

21世纪电脑学校



Visual C#程序设计 实用教程

王 辉 张大治 编著



清华大学出版社

21 世纪电脑学校

Visual C#程序设计实用教程

王辉 张大治 编著

清华大学出版社
北 京

内 容 简 介

C#是Microsoft在推出.NET Framework 1.0时提供的一种全新的计算机语言,其最新版本C# 2.0是.NET 2.0的一部分,它在原有基础上增加了一些更吸引人的功能。本书系统介绍了C#编程的相关内容,从基本语法讲起,由浅入深、循序渐进地介绍了C#语言的编程技巧和面向对象编程的精髓。主要内容包括C#的基本语法、面向对象编程、Windows应用程序设计、数据库编程、网络编程、Web应用程序及Web服务、异常处理机制以及程序的部署和打包等。本书内容丰富、结构清晰、图文并茂,实例具有极高的实用性和代表性,方便读者上机实践。

本书适合C#的初中、级读者,可作为高等院校师生的培训教材,也可供有一定C/C++编程基础的中、高级程序员阅读参考。

本书配套的电子教案和实例源代码可以到<http://www.tupwk.com.cn/21cn>网站下载。

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

Visual C#程序设计实用教程/王辉,张大治编著. —北京:清华大学出版社,2007.3

(21世纪电脑学校)

ISBN 978-7-302-14650-6

I. V… II. ①王…②张… III. C语言—程序设计—教材 IV. TP312

中国版本图书馆CIP数据核字(2007)第018896号

责任编辑:胡辰浩(huchenhao@263.net) 袁建华

装帧设计:孔祥丰

责任校对:成凤进

责任印制:王秀菊

出版发行:清华大学出版社

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

c-service@tup.tsinghua.edu.cn

社总机:010-62770175

投稿咨询:010-62772015

地 址:北京清华大学学研大厦A座

邮 编:100084

邮购热线:010-62786544

客户服务:010-62776969

印刷者:清华大学印刷厂

装订者:三河市溧源装订有限公司

经 销:全国新华书店

开 本:185×260 印 张:23.25 字 数:595千字

版 次:2007年3月第1版 印 次:2007年3月第1次印刷

印 数:1~5000

定 价:32.00元

本书如存在文字不清、漏印、缺页、倒页、脱页等印装质量问题,请与清华大学出版社出版部联系调换。

联系电话:010-62770177 转 3103

产品编号:022443-01

编审委员会

主任：郭 军 北京邮电大学信息工程学院教授
委员：(以下编委顺序没有先后，按照姓氏笔画排列)

王相林 杭州电子科技大学教授
王常吉 中山大学计算机科学系教授
王锁萍 南京邮电大学吴江职业学院教授
闪四清 北京航空航天大学教授
张孝强 南京邮电大学教授
张宗橙 南京邮电大学传媒技术学院教授
杜云贵 长城电脑学校老师
杜耀刚 北京电子科技学院基础部教授
赵树升 郑州大学升达经贸管理学院教授
郭清宇 中原工学院计算机系教授
崔洪斌 河北省科技大学教授
焦金生 《计算机教育》杂志社总编

执行委员：许书明 胡辰浩 李万红 荣春献

执行编辑：胡辰浩

丛书序

出版目的

电脑作为一种工具，已经广泛地应于到现代社会的各个领域，并改变着各行各业的生产方式以及人们的生活方式。在进入新世纪之后，不掌握电脑应用技能就跟不上时代，这已成为不争的事实。因此，如何快速地掌握电脑知识和使用技术，并应用于现实生活和实际工作中，就成为新世纪每个人迫切需要解决的新问题。

为适应需求，各高等院校、高职高专、中职中专以及培训学校都开设了计算机专业相关课程，另外，各类学校也将非计算机专业学生的电脑知识和技能教育纳入教学计划，并陆续出台了相应的教学大纲。基于以上情况，清华大学出版社组织了一批教学精英编写了这套“21世纪电脑学校”教材，以满足学校教学和学习电脑知识人员的需要。本套教材的作者均为各大院校的教学专家和业界精英，他们熟悉教学内容的编排，深谙学生的需求和接受能力，具有丰富的授课和写作经验，并将其充分融入本套教材的编写中。

读者定位

本丛书是为所有从事电脑教学的老师和自学人员编写的，可用做各类院校的教材以及电脑初、中级用户的自学参考书。

涵盖领域

本套教材内容涵盖了计算机各个应用领域，包括计算机硬件知识、操作系统、数据库、编程语言、文字录入和排版、办公软件、计算机网络、图形图像、三维动画、网页制作以及多媒体制作等。众多的图书品种可以满足不同读者、不同电脑课程设置的需要。

本丛书选用应用面最广的流行软件，对各个软件的讲解都从必备的基础知识和基本操作开始，使读者轻松入门，并以大量明晰的操作步骤和典型的应用实例教会读者更实用的软件技术和应用技巧，使读者真正对所学软件融会贯通、熟练掌握。

丛书特色

一、更为合理的学习过程

1、章节结构按照教学大纲的要求来安排，符合教学需要和电脑用户的学习习惯。

2、细化了每一章内容的分布。在每章的开始，有教学目标和理论指导，便于教师和学生掌握本章知识重点，每章最后还附带有上机实验和思考练习，读者不但可以锻炼实际操作能力，还可以复习本章内容，加深对所学知识的了解。

二、简练流畅的语言表述

语言精炼实用，不讲深奥的原理，也不涉及不常用的知识，只介绍学习电脑应用最需要的内容。

三、丰富实用的实例

以详细、直观的步骤讲解相关操作，每本图书都包含众多精彩实例。现在的计算机教学更加注重实际的动手操作，而且学校在教学过程中，也有很多的课时是进行实际的上机操作。因此，本丛书非常注意实例的选材，所选实例都具有较强的代表性。

四、简洁大方的版式设计

精心设计的版式简洁、大方，而且，对于标题、正文、注释和技巧等都设计了醒目的字体，读者阅读起来会感到轻松愉快。

周到体贴的售后服务

本丛书紧密结合自学与课堂教学的特点，针对广大初、中级读者电脑基础知识薄弱的现状，突出基础知识和实践指导方面的内容。每本教材配套的一些实例源文件、素材和教学课件均可在该丛书的信息支持网站 (<http://www.tupwk.com.cn/21cn>) 上下载或通过 Email(wkservice@tup.tsinghua.edu.cn)索取，读者在使用过程中如遇到了疑惑或困难可以在 <http://www.tupwk.com.cn/21cn> 的互动论坛上留言，本丛书的作者或技术编辑会提供相应的技术支持。

前言

C#是可用于创建要运行在.NET CLR 上的应用程序语言之一，它从 C 和 C++语言演化而来，是 Microsoft 专门为.NET 平台而创建的。C#的快速流行，使之成为使用.NET 的 Windows 和 Web 开发人员最佳的语言选择。另一方面，C++程序员不必丢弃以前的知识，就可以开始使用这种新的语言来开发程序。虽然有一些新的概念、技术和函数需要学习，但其语法和 C++是类似的。使用 C#开发应用程序比使用 C++更简单，因为其语法比较简单。为了让广大编程爱好者能够轻松愉快地掌握这门优秀的新语言，本着内容全面、通俗易懂的原则，我们精心策划并编写了本书。

全书共分 12 章，从.NET Framework 和 C#的关系讲起，由浅入深、循序渐进地介绍了 C#程序设计语言的编程技巧。第 1 章是.NET 和 C#概述，介绍了 C#与.NET 的关系，以及 C#语言发展、C# 2.0 的新增功能；第 2 章介绍了 Visual Studio 2005 开发环境，包括常用的工具窗口以及如何创建控制台应用程序和 Windows 应用程序，为后面的程序编译与调试奠定了基础；第 3 章和第 4 章介绍了 C#程序设计语言的基本语法，包括变量的定义与操作、流程控制结构以及一些高级的变量类型(如数组、结构等)、函数的使用等内容；第 5 章和第 6 章介绍面向对象编程技术，C#是一个完全面向对象的程序设计语言，本书用两章内容详细介绍了面向对象编程的思想，包括类和接口的定义与实现、继承与重载、集合与索引以及 C# 2.0 的一个新特性——泛型；第 7 章重点介绍了 Windows 窗体应用程序的开发和常用的.NET 控件，这些控件的使用都比较简单，本书只介绍了其中常用的几种控件，通过学习这些控件的使用方法，使读者能够自己尝试使用其他控件；第 8 章介绍的是数据库编程，围绕 ADO.NET 的结构，学习 ADO.NET 对象的使用方法，ADO.NET 支持与多种数据库类型连接，本章重点介绍了对 SQL Server 2005 数据库的读写操作，其他数据库类型的操作与之类似，读者应能举一反三，掌握 ADO.NET 访问常见数据库的方法；第 9 章介绍了网络编程，主要是 System.Net 和 System.Net.Sockets 命名空间的使用；第 10 章介绍了如何创建 ASP.NET 应用程序以及 Web 服务的开发与调用，其中包括 ASP.NET 网站的创建、Web 服务的创建与调用、综合应用数据库编程技术通过 ASP.NET 页面读写数据库；第 11 章介绍了 C#的异常处理机制以及使用 Visual Studio 2005 调试程序的一些技巧，这是编写健壮应用程序的基本保障，通过本章的学习，读者可以回过头来检查自己前面编写的代码，进一步完善应用程序；最后，在第 12 章介绍了程序的部署和打包技术，这通常是软件开发的最后一个过程，本章对 Windows 应用程序和 Web 应用程序分别介绍了几种部署方法。

本书内容丰富、结构清晰、通俗易懂，所选实例具有广泛的实用性和代表性，读者稍加修饰和完善就是一个软件或网站。本书适合 C#的初、中级读者，可作为各类社会培训机构及高等院校师生的培训教材，也可供有一定 C/C++编程基础的编程爱好者自学或参考。



本书是多人智慧的集成，除封面署名的作者外，参与编写和制作的人员还有王强、张民、王新、史新霞、周惠、杨玉敏、张凤霞、贺宝江、宋军山、李永奎、尚英强、王燕、陈丙离、张挂云、张极超和赵永魁等。由于作者水平有限，加之创作时间仓促，书中难免有疏漏和不足之处，欢迎广大读者批评指正。如果您在学习中发现任何问题，或者有更好的建议欢迎致函 E-mail: huchenhao@263.net。

作 者

2006 年 11 月

目

录

第 1 章 .NET 与 C#概述	1
1.1 .NET 概述	1
1.1.1 什么是.NET	1
1.1.2 Microsoft .NET 的意义	3
1.2 .NET Framework	4
1.2.1 .NET Framework 的内容	5
1.2.2 .NET Framework 的工作原理	6
1.3 C#简介	10
1.3.1 C#与 C++	10
1.3.2 C#与 Java	11
1.3.3 使用 C#编写程序	12
1.3.4 命名空间	12
1.3.5 C# 2.0 的新增功能	14
1.4 本章小结	15
1.5 思考和练习	16
第 2 章 Visual Studio 2005 开发环境	17
2.1 Visual Studio 2005 概述	17
2.1.1 为什么选择 Visual Studio 2005	17
2.1.2 Visual Studio 2005 Express 产品	18
2.1.3 VS 解决方案	18
2.2 环境介绍	19
2.2.1 环境设置	19
2.2.2 窗口布局	20
2.2.3 “选项”对话框	22
2.3 使用 VS 开发 C#应用程序	22
2.3.1 控制台应用程序	23
2.3.2 解决方案资源管理器	25
2.3.3 “属性”窗口	26
2.3.4 “错误列表”窗口	26
2.3.5 Windows 应用程序	27

2.3.6 使用智能提示	30
2.4 本章小结	30
2.5 思考和练习	30
第 3 章 C#程序设计基础	31
3.1 简单数据类型	31
3.1.1 设置单元格格式的方法	31
3.1.2 浮点数类型	32
3.1.3 其他数据类型	32
3.1.4 简单数据类型实例	33
3.2 变量	35
3.2.1 变量的声明	35
3.2.2 变量的命名规则	35
3.2.3 变量的赋值	36
3.3 运算符与表达式	38
3.3.1 算术运算符	38
3.3.2 赋值运算符	41
3.3.3 关系运算符	42
3.3.4 逻辑运算符	43
3.3.5 位运算符	44
3.3.6 布尔赋值运算符	47
3.3.7 运算符的优先级	47
3.4 类型转换	48
3.4.1 隐式转换	49
3.4.2 显式转换	50
3.4.3 选择 Convert 命令进行 显式转换	53
3.5 流程控制	55
3.5.1 条件分支	55
3.5.2 循环	62
3.5.3 跳转语句	64
3.6 本章小结	66
3.7 思考和练习	67



第4章 C#高级程序设计	69
4.1 复杂的变量类型	69
4.1.1 枚举	69
4.1.2 结构	72
4.1.3 数组	81
4.2 函数	87
4.2.1 定义和使用函数	88
4.2.2 返回值	89
4.2.3 参数	90
4.2.4 Main()函数	95
4.3 变量的作用域	98
4.3.1 其他结构中变量的作用域	100
4.3.2 参数与全局数据	101
4.4 函数重载	102
4.5 使用 unsafe 代码	104
4.6 类型操作符和 sizeof 操作符	107
4.6.1 类型操作符	107
4.6.2 sizeof 操作符	109
4.7 本章小结	111
4.8 思考和练习	111
第5章 面向对象编程基础	113
5.1 面向对象概述	113
5.2 OOP 相关的概念	114
5.2.1 类和对象	114
5.2.2 静态成员	116
5.2.3 对象的生命周期	117
5.2.4 继承	118
5.2.5 接口	119
5.2.6 多态性	120
5.2.7 对象之间的关系	122
5.2.8 运算符重载	123
5.2.9 引用类型和值类型	124
5.3 VS 2005 中的 OOP 工具	124
5.3.1 “类视图”窗口	124
5.3.2 对象浏览器设置	126
5.3.3 添加新类	126
5.3.4 生成 UML 类图	127
5.4 定义 C#类	128
5.4.1 类的声明	128
5.4.2 类基础规范	129
5.4.3 类主体	131
5.5 类成员	132
5.5.1 构造函数和析构函数	132
5.5.2 定义字段	138
5.5.3 定义方法	139
5.5.4 定义属性	146
5.5.5 使用类图添加成员	148
5.6 应用实例	150
5.7 本章小结	154
5.8 思考和练习	154
第6章 OOP 的高级技术	155
6.1 接口的定义与实现	155
6.1.1 接口的定义	155
6.1.2 接口的实现	156
6.2 委托与事件	158
6.2.1 委托	158
6.2.2 事件	160
6.3 集合与索引	165
6.3.1 提供集合功能的接口	165
6.3.2 使用集合	165
6.3.3 定义集合	166
6.3.4 索引	169
6.4 使用泛型	171
6.4.1 泛型概述	171
6.4.2 System.Collections.Generic 命名空间	172
6.4.3 可空类型	176
6.5 定义泛型	178
6.5.1 定义泛型类	178
6.5.2 定义泛型方法	184
6.5.3 定义泛型接口与泛型委托	185
6.6 本章小结	186



6.7 思考和练习	186
第7章 Windows 编程基础	189
7.1 Windows 应用程序概述	189
7.1.1 窗体与控件	189
7.1.2 控件的属性	190
7.1.3 控件的事件	191
7.1.4 添加事件处理函数	191
7.2 Button 控件	192
7.2.1 Button 控件的属性	192
7.2.2 Button 控件的事件	193
7.3 Label 和 LinkLabel 控件	193
7.4 文本框控件	194
7.4.1 TextBox 控件	195
7.4.2 应用实例：用户登录界面	196
7.4.3 RichTextBox 控件	197
7.5 RadioButton、CheckBox 和 GroupBox 控件	199
7.5.1 RadioButton 控件	199
7.5.2 CheckBox 控件	200
7.5.3 GroupBox 控件	200
7.6 TabControl 控件	201
7.6.1 TabControl 控件的属性	201
7.6.2 应用实例：信息登记	202
7.7 菜单	205
7.7.1 创建菜单	205
7.7.2 菜单项的常用属性与事件	206
7.7.3 弹出式菜单	207
7.8 通用对话框	208
7.8.1 文件对话框	208
7.8.2 打印对话框	210
7.8.3 字体和颜色对话框	212
7.8.4 应用实例：一个简单的文本 编辑器	213
7.9 本章小结	217
7.10 思考和练习	217

第8章 数据库编程	219
8.1 ADO.NET 概述	219
8.1.1 ADO.NET 架构	220
8.1.2 ADO.NET 的设计目标	221
8.2 ADO.NET 对象	222
8.2.1 提供者对象	222
8.2.2 数据集对象	223
8.3 使用 ADO.NET 访问数据库	224
8.3.1 使用 DataReader	225
8.3.2 使用数据集对象	226
8.3.3 使用 DataGridView 控件	232
8.3.4 修改数据库	234
8.3.5 在 DataSet 中访问多个表	238
8.4 XML 和 ADO.NET	240
8.5 调用 SQL 存储过程	241
8.6 本章小结	243
8.7 思考和练习	243
第9章 网络编程	245
9.1 Internet 概述	245
9.1.1 域名解析	246
9.1.2 统一资源标识符 URI	248
9.1.3 TCP 和 UDP	249
9.2 C# 网络编程初探	250
9.2.1 使用 WebClient 类	250
9.2.2 使用 WebRequest 和 WebResponse	251
9.3 Socket 编程	256
9.3.1 Socket 编程概述	256
9.3.2 聊天室服务器程序	258
9.3.3 聊天室客户端程序	263
9.4 本章小结	270
9.5 思考和练习	270
第10章 Web 程序设计	271
10.1 Web 编程基础	271
10.1.1 ASP.NET 介绍	271



10.1.2	ASP.NET 工作原理	272
10.1.3	Web 窗体页指令	273
10.1.4	ASP.NET 内部对象	274
10.2	创建 Web 站点	275
10.2.1	服务器控件	276
10.2.2	创建自己的 Web 站点	276
10.3	登录控件	293
10.4	什么是 Web 服务	294
10.4.1	Web 服务的定义	294
10.4.2	ASP.NET Web 服务体系	295
10.5	XML 和 SOAP	296
10.5.1	XML	296
10.5.2	SOAP	300
10.6	创建 Web 服务	303
10.6.1	WebService 类	303
10.6.2	创建 Web 服务项目	305
10.7	调用 Web 服务	308
10.7.1	调用 Web 服务的机制	308
10.7.2	使用 Windows 客户程序	310
10.7.3	创建 ASP.NET 客户程序	313
10.8	本章小结	315
10.9	思考和练习	315
第 11 章	异常处理和程序调试	317
11.1	异常处理	317
11.1.1	常见错误类型	317
11.1.2	异常	318
11.1.3	通用异常类	318
11.1.4	C#中的异常处理	319
11.2	调试 C#程序	326
11.2.1	非中断模式下的调试	326
11.2.2	中断模式下的调试	331
11.3	本章小结	337
11.4	思考和练习	337
第 12 章	程序的打包与部署	339
12.1	部署概述	339
12.2	部署 Windows 应用程序	340
12.2.1	ClickOnce 部署	340
12.2.2	安装和部署项目类型	346
12.2.3	为 Windows 应用程序创建 安装包	346
12.3	部署 Web 应用程序	353
12.3.1	复制 Web 站点	353
12.3.2	预编译 Web 站点	355
12.3.3	为 Web 应用程序创建 安装包	356
12.4	本章小结	357
12.5	思考和练习	357

第1章

.NET与C#概述

本章导读

本章首先为读者介绍.NET战略的提出及其意义,然后讲述了C#(发音为“C Sharp”)语言的特点以及.NET Framework的工作原理。.NET Framework是一种新技术,它包含的许多概念初看起来都不太容易理解,主要是因为该架构在应用程序开发环境中引入了一种执行操作的新方式。通过对本章的学习,读者将对.NET Framework和C#有一个概括性的认识,这对于理解如何利用C#进行编程是非常重要的。

重点和难点

- .NET战略和.NET的意义
- .NET Framework的工作原理
- C#和.NET Framework的含义
- C#与其他语言比较
- 命名空间
- C#2.0的新增功能

1.1 .NET 概述

随着网络经济的到来,微软公司希望帮助用户使其能够在任何时候、任何地方、并且利用任何工具都可以获得网络上的信息,并享受网络通信所带来的快乐。.NET战略就是为实现这样的目标而提出的。

1.1.1 什么是.NET

Microsoft .NET在提出概念的最初叫作NGWS(Next Generation Windows Services),它使Microsoft从以客户机为中心的传统单机模式转移到了以网络为中心的广阔模式,并以此为中心拟定了一个新的战略。不可否认的是,Microsoft花了较长一段时间才走到这一个新的立场和观念上来。早在十年前,Java的缔造者——Sun公司就在倡导“网络就是计算机”,并且以强健的Java语言作为核心,发展了一整套网络计算解决方案,也为其赢得了高额的利润。



Sun 在网络开发和应用,尤其是企业高端应用领域走在了 Microsoft 的前头,并且现在仍然在飞速地发展,而 Microsoft 是在 1998 年才提出了它的第一个支持终端服务的操作系统。网络是未来应用的重点,Microsoft 最终也意识到 Web 的重要性,并且接受了它。

在 .NET 战略中,Microsoft 宣布了 Windows 2000 DataCenter 上所做的一些改良工作:宣布了一个最终模型,即通过应用程序服务商(ASP)发放 Windows 应用程序许可;以及宣布在 Windows 2000 服务器操作系统的核心部分(包括终端服务)所作的决定。以前,使用个人计算机的公司较多接触的对象是 Microsoft,由于 Windows 操作系统占据了个人计算机操作系统的绝大部分。因此随着 .NET 概念的提出,使 Microsoft 有实力将它的市场延伸到网络高端应用。

Microsoft .NET 是一项革命性的技术框架。.NET 的核心技术包括分布式计算、XML、组件技术以及即时编译技术等。分布式计算是网络的本质;XML 奠定了新一代电子数据交换的标准,正是数据交换使网络计算成为可能;组件技术是软件技术多年来的发展成果,它使程序设计员从大量的 API 中解放出来,以采用面向对象和面向组件的技术来解决软件问题;即时编译技术使应用程序在运行时,能够根据主机的硬件和软件环境进行代码优化,并简化代码发放的过程。

.NET 平台包含以下内容:

- .NET Framework, 它包括 Common Language Runtime(CLR), 即通用语言运行环境,这是用于运行和加载应用程序的软件组件;以及新的类库,这些类库分级组织了开发者可以在他们的应用程序中用来显示图形用户界面、访问数据库和文件以及在 Web 上进行通信的代码集合。
- .NET 开发者工具, 包括: Visual Studio .NET 集成开发环境, 用来开发和测试应用程序; .NET 编程语言(例如 VB .NET 和 C#), 用来创建运行在 CLR 下并且使用类库的应用程序。
- ASP .NET, 一个取代以前的 Active Server Pages(ASP)的特殊类库, 用来创建动态的 Web 内容和 Web 服务器应用程序, 这些都将采用诸如 HTML、XML 和 Simple Object Access Protocol(SOAP)(简单对象访问协议)等 Internet 协议和数据格式。采用这项技术的网页请求速度更快, 编写更简单。

.NET 采用特殊的方式编译和执行程序, 先通过编译器编译, 将程序编译成微软中介语言文件, 也就是 MSIL(Microsoft Intermediate Language)文件, MSIL 将来被启动时会启动 MSIL 编译器, 将 MSIL 编译成机器码, 然后加载到 CPU 执行, 如图 1-1 所示。

由于 MSIL 文件的成分并非机器码, 因此每次执行都必须启动 MSIL 编译器, 这样会影响执行的效率。不过 MSIL 码十分接近机器码, 从 MSIL 编译成机器码然后加以执行的速度很快(微软把 MSIL 编译器叫做 JIT(Just In Time)编译器), 因此对执行效率的影响可以忽略。在 .NET 平台下, 不管哪一种程序语言, 最终都要编译成 MSIL, 因此 .NET 实现了在统一平台下开发应用程序。

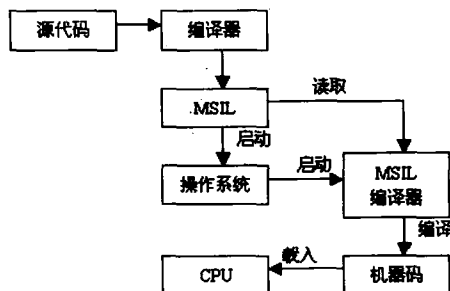


图 1-1 .NET 中的 MSIL 原理图

1.1.2 Microsoft .NET 的意义

.NET 的意义可以从三方面考虑：对开发人员的意义；对其他从业人员的意义；对企业的意义。

1. 对开发人员的意义

通过使用 .NET，开发人员能够创建出摆脱设备硬件束缚的应用程序，以便轻松实现互联网连接。.NET Framework 将互联网本身作为构建新一代操作系统的基础，而且对互联网和操作系统的设计进行了一定程度的延伸。

对于开发者而言，.NET 使他们能应用自己最熟悉的语言创建应用程序，并使得从异种数据源查询信息成为可能。开发者将不必在 COM 对象到 DLL 的不同编程模式中换来换去，不必应付不同的调试工具，也不必为 .NET 应用程序编写安装包。

.NET 对开发人员来说十分重要，因为 .NET 不但会改变开发人员的开发应用程序的方式，而且使得开发人员能创建出全新的各种应用程序。新型开发范例的核心是 Web 服务这个概念的引入。Web 服务是一种通过简单对象访问协议(SOAP)，在互联网上展露其功能特性的、极为公开的服务。SOAP 是一种基于可扩展标记语言(XML)制定的协议。

在过去，开发人员通过集成本地系统服务来构建应用程序。在这种模型下，开发人员可以访问丰富的开发资源并能严格控制应用程序的行为。现在，开发人员已在很大程度上挣脱了这种模型的束缚，并致力于构建具有复杂结构的 n 层化系统，这种系统能将网络上众多的应用程序一并进行集成，大大提升了应用程序的价值。这样，开发人员便可把精力集中在充分挖掘软件独特的商业价值上，而不是构建基本结构上。这样，软件投放市场的时间将大大缩短、开发人员的编程效率也明显提高。

n 层计算技术能够大幅度提高生产力，并且具有紧密耦合的特点，而 Web 概念具有面向消息、松散耦合的特点，.NET 将二者有机地糅合，巧妙地实现了上述构想。这种计算风格称为 Web 服务，它的出现标志着人类已经迈入应用程序开发技术的新纪元。Web 服务是一种应用程序，它可以通过编程并使用标准的 Internet 协议，就像超文本传输协议(HTTP)和 XML 一样，将功能展示在互联网和企业内部网上。还可将 Web 服务视作 Web 上的组件编程。



2. 对其他从业人员的意义

.NET 对计算机界其他从业人员来说也是非常重要的,因为计算机的功能将会得到大幅度提升,同时计算机操作也会变得非常简单。用户将完全摆脱人为的硬件束缚,可以通过任何桌面系统、任何便携式电脑、任何移动电话或 PDA 进行对互联网的访问,并可对其进行跨应用程序的集成。

Microsoft 推出的 Windows 2000 和 .NET Enterprise Servers 等操作系统为创建具有高度可管理性的、能迅速投入市场的应用程序奠定了坚实的基础。它们采用的是可扩展标记语言 (XML),随着 Web 体系结构的革新,在此平台上创建的程序仍然具有很高价值。

.NET 平台采用分门别类的方式来构建应用程序,并且达到了其前所未有的规模。该平台上的 Web 服务模型指的是:企业应用程序的中心业务要素通常由本地管理,而支持它们的服务(如用户认证、文件存储、用户首选项管理、日历和邮件等)却无需由本地管理,可以无缝订购。

3. 对企业的意义

Microsoft .NET 平台将侧重点从连接到互联网的单一网站或设备上转移到计算机、设备和服务群组上,并使其通力合作,以提供更广泛、更丰富的解决方案。用户将能够控制信息的传送方式、时间和内容。计算机、设备和服务将能够相辅相成,从而提供丰富的服务,而不是由用户提供唯一的集成。企业可以提供一种方式,允许用户将其产品和服务无缝地嵌入自己的电子架构中。这种思路将扩展对 20 世纪 80 年代 PC 功能的认识。

.NET 将保证完全消除当今计算技术中的所有缺陷;.NET 能确保用户在任何地点、使用任何设备都可访问其个人数据和应用程序的宏伟蓝图。除此之外,.NET 技术还可以实现多个应用程序在逻辑上的松散耦合链接和紧密耦合链接。

企业用户可以通过手写、语音和图像技术与个人数据进行交互。这些数据将安全地存放在互联网上,用户可以通过办公室(或家庭)PC、移动电话、PDA、甚至是新发明的移动电话+PDA+PC 联合设备访问这些数据。应用程序可进行灵活的功能调整,以适应用户所用设备的功能状况。应用程序可使用用户预定义的选项集和指令集,完全代替用户自动执行相应的操作。

1.2 .NET Framework

.NET Framework 是一种新的计算平台,它简化了在高度分布式 Internet 环境中的应用程序开发,是 .NET 平台的核心部分。.NET Framework 旨在实现下列目标:

- 提供一个一致的面向对象的编程环境,无论对象代码是在本地存储和执行,还是在本地执行但在 Internet 上分布,或是在远程执行。
- 提供一个将软件部署和版本控制冲突最小化的代码执行环境。
- 提供一个可消除脚本环境或解释环境的性能问题的代码执行环境。



- 使开发人员的经验在面对类型大不相同的应用程序(如基于 Windows 的应用程序和基于 Web 的应用程序)时保持一致。
- 提供一个保证代码(包括由未知的或不完全受信任的第 3 方创建的代码)安全执行的代码执行环境。
- 按照工业标准生成所有通信, 以确保基于 .NET Framework 的代码可与任何其他代码集成。

12.2.1 .NET Framework 的内容

.NET Framework 具有两个主要组件: 公共语言运行时(CLR)和 .NET Framework 类库。公共语言运行时是一个在执行时管理代码的代理, 它提供核心服务(如内存管理、线程管理和远程处理), 而且强制实施严格的类型安全, 还可确保安全性和可靠性的其他形式代码的准确性。

.NET Framework 可使开发人员更容易建立网络应用程序和网络服务。建立在操作系统最底层的服务, 是管理运行时代码需求的 CLR, 这些代码可以用任何现代编程语言编写。CLR 提供了许多服务, 这些服务有助于简化代码开发和应用程序的开发, 同时也将提高应用程序的可靠性。 .NET Framework 包括一套可被开发者用于任何编程语言的类库。在此之上是许多应用程序模板, 这些模板专门为开发网络站点和网络服务提供高级组件和服务。图 1-2 显示了公共语言运行时和类库与应用程序之间以及与整个系统之间的关系。另外还显示了托管代码如何在更大的结构内运行。

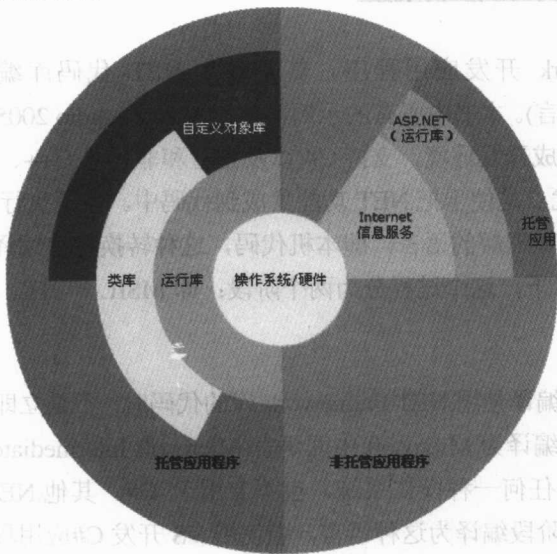


图 1-2 .NET Framework 内部关系

在 .NET Framework 中引入了组合体的概念。一个组合体是一组资源或类型, 包括这些资源或类型相关的元数据, 并且是被作为一个单元配制的。元数据被称为组合体的名单, 它包含类型和资源表等能在组合体外看得见的信息, 这个名单也包括有关从属关系的信息, 例如