

.net

.NET 平台研究与开发丛书

Visual C#.NET 开发实践

■ 前沿软件研究室 郑小平 编著

基于
Visual Studio.NET

综合Visual Basic的高生产率
和C++的行动力

使生产性能得到了提高

比C和C++更简单的语法

比Java更适合融入Microsoft的技术

带给人们
更强的生产力和更好的可靠性

人民邮电出版社
www.pptph.com.cn



附光盘
CD-ROM



.NET 平台研究与开发丛书

Visual C#.NET 开发实践

■ 前沿软件研究室 郑小平 编著

人民邮电出版社

图书在版编目 (CIP) 数据

Visual C# .NET 开发实践 / 温谦, 郑小平, 刘涛编著. —北京: 人民邮电出版社, 2001.8

(NET 平台研究与开发丛书)

ISBN 7-115-09618-X

I. V... II. ①温... ②郑... ③刘... III. C 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字 (2001) 第 057716 号

.NET 平台研究与开发丛书
Visual C# .NET 开发实践

145257/06

◆ 编 著 前沿软件研究室 郑小平
责任编辑 杨 璐

◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号
邮编 100061 电子函件 315@pptph.com.cn
网址 <http://www.pptph.com.cn>
读者热线: 010-67129212 010-67129211(传真)
北京汉魂图文设计有限公司制作
北京密云春雷印刷厂印刷
新华书店总店北京发行所经销

◆ 开本: 787 × 1092 1/16
印张: 26.25
字数: 619 千字 2001 年 8 月第 1 版
印数: 1-6 000 册 2001 年 8 月北京第 1 次印刷

ISBN 7-115-09618-X/TP·2442

定价: 48.00 元(附光盘)

本书如有印装质量问题, 请与本社联系 电话: (010) 67129223

内 容 提 要

软件业界的巨人微软公司在新世纪推出了恢宏的.NET 计划，C#语言是.NET 框架的核心语言。本书的目的就是让国内的开发者最先了解如何使用这门崭新的语言在.NET 平台上实现各种传统应用的开发，让国内的朋友最先领悟.NET 计划思想和前途。

本书详细介绍了使用 C#语言在 Visual Studio .NET 开发环境中开发 Windows 程序的各个方面，包括界面、图形图像、打印、文件、多任务和网络等高级主体。本书针对的是中高级开发者，也就是熟练使用 Windows 系统，并已经系统学习了 C#语言语法的读者。如果对这两个方面已经比较熟悉，那本书将使您成为.NET 开发方面的领先者。

丛书序言

2000年7月在美国奥兰多举行的专业开发者大会上,微软公司展示了它最新的.NET体系,众多的发言人向大会介绍了.NET的特征和构成。从此,软件业界就开始掀起了一股“.NET”热潮。通过.NET,微软公司公司为人们提供了一种全新的开发平台,这个平台将推动以新体系为基础的协同Web应用开发。微软公司推出 Visual Studio.NET 的目的就是要推进网络应用软件的开发,使用 Visual Studio.NET 开发系统可以编写最简单的基于 Windows 的单层应用软件,也可以编写 Windows 下的客户/服务器多层应用软件。无论是为 Windows 还是为 Web, Visual Studio 都在各个层面上作了大量的工作。

Visual Studio.NET 7.0 与以往的版本相比发生了巨大的变化,这种变化主要来自以下两个方面。

一是语言本身的改进和.NET平台的建立。无论是在网络应用还是在本地应用上,都对各种语言做了许多改进,特别是用崭新的C#语言代替了J++。C#具有简洁、灵活、安全、面向对象和兼容性强等许多优点,是.NET中的旗帜语言。此外, Visual Basic 第一次成为真正面向对象的语言; C++大大增强了对Web的集成和对ATL的支持。

二是开发环境的重大改变。首先,在一个开发环境中同时集成了 Visual Basic、Visual C++、C#和 FoxPro。其次,开发环境的适应性大大增强了,适用于快速Web应用软件的开发,可以轻松地运用XML和Web服务进行跨平台计算,也可以快速开发中间层商务组件。再次,调试功能增强了,可以在 Visual Studio.NET 中端对端地调试Web应用软件,可以同时调试用不同语言开发的程序,也可以调试工程、进程和存储等过程。这些新特性使用户的开发效率有了大幅度的提高。

为了让广大程序开发人员和编程爱好者们对 Visual Studio.NET 有比较系统和全面的了解,我们特编写此套丛书。丛书共分为以下5册:

◇ 《C#语言程序设计》 基于 Microsoft.NET Framework SDK,完整地介绍了C#语言的体系、语法和各种特性,力图通过本书使读者对C#语言本身有较深刻的理解。

◇ 《Visual C#.NET 开发实践》 基于 Microsoft Visual Studio.NET 7.0,具有很强的可操作性。通过本书的介绍,读者可以快速地掌握运用 Visual C#.NET 进行实际开发的方法。

◇ **《ASP.NET 高级专题与实例剖析》** 在有针对性地介绍 ASP.NET 的基础上, 分几个专题进行了讨论, 例如开发 User Control 和 Server Control 等, 并结合一个实际工程对开发的全过程进行了介绍。

◇ **《Visual C++.NET 核心编程与开发实例》** 讨论了 C++ 最核心的技术, 特别是对组件对象模型进行了深入的介绍, 并对 MFC 和 ATL 进行了非常具体的讲解。

◇ **《Visual Basic.NET 与数据库开发》** 介绍了 Visual Basic 的新特性, 并通过精选的实例详细讲解了用 Visual Basic.NET 开发几种典型数据库的完整方法。

编 者

编者的话

2000年6月,微软公司推出了“Microsoft .NET 下一代互联网软件和服务战略”,这是微软公司全面进军互联网的重大举措。2000年11月,微软公司在 Comdex 计算机展上发布了 Visual Studio.NET 软件,并展示了其.NET 发展战略的框架体系和开发工具的相关特性,加快了微软公司以.NET 技术进军市场的步伐。

Microsoft .NET 概述

Microsoft .NET 包含以下组成部分:

- Microsoft.NET 平台——包含.NET 基础结构和工具,以运行新一代服务程序;.NET 支持更加丰富的客户端;.NET 构造模块,支持新一代高度分布式超服务;还有.NET 设备软件,以支持新型智能化因特网设备。
- Microsoft .NET 产品和服务——包含带有核心构造模块服务的 Windows .NET、MSN .NET、个人订阅服务、Office .NET、Visual Studio®.NET 和 bCentral for .NET。
- 第三方.NET 服务——众多微软公司的商务伙伴和第三方开发商将有机会制造出基于.NET 平台的企业软件和垂直型服务程序。

Microsoft .NET 将把计算和通信带入一个丰富、合作和互动的环境中,远远胜过单向网络。由新型高级软件支持的 Microsoft .NET 将利用一系列的应用程序、服务程序和相关设备来创造一种个性化的数字体验,它能够不断地按照用户的需要自动调整。它意味着用户将能够通过一整套新型软件构成的整体服务程序来管理自己在因特网时代的生活和工作。

对于用户来说,Microsoft .NET 意味着简单化的整体服务;统一的信息浏览、编辑和授权;查看资料、工作和在线与离线媒体;一种整体的系统方案;随时随地的个性化。例如,无论是通过个人电脑、便携设备还是灵通卡,对于个人信息的任何修改都将即时和自动地通知到所有需要这些信息的地方。

对于知识工作者和企业,Microsoft .NET 意味着统一的信息浏览、编辑和授权;丰富的同步传播;密切的移动通信联系;得力的信息管理和电子商务工具,在基于内部网和因特网的服务程序之间灵活地切换,为动态商务伙伴关系的建立提供支持。

对于独立软件开发商,Microsoft .NET 意味着他们将得到更多的机会,为因特网时代创造更多的新型高级服务。这些服务可以自动从本地或远程取得和利用所需信息,无需为不同的工作环境重新编写程序。因特网上的一切都变成这种新一代服务程序的潜在构造模块,而每一种应用程序都可以在网上使用。

正如微软公司首席执行官鲍尔默所说:“Microsoft .NET 代表了一个集合、一个环境、一

个可以作为平台支持下一代 Internet 的可编程结构。” Microsoft .NET 的策略是把因特网本身作为构建新一代操作系统的基础，将因特网和操作系统的设计思想合理延伸。这样，开发人员必将创建出摆脱设备硬件束缚的应用程序，以便轻松实现因特网连接。作为中国 IT 业界的一分子，无论是开发者还是管理者，都有必要仔细地审视、学习和利用 Microsoft .NET 框架体系的每一个细节。

公用语言运行环境 (CLR)

Microsoft .NET 的核心是公用语言运行环境 (Common Language Runtime, CLR)，CLR 和 Java 虚拟机一样，也是一个运行环境，它负责资源管理（内存分配和垃圾收集），并保证应用和底层操作系统之间必要的分离。

为了提高平台的可靠性，并达到面向事务的电子商务应用所要求的稳定性级别，CLR 还要负责其他一些任务，比如监视程序的运行。按照 .NET 的说法，在 CLR 监视之下运行的程序属于“受管辖的” (Managed) 代码，而不在 CLR 监视之下且直接在裸机上运行的应用或者组件属于“非管辖的” (Unmanaged) 的代码。

CLR 将监视形形色色的常见编程错误。许多年来这些错误一直是软件故障的主要根源，其中包括访问数组元素越界，访问未分配的内存空间，由于数据体积过大而导致的内存溢出，等等。

公用语言运行环境由以下几个部分组成：

- 面向对象的编程模式（继承、多态和异常处理等）；
- 统一标准的类型系统；
- 所有类型运行时刻的元数据支持；
- 全面的 .NET Framework 类，覆盖了绝大多数的 Win32 API 和其他一些技术，比如数据库访问、XML 解析等；
- 高级的调试、发布支持；
- 受管辖的代码执行环境和高级内存管理机制；
- 系统范围内的垃圾回收器负责所有被管辖内存的生命期控制；
- 公布的安全模型。

C#语言

C#（读作 C-Sharp）是微软公司为它的 Microsoft .NET 计划推出的核心编程语言。该语言几乎综合了目前所有编程语言系统的优点，并结合因特网发展的需要，增加了丰富的新特性和增强功能。

C#语言面向实体的设计，可以用来构建从高水平的商务目标到体系标准应用程序的范围宽广的组件。使用 C#编程语言，程序员能够迅速地建造提供充分开拓计算和通信的工具和服务的新的 Microsoft.NET 平台。C#语言的组成部分使用简单的 C#语言结构体，能被转换成 Web 服务，允许人们通过 Internet 调用在任何操作系统上运行的任何语言。

C#作为一种编程语言，具有以下几个至关重要的特点：

- 兼容性；
- 灵活性；

- 简单性；
- 现代性；
- 面向实体；
- 类型安全；
- 版本控制。

读者如果需要全面地了解 C#语言的语法细节，可以参阅本套丛书的《C#语言程序设计》一书。

本书概述

本书的重点是介绍以 Visual Studio .NET 为开发平台使用 C#语言开发 Windows 程序。涉及的内容非常广泛，总体包括以下几个方面：

- 界面开发，包括开发各种常见的 Windows 用户界面元件，如按钮、菜单、工具条、状态栏、对话框、多文档应用程序等。
- 应用程序功能开发，包括文档打印、图形图像绘制和拖放剪贴板操作等。
- 系统开发，包括线程进程开发、文件系统和 XML 解析等。
- 网络开发，介绍了 .NET Framework 中提供的网络接口和应用。

特别说明

本书介绍的程序都在本书的随书光盘中给出。所有程序都在 Microsoft Visual Studio.NET 7.0 Beta2 版中运行通过。

读者对象

本书针对的是中高级读者，也就是对 Windows 系统的应用和开发、C#语言的基本语法知识比较熟悉的读者，所以本书不介绍任何 Windows 开发中的概念或 C#语言的语法细节，而直接介绍各种应用开发的方法和技巧。

如果读者对上述两个方面比较熟悉，学习本书就不会有难度。在学习的过程中，相信读者一定能领悟到 C#语言和 Visual Studio .NET 开发环境给程序员带来的强大功能和方便快捷的特性。

无论是 Visual Studio .NET 还是 C#语言，目前都处在不断的发展和改进中，我们非常希望能够和读者进行交流，尤其希望能够得到读者的批评和指正。我们的 E-mail 地址为 dragonpool@263.net。

编 者
2001.5

目 录

第 1 章 一个 Windows 程序	1
1.1 初识开发环境	1
1.2 “Start” 页面	2
1.2.1 “Get Started” 页面	2
1.2.2 “What's New” 页面	3
1.2.3 “Online Community” 页面	4
1.2.4 “Headlines” 页面	4
1.2.5 “Search Online” 页面	5
1.2.6 “My Profile” 页面	5
1.3 建立一个工程项目	6
1.3.1 创建工程框架	6
1.3.2 编译执行程序	8
1.4 Solution Explorer 窗口	9
1.4.1 Solution 概念	9
1.4.2 Project 概念	9
1.4.3 Solution Explorer 窗口	10
1.5 Code Editor 窗口	11
1.5.1 打开 Code Editor 窗口	11
1.5.2 Code Editor 新特性	12
1.6 分析代码	13
1.6.1 AssemblyInfo.cs 文件	13
1.6.2 Form1.cs 文件	15
1.7 编辑窗体	17
1.7.1 修改窗体属性	17
1.7.2 添加控件	19
1.7.3 新添加的代码	20
1.8 本章总结	21
第 2 章 基础 Win Forms 控件	23
2.1 Button (按钮) 控件	23

2.1.1 建立新工程.....	24
2.1.2 创建按钮并设置属性.....	25
2.1.3 实现按钮事件方法.....	25
2.1.4 演示按钮.....	28
2.2 TextBox (文本框) 控件.....	29
2.2.1 创建文本框并设置属性.....	29
2.2.2 访问 Text 属性.....	30
2.2.3 演示文本框.....	30
2.3 CheckBox (复选框) 控件.....	30
2.3.1 创建复选框并设置属性.....	31
2.3.2 实现复选框事件方法.....	31
2.3.3 演示复选框.....	32
2.4 RadioButton (单选按钮) 控件.....	32
2.4.1 创建单选按钮并设置属性.....	33
2.4.2 实现单选按钮事件.....	33
2.4.3 演示单选按钮.....	34
2.5 ListBox (列表框) 控件.....	35
2.5.1 创建列表框并设置属性.....	35
2.5.2 实现添加删除操作.....	36
2.5.3 演示列表框.....	38
2.6 ComboBox (组合框) 控件.....	38
2.6.1 创建组合框并设置属性.....	38
2.6.2 实现选择条目消息.....	40
2.6.3 演示组合框.....	40
2.7 ToolTip (工具提示) 控件.....	41
2.7.1 创建工具提示并设置属性.....	41
2.7.2 关联工具提示的控件.....	42
2.7.3 演示工具提示.....	42
2.8 本章总结.....	43
第3章 高级 Win Forms 控件	45
3.1 NumericUpDown 控件.....	45
3.1.1 创建包含 NumericUpDown 的窗体.....	45
3.1.2 控制 NumericUpDown 控件.....	46
3.1.3 实现 NumericUpDown 控件的事件方法.....	47
3.1.4 演示 NumericUpDown 控件.....	48
3.2 ProgressBar (进度条) 控件.....	48
3.2.1 创建包含 ProgressBar 的窗体.....	48
3.2.2 显示计算过程.....	49

3.2.3 演示 ProgressBar 控件	50
3.3 ListView (列表视图) 控件	51
3.3.1 创建包含 ListView 的窗体	52
3.3.2 实现地址簿功能	54
3.3.3 演示 ListView 控件	57
3.3.4 查看 ListView 的创建代码	58
3.4 TreeView (树形视图) 控件	59
3.4.1 创建包含 TreeView 的窗体	59
3.4.2 实现 TreeView 的事件方法	61
3.4.3 实现按钮的点击事件方法	62
3.4.4 演示 TreeView 控件	64
3.5 Splitter 控件	65
3.5.1 创建分隔窗体	66
3.5.2 演示 Splitter 控件	69
3.6 TabControl 控件	70
3.6.1 创建 TabControl 控件	70
3.6.2 创建标签页	71
3.6.3 分析代码	73
3.6.4 演示 TabControl 控件	74
3.7 本章总结	74
第 4 章 菜单设计	75
4.1 创建主菜单	75
4.1.1 添加主菜单条	75
4.1.2 添加“文件”菜单组	76
4.1.3 添加“新建”菜单项	76
4.1.4 添加“打开”和“保存”菜单项	77
4.1.5 添加菜单分隔条	78
4.1.6 添加“退出”菜单项	78
4.1.7 测试菜单功能	79
4.2 Check 和 RadioCheck 菜单	79
4.2.1 添加“查看”菜单组	80
4.2.2 添加 Check 类型的菜单项	80
4.2.3 添加 RadioCheck 类型的菜单项	81
4.2.4 测试 Check 和 RadioCheck 菜单功能	83
4.3 自画菜单	84
4.3.1 添加“图形”菜单组	84
4.3.2 添加自画菜单项	84
4.3.3 实现自画菜单事件方法	84

4.3.4 添加“圆形”和“直线”自画菜单项	86
4.3.5 测试自画菜单	89
4.4 创建上下文菜单	89
4.4.1 创建窗体的上下文菜单	90
4.4.2 创建控件上的上下文菜单	92
4.5 本章总结	94
第 5 章 工具栏和状态栏	95
5.1 创建工具栏	95
5.1.1 创建 ImageList 控件	95
5.1.2 创建 ToolBar 控件	96
5.1.3 添加工具栏按钮	96
5.1.4 实现工具栏按钮事件方法	97
5.1.5 测试工具栏	99
5.2 带有下拉菜单的工具栏按钮	99
5.2.1 创建 ContextMenu 控件	99
5.2.2 实现菜单项点击事件方法	100
5.2.3 把 ContextMenu 加入到【新建】工具栏按钮上	100
5.2.4 测试工具栏按钮下拉菜单	101
5.3 创建开关工具栏按钮	101
5.3.1 在 ImageList 控件中添加图像	101
5.3.2 创建 ToolBar 控件	101
5.3.3 实现按钮点击事件方法	102
5.3.4 测试开关工具栏按钮	103
5.4 创建状态栏	104
5.4.1 添加 StatusBar 控件	104
5.4.2 设置 StatusBar 属性	104
5.4.3 实现 PanelClick 事件	106
5.4.4 动态设置状态栏信息	106
5.4.5 演示状态栏	107
5.5 自画状态栏面板	107
5.6 本章总结	109
第 6 章 对话框	111
6.1 模式对话框	111
6.1.1 添加窗体类	111
6.1.2 编辑窗体	113
6.1.3 实现对话框功能	114
6.1.4 返回对话框关闭方式	116

6.1.5 定义访问对话框数据的属性.....	117
6.1.6 使用模式对话框.....	120
6.1.7 演示对话框.....	122
6.2 非模式对话框.....	123
6.2.1 创建应用程序.....	123
6.2.2 添加窗体类.....	123
6.2.3 编辑窗体.....	123
6.2.4 定义访问对话框数据的属性.....	124
6.2.5 实现控件事件.....	125
6.2.6 使用模式对话框.....	127
6.2.7 演示非模式对话框.....	129
6.3 通用对话框.....	129
6.3.1 OpenFileDialog.....	130
6.3.2 SaveFileDialog.....	132
6.3.3 ColorDialog.....	134
6.3.4 FontDialog.....	135
6.4 本章总结.....	137
第7章 多文档界面应用程序.....	139
7.1 创建主窗体及其控件.....	139
7.1.1 创建主窗体.....	139
7.1.2 添加控件.....	140
7.2 创建子窗体及其控件.....	142
7.2.1 创建子窗体.....	142
7.2.2 添加控件.....	142
7.3 实现子窗体功能.....	143
7.3.1 编辑功能.....	144
7.3.2 保存功能.....	144
7.4 实现主窗体功能.....	145
7.4.1 实现菜单栏功能.....	145
7.4.2 实现工具栏功能.....	149
7.5 演示多文档界面程序.....	151
7.6 本章总结.....	152
第8章 GDI+编程基础.....	153
8.1 GDI+简介.....	153
8.1.1 GDI+编程模式.....	153
8.1.2 GDI+的新特性.....	154
8.1.3 GDI+框架结构.....	155

8.2 绘图平面.....	155
8.2.1 创建 Graphics 对象.....	155
8.2.2 使用 Graphics 对象.....	156
8.3 Color 结构.....	157
8.3.1 创建颜色对象.....	157
8.3.2 预定义颜色.....	158
8.4 画笔 (Pen)	158
8.4.1 创建画笔对象.....	158
8.4.2 预定义画笔对象.....	159
8.4.3 画笔高级属性.....	160
8.5 刷子 (Brush)	165
8.5.1 SolidBrush 类.....	165
8.5.2 HatchBrush 类.....	166
8.5.3 TextureBrush 类.....	168
8.5.4 LinearGradientBrush 类.....	169
8.5.5 PathGradientBrush 类.....	172
8.6 字体 (Font)	176
8.7 混合和变换.....	177
8.7.1 混合 (Blend)	178
8.7.2 变换 (Transform)	180
8.8 本章总结.....	183
 第 9 章 图形和图像.....	 185
9.1 绘制图形.....	185
9.1.1 直线.....	185
9.1.2 矩形.....	186
9.1.3 椭圆.....	187
9.1.4 圆弧.....	188
9.1.5 圆饼.....	189
9.1.6 多边形.....	190
9.1.7 曲线.....	192
9.1.8 路径.....	196
9.2 操作图像.....	198
9.2.1 显示图像.....	198
9.2.2 保存图像.....	202
9.2.3 特效处理.....	204
9.3 本章总结.....	206

第 10 章 打印文档.....	207
10.1 打印组件.....	207
10.1.1 PrintDocument 类.....	207
10.1.2 PrintPreviewDialog 类.....	209
10.1.3 PrintDialog 类.....	210
10.1.4 PageSetupDialog 类.....	210
10.2 打印实例.....	211
10.2.1 创建工程.....	211
10.2.2 实现打印功能.....	212
10.2.3 实现打印设置功能.....	215
10.2.4 实现页面设置功能.....	216
10.2.5 实现打印预览功能.....	217
10.3 本章总结.....	218
第 11 章 拖放操作和剪贴板.....	219
11.1 拖放编程基础.....	219
11.1.1 DoDragDrop 方法.....	219
11.1.2 AllowDrop 属性.....	220
11.1.3 “目标”中的拖放事件.....	220
11.1.4 “源”中的拖放事件.....	222
11.2 进程内拖放操作.....	223
11.2.1 创建工程.....	223
11.2.2 实现 TreeView 的事件方法.....	224
11.2.3 实现“拖”功能.....	224
11.2.4 实现“放”功能.....	225
11.2.5 演示实例.....	231
11.3 进程间拖放操作.....	232
11.3.1 创建工程.....	232
11.3.2 添加显示功能.....	232
11.3.3 实现“拖”功能.....	234
11.3.4 实现“放”功能.....	234
11.3.5 演示实例.....	235
11.4 剪贴板操作.....	236
11.4.1 Clipboard 方法.....	236
11.4.2 剪贴板实例.....	237
11.5 本章总结.....	240
第 12 章 组件设计.....	241
12.1 相关概念.....	241

12.1.1 组件 (Component)	241
12.1.2 控件 (Control)	242
12.1.3 容器 (Container)	242
12.1.4 场所 (Site)	242
12.2 技术基础.....	242
12.2.1 属性 (Property)	242
12.2.2 事件 (Event) 和代表 (Delegate)	244
12.2.3 特性 (Attribute)	245
12.2.4 授权 (License) 组件和控件	248
12.3 设计 Windows Forms 控件	250
12.3.1 创建工程.....	250
12.3.2 添加属性.....	252
12.3.3 添加事件.....	256
12.3.4 覆盖基类方法.....	259
12.3.5 绘制控件.....	259
12.3.6 使用控件.....	261
12.4 本章总结.....	266
第 13 章 进程和线程.....	267
13.1 了解 AppDomain.....	267
13.1.1 Windows 32 进程模式	267
13.1.2 .NET 进程模式.....	268
13.1.3 AppDomain 类	268
13.2 操作进程.....	270
13.2.1 Process 类	271
13.2.2 应用实例.....	272
13.3 操作线程.....	277
13.3.1 Thread 类	278
13.3.2 多线程实例.....	280
13.4 多线程同步.....	286
13.4.1 加锁 (Lock)	286
13.4.2 监视器 (Monitor)	290
13.4.3 互斥体 (Mutex)	296
13.4.4 定时器 (Timer)	300
13.5 线程池.....	303
13.5.1 ThreadPool 类.....	303
13.5.2 ThreadPool 实例.....	304
13.6 本章总结.....	306