

///
.net

.NET 平台研究与开发丛书

C# 语言程序设计

■ 前沿软件研究室 温谦 郑小平 刘涛 编著

基于
Microsoft.NET Framework SDK

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

C# 语言程序设计

■ 前沿软件研究室 温谦 郑小平 刘涛 编著

人民邮电出版社

图书在版编目 (CIP) 数据

C#语言程序设计 / 前沿软件研究室编; 温谦, 郑小平, 刘涛编著.
—北京: 人民邮电出版社, 2001.7

ISBN 7-115-09469-1

I. C... II. ①前... ②温... ③郑... ④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 43418 号

.NET 平台研究与开发丛书

C#语言程序设计

-
- ◆ 编 著 前沿软件研究室
温 谦 郑小平 刘 涛
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
北京顺义振华印刷厂印刷
新华书店总店北京发行所经销
- ◆ 开本: 787×1092 1/16
印张: 27
字数: 635 千字 2001 年 7 月第 1 版
印数: 1-6 000 册 2001 年 7 月北京第 1 次印刷
- ISBN 7-115-09469-1/TP·2339
-

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

Microsoft 公司新推出的 C#语言具有诸多新的特性,它舍弃了 C 和 C++ 中很多使用复杂、容易出现编程错误的特性和功能,以崭新的姿态呈现在世人面前。不久的将来, C#会成为每一位资深程序员首选的编程语言。

本书分为 3 个部分。入门部分介绍了 C#语言的新特性以及 C#语言的开发工具;语法部分展示了 C#语言的各个细节,每一章介绍一个主题;提高部分讲解了 Microsoft.NET 平台中常用的类和服务。

本书体系结构清晰,解释说明详尽,开发实例丰富。本书附赠一张光盘,其中包括书中所有实例的源代码。

本书适用于对面向对象程序语言有所了解的读者,同时可作为计算机爱好者系统学习 C# 语言的教材。

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, Microsoft 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, Microsoft 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。Microsoft 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、现代、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应用程序, 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

什么是 C#语言

在介绍 C#（读作 C Sharp）之前，先介绍一个人，Anders Hejlsberg。他是 C#语言的首席设计师，并领衔设计了 .NET Framework 体系结构。中国的程序员可能对这个名字不是非常熟悉，但是他做过的工作是无人不晓的。他曾是著名的 Borland 公司的第一代程序员，是 Turbo Pascal 的创始人，以及 Delphi 产品线的首席设计师。1996 年 Anders 加盟 Microsoft 公司，领导了 Visual J++ 和 WFC（Windows Foundation Classes）的开发。因为他的贡献突出，所以在 1999 年被授予“杰出工程师”（Distinguished Engineer）的殊荣。C#作为一个新语言，基于对 Anders Hejlsberg 的了解，更增加了我们对 C#的信任。

Microsoft 公司在职业开发人员技术大会上正式发布了 C#语言，掀开了高级语言的新篇章。C#是提高生产性能的程序设计语言。开发者的目的是为了能够以更少的源代码完成更多的事情。因为在 C#中，可以灵活应用在 C 或 C++中已经熟练了的技巧，从而可以削减开发成本并缩短开发时间。

C#是一种以 C 和 C++为基础的语言，它适用于 Microsoft 公司公布的基于 .NET 平台的 XML 基础应用开发业务。利用 C#可以开发任何古典风格的 Windows 程序。不仅如此，除了硬件驱动程序，不管是控制程序、图形程序、服务程序，还是普通组件，甚至是 Web 页面，都可以用 C#开发出来。

为什么要使用 C#

长期以来，C/C++都是广泛地应用于商用软件开发中的开发语言，但其强大的功能和灵活性也同时伴有开发的复杂性。Java 是一个开放、标准、通用的网络运算平台，它具有强大的兼容性和跨平台性，现已经成为在互联网技术领域广泛采用的成熟的技术平台。但因为纯 Java 编程的应用系统运行速度太慢，且据 Forrest 的调查报告显示，目前基于 Java 开发的应用系统并没有实现百分之百的跨平台，所以 Java 仍旧未能完全取代 C/C++。此时，Microsoft 公司推出了全新的 C#语言。Microsoft 公司对 C#的定义是：“C#是一种类型安全的、现代的、简单的，由 C 和 C++衍生出来的，面向对象的编程语言，它牢牢根植于 C 和 C++语言中，并可很快被 C 和 C++的使用者所熟悉。C#的目的就是综合 Visual Basic 的高生产率和 C++的行动力。”这个定义是恰如其分的，因为在 Visual Studio.NET 中，用户已经可以利用 C#极其简便地开发出强大的企业级分布式应用系统了。

C#是 Microsoft 公司为构造基于 Web 的应用程序的“.NET”平台而设计的，它将与 Internet

标准很好地协调工作,特别是在数据传送的 XML 中,以及为在各网站之间连接各代码的 Soap (简单对象存取协议) 协议中。C#和 C、C++的差别在于 C#有更简单的语法、紧密集成的组件和函数、类型安全等。它将比 Java 更适合融入 Microsoft 的技术。C#在增加生产力和可靠性功能方面仿照了 Java 的做法,比如为防止在各种变量之间错误地搭配而采取的强大的类型执行,以及采集无用信息来自动地进行内存分配。

在 Microsoft 公司即将发布的 Visual Studio 7.0 开发软件包中,C#取代了 J++,并且 Microsoft 公司已经将 C#技术规格呈送给 ECMA (欧洲标准机构) 批准。

本书结构

本书大致可分为入门、语法和提高 3 个部分。

- 入门

第 1 章和第 2 章属于入门部分。第 1 章介绍了 C#语言的新特性及其与其他程序设计语言特别是 C 和 C++的比较;第 2 章介绍了 C#语言的开发工具。根据本书的特点,重点介绍了 Microsoft.NET Framework SDK 中编译工具的使用,这是阅读本书的基础。

- 语法

第 3 章到第 18 章属于语法部分。这是本书的主体部分,也是本书的核心部分。这一部分内容展示了 C#语言的各个细节,每一章是一个主题。无论是 C#语言的初学者还是已经有一定基础的读者,都会发现这部分内容在进行学习和开发过程中是必需的。本书简单扼要的用实例对 C#语言的运用进行了示范,这也许是本书最大的特点。程序设计语言的掌握水平往往是在大量实例的练习中不断提高的。

- 提高

第 19 章到第 22 章属于提高部分。这几章着重介绍了 Microsoft.NET 平台中常用的类和服务,包括时间和计时器的运用、文件操作、多线程以及 Windows 界面设计。虽然这部分的内容很基础,只是 C#开发应用程序的极小的一部分,但通过学习,读者会体会到在 Microsoft.NET 平台上使用 C#语言将会有很大的收获。

本书特色

本书全面且系统地介绍了 C#语言的语法,并以大量的实例演示了每个语法现象的实际应用和效果。这不但可以让初学者系统而深入的学习,也可以让对 C#语言有一定了解的读者加深对编程细节的印象。

对于一位有追求和创造力的程序员来说,学习编程语言绝对不是目的,因此本书还介绍了 C#语言在应用软件设计方面的应用,包括文件操作、GUI 编程等,这虽然不是本书的重点,却是从学习语言转向开发程序的起点。

总之,清晰的体系结构、详尽的解释说明、丰富的编程实例是本书的 3 个特色。

适用读者

如果读者对 C++有一定程度的了解,那么学习 C#会很容易。读者也可将本书作为参考手册,随时查阅。

如果读者熟悉其他面向对象语言，例如 Java、Delphi 等，也就是说熟悉了“面向对象”的思想，就只需跟随本书学习如何用 C# 实现这种思想。

对初涉这个领域的读者来说，本书可以作为系统学习 C# 语言的教材。要掌握这门语言，包括两个最重要的方面：一是理解和建立起成熟的编程思想方法。任何一种程序设计语言在本质上同样是一个外壳，它封装的是程序员的思想。因此读者要逐渐地掌握并建立起这种成熟的思想方法；另一方面就是，在建立起思想方法的同时，不能忽视语言本身的作用，没有扎实的语言基础是不会成为编程高手的。

特别说明

对于本书介绍的源程序都在本书的随书光盘中给出，使用 Microsoft.NET Framework SDK Beta2 编译运行。

考虑到部分读者仍在使用 Beta1 版，我们在随书光盘中给出了在 Beta1 中可以编译运行的源代码。两个版本的源程序分别为 Beta1 和 Beta2 子目录中。Beta1 和 Beta2 版语言本身没有任何改变，但是类库有少量改变，因此需要读者注意。

对于本书介绍的内容以及源程序，不需要 Visual Studio.NET 7.0 开发环境，仅使用 Microsoft.NET Framework SDK 通过命令行方式编译即可。但是可能很多读者对 Visual Studio.NET 7.0 很感兴趣，因此我们在附录中对此进行了介绍，如果读者希望详细了解有关内容，请参阅本套丛书的相关书籍。

关于作者

本书由温谦、郑小平、刘涛编著，刘胜雷等担任了全书的文档处理工作。在本书的编写过程中，得到了中国原子能科学研究院的乔立严研究员以及清华大学张文俊教授的热情指导及多方面的帮助，在此表示诚挚的感谢！

由于时间仓促，水平有限，书中难免有错误疏漏之处，恳请专家、同行和广大读者指正，我们会全力改进，在后续的工作中不断提高。

编 者
2001.5

目 录

第 1 章 C#语言概述	1
1.1 C#语言简介	1
1.2 C#语言的特点	2
1.2.1 简单 (Simplicity)	3
1.2.2 现代 (Modernity)	4
1.2.3 面向对象 (Object-orientation)	4
1.2.4 类型安全 (Type-safety)	5
1.2.5 版本可控 (Versionable)	5
1.2.6 兼容 (Compatibility)	5
1.2.7 灵活 (Flexibility)	6
1.3 C#和 C、C++ 的比较	6
1.3.1 定义全局变量	6
1.3.2 指针 (Pointer)	6
1.3.3 内存管理	7
1.3.4 数据类型的支持	7
1.3.5 类型转换	7
1.3.6 头文件	7
1.3.7 预处理	7
第 2 章 开发工具	9
2.1 SDK 的安装	9
2.1.1 安装准备工作	9
2.1.2 开始安装	10
2.2 开发工具的使用	13
2.2.1 Hello World 程序	13
2.2.2 编译动态连接库	14
2.2.3 使用动态连接库	15
第 3 章 数据类型	17
3.1 数值类型	17

3.1.1	bool 类型	18
3.1.2	char 类型	20
3.1.3	byte 类型	20
3.1.4	sbyte 类型	22
3.1.5	short 类型	23
3.1.6	ushort 类型	25
3.1.7	int 类型	26
3.1.8	uint 类型	27
3.1.9	long 类型	28
3.1.10	ulong 类型	29
3.1.11	float 类型	31
3.1.12	double 类型	32
3.1.13	decimal 类型	33
3.1.14	enum 类型	35
3.1.15	struct 类型	37
3.2	装箱和拆箱	40
3.2.1	装箱	40
3.2.2	拆箱	41
3.3	引用类型	43
3.3.1	class 类型	43
3.3.2	delegate 类型	46
3.3.3	interface 类型	53
3.3.4	object 类型	60
3.3.5	string 类型	61
3.4	类型参考表	62
3.4.1	内建类型表	62
3.4.2	整数类型表	63
3.4.3	浮点类型表	64
3.4.4	缺省值表	64
3.4.5	数值类型表	64
3.4.6	隐式数字转换表	65
3.4.7	显式数字转换表	65
第 4 章	修饰符	67
4.1	访问修饰符	67
4.1.1	访问属性级别	67
4.1.2	访问属性范围	68
4.1.3	使用访问属性的限制	70
4.1.4	internal 修饰符	72
4.1.5	private 修饰符	73

4.1.6	protected 修饰符	74
4.1.7	public 修饰符	75
4.2	abstract 修饰符	76
4.2.1	修饰符描述	76
4.2.2	示例	77
4.3	const 修饰符	79
4.3.1	修饰符描述	79
4.3.2	示例	80
4.4	event 修饰符	81
4.4.1	修饰符描述	81
4.4.2	示例	82
4.5	extern 修饰符	88
4.5.1	修饰符描述	88
4.5.2	示例	88
4.6	override 修饰符	90
4.6.1	修饰符描述	90
4.6.2	示例	90
4.7	readonly 修饰符	92
4.7.1	修饰符描述	92
4.7.2	示例	92
4.8	sealed 修饰符	93
4.8.1	修饰符描述	93
4.8.2	示例	93
4.9	static 修饰符	94
4.9.1	修饰符描述	94
4.9.2	示例	95
4.10	virtual 修饰符	96
4.10.1	修饰符描述	96
4.10.2	示例	97
 第 5 章 语句		101
5.1	选择语句	101
5.1.1	if-else 语句	101
5.1.2	switch-case 语句	105
5.2	循环语句	107
5.2.1	do-while 语句	107
5.2.2	for 语句	109
5.2.3	foreach-in 语句	110
5.2.4	while 语句	117
5.3	跳转语句	118

5.3.1 break 语句	118
5.3.2 continue 语句	120
5.3.3 goto 语句	121
5.3.4 return 语句	124
5.4 异常处理语句	125
5.4.1 throw 语句	125
5.4.2 try-catch 语句	126
5.4.3 try-finally 语句	129
5.4.4 try-catch-finally 语句	131
5.5 检查和非检查	132
5.5.1 checked 语句	132
5.5.2 unchecked 语句	135
5.6 非保护和固定	138
5.6.1 unsafe 语句	138
5.6.2 fixed 语句	141
5.7 锁定语句	143
5.7.1 语法规则	143
5.7.2 注解	143
5.7.3 示例 1	143
5.7.4 示例 2	144
第 6 章 方法参数	147
6.1 params 关键词	147
6.2 ref 关键词	148
6.3 out 关键词	149
第 7 章 名字空间	151
7.1 namespace 关键词	151
7.1.1 语法规则	151
7.1.2 注解	151
7.1.3 示例	152
7.2 using 关键词	153
7.2.1 语法规则	153
7.2.2 注解	153
7.2.3 示例 1	153
7.2.4 示例 2	154
第 8 章 操作符关键词	157
8.1 as 关键词	157
8.1.1 语法规则	157

8.1.2 注解	157
8.1.3 示例	158
8.2 is 关键词	159
8.2.1 语法规则	159
8.2.2 注解	159
8.2.3 示例	159
8.3 new 操作符	161
8.3.1 语法规则	161
8.3.2 注解	161
8.3.3 示例	161
8.4 new 修饰符	163
8.4.1 语法规则	163
8.4.2 示例 1	164
8.4.3 示例 2	165
8.5 sizeof 关键词	166
8.5.1 语法规则	166
8.5.2 注解	166
8.5.3 示例	166
8.6 typeof 关键词	167
8.6.1 语法规则	167
8.6.2 注解	167
8.6.3 示例 1	167
8.6.4 示例 2	169
8.7 true 关键词	169
8.7.1 注解	169
8.7.2 示例	170
8.8 false 关键词	170
8.8.1 注解	170
8.8.2 示例	171
8.9 stackalloc 关键词	171
8.9.1 语法规则	171
8.9.2 注解	171
8.9.3 示例	171
第 9 章 转换关键词	173
9.1 explicit 关键词	173
9.1.1 注解	173
9.1.2 示例	174
9.2 implicit 关键词	174
9.2.1 注解	175

9.2.2 示例	175
9.3 operator 关键词	176
9.3.1 语法规则	176
9.3.2 注解	177
9.3.3 示例	177
第 10 章 访问关键词	179
10.1 base 关键词	179
10.1.1 注解	179
10.1.2 示例 1	179
10.1.3 示例 2	180
10.2 this 关键词	182
10.2.1 注解	182
10.2.2 示例	182
第 11 章 操作符	185
11.1 操作符分类	185
11.2 算术溢出	185
11.3 操作符详解	186
11.3.1 可重载操作符	186
11.3.2 []操作符	186
11.3.3 ()操作符	187
11.3.4 .操作符	188
11.3.5 +操作符	189
11.3.6 -操作符	190
11.3.7 *操作符	191
11.3.8 /操作符	191
11.3.9 %操作符	192
11.3.10 &操作符	192
11.3.11 !操作符	193
11.3.12 ^操作符	194
11.3.13 !操作符	194
11.3.14 ~操作符	195
11.3.15 =操作符	195
11.3.16 <操作符	196
11.3.17 >操作符	196
11.3.18 ?:操作符	197
11.3.19 ++操作符	198
11.3.20 --操作符	199
11.3.21 &&操作符	200

11.3.22	操作符.....	202
11.3.23	<<操作符.....	203
11.3.24	>>操作符.....	204
11.3.25	==操作符.....	205
11.3.26	!=操作符.....	206
11.3.27	<=操作符.....	207
11.3.28	>=操作符.....	207
11.3.29	+=操作符.....	208
11.3.30	-=操作符.....	208
11.3.31	*=操作符.....	209
11.3.32	/=操作符.....	209
11.3.33	%=操作符.....	210
11.3.34	&=操作符.....	210
11.3.35	=操作符.....	211
11.3.36	^=操作符.....	212
11.3.37	<<=操作符.....	212
11.3.38	>>=操作符.....	213
11.3.39	->操作符.....	213
11.4	操作符的重载.....	214
11.4.1	复数类.....	214
11.4.2	三值逻辑类型.....	215
第 12 章 特性.....		219
12.1	特性概述.....	219
12.1.1	使用特性.....	219
12.1.2	全局特性.....	220
12.1.3	创建定制特性.....	221
12.1.4	获取特性信息.....	222
12.2	AttributeUsage 特性.....	224
12.2.1	语法规则.....	224
12.2.2	注解.....	224
12.3	Conditional 特性.....	224
12.3.1	语法规则.....	224
12.3.2	注解.....	225
12.3.3	示例.....	225
12.4	Obsolete 特性.....	226
12.4.1	语法规则.....	226
12.4.2	注解.....	227
12.4.3	示例.....	227
12.5	应用实例.....	227

12.5.1	声明特性类	228
12.5.2	使用特性类	229
12.5.3	综合示例	230
第 13 章 预处理指令		237
13.1	#if 预处理指令	237
13.1.1	语法规则	237
13.1.2	注解	238
13.1.3	示例	238
13.2	#else 预处理指令	239
13.2.1	语法规则	239
13.2.2	注解	239
13.2.3	示例	239
13.3	#elif 预处理指令	239
13.3.1	语法规则	239
13.3.2	注解	240
13.3.3	示例	240
13.4	#endif 预处理指令	240
13.4.1	语法规则	240
13.4.2	注解	240
13.4.3	示例	240
13.5	#define 预处理指令	240
13.5.1	语法规则	240
13.5.2	注解	241
13.5.3	示例	241
13.6	#undef 预处理指令	241
13.6.1	语法规则	241
13.6.2	注解	241
13.6.3	示例	241
13.7	#warning 预处理指令	242
13.7.1	语法规则	242
13.7.2	注解	242
13.7.3	示例	242
13.8	#error 预处理指令	243
13.8.1	语法规则	243
13.8.2	注解	243
13.8.3	示例	243
13.9	#line 预处理指令	244
13.9.1	语法规则	244
13.9.2	注解	244