

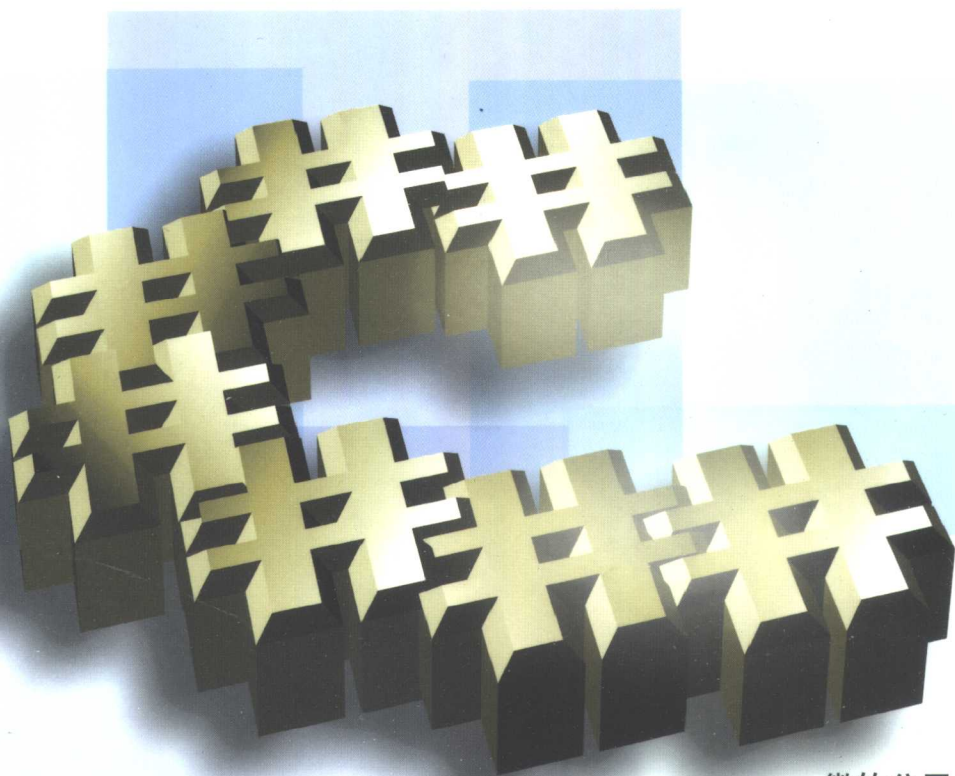
Microsoft
.net

微软 Visual Studio.NET
程序开发系列丛书

Microsoft® Press



C# 编程技术



微软公司
东方人华 编著



清华大学出版社
<http://www.tup.tsinghua.edu.cn>

微软 Visual Studio.NET 程序开发系列丛书

C# 编程技术

微软公司
东方人华 编著

清华大学出版社

(京)新登字 158 号

内 容 简 介

C#是编写企业应用程序的一种简单而功能强大的编程语言。本书介绍了 C#语言的特性、C#程序的创建方法、C#代码向导、C#指南、C#关键字、C#操作符、C#属性,以及 C#预处理程序指示。另外,本书还介绍了 C#语言的优点。本书对每一部分均给出了其相关操作符和关键字等的详尽代码举例。

本书结构层次简单清晰,举例详尽,可作为广大 C#语言编程者及其爱好者的助手,也可供相关语言爱好者参考使用。

C# Programming

Microsoft Press

Copyright © 2000 by Microsoft Corporation

Original English language edition published by Microsoft Press, a Division of Microsoft Corporation.

All rights reserved.

No part of the contents of this book may be reproduced or transmitted in any form or by any means without the written permission of the publisher. For sale in the People's Republic of China only.

北京市版权局著作权合同登记号: 图字 01-2001-0130 号

本书中文简体版由 Microsoft Press 授权清华大学出版社出版发行, 未经出版者书面许可, 不得以任何方式复制或抄袭本书的任何部分。

版权所有, 翻印必究。

本书封面贴有清华大学出版社激光防伪标签, 无标签者不得销售。

书 名: C# 编程技术

作 者: 微软公司 东方人华

出 版 者: 清华大学出版社(北京清华大学学研大厦, 邮编 100084)

<http://www.tup.tsinghua.edu.cn>

印 刷 者: 北京市密云胶印厂

发 行 者: 新华书店总店北京发行所

开 本: 787×960 1/16 印 张: 24.5 字 数: 536 千字

版 次: 2001 年 6 月第 1 版 2001 年 6 月第 1 次印刷

书 号: ISBN 7-302-04389-2/TP·2579

印 数: 0001~4000

定 价: 42.00 元

《微软 Visual Studio .NET 程序开发系列丛书》序

.NET 技术简介

.NET 技术是 2000 年微软公司推出的全新概念，对于它很难做出一个明确的定义，它代表了一个集合、一个环境、一个编程的基础结构。其目的是将互联网本身作为构建新一代操作系统的基础，对互联网和操作系统的设计思想进行延伸。具体地说，.NET 技术就是要在不同的网站之间建立起协定，促使网站之间的协同合作，实现信息的自动交流，从而帮助用户最大限度地获取信息、并对他们的数据进行简单、高效的管理。

对于最终用户来说，.NET 技术的实现将使计算机的功能将得到大幅度的提升，而计算机的操作却变得更加简单。他们将完全摆脱人为的硬件束缚，利用任何设备、通过任何系统、在任何时间、任何地点访问互联网的多维时空，并对其进行跨应用程序的集成。用户对个人信息的任何修改——无论是通过个人电脑、便携设备还是灵通卡——将即时和自动地通知到所有需要这些信息的地方。

对于应用程序的开发人员来说，.NET 的意义也许更为重大。通过 .NET 技术，程序开发人员不必再像过去那样通过集成本地系统服务来构建应用程序，花费大量的精力来考虑如何构建基本的结构，而无法专注于如何实现软件所特有的商业价值。现在，程序开发人员可以访问 Microsoft .NET 所提供的内容丰富的类库，直接使用各种各样来自于 Internet 的模块来构建自己的应用程序，而无需为不同的工作环境重新编写程序，快速、高效地开发应用程序。

对于网络开发人员来说，借助 .NET 技术所提供的、基于 XML 的松散耦合技术，他们终于可以真正地将应用程序逻辑分发到网络上，而不必顾及服务端与客户端是否属于同一种结构、执行机制是否相同、接口是否对应。Microsoft .NET 将把 ASP 和程序集成带到一个新的水平上，将集成程序和其他的集成或非集成程序整合；把这些程序风格化；形成针对这些应用程序编写新程序的能力；并可以在离线状态下运行它们。

Visual Studio.NET 简介

Visual Studio.NET 是 Microsoft 为实现其 .NET 技术而开发的一整套工具组件。它把开发功能强大、性能可靠的企业网络解决方案进行了简化。通过提供终端到终端的网络开发能力以及可伸缩、可复用的服务器端组件, Visual Studio.NET 将会大大地提高生产率, 并促使商务活动更加有效地关注快速多变而又充满竞争的市场需求。

Visual Studio.NET 具有许多令人激动人心的特性, 其中的某些特性是对早期 Visual Studio®版本的增强, 而另外一些特性则是全新的。这些新特性包括: 一个单一的集成化开发环境、Visual Basic.NET 当中面向对象的新特性、以及大量在产品开发生命周期中使用的工具。

Visual Studio.NET 的主要作用在于:

- **提供加速开发过程的高效工具**

Visual Studio.NET 提供了一个统一的、紧密集成的可视化编程环境, 以帮助您简化开发网络应用程序的过程、缩短学习使用方法的时间。通过共享的 HTML, XML 和样式单编辑器, 您可以轻轻松松地借助包括 C#在内的任何一种 Visual Studio 语言来开发网络应用程序。

- **提供对各种各样的网络应用程序的快速设计能力**

借助于 Web Forms, 程序员们可以使用他们在开发基于窗体的桌面应用程序时所使用技巧来创建跨平台、跨浏览器的网络应用程序。Web Forms 被用于网络的服务器端, 它能高速地运行, 并生成符合 HTML 3.2 规范、能够在各种浏览器上阅读的文档。

- **利用 XML 和 Web Services 来简化分布式计算**

Web Services 借助标准的 Internet 协议在网络上调用商务逻辑。HTTP 被作为 Web Services 传输的基础协议, 该协议使得对功能的请求能够穿越各种团体所使用的防火墙。XML 被用来对上述功能请求的参数进行格式统一, 从而使得这些请求能够适用于所有的软件和硬件。这样使得对 Web Services 的访问可以通过任何一种语言、使用任何一种组件模型、在任何一种操作系统上实现。

- **快速构建中间层商务组件**

Visual Studio 的一个核心目标就是要为基于服务器的应用程序提供应用程序快速部署(Rapid Application Development, RAP)工具。Visual Studio.NET 在创建中层商务逻辑时所表现出的效率丝毫不逊于 Visual Basic®开发窗体程序时的效率。利用 Visual Studio.NET 创建的组件将为您的商务运作提供足够的功能和伸缩性。

- **构建可靠的可伸缩解决方案**

利用 Visual Studio.NET, 您可以非常轻松地创建具有自动伸缩能力的可靠的应用程序和组件。例如, 如果使用 Visual C++来编写网络应用程序, 新推出的 ALT Server 将会极大地提高性能。

Visual Studio.NET 的其他新性能包括:

- **一种新的语言: C#**

C#语言是一种优雅、简单、类型安全、面向对象的编程语言。它在向 C 和 C++的用户提供 RAD 的同时保持了 C/C++所特有的强大功能和控制能力。C#的关键特性包括:

- 具有与 C++十分相似的模型和语法。99%的语句、表达式和运算符都与 C++的相同
- 具有与 COM+服务完全的互换性
- 提供对 COM 和平台的完全支持能力, 使您能够轻松地迁移现有的代码
- 提供自动的无用单元收集能力
- 提供可扩展的分类元数据, 使您可以声明新类型和各种元数据
- 对基于网络的组件交互提供 XML 支持

Visual Studio.NET 的新特性使之为基于微软 .NET 框架(Microsoft .NET Framework, 微软下一代的网络应用程序开发平台)的应用程序提供了一个完整的开发环境。Visual Studio.NET 提供的关键性基础技术简化了安全、可伸缩、高度可用的网络应用程序和 Web Services 的创建、部署和更新的方法, 并为程序员们施展他们的技巧提供了广阔的空间。

除此以外, Microsoft .NET Framework 还帮助网络开发人员使用自己所喜爱的语言按照使用本地对象的方式来使用 Web Services。程序开发人员可以把注意力投向如何增强产品所特有的功能, 以为公司赢得竞争优势。这样的结果是更快的产品开发速度、更高的开发效率和更高的产品质量。

本套丛书的内容与安排

本套丛书取材于由微软出版社提供、独家授权的大量第一手资料, 本套丛书目前包括五册, 分别为:

- 《C#语言参考手册》

深入详细地介绍 C#语言的语法规则和例外情况, 为学习者提供从 C/C++到 C#的快速入门指南。

- 《C#编程技术》

用大量的程序代码来比较 C#与 C/C++的异同, 并具体地说明如何利用 C#访问

Microsoft .NET 类库，编写功能强大的应用程序。

- 《Visual Studio.NET 开发环境使用指南》

详细地介绍新一代的应用程序开发环境的使用方法和技巧，为您对 Visual Studio.NET 开发环境的亲身体验提供全面详细的使用指南。

- 《Visual Basic.NET 程序设计》

在对 Visual Basic 6.0 和 Visual Basic.NET 比较的基础上，详细地介绍后者的新特性，以及这些特性如何使得大型复杂的应用程序的开发变得轻松容易。

- 《XML 3.0 技术内幕》

详细深入地介绍最新版本的 XML (Extensible Markup Language, 可扩展标记语言)的语言规范和使用方法，帮助读者快速深入地了解这项当今网络的核心技术。

随着技术的发展，并根据读者的需要，我们将陆续增加新的书目。

本套书在编写的过程中面临着如何迅速地把最新的 IT 技术传播给国内的广大读者的巨大压力。迫于时限，在章节篇幅的安排上秉承“形式服从于内容”的宗旨，因而造成某些章节篇幅远远超出其他章节的情况。另外，由于本书所讨论技术的新颖性，使得可资借鉴的中文资料相当稀少，加上编者们的水平有限，故而某些术语的译称可能并不尽如人意，甚至可能还会有不少错误，对于这些问题，恳请广大读者批评指正。

编者

2001 年 4 月

目 录

第 1 章 C# 语言概述	1
1.1 C# 特性的简单概述.....	1
1.2 使用 C# 的原因.....	1
1.3 Hello World! 程序的 C# 版本.....	2
1.3.1 注释.....	2
1.3.2 Main 方法.....	3
1.3.3 输入和输出.....	3
1.3.4 编译和执行.....	4
1.4 C# 程序的一般结构.....	4
1.5 使用完全合格名称.....	5
1.6 创建和使用 C# DLL.....	6
1.7 C++ 和 C# 的比较.....	8
1.8 C# 文档映射.....	10
第 2 章 创建 C# 程序	11
2.1 从命令行创建.....	11
2.2 C# 编译选项.....	12
2.3 编译错误.....	33
第 3 章 C# 代码向导	35
3.1 添加 C# 类.....	35
3.1.1 通过 Solution Explorer 添加类.....	35
3.1.2 通过 Class View 添加类.....	35
3.1.3 C# 添加类向导.....	36
3.2 向 C# 类中添加方法.....	37
3.3 向 C# 类中添加属性.....	38
3.4 向 C# 类中添加变量.....	40
3.5 向 C# 类中添加索引器.....	41

3.6 向 C# 界面中添加 Method(方法).....	42
3.7 向 C# 界面中添加属性.....	43
3.8 向 C# 界面中添加索引器.....	43
3.9 设置 C# 工程属性.....	44
3.9.1 Build、Configuration Properties 及 <Projectname>Property Pages 对话框.....	44
3.9.2 Debug、Configuration Properties 及 <Projectname>Property Pages 对话框.....	45
第 4 章 C# 指南.....	47
4.1 Hello World.....	48
4.2 命令行参数.....	50
4.3 数组.....	52
4.3.1 数组概况.....	52
4.3.2 声明数组.....	52
4.3.3 初始化数组.....	54
4.3.4 访问数组成员.....	55
4.3.5 数组是对象.....	56
4.3.6 在数组中使用 foreach.....	56
4.4 属性.....	56
4.5 库.....	62
4.6 版本管理.....	66
4.7 集合类.....	68
4.8 结构体.....	73
4.8.1 结构体和类的比较.....	74
4.8.2 堆或堆栈.....	74
4.8.3 构造器和继承.....	76
4.8.4 结构体的有关属性.....	76
4.9 索引器.....	77
4.10 用户自定义转换.....	84
4.11 操作符重载.....	87
4.12 代表.....	92
4.13 事件.....	98
4.13.1 事件和继承.....	102
4.13.2 界面中的事件.....	102
4.13.3 .NET Framework 方针.....	102
4.14 显式界面的实现.....	105

4.15 条件方法.....	109
4.16 XML 文档.....	112
4.17 平台调用.....	116
4.18 COM InterOp	120
4.19 属性.....	131
4.20 安全.....	138
4.21 非安全代码.....	145
4.22 ADO.....	147
第 5 章 C# 关键字.....	151
5.1 类型.....	152
5.1.1 值类型.....	152
5.1.2 引用类型.....	174
5.1.3 void.....	183
5.1.4 类型引用表.....	183
5.2 限定符.....	188
5.2.1 访问限定符.....	189
5.2.2 abstract(抽象).....	198
5.2.3 const(常量).....	200
5.2.4 event(事件).....	202
5.2.5 extern(外部).....	208
5.2.6 override(覆盖).....	210
5.2.7 readonly(只读).....	211
5.2.8 sealed(封装).....	213
5.2.9 static.....	214
5.2.10 virtual.....	216
5.3 语句.....	218
5.3.1 选择语句.....	219
5.3.2 重复语句.....	224
5.3.3 转移语句.....	234
5.3.4 例外处理语句.....	240
5.3.5 Checked 和 Unchecked.....	246
5.3.6 Unsafe 和 fixed	252
5.3.7 Lock 语句.....	255
5.4 方法参数.....	258

5.4.1	params	258
5.4.2	ref.....	260
5.4.3	out.....	261
5.5	名称空间.....	262
5.5.1	namespace	262
5.5.2	using	264
5.6	操作符关键字.....	267
5.6.1	as.....	267
5.6.2	is	269
5.6.3	new	270
5.6.4	sizeof	275
5.6.5	typeof.....	276
5.6.6	true.....	278
5.6.7	false	279
5.6.8	stackalloc	279
5.7	转换关键字.....	281
5.7.1	explicit.....	281
5.7.2	implicit.....	282
5.7.3	operator.....	283
5.8	访问关键字.....	285
5.8.1	base.....	286
5.8.2	this	288
5.9	Literal 关键字	290
5.10	ull	290
第 6 章	C# 操作符.....	291
6.1	算术溢出.....	291
6.2	可过载操作符.....	292
6.3	[]操作符	293
6.4	()操作符	294
6.5	操作符.....	294
6.6	+ 操作符	295
6.7	- 操作符	296
6.8	* 操作符.....	297
6.9	/操作符.....	297

6.10	% 操作符	298
6.11	& 操作符	299
6.12	操作符	299
6.13	^ 操作符	300
6.14	! 操作符	301
6.15	~ 操作符	301
6.16	= 操作符	302
6.17	< 操作符	303
6.18	> 操作符	303
6.19	?: 操作符	304
6.20	++ 操作符	305
6.21	-- 操作符	306
6.22	&& 操作符	307
6.23	操作符	308
6.24	<< 操作符	309
6.25	>> 操作符	311
6.26	== 操作符	312
6.27	!= 操作符	313
6.28	<= 操作符	314
6.29	>= 操作符	315
6.30	+= 操作符	315
6.31	-= 操作符	316
6.32	*= 操作符	317
6.33	/= 操作符	317
6.34	%= 操作符	318
6.35	&= 操作符	319
6.36	= 操作符	319
6.37	^= 操作符	320
6.38	<<= 操作符	321
6.39	>>= 操作符	321
6.40	-> 操作符	322
第 7 章 C# 属性		323
7.1	属性介绍	323
7.1.1	使用属性	323

7.1.2 全局属性	324
7.1.3 创建自定义属性	325
7.1.4 检索属性信息	326
7.2 AttributeUsage	328
7.3 条件	329
7.3.1 参数	329
7.3.2 应用范围	329
7.3.3 说明	329
7.4 obsolete	331
第 8 章 C# 预处理程序指示	333
8.1 #if	333
8.2 #else	334
8.3 #elif	335
8.4 #endif	336
8.5 #define	336
8.6 #undef	337
8.7 #warning	337
8.8 #error	338
8.9 #line	339
8.10 #region	339
8.11 #endregion	340
第 9 章 C# 语言特征	341
9.1 数组	341
9.1.1 一维数组	341
9.1.2 多维数组	344
9.2 Main	345
9.2.1 返回值	345
9.2.2 命令行参数值	346
9.3 属性	348
9.3.1 属性声明	348
9.3.2 存取器	351
9.3.3 界面属性	356
9.4 索引器	359

9.4.1 索引器声明	359
9.4.2 属性和索引器的比较	362
9.4.3 界面索引器	363
9.5 XML 文件	365
9.5.1 文件注释标记	366
9.5.2 处理 XML 文件	376

第 1 章 C# 语言概述

本章将对 C# 语言作一个简单的概述，还将介绍一个简单的 C# 程序，以及这个程序的结构和重要特性。

下面是本章将讨论的主题：

- C# 特性的概述
- 使用 C# 的原因
- Hello World! 程序的 C# 版本
- C# 程序的一般结构
- 使用完全合格名称
- 创建和使用 DLL
- C++ 和 C# 的比较

1.1 C# 特性的简单概述

C# 非常简单，但却是编写企业应用程序的一种功能非常强大的编程语言。

C# 语言是在 C 及 C++ 语言的基础上发展起来的。它在语句、表达式和操作符方面使用了很多 C++ 的特性。

C# 在类型安全(type safety)、版本、事件和垃圾收集程序等方面又有相当多的改进和创新。

C# 能够访问通常的 API 样式：.NET、.COM、Automation 和 C 风格的 API。它还支持非安全模式，这样用户就可以使用指针来操作不由垃圾收集程序控制的内存。

1.2 使用 C# 的原因

C# 是一种良好、简单、类型安全、面向对象的语言，企业程序开发者可以使用它来创建多种应用程序。

因为具有以下优点，C# 可以为您提供创建稳定的系统级组件的能力：

- 对现存代码集成的完全 COM/Platform(COM/平台)支持。

- 因为垃圾收集和类型安全而具备的健壮性。
- 由内部代码信任机制所提供的安全性。
- 对可扩展元数据概念的完全支持。
- 用户还可以利用 C# 的下列特性，实现 C# 同其他语言、跨平台以及与传统数据的交互操作：
 - 实现对 COM+ 1.0 和 .NET 服务与紧密库访问之间的交互操作性的完全支持。
 - 对网站组件交互作用的 XML 支持。
 - 能够提供简易的管理和部署。

1.3 Hello World! 程序的 C# 版本

下面的控制台程序是传统的“Hello World!”程序的 C# 版本，这个程序显示字符串 Hello World!。

```
// A "Hello World!" program in C#
class Hello
{
    static void Main()
    {
        System.Console.WriteLine("Hello World!");
    }
}
```

下面是这个程序的一些要点：

- 注释
- Main 方法
- 输入和输出
- 编译和执行

1.3.1 注 释

第一行包含下列注释：

```
// A "Hello World!" program in C#
```

符号“//”将该行其余的部分转换为注释语句。用户也可以通过将一个文本块放在符号/*和*/之间来注释整个文本块，例如：

```
/* A "Hello World!" program in C#.
This program displays the string "Hello World!" on the screen. */
```

1.3.2 Main 方法

C# 程序必须包含一个 Main 方法(Main method), 该方法是控制开始和结束之处。用户在 Main 方法中创建对象和执行其他方法。

Main 方法是一种静态方法, 它位于一个类或者一个结构体中。在“Hello World!”例子中, Main 方法位于 Hello 类中。声明 Main 方法的方式有三种:

- 可以返回一个空值:

```
static void Main()
{
    ...
}
```

- 也可以返回一个整数:

```
static int Main()
{
    ...
    return 0;
}
```

- 还可以带有参数值:

```
static int Main(string[] args)
{
    ...
    return 0;
}
```

Main 方法的参数是一个字符串数组, 这个数组代表调用程序所使用的命令行参数值。注意, 不同于 C++ 的是, 这个数组不包括可执行文件(exe)名。

1.3.3 输入和输出

C# 程序通常使用由 .NET Framework 的运行时间库提供的输入/输出服务。语句:

```
System.Console.WriteLine("Hello World!")
```

使用了 WriteLine 方法, 这是运行时间库中 Console 类的一种输出方法。它在标准输出流上显示其字符串参数, 并且后跟一个新行。其他 Console 方法用于不同的输入和输出操作。如果用户在程序开始使用了下面的语句:

```
using System
```