使软件能够模仿建立客观世界的模型，所以对系统的复杂性进行了概括、抽象和分类，使软件的设计与实现形成一个由抽象到具体、由简单到复杂的循序渐进的过程，从而解决了大型软件开发中存在的效率低、质量无法保证、调试复杂、维护困难等一系列问题。也正是如此，面向对象的程序设计方法得到了广泛的应用。

当然，面向对象的程序设计并不是要完全抛弃结构化程序设计的方法，二者并不矛盾。面向对象的程序设计的分解为低级模块时，仍需要结构化编程技巧。只是它在分解一个大问题为小问题时采取的思路是不同的。

结构化程序设计的分解突出过程，即怎样做。它强调代码的功能是如何得以实现的；面向对象的分解突出客观世界和抽象的对象，即做什么。它将大量的工作由相应的对象来完成，程序员在应用程序中只需说明要求对象完成的任务是什么即可。

1.1.2 程序设计语言的发展

无论是传统的编程方法、结构化程序设计方法，还是面向对象的程序设计方法，都需要某种计算机语言的支持。

1. 机器语言和汇编语言

计算机诞生之初人们使用二进制形式的机器语言编写程序。它是一个面向机器的计算机语言，要求程序设计人员必须精通计算机原理，熟知计算机底层资源的分配及调用方法。而且，使用二进制编写出来的代码可读性差，记忆也十分困难。

人们为了解决机器语言可读性差和不易记忆的问题，开始使用汇编语言。它使用一些助记符来代替冗长的二进制编码，这样的程序计算机是不能直接执行的，需要使用专门的软件（编译系统）将汇编程序“翻译”成计算机可识别的二进制指令形式，然后交给计算机执行。但由于汇编语言仍是面向机器的，不利于计算机编程技术向其他行业中发展。

2. 高级语言

1954 年第一个高级语言程序设计语言——FORTRAN 的诞生，宣告了程序设计的一个新时代的开始。高级语言以人类自然语言和数学语言为基础，更接近人们的习惯，更容易被人们所接受，从而使更多的人参与到计算机程序设计的行列中来，为计算机应用的普及起到了重要的作用。使用较广泛的高级语言有 BASIC、FORTRAN、C、C++、PASCAL、JAVA 等。

随着 Windows 操作系统的发展，人们迫切需要迅速从 DOS 开发环境中过渡到 Windows 开发环境，开发出能在 Windows 系统中运行的、可视化的应用程序。Microsoft Visual Studio 6.0

(包括 Visual Basic、Visual C++、Visual FoxPro 等) 及 PowerBuilder、Delphi 等可视化的开发环境均可较好的实现这一目的。

3. 面向对象的程序设计语言

在计算机网络及其应用迅速发展的今天,美国 Microsoft 公司为适应潮流推出了以 .NET 为核心的 Visual Studio .NET 系列产品 (Visual Studio 2003 和 Visual Studio 2005)。ASP.NET 就是其中重要的一个组成部分,它包含了几乎所有先进的编程技术,而且相对简单易学,可以说它为用户提供了 Microsoft .NET 的简单切入点。

目前在 Windows 环境中常用的面向对象程序设计语言有: Visual Studio 2003/2005 中包含的 C#、Visual Basic .NET 等,以及 Java、PowerBuilder、Delphi 等。虽然它们的风格各异,语法结构有很大的不同,但都具有相同的思维方式和编程模式。

本书主要介绍在 Visual Studio 2005 环境中,使用 C#语言进行面向对象的 ASP.NET 2.0 网站设计的方法。选择 C#作为编程的入门语言不但可以学会基本的编程方法和技巧,还可以掌握非常实用的编程语言和工具,可以在相当长的时间内拥有当前最先进的编程技术。

传统的编程方法使用的是面向过程、按顺序进行的机制,其缺点是程序员始终要关心什么时候发生什么事件。如果用这种方法来处理 Windows 环境下的事件驱动方式工作量实在是太大了。Visual Studio 2005 采用面向对象的事件驱动编程机制,程序员只需编写响应用户动作(事件)的程序,如用户移动鼠标、单击或双击某控件(按钮、文本框、列表框等)的“事件”发生时,应该做什么,而不必考虑按精确次序执行的每个步骤,编写代码相对较少。这样就可以快速创建强大的应用程序,而无需考虑过多的细节。

1.1.3 C/S 和 B/S 架构体系

目前在程序开发领域中,主要有两大编程体系:一是基于操作系统平台的 C/S 结构,另一种是基于浏览器的 B/S 结构。

1. C/S 架构体系

在 2000 年以前 C/S 架构占据着开发领域的主流地位,其结构如图 1-1 所示。通常程序员将开发完成的软件安装在某计算机(客户机)中,将数据库安装在专用的服务器(数据库服务器)中,用户通过安装在客户机中的软件和网络进行各种数据库操作。这种架构要求客户机中必须安装客户端程序,否则无法工作。再有,在 C/S 架构中主要的数据分析处理工作需要客户机中完成,这就要求客户机有较高的硬件配置。常用的聊天工具如 QQ、MSN 及一些网络游戏都属于 C/S 架构的应用程序。

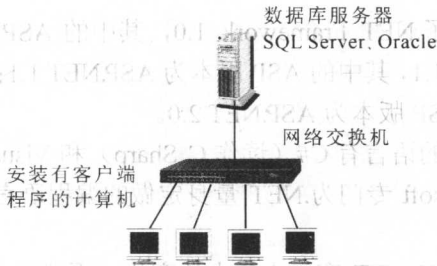


图 1-1 C/S 架构体系

2. B/S 架构体系

B/S 架构体系如图 1-2 所示，它由客户机、应用服务器和数据库服务器 3 部分组成。在中小型应用系统中，Web 服务器可以与数据库服务器安装在同一台服务器中。与 C/S 架构相比，它不需要在客户机上安装专门的客户端软件，用户在使用程序时仅需要通过安装在客户机上的浏览器访问指定的 Web 服务器即可。目前绝大多数微机都在使用集成了 Internet Explorer (IE 浏览器) 的 Windows 操作系统，也就是说只要客户机能够通过网络访问指定的 Web 服务器，即可使用 B/S 架构的应用程序。此外，在 B/S 架构中，主要的数据分析处理工作是在应用服务器中完成的，客户端主要用来下达指令和接收结果，所以客户机的配置要求不高，B/S 架构非常适合“瘦客户端”的运行环境。

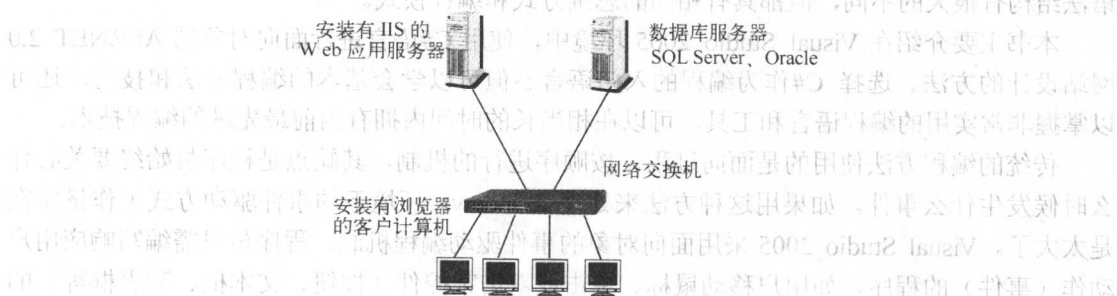


图 1-2 B/S 架构体系

3. B/S 架构编程语言

目前 B/S 架构应用程序开发，主要使用 4 种语言：ASP、ASP.NET、PHP 和 JSP。

(1) ASP

ASP (Active Server Pages) 所使用的 VBScript 脚本语言直接来源于 Visual Basic 语言，秉承了 Visual Basic 语言的简单易用特点，可以将脚本语言直接嵌入 HTML 文档中，不需要编译就可以直接运行。由于 ASP 程序是在服务器端运行的，当客户端浏览器访问 ASP 网页时，服务器将网页解释成标准的 HTML 代码发送给客户端，所以不存在浏览器兼容的问题。但因为每当客户端打开一个 ASP 页面时，服务器都会将该 ASP 程序解释一遍，最后生成标准的 HTML 代码发送到客户端，从而影响了 ASP 程序的运行速度；再有，ASP 不能运行在 Linux、FreeBSD 等操作系统中。因此一般情况下，用 ASP 开发的程序只能运行在 Windows 操作系统的 IIS 环境中。

(2) ASP.NET

2002 年 Microsoft 发布了 .NET Framework 1.0，其中的 ASP 版本为 ASP.NET 1.0；2003 年又发布了 .NET Framework 1.1，其中的 ASP 版本为 ASP.NET 1.1；2005 年推出了最新的 .NET Framework 2.0 版，其中的 ASP 版本为 ASP.NET 2.0。

ASP.NET 目前主要支持的语言有 C# (读作 C Sharp) 和 Visual Basic .NET。其中本教材重点介绍的 C# 语言是 Microsoft 专门为 .NET 量身定做的编程语言，它与 .NET 有着密不可分的联系。

与早期的 ASP 相比，ASP.NET 有了本质上的变化，不能将 ASP.NET 看成是 ASP 的简单升级，ASP.NET 的主要优点有以下几个方面：

1) 使用.NET 提供的所有类库, 全面支持面向对象的程序设计, 可以实现以往 ASP 所不能实现的许多功能。

2) 引入了服务器端控件的概念, 这样使开发交互式网站更加方便。

3) 引入了 ADO.NET 数据访问接口, 大大提高了数据库访问效率。

4) 使用 Visual Studio .NET 可以在可视化开发环境中创建 ASP.NET 应用程序, 进一步提高了编程效率。

5) 由于 ASP.NET 应用程序的核心部分在发布到 IIS 网站前已被编译成了 .dll 文件, 所以执行速度更快。但 ASP.NET 目前只能运行在 Windows 操作系统的 IIS 环境中。

(3) JSP

JSP 页面是由 HTML 代码和嵌入其中的 Java 代码组成, 具有良好的跨平台性。在页面被客户端请求时, 服务器对其中的 Java 代码进行处理, 然后将生成的标准 HTML 页面发送到客户端。与 ASP 不同的是 JSP 页面第一次被访问时, 服务器将 JSP 编译成二进制代码, 并保存起来, 以后当客户机再次访问该页面时, 这些二进制代码被直接调用, 所以 JSP 较 ASP 具有更高的执行效率。

(4) PHP

PHP 程序最初是用 Perl 语言编写的简单程序, 后来经其他程序员不断完善, 于 1997 年发布了功能基本完善的 PHP3。PHP 程序可以运行在 UNIX、Linux 和 Windows 操作系统中, 对客户端浏览器也没有特殊的要求。PHP 也是将脚本语言嵌入到 HTML 文档中, 它大量采用了 C、Java 和 Perl 语言的语法, 并加入了 PHP 自己的特征。

PHP 在 1999 年到 2000 年期间应用较为普遍, 由 Linux + PHP + MySQL 构成的完全开源而且非常稳定的应用平台曾经风靡一时, 但因 PHP 语言更新较慢加上没有很好的技术支持, 目前 PHP 正在逐步退出 B/S 架构的开发领域。

1.1.4 多层开发架构

在构建企业级应用时, 通常需要大量的代码, 而且这些代码一般分布在不同的计算机上。这种使程序代码分布在多台计算机上的设计理念称为“多层设计”理念。企业级应用系统通常分为两层、三层或 N 层架构。使用这种多层设计方法, 可以使开发的程序层次结构更加清晰, 特别适合用于大型软件的团队开发。

1. 两层架构

传统的两层应用包含有用户接口和后台程序两部分, 其中后台程序一般是一个数据库, 用户接口可直接与数据库通信。通常可使用 ASP、ASP.NET、JSP 等语言编写这类程序。图 1-3 表现了两层架构的概况。

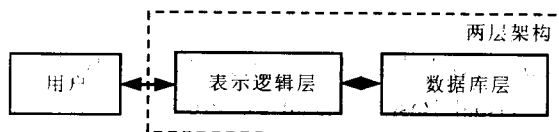


图 1-3 两层应用架构

两层应用架构中的表示逻辑层一般有 HTML、ASP、ASP.NET、JSP 页面实现, 通过 ODBC