

C#语言参考手册

微软公司 编著
东方人华

清华大学出版社

(京)新登字 158 号

内 容 简 介

C#是 C 和 C++语言家族的新成员,由 C 和 C++发展而来。C#是一种简单、现代、面向对象、类型安全的编程语言,其目标是将 Visual Basic 的高效率和 C++的强大功能结合在一起。

本书介绍了C#语言的基本特点以及 C#语言的语法规则和异常,并使用了大量示例代码以便于读者理解和操作。对于 C#语言的初学者及专业程序开发人员而言,本书无疑是一本理想的学习教材和参考用书。

C# Language Reference

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 01-2001-0123 号

本书中文简体字版由 Microsoft Press 授权清华大学出版社出版,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#语言参考手册

作 者: 微软公司 东方人华

责任编辑: 向 璐

出 版 者: 清华大学出版社(北京市海淀区清华大学学研大厦,邮编 100084)

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

印 刷 者:

发 行 者: 新华书店总店北京发行所

开 本: 787 × 960 1/16 印张: 27.5 字数: 592 千字

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

书 号: ISBN 7-302-04434-1/TP · 2609

印 数: 0001 ~ 4000

定 价: 49.50 元

《微软 Visual Studio .NET 程序开发系列丛书》序

删除的内容:

.NET 技术简介

删除的内容:

.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。

删除的内容:

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。

删除的内容: ——

删除的内容: ——

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。

删除的内容: 不用

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。

删除的内容: 把

删除的内容: 了

删除的内容:

删除的内容: 编程技术

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。

删除的内容:

删除的内容:

删除的内容: 加以

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。

删除的内容: 的

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、缩短学习使用的时间。通过共享的 HTML、XML 和样式单编辑器, 您可以轻松地借助包括 C#在内的任何一种 Visual Studio 语言来开发网络应用程序。

带格式的: 项目符号和编号

删除的内容:

带格式的

删除的内容: 来

- **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。

带格式的

带格式的: 项目符号和编号

删除的内容:

- **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。

删除的内容:

带格式的

带格式的: 项目符号和编号

删除的内容: 用

- **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 创建的组件将为您商务运作提供足够的功能和伸缩性。

带格式的

带格式的: 项目符号和编号

删除的内容:

删除的内容: 色

删除的内容: ii

● **构建可靠的伸缩解决方案**

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 的新特性使之成为基于微软 .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

删除的内容: iii

删除的内容: iii

带格式的

带格式的: 项目符号和编号

删除的内容:

删除的内容:

带格式的

带格式的: 项目符号和编号

删除的内容:

带格式的: 项目符号和编号

删除的内容:

删除的内容:

带格式的: 项目符号和编号

删除的内容:

带格式的: 项目符号和编号

删除的内容: iii

删除的内容: 编程技术

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。

删除的内容: NET

删除的内容: 应用程序开发环境使用指南

● 《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。

删除的内容: 指南

● 《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。

删除的内容: Extensible

删除的内容: 门

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。

删除的内容: 内容

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。

删除的内容: 能

编者

2001 年 4 月

删除的内容: ii

目 录

第 1 章 简介	1
1.1 Hello, world 程序	1
1.2 类型	2
1.2.1 预定义类型	4
1.2.2 转换	7
1.2.3 数组类型	8
1.2.4 类型系统的统一	10
1.3 变量和参量	11
1.4 自动内存管理	15
1.5 表达式	17
1.6 语句	18
1.7 类	23
1.7.1 常量	25
1.7.2 字段	26
1.7.3 方法	27
1.7.4 特性	29
1.7.5 事件	29
1.7.6 运算符	31
1.7.7 索引函数	32
1.7.8 实例构造函数	34
1.7.9 析构函数	35
1.7.10 静态构造函数	35
1.7.11 继承	36
1.8 结构体	38
1.9 接口	39
1.10 代表	40
1.11 枚举	41
1.12 名称空间与汇编	42

1.13	版本进化	44
1.14	属性	47
第 2 章	词法结构	51
2.1	翻译步骤	51
2.2	语法符号	51
2.3	预处理	52
2.3.1	预处理声明	53
2.3.2	#if、#elif、#else 和 #endif	55
2.3.3	预处理控制行	56
2.3.4	#line	57
2.3.5	预处理标识符	57
2.3.6	预处理表达式	58
2.3.7	与空白的相互作用	58
2.4	词法分析	59
2.4.1	输入	59
2.4.2	输入字符	60
2.4.3	行结束符	60
2.4.4	注释	60
2.4.5	空白	61
2.4.6	权标	62
2.5	句法分析	62
2.5.1	标识符	62
2.5.2	关键字	64
2.5.3	文字	65
2.5.4	运算符和标点符号	70
2.5.5	Unicode 字符转义序列	70
第 3 章	基本概念	73
3.1	声明	73
3.2	成员	76
3.2.1	名称空间成员	76
3.2.2	结构体成员	76
3.2.3	枚举成员	77
3.2.4	类成员	77

3.2.5 接口成员	77
3.2.6 数组成员	77
3.2.7 代表成员	77
3.3 成员访问	78
3.3.1 访问性声明	78
3.3.2 访问域	79
3.3.3 受保护的访问	81
3.3.4 访问性约束	82
3.4 签名和重载	83
3.5 作用域	84
3.6 名称空间名称和类型名称	90
第 4 章 类型	93
4.1 值类型	93
4.1.1 默认构造函数	94
4.1.2 结构体类型	95
4.1.3 简单类型	96
4.1.4 整数类型	97
4.1.5 浮点数类型	98
4.1.6 十进制数类型	99
4.1.7 布尔类型	100
4.1.8 枚举类型	100
4.2 引用类型	100
4.2.1 类类型	101
4.2.2 object 类型	102
4.2.3 string 类型	102
4.2.4 接口类型	102
4.2.5 数组类型	102
4.2.6 代表类型	102
4.3 装箱和开箱	103
4.3.1 装箱转换	103
4.3.2 开箱转换	104
第 5 章 变量	107
5.1 变量分类	107

5.1.1	静态变量	108
5.1.2	实例变量	108
5.1.3	数组元素	108
5.1.4	值参量	108
5.1.5	引用参量	109
5.1.6	输出参量	109
5.1.7	局部变量	110
5.2	默认值	110
5.3	明确赋值	110
5.3.1	初始化赋值变量	113
5.3.2	初始化未赋值变量	113
5.4	变量引用	113
第 6 章	转换	115
6.1	隐式转换	115
6.1.1	同一转换	115
6.1.2	隐式数字转换	115
6.1.3	隐式枚举转换	116
6.1.4	隐式引用转换	116
6.1.5	装箱转换	117
6.1.6	隐式常量表达式转换	117
6.1.7	用户自定义隐式转换	117
6.2	显式转换	117
6.2.1	显式数字转换	118
6.2.2	显式枚举转换	119
6.2.3	显式引用转换	119
6.2.4	开箱转换	120
6.2.5	用户自定义显式转换	120
6.3	标准转换	120
6.3.1	标准隐式转换	120
6.3.2	标准显式转换	121
6.4	用户自定义转换	121
6.4.1	允许的用户自定义转换	121
6.4.2	评定用户自定义转换	121
6.4.3	用户自定义隐式转换	123

6.4.4 用户自定义显式转换	123
第 7 章 表达式	125
7.1 表达式分类	125
7.2 运算符	126
7.2.1 运算符的优先级和结合性	127
7.2.2 运算符重载	128
7.2.3 一元运算符重载方案	129
7.2.4 二元运算符重载方案	129
7.2.5 用户自定义候选运算数	130
7.2.6 数字升级	130
7.3 成员查找	132
7.4 函数成员	133
7.4.1 自变量列表	135
7.4.2 重载方案	138
7.4.3 函数成员调用	138
7.5 基本表达式	139
7.5.1 文字	140
7.5.2 简单名称	140
7.5.3 括号表达式	142
7.5.4 成员访问	143
7.5.5 调用表达式	144
7.5.6 元素访问	146
7.5.7 This 访问	148
7.5.8 Base 访问	149
7.5.9 后增量和后减量运算符	149
7.5.10 运算符 new	150
7.5.11 typeof 运算符	155
7.5.12 checked 和 unchecked 运算符	156
7.6 一元表达式	159
7.6.1 一元加法运算符	159
7.6.2 一元减法运算符	160
7.6.3 逻辑非运算符	160
7.6.4 按位反运算符	161
7.6.5 前增量和前减量运算符	161

7.6.6 转换表达式	162
7.7 算术运算符	163
7.7.1 乘法运算符	163
7.7.2 除法运算符	164
7.7.3 取余运算符	166
7.7.4 加法运算符	167
7.7.5 减法运算符	168
7.8 移位运算符	170
7.9 关系运算符	171
7.9.1 整数比较运算符	172
7.9.2 浮点数比较运算符	173
7.9.3 十进制数比较运算符	174
7.9.4 布尔值比较运算符	174
7.9.5 枚举比较运算符	174
7.9.6 .ncc 类型比较运算符	175
7.9.7 字符串比较运算符	176
7.9.8 代表比较运算符	177
7.9.9 运算符 is	177
7.10 逻辑运算符	178
7.10.1 整数逻辑运算符	178
7.10.2 枚举逻辑运算符	179
7.10.3 布尔逻辑运算符	179
7.11 条件逻辑运算符	179
7.11.1 布尔条件逻辑运算符	180
7.11.2 用户自定义条件逻辑运算符	181
7.12 条件运算符	181
7.13 赋值运算符	182
7.13.1 简单赋值	183
7.13.2 复合赋值	185
7.14 表达式	186
7.15 常量表达式	186
7.16 布尔表达式	188
第 8 章 语句	189
8.1 结束点和到达性	190

8.2 块	191
8.3 空语句	192
8.4 标号语句	193
8.5 声明语句	194
8.5.1 局部变量声明	194
8.5.2 局部常量声明	195
8.6 表达式语句	196
8.7 选择语句	196
8.7.1 if 语句	196
8.7.2 switch 语句	197
8.8 循环语句	202
8.8.1 while 语句	202
8.8.2 do 语句	203
8.8.3 for 语句	203
8.8.4 foreach 语句	205
8.9 跳转语句	206
8.9.1 break 语句	207
8.9.2 continue 语句	208
8.9.3 goto 语句	208
8.9.4 return 语句	209
8.9.5 throw 语句	210
8.10 try 语句	211
8.11 checked 和 unchecked 语句	214
8.12 lock 语句	214
8.13 using 语句	215
第 9 章 名称空间	219
9.1 编译单元	219
9.2 名称空间声明	220
9.3 Using 指示符	221
9.3.1 Using 别名指示符	222
9.3.2 Using 名称空间指示符	225
9.4 名称空间成员	227
9.5 类型声明	228
第 10 章 类	229

10.1	类声明	229
10.1.1	类修饰符	229
10.1.2	类基指定	231
10.1.3	类体	231
10.2	类成员	231
10.2.1	继承性	233
10.2.2	修饰符 new	233
10.2.3	访问修饰符	234
10.2.4	要素类型	234
10.2.5	静态成员和实例成员	234
10.3	常量	235
10.4	字段	238
10.4.1	静态字段和实例字段	239
10.4.2	只读字段	240
10.4.3	字段初始化	241
10.4.4	变量初始化符	242
10.5	方法	244
10.5.1	方法的参量	245
10.5.2	静态方法和实例方法	252
10.5.3	虚方法	252
10.5.4	重载方法	255
10.5.5	抽象方法	257
10.5.6	外部方法	259
10.5.7	方法体	260
10.5.8	方法重载	261
10.6	特性	261
10.6.1	静态特性	262
10.6.2	虚特性	262
10.6.3	重载特性	263
10.6.4	抽象特性	264
10.6.5	访问函数	265
10.7	事件	271
10.8	索引函数	274
10.8.1	索引函数重载	278
10.8.2	虚索引函数	278

10.8.3 重载索引函数	278
10.8.4 抽象索引函数	279
10.9 运算符	280
10.9.1 一元运算符	281
10.9.2 二元运算符	281
10.9.3 转换运算符	282
10.10 实例构造函数	283
10.10.1 构造函数初始化符	284
10.10.2 实例变量初始化符	285
10.10.3 默认构造函数	288
10.10.4 私有构造函数	289
10.10.5 可选的构造函数参量	289
10.11 析构函数	290
10.12 静态构造函数	290
第 11 章 结构体	295
11.1 结构体声明	295
11.1.1 结构体修饰符	295
11.1.2 接口	295
11.1.3 结构体主体	295
11.2 结构体成员	296
11.3 结构体范例	296
11.3.1 数据库整数类型	296
11.3.2 数据库布尔类型	299
第 12 章 数组	303
12.1 数组类型	303
12.2 数组创建	304
12.3 数组元素访问	305
12.4 数组成员	305
12.5 数组协方差	305
12.6 数组初始化符	306
第 13 章 接口	309
13.1 接口声明	309
13.1.1 接口修饰符	309

13.1.2	基本接口	310
13.1.3	接口体	311
13.2	接口成员	311
13.2.1	接口方法	312
13.2.2	接口特性	312
13.2.3	接口事件	313
13.2.4	接口索引函数	313
13.2.5	接口成员访问	314
13.3	接口成员的全限定名	316
13.4	接口实现	317
13.4.1	显式接口成员实现	318
13.4.2	接口映射	320
13.4.3	接口实现继承	324
13.4.4	接口重实现	326
13.4.5	抽象类和抽象接口	328
第 14 章	枚举	329
14.1	枚举声明	329
14.2	枚举成员	330
14.3	枚举值和操作	333
第 15 章	代表	335
15.1	代表声明	335
15.2	代表实例化	336
15.3	多重转换代表	337
15.4	代表调用	337
第 16 章	异常	339
16.1	发生异常的原因	339
16.2	System.Exception 类	339
16.3	如何处理异常	340
16.4	常用的异常类	340
第 17 章	属性	343
17.1	属性类	343
17.1.1	AttributeUsage 属性	343

17.1.2 位置参量和命名参量	344
17.1.3 属性参量类型	345
17.2 属性指定	346
17.3 属性实例	348
17.3.1 属性的编译	349
17.3.2 属性实例的运行期回溯	349
17.4 保留属性	350
17.4.1 AttributeUsage 属性	350
17.4.2 Conditional 属性	351
17.4.3 Obsolete 属性	354
第 18 章 不安全代码	355
18.1 指针类型	355
18.2 指针操作	355
18.2.1 指针成员访问	356
18.2.2 指针元素访问	356
18.2.3 sizeof 运算符	356
18.2.4 stackalloc 运算符	356
18.2.5 间接指针	356
18.2.6 取址运算符	356
18.3 固定语句	356
第 19 章 互用性	359
19.1 COMImport 属性	359
19.2 COMRegisterFunction 属性	360
19.3 COMSourceInterfaces 属性	360
19.4 COMVisible 属性	361
19.5 DispId 属性	361
19.6 DllImport 属性	361
19.7 FieldOffset 属性	363
19.8 GlobalObject 属性	363
19.9 Guid 属性	364
19.10 HasDefaultInterface 属性	364
19.11 ImportedFromTypeLib 属性	365
19.12 In 和 Out 属性	365