

- 详细讲述C#.NET网络程序设计方法
- 内容安排和叙述风格采用“教与学面对面”方式
- 理论结合实际，通过具体示例巩固所学知识
- 免费提供PPT电子课件以及书中示例的源代码

C#.NET

网络程序设计实用教程

于国防 李 剑 编著

C#.NET

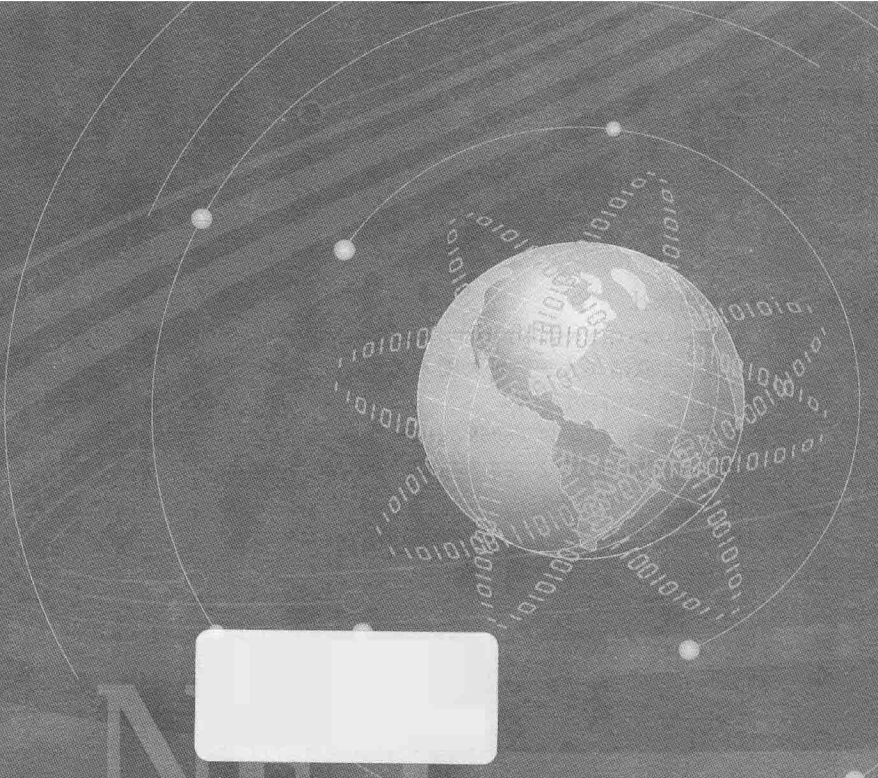


清华大学出版社

C#.NET

网络程序设计实用教程

于国防 李 剑◎编著



清华大学出版社
北 京

内 容 简 介

本书以 C#语言和 Visual Studio.NET 平台为技术基础,以网络程序设计为主题,结合应用示例,将全部内容统筹规划为 3 篇共 19 章,先后讲述 Windows 网络程序设计和 ASP.NET 网站创建及其程序设计。两种程序设计同书对比、前后衔接,更有利于知识的分析理解、融会贯通,达到事半功倍的学习效果。对于全书的技术阐述,基础知识点到为止,专业技术简明实用;内容描述力求通俗易懂,避免专业晦涩。

本书既可作为高等学校计算机网络或通信工程等相关专业的教材,也可作为 C#.NET 程序设计人员入门与提高的参考书。

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#.NET 网络程序设计实用教程/于国防,李剑编著. —北京:清华大学出版社,2013

ISBN 978-7-302-34431-5

I. ①C… II. ①于… ②李… III. ①C 语言-程序设计-教材 IV. ①TP312

中国版本图书馆 CIP 数据核字(2013)第 265937 号

责任编辑:朱英彪 贾小红

封面设计:刘 超

版式设计:文森时代

责任校对:王 云

责任印制:何 芊

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 装 者:三河市李旗庄少明印装厂

经 销:全国新华书店

开 本:185mm×260mm

印 张:22

字 数:520 千字

版 次:2013 年 12 月第 1 版

印 次:2013 年 12 月第 1 次印刷

印 数:1~3800

定 价:39.00 元

产品编号:053088-01

前言

Microsoft.NET 与 Visual C#语言在程序设计中的应用,可谓珠联璧合,并且正逐渐成为业界程序设计人员的首选。而网络技术的飞速发展和网络应用的广泛普及,也更加促进了对于 C/S 和 B/S 模式的 Windows 和 Web 网络程序设计的需求。特别是微软的 ASP.NET 技术,将客户端浏览器网页中的程序(原本为脚本),转到了服务器端运行,使之成为了真正的程序。也正因为如此,对于基于.NET 技术的开发,Windows 与 Web 程序设计之间必然存在着密不可分的联系或相似之处,所以将二者的学习有机地融入一书分析讲解,不能不说是一个事半功倍的重要举措,这也正是我们编著此书、联合讲解基于.NET 和 C#的网络程序设计的初衷。

全书分 3 篇共 19 章,章节内容前后衔接,技术要点逐层深入;示例分析图文并茂;代码注释详尽明晰;重点问题随时“说明”,引申知识及时“提示”。总之,全书的编排讲解将最大限度地降低读者的学习难度,提高读者的学习效率。各章内容概括如下。

第 1 章:以有限的篇幅介绍了.NET 和 C#的基本知识,为后续相关技术的学习奠定基础。

第 2 章:理解网络系统的体系结构,就能更好地学习和掌握本书将要学习的 Windows 和 Web 程序设计。同时,Visual Studio 2010 的功能介绍及其安装方法,也为后续的程序设计实战准备好了称手的兵器。

第 3 章:简要介绍面向对象编程的基本知识,重点引入了.NET 的命名空间概念及其用法说明,此乃 C#程序设计的重要技术基础。

第 4 章:变量、数据类型、运算符以及流程控制等可谓任何编程语言的必备基础,C#语言当然亦不例外,所以,本章既有侧重性,又简明扼要地分析讲解了这些基础知识。

第 5 章:以一个简单的 Windows 上机自测系统的开发为例,将窗体(含控件)、菜单以及工具栏的创建及其相应的程序设计有机地融为一体。轻松的设计、实际的操作,令初学者自信地跨入 C#语言程序设计的大门!

第 6 章:进程和线程是实时或多任务应用程序开发的技术基础。结合形象、直观的示例程序,并采用类比式的分析方法,深入浅出,重点讲述多线程开发,并为后续的网络程序设计奠定相关的技术基础。

第 7 章:重点讲述最基本、最常用的文件读/写程序设计,为后续的网络文件流操作奠定技术基础。

第 8 章:网络编程是学习的重点,也是学习的难点。遵循由浅入深的原则,结合具体示例,分别讲解了基于 TCP 协议、FTP 和 UDP 协议的网络程序设计。文中的具体分析、代码中的详细注释,都极大地增强了读者对本章知识的理解和掌握。

第 9 章:首先用精简的篇幅以 SQL Server 为主介绍数据库的基本知识,然后在此基础上逐步分析讲解了 ADO.NET 数据库访问技术,最后将该技术用于示例网站的程序设计中。

第 10 章：简要介绍邮件发送和接收的基本原理及其相应的协议，分别结合微软 .NET Framework 的 System.Net.Mail 类和第三方的 LumiSoft.Net 组件，分析讲解了邮件发送和接收的 Windows 程序设计。

第 11 章：用有限的篇幅和详细步骤，介绍了最简单、实用的 Windows 程序的安装和部署方法。

第 12 章：何为 Web 应用程序，为何采用 ASP.NET 开发技术，以及如何选用及配置相应的 Web 服务器来构建程序的运行环境，通过本章的学习将使读者对此一目了然。

第 13 章：功能网页内容、主页风格以及页面超链接跳转设计规划后，合理地运用母版页、目录树以及网站地图，一个看似简单却功能实用的示例网站就初见端倪了。从中足见 Visual Studio 2010 工具使用之简单，ASP.NET 网站开发功能之强大，怎能不使读者的学习信心倍增？

第 14 章：正如 Windows 程序设计时会用到许多窗体控件一样，Web 程序设计也要用到一些 Web 服务器控件，这些控件与窗体控件有许多相似之处，所以，本章将在相关知识的基础上，对比式地、有侧重地分析介绍，最后将其运用于示例网站内容页建设中。

第 15 章：网页之间经常要进行诸如参数传递、数据输出或信息记录等操作，这些功能的实现就需要借助 ASP.NET 的内置对象。

第 16 章：ADO.NET 在数据库访问的 Windows 与 Web 程序设计中的原理与用法几乎一样，所以，本章仅在现有知识的基础上，补充性地拓展相关技术，重点是进一步完善示例网站中各个内容页的相关程序设计。

第 17 章：虽然基于 Windows 与 Web 环境的邮件发送与接收的原理相同，但二者的程序设计及实现过程还是有一定的差别。所以，本示例网站中又融入了邮件应用方面的程序设计，并且为了拓展这方面的相关知识，特别介绍了 JMail 组件的邮件接收。

第 18 章：AJAX 技术可谓 Web 程序设计中不得不学、不得不会的一项基本开发技能，所以，在介绍了 AJAX 的技术原理后，书中结合示例网站一个内容页的功能改进的例子，对其进行讲解，使读者在一定程度上对 AJAX 的特点和用法有所了解。

第 19 章：犹如 Windows 程序开发完成后还需要安装和部署一样，Web 程序也需要生成，特别是需要发布。不过，若仅用于程序在 Web 服务器上的移植，这一过程相比于 Windows 程序来说要简单许多。

本书由于国防和李剑共同编著。李剑编写了第 1~3、7、11、12 和 14 章，于国防编写了其他 12 章，全书由于国防统稿。在本书的编写过程中，还得到了张玉杰、杜文龙和庄立运几位老师的协助，在此表示衷心的感谢！

作为本书的编者，并身为高校教师，我们深知编书助教的责任之重，所以，我们为此书的编写竭尽全力、精益求精，但即便如此，仍不无谦虚地承认：由于水平所限，书中可能会有疏漏或不妥之处，殷切希望广大读者、同仁批评指正。

本书提供所有示例的源程序、多媒体教学 PPT 电子教案以及全部实验的参考源程序，可到 www.tup.com.cn 网站下载。

编 者

教学（实验）计划

教 学	学 时	实 验	学 时
第 1 篇 基础知识	4		
第 1 章 .NET 与 C#简介	0.5		
第 2 章 网络系统体系结构与 Visual Studio.NET	0.5		
第 3 章 面向对象程序设计基础	1		
第 4 章 C#语言基础	2		
第 2 篇 Windows 程序设计	20		/8
第 5 章 Windows 窗体与控件应用及其程序设计	4	实验 1	/1
第 6 章 Windows 进程与线程程序设计	4	实验 2	/1
第 7 章 输入与输出	2	实验 3	/1
第 8 章 Windows 网络程序设计	6	实验 4	/2
第 9 章 数据库访问 Windows 程序设计	1	实验 5	/1
第 10 章 邮件发送与接收 Windows 程序设计	2.5	实验 6	/2
第 11 章 Windows 程序的安装和部署	0.5		
第 3 篇 Web 程序设计	14		/6
第 12 章 Web 程序、ASP.NET 及 Web 服务器	1		
第 13 章 ASP.NET 网站设计基础	3		
第 14 章 常用 Web 服务器控件分析与应用	2	实验 7	/1
第 15 章 常用 ASP.NET 内置对象	2		
第 16 章 数据库访问 ASP.NET 程序设计	2	实验 8	/2
第 17 章 示例网站的邮件程序设计	2	实验 9	/2
第 18 章 AJAX 简介与应用	1.5	实验 10	/1
第 19 章 ASP.NET 网站的生成与发布	0.5		
小计	38	小计	14
合计：52			

注：

① 以上教学计划是按照 52 总学时制定的，如果实际教学为 32 或 56 学时，可适当增减实验学时数，本计划中未列出的本书其他实验可作为选做或者附加内容。

② 根据对 Windows 或 Web 程序设计的要求或侧重点的不同，实际教学过程中的内容也可灵活调整。

③ 实验学时数前边的斜杠（如/2），表示该实验可能被选做，选做的实验总学时为表中的“小计”数量。

目 录

第1篇 基础知识

第1章 .NET 与 C#简介..... 1

1.1 .NET 简介..... 1

1.1.1 .NET 平台..... 1

1.1.2 .NET 框架..... 2

1.1.3 公共语言运行库 CLR..... 3

1.1.4 .NET Framework 类库..... 4

1.2 C#简介..... 5

1.2.1 C#语言的由来..... 5

1.2.2 C#语言的特点..... 6

习题..... 7

第2章 网络系统体系结构与 Visual Studio.NET..... 8

2.1 网络系统体系结构简介..... 8

2.2 Visual Studio.NET 简介及 Visual Studio 2010 安装..... 9

2.2.1 Visual Studio.NET 的发展历史和 优点..... 9

2.2.2 Visual Studio 2010 集成开发环境 安装..... 10

习题..... 16

第3章 面向对象程序设计基础..... 17

3.1 面向对象编程简介..... 17

3.1.1 类和对象..... 17

3.1.2 面向对象程序设计的相关概念..... 17

3.2 类..... 18

3.2.1 类的声明与实例..... 18

3.2.2 类成员..... 19

3.3 方法..... 21

3.3.1 方法的声明与调用..... 21

3.3.2 方法的参数..... 21

3.3.3 方法重载..... 23

3.4 C#语言编程初试..... 23

3.4.1 项目与解决方案..... 23

3.4.2 第一个简单的 C#程序设计..... 24

3.4.3 命名空间..... 25

3.4.4 Main()方法..... 26

3.4.5 代码注释..... 26

3.5 事件..... 27

习题..... 28

第4章 C#语言基础..... 29

4.1 标识符和关键字..... 29

4.1.1 标识符..... 29

4.1.2 关键字..... 30

4.2 值类型..... 30

4.2.1 简单值类型..... 31

4.2.2 结构类型..... 33

4.2.3 枚举类型..... 34

4.3 引用类型..... 35

4.3.1 object 类型..... 36

4.3.2 类类型..... 36

4.3.3 字符串类型..... 37

4.3.4 数组类型..... 38

4.4 类型转换..... 39

4.5 常量和变量..... 41

4.5.1 常量..... 41

4.5.2 变量..... 41

4.6 表达式和运算符..... 43

4.6.1 表达式..... 43

4.6.2 运算符..... 43

4.7 流程控制语句	46	4.7.3 跳转语句	50
4.7.1 条件语句	46	4.8 异常处理语句	52
4.7.2 循环语句	48	习题	53

第 2 篇 Windows 程序设计

第 5 章 Windows 窗体与控件应用及

其程序设计

5.1 窗体与控件简介	54
5.1.1 窗体	54
5.1.2 控件	55
5.2 一个基于 Win Forms 的上机自 测系统设计	55
5.2.1 Windows 窗体控件分析与应用	57
5.2.2 PictureBox 窗体控件分析与应用	58
5.2.3 Label 窗体控件分析与应用	60
5.2.4 添加新窗体	61
5.2.5 TextBox 窗体控件分析与应用	62
5.2.6 RadioButton 与 Panel 窗体控件分析 与应用	63
5.2.7 ListBox 与 ComboBox 窗体控件分析 与应用	65
5.2.8 CheckBox 窗体控件分析与应用	66
5.2.9 Timer 窗体控件分析与应用	67
5.2.10 Button 窗体控件分析与应用	69
5.2.11 启动窗体设置	71
5.2.12 MenuStrip 窗体控件分析与应用	72
5.2.13 ToolStrip 窗体控件分析与应用	74
5.3 Windows 窗体的文档界面设置及 其管理	75
5.3.1 基于单文档 (SDI) 的应用程序	76
5.3.2 基于多文档 (MDI) 的应用程序	76
5.3.3 自测系统多文档管理	76
5.4 Windows 窗体的显示模式分析与 应用	79
5.4.1 模态窗体	79
5.4.2 非模态窗体	80
5.5 Windows 通用对话框简介	81

5.5.1 MessageBox 消息对话框	81
5.5.2 OpenFileDialog 对话框控件	83
5.5.3 SaveFileDialog 对话框控件	84
习题	85

第 6 章 Windows 进程与线程程序

设计

6.1 进程与线程简介	86
6.2 进程程序设计	87
6.3 线程程序设计基础知识	89
6.4 多线程程序设计	94
6.4.1 多线程简介	94
6.4.2 多线程互斥与同步简介	95
6.4.3 多线程互斥程序设计	95
6.4.4 多线程同步程序设计	106
6.5 Windows 窗体控件的跨线程 访问	109
6.5.1 非线程安全的窗体控件访问	109
6.5.2 线程安全的窗体控件访问	109
6.5.3 线程安全的窗体控件访问程序 设计	110
习题	112

第 7 章 输入与输出

7.1 文件与流概述	113
7.2 System.IO 命名空间	113
7.3 文件对话框类	114
7.3.1 OpenFileDialog 类	114
7.3.2 SaveFileDialog 类	115
7.4 文件读写与流操作	115
7.4.1 文件流操作	115
7.4.2 网络流操作	120
习题	121

9.3	ADO.NET 数据库访问基础	170
9.3.1	ADO.NET 数据库访问步骤	171
9.3.2	SqlConnection 类与数据库连接	172
9.3.3	DataSet 数据集与 DataAdapter 数据适配器	174
9.3.4	SqlCommand 类与 SQL Server 数据库访问	177
9.3.5	利用 DataReader 类获取只读数据	181
9.3.6	利用 DataTable 类和 DataView 类 更新数据及控制视图	182
习题		186

10.1	电子邮件的发送	187
10.1.1	SMTP 协议简介	187
10.1.2	利用 System.Net.Mail 发送邮件的 程序设计	188
10.2	电子邮件的接收	193
10.2.1	POP3 协议简介	193
10.2.2	利用 LumiSoft.Net 接收邮件的程 序设计	193
10.3	IMAP 协议简介	204
习题		204

11.1	Windows 程序安装和部署 简介	205
11.2	Windows 程序安装和部署 方法	205
习题		210

第3篇 Web 程序设计

12.2	ASP.NET 简介.....	212
12.3	Web 服务器.....	213
12.3.1	Web 服务器简介.....	213

12.3.2 IIS 安装与配置.....	214	14.5.1 Image 服务器控件.....	252
习题.....	217	14.5.2 ImageMap 服务器控件.....	253
第 13 章 ASP.NET 网站设计基础.....	218	14.6 容器类服务器控件.....	254
13.1 一个邮件收、发及其用户管理		14.6.1 Panel 服务器控件.....	254
示例网站.....	218	14.6.2 View 和 MultiView 服务器控件.....	255
13.1.1 示例网站规划.....	218	14.7 增强类服务器控件.....	256
13.1.2 示例网站创建.....	221	14.7.1 Calendar 服务器控件.....	256
13.2 添加母版页.....	222	14.7.2 FileUpload 服务器控件.....	257
13.3 添加站点地图.....	225	14.8 验证类服务器控件.....	259
13.4 SiteMapDataSource、TreeView 及 SiteMapPath 控件分析与应用.....	227	14.8.1 RequiredFieldValidator 服务器 控件.....	259
13.4.1 SiteMapDataSource 控件分析与 应用.....	227	14.8.2 RangeValidator 服务器控件.....	260
13.4.2 TreeView 控件分析与应用.....	227	14.8.3 CompareValidator 服务器控件.....	261
13.4.3 SiteMapPath 控件分析与应用.....	229	14.8.4 RegularExpressionValidator 服务 器控件.....	262
13.5 添加网站主页.....	231	14.8.5 ValidationSummary 服务器控件.....	263
13.6 示例网站试运行.....	233	14.8.6 屏蔽验证.....	264
13.7 系统环境文件 Web.config 简介.....	235	14.9 服务器控件在示例网站中的 应用.....	264
习题.....	239	14.9.1 用户注册内容页的控件添加及 其布局.....	265
第 14 章 常用 Web 服务器控件分析与 应用.....	241	14.9.2 用户编辑内容页的控件添加及 其布局.....	266
14.1 文本类服务器控件.....	241	14.9.3 用户登录内容页的控件添加及 其布局.....	267
14.1.1 Label 服务器控件.....	241	14.9.4 邮件发送内容页的控件添加、 布局及程序设计.....	269
14.1.2 TextBox 服务器控件.....	242	14.9.5 邮件接收列表内容页的控件添加、 布局及程序设计.....	271
14.1.3 HyperLink 服务器控件.....	243	14.9.6 邮件详细信息内容页的控件添加 及其布局.....	272
14.2 按钮类服务器控件.....	244	习题.....	273
14.2.1 Button 服务器控件.....	244	第 15 章 常用 ASP.NET 内置对象.....	274
14.2.2 LinkButton 服务器控件.....	245	15.1 Response 对象.....	274
14.2.3 ImageButton 服务器控件.....	245	15.1.1 Response 对象的属性和方法.....	274
14.3 选择类服务器控件.....	246	15.1.2 Response 对象的常用方法应用 示例.....	275
14.3.1 CheckBox 服务器控件.....	246		
14.3.2 RadioButton 服务器控件.....	247		
14.4 列表类服务器控件.....	248		
14.4.1 ListBox 服务器控件.....	248		
14.4.2 DropDownList 服务器控件.....	250		
14.5 图像类服务器控件.....	252		

15.2 Request 对象	276	16.3.1 GridView 服务器控件	295
15.2.1 Request 对象的属性和方法	276	16.3.2 Repeater 服务器控件	295
15.2.2 利用 Request 对象读取表单提交 数据	277	16.4 示例网站用户管理程序设计	297
15.2.3 利用 Request 对象获取用户传递 数据	277	16.4.1 用户注册内容页程序设计	297
15.3 Server 对象	278	16.4.2 用户编辑内容页程序设计	300
15.3.1 Server 对象的属性和方法	278	16.4.3 用户登录内容页程序设计	306
15.3.2 Server 对象的常见应用	279	16.5 示例网站内容页登录及权限 设置程序设计	308
15.4 Application 对象	283	习题	310
15.4.1 Application 对象的属性、方法和 事件	283	第 17 章 示例网站的邮件程序设计	311
15.4.2 Application 对象变量值的存取	284	17.1 JMail 邮件组件简介	311
15.5 Session 对象	285	17.1.1 JMail 组件的安装与卸载	311
15.5.1 Session 对象的属性、方法和 事件	285	17.1.2 JMail 组件的引用	313
15.5.2 Session 对象变量值的存取	286	17.1.3 JMail 组件的常用对象及其常用 属性和方法	313
15.6 Cookie 对象	286	17.2 邮件收、发及信息内容页的程序 设计	315
15.6.1 Cookie 对象的属性和方法	286	17.2.1 邮件发送内容页程序设计	315
15.6.2 Cookie 对象的设置	287	17.2.2 邮件接收列表内容页程序设计	318
习题	287	17.2.3 邮件详细信息内容页程序设计	322
第 16 章 数据库访问 ASP.NET 程序 设计	288	习题	326
16.1 ASP.NET 的数据库访问对象	288	第 18 章 AJAX 简介与应用	327
16.1.1 Connection 对象	288	18.1 AJAX 技术简介	327
16.1.2 Command 对象	290	18.2 AJAX 应用示例	328
16.1.3 DataReader 对象	290	习题	331
16.1.4 DataTable 对象	291	第 19 章 ASP.NET 网站的生成与 发布	332
16.1.5 DataAdapter 对象	292	19.1 生成网站与重新生成网站	332
16.1.6 DataSet 对象	293	19.2 网站发布	332
16.1.7 DataView 对象	293	习题	334
16.2 DropDownList 与 ListBox 服务器 控件的数据绑定	294	参考文献	335
16.3 常用数据服务器控件	295	附录 A 实验参考	336

第1篇 基础知识

第1章 .NET 与 C#简介

学习要点

- 了解.NET平台和.NET Framework组织结构。
- 了解CLR的优点及其工作流程。
- 了解C#语言的由来及其特点。

1.1 .NET 简介

1.1.1 .NET 平台

微软公司总裁史蒂夫·鲍尔默对“什么是.NET”的解释是：.NET 代表了一个集成、一个环境、一个编程的基本结构，它可作为一个平台来支持下一代的互联网，是一个可实现的环境。比尔·盖茨对.NET 的评价是：这是一个崭新的平台，它将影响我们今后编写的每一行应用程序代码，它将重新定义用户界面，屏幕内容以及人机交互方式都将大为改观，如同从 DOS 到 Windows 的转变一样，微软的所有产品都将受到它的影响。他在 2000 年就明确了 Microsoft .NET 的战略：未来 5 年，我们的目标就是超越今天各自为营的 Web 站点，把 Internet 建成一个可以互相交换组件的地方。

诚如微软公司的预期和努力，如今.NET 已是一个可以支持下一代互联网服务和运营的平台。这个平台包含微软新一代的操作系统、大量的互联网服务软件、对各种设备（移动通信设备、机顶盒和信息家电等）的支持和应用软件开发套件——Visual Studio.NET 等。在这个平台下，从用户的角度看，只需发出请求，无论在什么设备上运行着什么操作系统，只要安装了.NET Framework，就可以运行.NET 可执行程序，用户就能获得所需要的服务，而无须关心后台的复杂操作；从开发人员的角度看，平台对开发进行强有力的支持，方便创建各式各样的应用软件（无论是传统的 Windows 应用软件还是 Web 站点），而且部署和发布应用程序更加简便。

.NET 平台结构如图 1-1 所示，它采用开放式的体系结构，集中体现了微软公司在软件设计领域的先进技术成就，其核心技术包括.NET Framework、.NET 企业服务器、模块服务、Visual Studio.NET。



图 1-1 .NET 平台结构

.NET Framework 是 .NET 战略的核心，它为 .NET 平台下应用程序的开发和执行提供了基本的环境架构。

.NET 企业服务器是一系列的技术服务产品，为用户提供了数据通信、协作、交换和存储等服务，包括 SQL Server、BizTalk Server、Office Communications Server、Host Integration Server 和 Exchange Server 等。

积木块服务主要是微软公司提供的 COM+ 组件服务和 XML Web 服务技术，在应用程序中作为功能模块调用，以便快速完成开发。

1.1.2 .NET 框架

.NET Framework (.NET 框架) 是支持生成和运行 .NET 应用程序以及 XML Web Services 的内部 Windows 组件。为了创建和运行基于 .NET 平台的应用程序，.NET Framework 提供了一个多语言组件开发、编译和运行的环境。图 1-2 是 .NET Framework 的组织结构图。

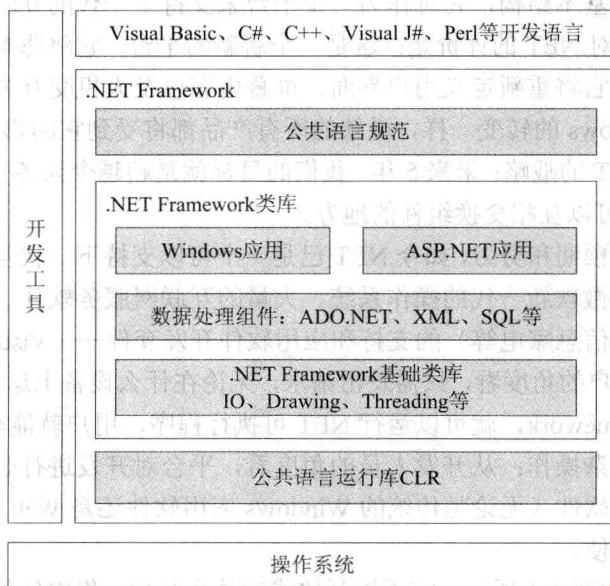


图 1-2 .NET Framework 的组织结构图

.NET Framework 主要包括 3 个部分：公共语言规范、.NET Framework 类库和公共语言

运行库。其中，公共语言运行库是 .NET Framework 的基础，其作用是托管和执行代码，并向托管代码提供内存管理、线程管理等服务；.NET Framework 类库是一个综合性的、面向对象的可重用类型集合，利用它可以开发多种应用程序；公共语言规范是公共语言运行库支持的语言功能的子集，包括几种面向对象的编程语言的通用功能。符合公共语言规范的组件和工具能够保证与其他符合公共语言规范的组件和工具进行交互操作。

.NET Framework 实现的目标如下：

- ☑ 对于不同的编程语言而言，提供一个统一的面向对象的编程环境；而对于同一种编程语言，无论对象代码是在本地存储和执行，还是在本地执行，在 Internet 上分布，或者是在远程执行，其编程环境是基本一致的。
- ☑ 代码执行环境将软件部署和版本控制冲突最小化。
- ☑ 代码执行环境能够提高代码执行的安全性。
- ☑ 使开发人员能够轻松地设计不同类型的应用程序，如传统的命令行、图形化的 Windows GUI（即 Windows 窗体应用程序）、ASP.NET 应用程序、XML Web Services 以及 Windows 服务等。
- ☑ 能够按照工业标准生成所有应用，以确保基于 .NET Framework 的代码可与其他任何代码集成。

.NET Framework 正式发布的版本从 1.0 开始，逐步升级为 1.1、2.0、3.0、3.5 和 4.0，并且仍在不断地升级中。.NET Framework 1.0、1.1、2.0 这 3 个版本是彼此完全独立的，可以同时位于一台计算机上，每个版本都有自己的公共语言运行库、类库和编译器等。需要注意的是，虽然 2.0 版本以上的 .NET Framework 具有向下兼容性，但是有的软件可能无法运行于高版本的 .NET Framework 中，所以仍需要安装相应版本的 .NET Framework。

1.1.3 公共语言运行库 CLR

公共语言运行库（Common Language Runtime，CLR）是 .NET Framework 的一大特色。CLR 是一个执行并管理代码的中枢，提供了内存管理、线程管理和远程处理等核心服务，并且强制实施严格的类型安全检查，实施提高安全性和可靠性的其他形式的动作。代码管理的概念是 CLR 的基本概念。以运行库为目标的代码称为托管代码，即在 CLR 下运行的代码都是托管代码，而不以运行库为目标的代码称为非托管代码。C# 语言中代码也能使用非托管方式，但是考虑到代码的安全性，除非必要，不建议这么做。托管代码具有许多优点，如跨语言集成、跨语言异常处理、安全性增强、版本控制和部署支持、简化的组件交互模型、调试和分析服务等。

CLR 的功能是通过编译器和工具公开的，因此，只要有基于 CLR 的语言编译器，开发人员就可以编写利用此托管执行环境的代码，而不管使用的是哪一种编程语言。

CLR 主要提供内存管理、线程管理、远程处理、异常处理、类加载、自动垃圾收集、安全和认证、通过 BCL 得到广泛的编程功能（包括 Web 服务和数据服务等）。

1. CLR 的优点

CLR 主要具有如下优点：

- ☑ 性能提高。程序执行的机器代码是通过 JIT 实时编译器生成的。
- ☑ 语言的互操作性。能够轻松使用其他语言开发的组件。
- ☑ 与使用的开发平台无关，可以使用第三方开发平台。
- ☑ 新的语言功能，如面向对象的编程的继承、接口和重载；结构化异常处理和自定义属性支持。

2. CLR 的工作流程（托管代码的执行过程）

CLR 的工作流程如下：

(1) 选择编译器。CLR 是一个多语言执行环境，支持各种数据类型和语言功能，开发人员可能使用不同的编程语言编写代码，因此必须选择针对该编程语言的编译器或第三方编译器。为了能够正确使用 CLR 的功能，开发人员使用的语法必须符合公共语言规范 CLS。

(2) 将代码编译为 Microsoft 中间语言 (MSIL)。当源代码编译为托管代码时，编译器将源代码编译为 Microsoft 中间语言，这是一组可以有效地转换为本机代码且独立于 CPU 的指令，不是 CPU 能够直接执行的代码。MSIL 包括用于加载、存储和初始化对象以及对对象调用方法的指令，还包括用于算术和逻辑运算、控制流、直接内存访问、异常处理和其他操作的指令。

(3) 将 MSIL 编译为本机代码。在执行时，实时 (JIT) 编译器将 MSIL 编译为本机代码。这种本机代码是针对特定 CPU 结构的 JIT 编译器生成的 (CLR 为每种 CPU 结构都提供了 JIT 编译器)。JIT 编译器并不是把整个应用程序一次编译完 (这样会有很长的启动时间)，而是只编译它调用的那部分代码 (这是其名称由来)。代码编译过一次后，就将得到的内部可执行代码存储起来，直到退出该应用程序为止，这样在下次运行这部分代码时就不需要重新编译了。Microsoft 认为这个过程要比一开始就编译整个应用程序代码的效率高得多，因为一个应用程序在运行过程中，其大部分代码实际上并不是在每次运行过程中都执行的。在此编译过程中，代码必须通过类型安全的验证。

(4) 运行代码。CLR 提供能够使执行发生以及可在执行期间使用的各种服务的结构。这些服务涉及垃圾回收、安全性、与非托管代码的互操作性、跨语言调试支持、增强的部署以及版本控制支持等。

整个 CLR 执行的工作流程如图 1-3 所示。



图 1-3 CLR 的工作流程

1.1.4 .NET Framework 类库

.NET 提供了强大的类库，如同 Visual C++ 有 MFC 类库、Delphi 有 VCL 类库、Java 有 Swing 和 AWT 等类库一样，这些类库封装了系统底层的功能，并提供了很好的操作方式，

可以使开发者轻松地构建应用程序。.NET Framework 类库是一个由 Microsoft.NET Framework 中包含的类、接口和值类型组成的库,它封装了对 Windows、网络、文件和多媒体处理的功能,是所有支持.NET 的编程语言都能够使用的类库。类库的源代码保存在称为程序集的.DLL 文件中。

.NET Framework 类库中大约有 7000 多个类(每个类可能会有上百个方法或属性),这些类库被分类打包,功能相似的类库包含在同一个命名空间下,通过调用不同的命名空间来使用类库,以完成不同的功能。

在.NET 平台中进行开发的任何编程语言都使用同样的类库,由于统一了开发模式,开发者再也不必也没有必要掌握全面的开发技术(如 ATL、COM、MFC、ASP、VB 和 C++ 等)。所以,在.NET Framework 下,无论是开发桌面程序还是开发 Web 程序,因为使用的是同样的类库和几乎同样的技术,使得应用程序的开发因此变得非常简单。

学习类库时,不需要掌握其全部应用技术,了解类库的基本架构和实现原理,并能将其正确地运用于实际程序的开发中才是最重要的。

1.2 C#简介

C#(读作“C sharp”)是目前十分流行的一种编程语言。作为一种类型安全、面向对象的语言,C#的应用简单,功能却强大,可使程序设计人员开发出应用广泛的各类程序。

1.2.1 C#语言的由来

随着软件开发技术的不断发展,程序员们不但要解决各种技术平台上的组件兼容问题,还需要解决由不同编程语言开发的组件间的集成问题;同时 Web 应用已成为一种趋势,编程语言还应该能够快速地进行网络应用的开发。为了满足这些需求,2000 年微软公司发布了 C#编程语言,该语言是由 Anders Hejlsberg 和 Scott Wiltamuth 领导的小组专门为.NET 平台设计的运行于.NET Framework 之上的编程语言。这种语言作为微软公司.NET 战略的一部分,是.NET 平台应用的首选语言,其版本随着 Visual Studio.NET 版本的升级而逐步升级。

对 C#的由来有两种解释:从字面意义来解释,C#是 C 语言的开发利器;而微软公司给出的解释是:C#是 C++的升级语言,具有比 C++更优越的开发特性。C#在表达式、运算符和语句等方面沿用了 C/C++的许多特性,而在类型安全、错误处理、版本转换、事件和垃圾回收等方面做了很大的改进和创新。

C#从诞生起就很好地体现了面向对象的设计精神,在 C#语言中不仅使用了类、继承和多态的概念,还使用了属性、方法、接口和事件等概念,很好地描述了对对象。同时,C#还通过值类型和引用类型的概念形成了统一的类型系统,所有的类型只有实例化为对象才能被使用。

C# 2.0 增加了泛型、匿名方法和迭代等新特性,增强了语言的表达能力。C# 2.0 已经被国际标准化组织定为高级语言开发标准。

C# 3.0 又增加了查询关键字、Lambda 表达式、匿名类型、扩展方法、自动实现的属性、

隐式类型的本地变量和数组等功能。这些新的语言构造可以分别用在各种上下文中，并且可以共同完成语言集成查询（LINQ）功能。

C# 4.0 新特性包括 dynamic 类型、命名和可选参数、COM 的互操作、协变和逆变以及与编译器相关的 API 等。

1.2.2 C#语言的特点

相对于其他常用的编程语言，C#的特点可概括如下。

1. 简洁的语法

相对 C++ 语言而言，C# 语言简单易学、容易入门。C# 语言淘汰了 C++ 语言中繁乱的表示符号和伪关键字，使用了有限的、统一的操作符、修饰符和运算符。另外，C# 语言很少使用 C++ 语言中功能强大却难以掌握的指针，使用指针可能会带来内存泄漏以及管理漏洞等不安全因素，会使开发和维护的难度大大增加。而 C# 语言中操作的基本上是实例的对象，只有部分类类型支持指针，因此不建议使用指针。

2. 面向对象的设计

作为一种完全按照面向对象的思想而设计的新语言，C# 语言支持面向对象的所有关键特性，如封装、继承和多态等，是真正纯粹的面向对象的编程语言。而 C++、Java 等语言并不是纯粹的面向对象的编程语言。

C# 语言以类和结构为基础构建所有的类型，每种类型都是一个对象。C# 语言通过命名空间对代码进行层次化的管理，所有的常量、变量、属性、方法和事件等都被封装在类中，从而使程序具有更好的可读性，并且避免了发生命名冲突的可能。

3. 完备的安全性

默认情况下，代码工作在一种受托管的环境中，在托管环境下不允许进行类似直接存取内存的不安全操作。C# 语言可以消除许多软件开发中常见的编程错误，如忘记初始化变量和数组越界等，提供包括类型安全在内的严格的安全检查。此外，不允许网络上下载的代码直接访问本地文件和资源；提供了边界检查和溢出检查等功能；使用垃圾回收机制减轻内存管理的负担；通过使用 CLR 提供的代码，在程序中配置安全等级和用户权限等。这些安全机制保证了 C# 语言具有完备的安全性，使得程序员能够把注意力集中在其他方面。

4. 版本控制

版本问题一直就是比较棘手的问题。例如，由于多个应用程序都安装了名字相同的 DLL 的不同版本，有时应用程序能够正常运行，更多的时候会中断运行。C# 语言内置了版本控制功能，可以很好地支持版本管理，从而使得 C# 语言开发复杂的软件可以不断地进行更新和升级。同时，C# 语言的标准开发环境 Visual Studio 集成的发布和其部署功能，能够确保安装和卸载应用程序时不会影响其他应用程序的使用。

5. 良好的兼容性

C# 语言凭借 .NET Framework 平台对 COM+ 组件、XML Web 服务和 MSMQ（微软消息