



普通高等教育“十一五”国家级规划教材

王振武 编著

C# Web程序设计

21世纪
计算机
科学
与
技术
实
践
型
教
程

丛书主编
陈明

清华大学出版社





普通高等教育“十一五”国家级规划教材

王振武 编著

C# Web程序设计

21世纪
计算机
科学
与
技术
实
践
型
教
程

丛书主编
陈明

清华大学出版社
北京

内 容 简 介

本书对如何使用 C# 语言开发 Web 应用程序进行了详细的介绍。在对开发 Web 应用程序所涉及的基础知识进行全面系统阐述的同时,以用户信息表(UserInfo 表)为例,分别给出了使用 ADD.NET 和 LINQ to SQL 对数据库表进行增加、删除、修改和查询操作的通用代码,读者可参照该代码的编写思路很容易地实现对其他数据库表的操作,这种理论和实践相结合的方式大大方便了初学者理解和掌握使用 C# 开发 Web 应用程序的方法和步骤。本书不仅对如何使用 C# 开发 Web 应用程序进行了介绍,而且还介绍了 ASP.NET 程序分层设计的方法和步骤。

全书共 13 章,包括 .NET 简介、C# 语法基础、C# 面向对象基础、C# 泛型编程、LINQ 技术、Web 程序设计基础、ASP.NET 服务器控件、ASP.NET 内置对象、数据库知识简介、使用 ADO.NET 访问数据库、使用 LINQ to SQL 访问数据库、ASP.NET 应用程序配置、MVC 及 ASP.NET 分层程序设计等内容。

本书可作为高等院校 C# Web 程序设计的课程教材,也可作为使用 C# 开发 Web 应用程序的技术人员的参考书。

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

C# Web 程序设计/王振武编著. —北京:清华大学出版社,2012.9

21 世纪计算机科学与技术实践型教程

ISBN 978-7-302-29274-6

I. ①C… II. ①王… III. ①C 语言—程序设计 IV. ①TP312

中国版本图书馆 CIP 数据核字(2012)第 151591 号

责任编辑:谢 琛

封面设计:傅瑞学

责任校对:白 蕾

责任印制:王静怡

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦 A 座 邮 编:100084

社 总 机:010-62770175 邮 购:010-62786544

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

印 刷 者:清华大学印刷厂

装 订 者:三河市新茂装订有限公司

经 销:全国新华书店

开 本:185mm×260mm 印 张:20

字 数:454 千字

版 次:2012 年 9 月第 1 版

印 次:2012 年 9 月第 1 次印刷

印 数:1~2500

定 价:36.00 元

产品编号:048467-01

《21 世纪计算机科学与技术实践型教程》

序

21 世纪影响世界的三大关键技术：以计算机和网络为代表的信息技术；以基因工程为代表的生命科学和生物技术；以纳米技术为代表的新型材料技术。信息技术居三大关键技术之首。国民经济的发展采取信息化带动现代化的方针，要求在所有领域中迅速推广信息技术，导致需要大量的计算机科学与技术领域的优秀人才。

计算机科学与技术的广泛应用是计算机学科发展的原动力，计算机科学是一门应用科学。因此，计算机学科的优秀人才不仅应具有坚实的科学理论基础，而且更重要的是能将理论与实践相结合，并具有解决实际问题的能力。培养计算机科学与技术的优秀人才是社会的需要、国民经济发展的需要。

制订科学的教学计划对于培养计算机科学与技术人才十分重要，而教材的选择是实施教学计划的一个重要组成部分，《21 世纪计算机科学与技术实践型教程》主要考虑了下述两方面。

一方面，高等学校的计算机科学与技术专业的学生，在学习了基本的必修课和部分选修课程之后，立刻进行计算机应用系统的软件和硬件开发与应用尚存在一些困难，而《21 世纪计算机科学与技术实践型教程》就是为了填补这部分空白。将理论与实际联系起来，使学生不仅学会了计算机科学理论，而且也学会了应用这些理论解决实际问题。

另一方面，计算机科学与技术专业的课程内容需要经过实践练习，才能深刻理解和掌握。因此，本套教材增强了实践性、应用性和可理解性，并在体例上做了改进——使用案例说明。

实践型教学占有重要的位置，不仅体现了理论和实践紧密结合的学科特征，而且对于提高学生的综合素质，培养学生的创新精神与实践能力有特殊的作用。因此，研究和撰写实践型教材是必需的，也是十分重要的任务。优秀的教材是保证高水平教学的重要因素，选择水平高、内容新、实践性强的教材可以促进课堂教学质量的快速提升。在教学中，应用实践型教材可以增强学生的认知能力、创新能力、实践能力以及团队协作和交流表达能力。

实践型教材应由教学经验丰富、实际应用经验丰富的教师撰写。此系列教材的作者不但从事多年的计算机教学，而且参加并完成了多项计算机类的科研项目，他们把积累的经验、知识、智慧、素质融于教材中，奉献给计算机科学与技术的教学。

我们在组织本系列教材过程中，虽然经过了详细的思考和讨论，但毕竟是初步的尝试，不完善甚至缺陷不可避免，敬请读者指正。

本系列教材主编 陈明

2005 年 1 月于北京

前 言

C# (C Sharp)语言是微软公司专为.NET Framework 量身定做的编程语言,它广泛应用于 Windows 窗体程序设计、Web 应用程序设计、Web Service 开发以及 COM 组件开发等多个领域。对于普通高等院校计算机科学与技术、信息安全、软件工程等专业的学生,掌握 C# 程序设计不但可以快速高效地开发软件,还将有助于提高自身的就业竞争力。

本书的创作契机

近几年来,我在给计算机科学与技术专业的学生讲授“C# Web 程序设计”的同时,也一直在使用 C# 语言进行实际项目的开发。目前市面上有关 C# 的书籍要么页数很多、内容繁杂,初学者很难读懂;要么内容组织不系统,无法作为教材使用,因此在给学生讲授“C# Web 程序设计”课程的时候,很难找到一本合适的教材。与此同时,使用 C# 语言开发 Web 应用程序,相当一部分的工作都是对数据库表进行增加、删除、修改和查询等操作,而随着 C# 语言的不断发展,操作数据库的技术日益成熟和丰富。因此,我们能否针对数据库表的增加、删除、修改和查询操作给出一组通用而规范的代码作为开发 Web 应用程序的范例呢?出于上述原因,我决定编写本书。

本书的特点

本书不仅包含 C# 语言的基础知识,而且对如何使用 C# 语言进行 ASP.NET 程序设计也进行了详细的讲解。本书结合作者多年的开发和教学经验,对 C# 编程从最基本的理论概念到实践案例,都进行了循序渐进的精心设计,其特点主要体现在以下几个方面。

(1) 内容的全面性。本书对 C# 语言以及使用 C# 语言开发 ASP.NET 应用程序过程中涉及的知识做了全面而系统的介绍,非常适合初学者从头开始学习。

(2) 代码的通用性。本书的内容循序渐进,在对相关基础知识的介绍后,针对 ADO.NET 和 LINQ to SQL 技术分别给出了一组使用 C# 语言对 SQL Server 数据库表进行增加、删除、修改和查询操作的通用代码,读者可简单地对该代码进行修改并在新的软件项目中加以应用。这部分内容为初学者省去了漫长而曲折的摸索过程,极大地提高读者的学习和开发效率;对于有一定 C# 语言基础的读者也具有很好的提高作用。

(3) 开发方法的规范性。本书在给出了开发 ASP.NET 应用程序的 C# 通用代码之后,又进一步对如何开发分层的 Web 应用程序做了详细介绍,重点介绍了 MVC 模式。对应用程序进行分层开发是企业开发较大型软件的常用方法。

(4) 代码的便捷性。书中所有的代码都可在清华大学出版社网站上下载,读者可直

接运行它来进行学习。

本书的章节安排

本书共包含 13 章,循序渐进地讲述了 C# 语言、Web 程序设计基础、数据库基础以及 ASP.NET 程序设计的有关知识。

第 1 章对 .NET 的基础知识进行介绍。具体介绍 .NET Framework 的基本理论、Visual Studio.NET2008 的安装和 ASP.NET 应用程序的开发流程,为后续章节的学习打下一个良好的基础。

第 2、3 章详细介绍 C# 1.1 的知识。其中第 2 章介绍 C# 语言的基本语法知识,第 3 章介绍 C# 语言面向对象的知识。第 2、3 章的内容是学习后续章节的基础。

第 4 章对 C# 2.0 中的新增特性进行系统全面的介绍。泛型是 C# 2.0 中引入的最重要的特性,使用它可以方便地开发高效安全的 C# 代码。除泛型之外,还介绍 C# 2.0 中引入的其他新特性。本章的内容是学习后续章节的基础。

第 5 章对 C# 3.0 中的新增特性进行系统全面的介绍。LINQ 是 C# 3.0 中引入的最重要的特性,使用 LINQ 技术可以非常方便地操作数据集。除 LINQ 之外,本章还介绍 C# 3.0 中引入的其他新特性。本章的内容是学习后续章节的基础。

第 6 章对 Web 程序设计的基础知识进行了介绍,具体包括 C/S 模式、B/S 模式、Web 工作原理、HTML、JavaScript 和 CSS 等。这些知识对于理解 Web 程序设计是必不可少的内容。

第 7、8 章分别对 ASP.NET 服务器控件和内置对象进行了详细的介绍,服务器控件和内置对象都是开发 ASP.NET 应用程序的常用元素。

第 9 章介绍数据库的基本知识,目的是为在第 10 章讲述如何使用 ADO.NET 访问数据库以及第 11 章如何使用 LINQ to SQL 访问数据库做准备。

第 10、11 章的内容是对前面章节内容的综合应用,分别针对 ADO.NET 和 LINQ to SQL 给出了一组使用 C# 语言对 SQL Server 数据库表进行增加、删除、修改和查询操作的通用代码。这些代码无论是对于初学者还是具有一定 C# 语言基础的读者都是十分有益的,读者通过简单修改可应用在自己开发的项目中。

第 12 章分别对 XML 和 ASP.NET 应用程序的配置进行介绍,其中 XML 是学习 ASP.NET 应用程序配置的基础知识。

第 13 章对 MVC 设计模式和 ASP.NET 应用程序的分层设计进行介绍。本章内容对于初学者理解如何设计较大型的 ASP.NET 程序有较好的帮助。这部分内容是在前面内容基础之上的再提高。

本书由王振武主编,参加本书编写的还有王振华、孙绪林、闫志波、冯永刚等。本书提供教学课件和程序代码,读者可登录 <http://edu.tqbooks.net> 网站自行下载。由于编者水平有限,本书必定存在不少缺点和不足之处,恳请专家和读者批评指正。

作 者

2012 年 6 月

目 录

第 1 章 Microsoft.NET 简介	1
1.1 Microsoft.NET 概述	1
1.1.1 什么是 Microsoft.NET	1
1.1.2 Microsoft.NET 与 Java EE 比较	2
1.2 Microsoft.NET Framework 介绍	3
1.2.1 公共语言运行库	3
1.2.2 .NET Framework 各版本之间的关系	5
1.2.3 Microsoft.NET Framework 类库	7
1.2.4 Microsoft.NET Framework 与 Visual Studio.NET 的关系	8
1.3 Visual Studio.NET IDE 介绍	9
1.3.1 Visual Studio.NET 2008 开发环境简介	9
1.3.2 利用 Visual Studio.NET 2008 开发 ASP.NET 应用程序	10
小结	12
习题	12
第 2 章 C# 语法基础	13
2.1 C# 概述	13
2.1.1 C# 的产生	13
2.1.2 C# 与 Microsoft.NET 的关系	14
2.1.3 C# 的语法元素	14
2.1.4 C# 的特点	15
2.2 变量和常量	16
2.2.1 变量	16
2.2.2 常量	17
2.3 数据类型	17
2.3.1 值类型	18
2.3.2 引用类型	22
2.3.3 类型转换	24

2.4	运算符和表达式	26
2.4.1	算术运算符和算术表达式	26
2.4.2	赋值运算符和赋值表达式	28
2.4.3	关系运算符和关系表达式	28
2.4.4	条件运算符和条件表达式	29
2.4.5	逻辑运算符和逻辑表达式	30
2.4.6	其他运算符和表达式	30
2.4.7	运算符的优先级	32
2.5	流程控制语句	33
2.5.1	条件语句	33
2.5.2	循环语句	34
2.5.3	跳转语句	36
2.6	异常处理	37
2.6.1	异常处理机制	37
2.6.2	常见的异常类	38
2.6.3	抛出异常	38
	小结	39
	习题	39
第3章	C# 面向对象基础	40
3.1	面向对象编程概述	40
3.1.1	面向对象编程方法学	40
3.1.2	面向对象的程序设计步骤	40
3.2	命名空间	42
3.2.1	命名空间的概念	42
3.2.2	命名空间的定义和引用	42
3.3	类和对象	43
3.3.1	类和对象的关系	43
3.3.2	类的定义	43
3.3.3	类的成员和访问控制	44
3.4	属性和索引器	45
3.4.1	属性	46
3.4.2	索引器	47
3.5	方法	49
3.5.1	方法的声明	49
3.5.2	返回类型和形式化参数	50
3.5.3	方法的重载	50
3.6	接口	51

3.6.1 接口的声明	52
3.6.2 接口的实现	52
3.7 继承与多态	53
3.7.1 继承	53
3.7.2 多态	55
3.8 委托与事件	56
3.8.1 委托	56
3.8.2 事件	58
小结	61
习题	61
第4章 泛型编程	62
4.1 C#泛型概述	62
4.1.1 为什么需要泛型	62
4.1.2 C#泛型的定义	65
4.1.3 C#泛型编译机制	65
4.2 C#泛型类型	66
4.2.1 C#泛型类与结构	66
4.2.2 C#泛型接口	69
4.2.3 C#泛型委托	69
4.3 C#泛型方法	70
4.3.1 C#泛型方法简介	70
4.3.2 C#泛型方法的声明与调用	70
4.3.3 C#泛型方法的重载	70
4.3.4 C#泛型方法的重写	71
4.4 C#泛型约束	71
4.4.1 C#泛型约束简介	71
4.4.2 基类约束	72
4.4.3 接口约束	72
4.4.4 默认构造器约束	73
4.4.5 值类型约束	73
4.4.6 引用类型约束	74
4.5 泛型集合类	74
4.5.1 泛型集合接口	74
4.5.2 列表	76
4.5.3 堆栈和队列	76
4.5.4 哈希表	77
4.6 匿名方法与迭代器	77
4.6.1 为什么需要匿名方法	77

4.6.2	匿名方法的语法	78
4.6.3	匿名方法的运行机制	79
4.6.4	什么是迭代器	81
4.6.5	迭代器的运行机制	82
4.7	C# 2.0 中的其他新特性	82
4.7.1	分部类型	82
4.7.2	空属类型	84
4.7.3	静态类	84
小结		85
习题		85
第 5 章	LINQ 技术	86
5.1	LINQ 简介	86
5.1.1	为什么需要 LINQ	86
5.1.2	什么是 LINQ	87
5.1.3	LINQ 简单示例	88
5.2	C# 3.0 中新的语法特性	88
5.2.1	自动属性	89
5.2.2	对象与集合的初始化器	90
5.2.3	隐式类型的局部变量	91
5.2.4	匿名类型	92
5.2.5	扩展方法	93
5.2.6	Lambda 表达式	94
5.3	LINQ 查询简介	96
5.3.1	获取数据源	97
5.3.2	创建查询	97
5.3.3	执行查询	98
5.4	LINQ 标准查询操作符	99
5.4.1	from 操作符	100
5.4.2	where 操作符	104
5.4.3	select 操作符	104
5.4.4	group 操作符	105
5.4.5	into 操作符	106
5.4.6	排序操作符	106
5.4.7	let 操作符	107
5.4.8	join 操作符	107
5.5	查询语法和方法语法	107
5.5.1	两种方法的比较	107

5.5.2 两种方法的选择·····	109
小结·····	109
习题·····	109
第6章 Web 程序设计基础 ·····	110
6.1 软件体系结构的演变 ·····	110
6.1.1 软件体系结构·····	110
6.1.2 C/S 模式的特点·····	110
6.1.3 B/S 模式的特点·····	111
6.2 Web 工作原理 ·····	111
6.2.1 什么是 Web ·····	111
6.2.2 Web 浏览器的工作原理 ·····	112
6.2.3 服务器端技术·····	113
6.2.4 客户端技术·····	114
6.3 HTML 基础知识·····	115
6.3.1 HTML 标记简介 ·····	115
6.3.2 HTML 文档示例 ·····	118
6.4 JavaScript 简介 ·····	120
6.4.1 JavaScript 在客户端的应用 ·····	121
6.4.2 在 HTML 中使用 JavaScript ·····	121
6.5 CSS 简介 ·····	122
6.5.1 在 HTML 文档中引用 CSS ·····	122
6.5.2 规则和样式表·····	122
6.5.3 对选择器和规则进行分组·····	123
小结·····	123
习题·····	124
第7章 ASP.NET 服务器控件 ·····	125
7.1 ASP.NET 服务器控件概述 ·····	125
7.1.1 ASP.NET 服务器控件简介 ·····	125
7.1.2 ASP.NET 服务器控件的特点 ·····	126
7.2 文本输入控件 ·····	126
7.2.1 Label 控件·····	126
7.2.2 TextBox 控件·····	126
7.2.3 ListBox 控件 ·····	128
7.3 选择控件 ·····	129
7.3.1 RadioButton 控件 ·····	129
7.3.2 CheckBox 控件 ·····	131

7.3.3	DropDownList 控件	131
7.3.4	Calendar 控件	133
7.4	按钮控件	134
7.4.1	Button 控件	134
7.4.2	LinkButton 控件	136
7.4.3	ImageButton 控件	137
7.5	导航控件	138
7.5.1	SiteMapPath 控件	138
7.5.2	Menu 控件	139
7.5.3	TreeView 控件	141
7.6	验证控件	142
7.6.1	RequiredFieldValidator 控件	142
7.6.2	CompareValidator 控件	143
7.6.3	RangeValidator 控件	145
7.6.4	RegularExpressionValidator 控件	145
7.6.5	CustomValidator 控件	146
7.6.6	ValidationSummary 控件	147
小结		150
习题		150
第 8 章	ASP.NET 内置对象	151
8.1	ASP.NET 内置对象概述	151
8.2	Page 对象	152
8.3	Request 对象	154
8.3.1	Form 属性	155
8.3.2	QueryString 属性	156
8.3.3	ServerVariables 属性	157
8.3.4	Browser 属性	159
8.3.5	ClientCertificate 属性	160
8.4	Response 对象	162
8.4.1	Write()方法和 Redirect()方法	163
8.4.2	与缓冲区有关的属性和方法	164
8.5	Server 对象	165
8.5.1	ScriptTimeout 属性	166
8.5.2	Execute 方法与 Transfer 方法	166
8.5.3	MapPath 方法	166
8.6	Session 对象	167
8.7	Cookie 对象	171

8.8 Application 对象	175
小结	178
习题	178
第 9 章 数据库知识简介	179
9.1 数据库知识概述	179
9.1.1 数据库管理系统	179
9.1.2 数据库对象	180
9.1.3 什么是 SQL	181
9.2 操作基本表	181
9.2.1 创建基本表	181
9.2.2 插入数据	182
9.2.3 修改数据	183
9.2.4 查询数据	183
9.2.5 删除数据	184
9.3 操作视图	184
9.3.1 创建视图	184
9.3.2 删除视图	185
9.4 索引	185
9.4.1 创建索引	186
9.4.2 删除索引	186
9.5 存储过程	186
9.5.1 什么是存储过程	186
9.5.2 创建存储过程	187
小结	188
习题	188
第 10 章 使用 ADO.NET 访问数据库	189
10.1 ADO.NET 简介	189
10.1.1 ADO.NET 概述	189
10.1.2 Connection 类	191
10.1.3 Command 类	192
10.1.4 DataReader 类	193
10.1.5 DataAdapter 类	194
10.1.6 DataSet 类	194
10.2 使用 ADO.NET 对数据库进行操作	195
10.2.1 利用 insert 语句插入记录	197
10.2.2 利用 select 语句查询记录	199

10.2.3	利用 update 语句更新记录	202
10.2.4	利用 delete 语句删除记录	205
10.3	使用 GridView 控件操作数据库	207
10.3.1	绑定查询结果功能	207
10.3.2	更新和删除记录功能	210
10.3.3	分页与排序功能	215
小结		217
习题		217
第 11 章	使用 LINQ to SQL 访问数据库	218
11.1	LINQ to SQL 简介	218
11.1.1	LINQ to SQL 概述	218
11.1.2	LINQ to SQL 查询过程	220
11.2	实体类的建立	220
11.2.1	LINQ to SQL 对象模型	220
11.2.2	实体类建立的方法	224
11.2.3	跟踪和提交实体变更	230
11.2.4	并发变更和并发冲突	230
11.3	使用 DataContext 操作数据库	231
11.3.1	DataContext	231
11.3.2	强类型的 DataContext	232
11.3.3	操作数据库单一表格	235
11.3.4	操作数据库一对多表格	238
11.3.5	使用事务	244
11.4	LINQ to SQL 的查询方式	245
11.4.1	查询的延迟执行与立即执行	245
11.4.2	远程查询执行与本地查询执行	245
11.4.3	已编译查询	246
小结		247
习题		247
第 12 章	ASP.NET 应用程序配置	248
12.1	XML 概述	248
12.1.1	什么是 XML	248
12.1.2	XML 的语法结构	249
12.2	格式良好的 XML 和有效的 XML	251
12.2.1	格式良好的 XML	251
12.2.2	有效的 XML	251

12.3	XML 与 DataSet 对象	253
12.3.1	读取 XML 文档	253
12.3.2	写入 XML 文档	253
12.4	Web.config 文件的配置	255
12.4.1	使用 Web.config 文件放置常量	256
12.4.2	使用 Web.config 文件进行安全性配置	256
12.4.3	使用 Web.config 文件配置其他内容	264
12.5	Global.asax 文件的配置	265
12.6	ASP.NET 缓存技术	266
12.6.1	页面输出缓存	267
12.6.2	应用程序数据缓存	267
小结		269
习题		269
第 13 章	MVC 及 ASP.NET 分层程序设计	270
13.1	设计模式简介	270
13.1.1	什么是设计模式	271
13.1.2	为什么需要设计模式	271
13.1.3	常见的设计模式	271
13.2	MVC 设计模式简介	272
13.2.1	MVC 模式原理	272
13.2.2	ASP.NET Web 窗体与 MVC 模式	273
13.2.3	MVC 模式的优点和缺点	273
13.3	ASP.NET 中常用的 MVC 框架	275
13.3.1	NHibernate 框架	275
13.3.2	Spring.NET 框架	275
13.3.3	ASP.NET MVC 框架	275
13.4	ASP.NET 三层架构程序设计	275
13.4.1	创建表示层	276
13.4.2	创建业务逻辑层	279
13.4.3	创建数据访问层	294
小结		297
习题		298
参考文献		299

第 1 章 Microsoft.NET 简介

通过本章的学习,读者可以理解 Microsoft.NET 的含义,并对 Microsoft.NET 与 Java EE 的区别有清晰的认识;在此基础上,读者将循序渐进地学习 .NET Framework 的原理,它是 Microsoft.NET 的核心部分;最后读者将了解如何使用 Visual Studio.NET IDE 以及它与 .NET Framework 的关系。这些内容都是从宏观上介绍 Microsoft.NET 技术,通过该章的学习可为后续章节打下基础。

学习目标:

- 了解 Microsoft.NET 的含义。
- 理解 .NET Framework 的原理。
- 掌握 VS.NET2008 的安装以及 ASP.NET 应用程序的开发流程。

1.1 Microsoft.NET 概述

1.1.1 什么是 Microsoft.NET

Microsoft.NET(以下简称.NET)是微软公司推出的下一代计算计划,该计划使微软现有的软件在 Web 时代不仅适用于传统的 PC 设备,而且也能满足新设备、新业务的要求。如今,.NET技术和相应的产品已经得到了广泛应用,从中小型的应用程序到大型的企业级应用软件,从桌面应用程序、Web 应用程序、Mobile 应用程序、Web 服务到操作系统开发和嵌入式设备开发等,都能看到.NET的身影。

对.NET可以从两个方面理解。首先,.NET是一个开发平台。它对微软之前的各种主要开发平台进行了集成,提供了一套全新的 Windows 平台实现。例如,在.NET平台下不仅可以从事 VB、C++ 程序的开发,而且还可以使用特别为.NET平台开发的 C# 语言进行编程。.NET平台要做的就是消除互连环境中不同软硬件以及服务的差异,使得不同设备和系统都可以相互通信,使得不同的程序和服务之间都可以相互调用。其次,.NET是一组规范。.NET平台本身就基于一系列规范,其中有些规范是由微软以外的其他组织来维护。例如定义诸如 C#、VB.NET 和 IL 语言的规范,定义数据交换格式的规范,如 SOAP 等。.NET平台的组成如图 1-1 所示。

图 1-1 的最底层的是操作系统,它是各种设备得以运行的基础。这里说的操作系统不但可以是 Windows 系列操作系统,也可以是其他操作系统。但需要说明的是,微软出

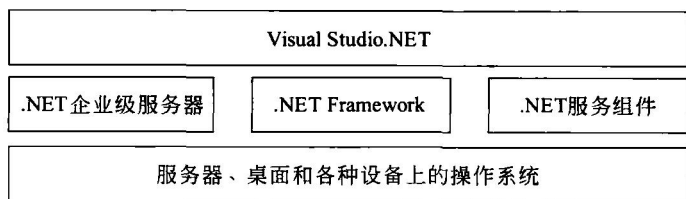


图 1-1 Microsoft.NET 平台的组成

于商业考虑,并未跨域其他操作系统,但从原理上讲,.NET跨越平台是完全可行的。微软全新的操作系统(如 Windows 7、Windows 8)中都带有 .NET Framework,如果想在其他操作系统上运行 .NET 应用程序,需要先在該操作系统上安装公共语言运行库。

.NET 企业级服务器是 .NET 平台的另一个组成部分。这些企业级服务器包括 Application Center、BizTalk Server、Commerce Server、Exchange Server、Host Intergration Server、Internet Security and Acceleration Server 和 SQL Server 等。这些服务器都是为特定的企业级应用所设计,在互联的、分布式环境中它们必不可少。当然,.NET 技术也支持其他厂商的服务器产品,如 Oracle 数据库等。

.NET Framework 是 .NET 平台的核心部分,.NET 应用程序都必须运行在 .NET Framework 下,关于 .NET Framework 的详情在 1.2 节中介绍。.NET 服务组件指的就是 XML Web Service,它是 .NET 平台中的关键性技术。Web 服务作为一种全新的开发模式,是构建下一代互联网的关键技术。

Visual Studio.NET 是微软推出的全新的 .NET 开发工具,最新的版本是 Visual Studio.NET 2010。Visual Studio.NET 对微软之前的主要开发工具做了全新集成并有了质的飞跃,它内置支持 Visual Basic.NET、Visual C#.NET、Visual C++.NET 和 Visual J#.NET 等多种语言,并且各种语言拥有统一的开发环境,可以进行跨语言调试和跨语言调用。需要说明的是 Visual Studio.NET 并不是对 Microsoft Visual C++、Microsoft Visual Basic 等语言的简单集成,Visual C++.NET、Visual Basic.NET 等语言是基于全新的设计思想开发的。当然,Visual Studio.NET 并不是开发人员的唯一选择,其他开发工具(如 C# Builder)也可以选择。

1.1.2 Microsoft.NET 与 Java EE 比较

Java EE(Java Platform, Enterprise Edition)是 Sun 公司推出的一套开发企业级应用的技术架构,用它可以设计、开发、装配及部署 Java 类型的应用程序,并且开发的应用程序具有跨平台的特性,它的早期版本称为 J2EE。而对于 .NET 而言,只要在任何一个操作系统上安装了公共语言运行库(CLR)就可以运行 .NET 应用程序。所以从这一点上来说,.NET 与 Java EE 非常相似,但这两种技术的实现方法和具体实现技术还是存在很大差别,具体区别如下:

(1) 两者支持的开发语言种类不同。.NET 平台支持 C++、VB、C# 和 J# 等语言,开发人员可以很容易找到适合自己的语言,而 Java EE 只支持 Java 语言,开发人员选择面