

精通.NET编程系列

精通C#编程

黎晓冬 李华飏 编著

 科学出版社
www.sciencep.com

精通.net 编程系列

精通 C#编程

黎晓冬 李华飏 编著

科学出版社

北 京

内 容 简 介

C#语言是一种现代、简单、完全面向对象的编程语言,同时,也是编写企业应用程序的一种功能强大的编程工具。本书详细介绍了C#的语法、数据类型、关键字、操作符、语言特性以及面向对象的编程,包括C#的类、方法、事件和索引等。另外,本书还介绍了C#预处理程序指令和XML相关的知识,书中对每一部分都有详细的示例代码,帮助理解和应用各个知识点。

本书结构清晰,层次分明,示例详尽,可作为广大C#语言编程爱好者的助手,也可以作为C#语言编程的参考之用。

图书在版编目(CIP)数据

精通C#编程/黎晓冬,李华飏编著. —北京:科学出版社,2003

(精通.net编程系列)

ISBN 7-03-010899-X

I.精... II.①黎...②李... III.C语言—程序设计 IV.TP312

中国版本图书馆CIP数据核字(2002)第080469号

责任编辑:陈红英 孟战龙 / 责任校对:赵慧玲

责任印制:吕春珉 / 封面设计:郭建

科学出版社 出版

北京东黄城根北街16号

邮政编码:100717

<http://www.sciencep.com>

新蕾印刷厂 印刷

科学出版社发行 各地新华书店经销

*

2003年1月第一版 开本:720×1000 1/16

2003年1月第一次印刷 印张:30

印数:1—4 000 字数:604 000

定价:48.00元(含光盘)

(如有印装质量问题,我社负责调换〈环伟〉)

前 言

Visual Studio.NET 是微软实现 .NET 技术而开发的一套组件，通过提供终端到终端的网络开发能力和可伸缩、可重用的服务器端组件功能，将开发功能强大、可靠性高的企业网络解决方案简化，提高了效率。C#语言作为用于该平台的主要编程语言，保持了 C/C++语言中特有的强大功能和控制能力，同时，又提供了诸如与 COM+服务的完全的互换性、自动垃圾回收功能、对基于网络的组件提供 XML 支持等新特性，该语言将成为未来 C/C++程序员的首选编程工具。

本书全面系统地介绍 C#语言的基础知识、语言特性及语言的应用，使用了丰富的示例，深入浅出地讲解了 C#语言的各个知识点，从基础语法的数据类型、操作符、关键字到面向对象的高级编程，层层深入。通过阅读本书，你可以全面了解 C#语言的各种特性并掌握和熟练应用 C#语言进行开发工作。

全书共分 15 章：

第 1 章 C#语言概述：讲述了 C#语言的发展历程，介绍了 C#语言的新特性，并与 C/C++和 Java 语言进行比较，描述了它们的差异性，使读者对 C#语言有更深入的了解。最后，还介绍了 C#语言使用平台的搭建。

第 2 章 数据类型：介绍了 C#的数据类型、常量及变量的使用。

第 3 章 C#操作符：讲述了 C#语言中各种操作符的使用。

第 4 章 数组：介绍 C#语言中数组的使用。

第 5 章 流程控制语句：讲解了选择语句、循环语句、检查和非检查语句及锁定语句的应用。

第 6 章 C#关键字：介绍 C#中关键字的使用。

第 7 章 拆箱和装箱：拆箱和装箱是 C#语言类型系统中的一个核心概念，本章重点介绍拆箱和装箱的原理及应用。

第 8 章 面向对象编程：介绍 C#语言中主要的特性，面向对象的特性。

第 9 章 类：介绍了面向对象编程中类的相关知识及类在面向对象编程中的应用。

第 10 章 异常处理：异常是程序执行过程中产生的意外，C#语言提供了对异常情况处理的支持，称为异常处理，本章介绍如何应用异常处理应付程序中出现的异常。

第 11 章 C#预处理程序指令：介绍 C#语言包含的程序预处理指令，该指令让编译器具备了预处理功能。

第 12 章 事件和索引：介绍面向对象编程中的事件和索引的应用。

第 13 章 C#特性：特性是 C#语言提供了一种定义声明标志的机制，本章介绍 C#特性的语法及其应用。

第 14 章 XML 文档：用户可以在 C#语言中使用 XML 文档编写的代码，这是 Visual Studio.NET 中惟一具有这种特性的编程语言，本章重点是如何在 C#语言中使用 XML 文档编写的代码。

第 15 章 客票查询系统：本章介绍了一个网上客票查询系统的制作过程。

本书主要面向初级和中级的读者，引领读者走入 C#的世界，体验利用 C#编程的乐趣。同时本书对高级用户也有参考意义，可以在其中查找相关的信息。

书中还配有光盘，包含书里应用到的示例的代码。

本书由逸枫工作室黎晓冬、李华飏等人编写，同时，参与编写的人员还有王福水、王涛、文创、鲁峥、李方、杨松、吴永朝、刘勇、王兵、周同、李易、蒋兵、吴爽等。在此表示感谢。由于时间仓促，书中难免有不足之处，敬请专家、广大读者指正。

编 者

2002 年 8 月

目 录

第 1 章 C#语言概述	1
1.1 C#特性概述	2
1.2 C#与 C++的比较	4
1.3 C#与 Java 的比较	5
1.3.1 C#与 Java 的共同点	5
1.3.2 C#与 Java 的区别	6
1.4 Visual Studio.NET 的安装	7
1.4.1 软硬件环境需求	7
1.4.2 安装	7
第 2 章 C#数据类型	11
2.1 数据类型	11
2.1.1 值类型及示例分析	11
2.1.2 引用类型	38
2.1.3 类型引用表	55
2.2 常量和变量	59
2.2.1 变量	60
2.2.2 常量	62
第 3 章 C# 操作符	64
3.1 算术溢出	64
3.2 [] 操作符	65
3.3 () 操作符	67
3.4 . 操作符	69
3.5 + 操作符	71
3.6 - 操作符	72
3.7 * 操作符	73
3.8 / 操作符	75
3.9 % 操作符	77
3.10 & 操作符	78
3.11 操作符	79
3.12 ^ 操作符	80

3.13	! 操作符.....	81
3.14	~ 操作符.....	82
3.15	= 操作符.....	84
3.16	< 操作符.....	85
3.17	> 操作符.....	86
3.18	?: 操作符.....	87
3.19	++ 操作符.....	89
3.20	-- 操作符.....	91
3.21	&& 操作符.....	92
3.22	操作符.....	94
3.23	<< 操作符.....	95
3.24	>> 操作符.....	96
3.25	== 操作符.....	98
3.26	!= 操作符.....	100
3.27	<= 操作符.....	101
3.28	>= 操作符.....	102
3.29	+= 操作符.....	104
3.30	-= 操作符.....	105
3.31	*= 操作符.....	106
3.32	/= 操作符.....	107
3.33	%= 操作符.....	108
3.34	&= 操作符.....	110
3.35	= 操作符.....	111
3.36	^= 操作符.....	112
3.37	<<= 操作符.....	114
3.38	>>= 操作符.....	115
3.39	-> 操作符.....	116
3.40	操作符的重载.....	118
第4章	数组.....	122
4.1	什么是数组.....	122
4.2	数组的声明和初始化.....	122
4.3	访问数组成员.....	124
4.4	一维数组和多维数组.....	126
4.4.1	一维数组.....	126

4.4.2 多维数组	127
4.5 数组的遍历	128
4.6 数组查询	131
第 5 章 流程控制语句	133
5.1 选择语句	133
5.1.1 if-else 语句	133
5.1.2 switch-case 语句	141
5.2 循环语句	146
5.2.1 do-while 语句	147
5.2.2 for 语句	149
5.2.3 foreach-in 语句	151
5.2.4 while 语句	157
5.3 跳转语句	158
5.3.1 break 语句	158
5.3.2 continue 语句	161
5.3.3 goto 语句	162
5.3.4 return 语句	166
5.4 检查和非检查	168
5.4.1 checked 语句	168
5.4.2 unchecked 语句	170
5.5 非保护和固定	173
5.5.1 unsafe 语句	173
5.5.2 fixed 语句	175
5.6 锁定语句	177
第 6 章 C#关键字	183
6.1 限定符	183
6.1.1 访问限定符	183
6.1.2 internal 限定符	188
6.1.3 private 限定符	192
6.1.4 protected 限定符	195
6.1.5 public 限定符	197
6.1.6 abstract 限定符	198
6.1.7 const 限定符	205
6.1.8 event 限定符	208

6.1.9	extern 限定符	212
6.1.10	override 修饰符	215
6.1.11	readonly 限定符	220
6.1.12	sealed 限定符	224
6.1.13	static 限定符	227
6.1.14	virtual 限定符	232
6.2	名称空间	235
6.2.1	namespace 关键字	235
6.2.2	using 关键字	237
6.3	操作符关键字	240
6.3.1	as 关键字	241
6.3.2	is 关键字	242
6.3.3	new 关键字	246
6.3.4	sizeof 关键字	253
6.3.5	typeof 关键字	255
6.3.6	true 关键字	256
6.3.7	false 关键字	258
6.3.8	stackalloc 关键字	261
6.4	转换关键字	263
6.4.1	explicit 关键字	263
6.4.2	implicit 关键字	265
6.4.3	operator 关键字	267
第 7 章	装箱与拆箱	271
7.1	装箱	271
7.2	拆箱	273
第 8 章	面向对象的编程	276
8.1	对象	276
8.2	属性	277
8.2.1	属性的声明	277
8.2.2	访问器	281
8.2.3	接口属性	288
8.2.4	静态属性	290
8.2.5	覆盖属性	292
8.2.6	虚拟属性	295

8.2.7 抽象属性	295
8.3 域	297
8.3.1 域的声明	297
8.3.2 域的初始化	300
8.3.3 静态域	301
8.3.4 非静态域	302
8.3.5 只读域	305
8.4 方法	309
8.4.1 方法的声明	309
8.4.2 方法参数	313
8.4.3 静态和非静态方法	319
8.4.4 方法的返回值	322
8.4.5 方法的重载	323
8.4.6 Main 方法	325
8.4.7 构造函数和析构函数	327
8.4.8 this 关键字	331
8.5 继承	335
8.5.1 继承概述	336
8.5.2 方法的覆盖	339
8.5.3 base 关键字	340
8.5.4 多态性	343
第 9 章 类	345
9.1 类的声明	345
9.2 类的成员	347
9.2.1 对类的成员的访问	348
9.2.2 静态成员和非静态成员	352
9.3 抽象类	353
9.4 密封类	356
第 10 章 异常处理	358
10.1 throw 语句	358
10.2 try-catch 语句	360
10.3 try-finally 语句	364
10.4 try-catch-finally 语句	365
10.5 try 语句执行情况分析	366

第 11 章 C#预处理程序指令	368
11.1 #if.....	368
11.2 #else.....	370
11.3 #elif.....	372
11.4 #endif.....	373
11.5 #define	373
11.6 #undef.....	375
11.7 #warning	376
11.8 #error	378
11.9 #line	379
11.10 #region.....	380
11.11 #endregion	382
第 12 章 事件和索引.....	383
12.1 事件.....	383
12.1.1 事件的声明	383
12.1.2 静态事件	385
12.2 索引.....	385
12.2.1 索引的声明	385
12.2.2 属性和索引的比较	388
12.2.3 接口索引	388
第 13 章 C#特性	392
13.1 特性用法	392
13.2 全局特性	392
13.3 创建自定义特性	393
13.4 检索特性信息	393
13.5 AttributeUsage 特性.....	396
13.6 Conditional 特性	398
13.7 Obsolete 特性.....	400
第 14 章 XML 文档	403
14.1 文档注释标记	403
14.1.1 <c>标记	403
14.1.2 <code>标记	405
14.1.3 <example>标记	407
14.1.4 <exception>标记	409

14.1.5	<list>标记	410
14.1.6	<para>标记	413
14.1.7	<param>标记	415
14.1.8	<paramref>标记	416
14.1.9	<permission>标记	418
14.1.10	<remarks>标记	419
14.1.11	<returns>标记	421
14.1.12	<see>标记	422
14.1.13	<seealso>标记	424
14.1.14	<summary>标记	426
14.1.15	<value>标记	428
14.2	处理 XML 文档	429
14.3	XML 应用示例	431
第 15 章	客票查询系统	437
15.1	设计综述	437
15.1.1	信息查询	437
15.1.2	预定车票	437
15.2	设计步骤	438
15.3	查询系统功能介绍	448
附录	误码	454

第 1 章 C#语言概述

在过去的 20 年里，C 和 C++ 已经成为广泛应用在商用软件开发的开发语言。但是 C 和 C++ 都提供了一些容易使开发者产生错误的特性，或者说 C 和 C++ 的灵活性是以牺牲开发效率为代价的。如果将它们与其他的开发语言相比（比如说 VB），相同功能的 C 和 C++ 软件通常会需要更长的开发周期。正是由于 C 和 C++ 开发的复杂性和较长的开发周期，许多 C 和 C++ 开发人员都在寻找一种可以在功能和开发效率间提供更多平衡的开发语言。

目前有一些开发语言通过牺牲 C 和 C++ 语言的灵活性（一些必要的灵活性）来换取开发效率。还有些语言对开发人员产生了过多的限制（比如，限制使用底层控制代码），并且只提供更少的通用命名能力。因此，这些语言不能轻易地与现有的系统相结合，并且不能与当前的 Web 开发相结合。

一种合理的 C 和 C++ 替代语言应该能够对现有和潜在平台上的高效开发提供有效和有力的支持，并可以使 Web 开发方便地与现有的应用开发相结合。而且 C 和 C++ 开发人员都倾向于在必要的时候使用底层代码。C#（发音为 CSharp）语言就是在这种技术发展需求下应运而生了。C# 语言是一种先进的、面向对象的语言，通过 C# 可以使开发人员快速地建立大范围的基于 MS 网络平台的应用，并且提供大量的开发工具和服务，帮助开发人员开发基于计算和通信的各种应用。

由于 C# 是一种面向对象的开发语言，所以 C# 可以大范围的适用于高层商业应用和底层系统的开发。即使是通过简单的 C# 结构也可使各种组件方便的转变为基于 Web 的应用，并且能够通过 Internet 被各种系统或是其他语言开发的应用程序所调用。

我们抛开上面所提到的优点，C# 还具有为 C 和 C++ 开发人员提供快速的开发手段而不需要牺牲 C 和 C++ 语言的特点。从继承角度来看，C# 在更高层次上重现了 C 和 C++，熟悉 C 和 C++ 开发的人员可以很快地转变为 C# 开发人员。

目前各种基于 Web 应用的软件开发向传统的商业应用软件开发提出了挑战，开发者被组织起来开发具有更短开发周期的各种应用程序，并且需要能够提供更好的可修正性，而不是建立一个可以长久使用的软件系统。C# 的设计正是充分考虑了这些因素。C# 会帮助开发者通过更少的代码完成相同的功能，并且能够更好地避免错误发生。

在编程语言中真正的霸主多年来一直是 C++，所有的操作系统和绝大多数的

商品软件都是用 C++ 作为主要开发语言的。Java 的程序员绝大多数也是 C++ 的爱好者，PHP 的成功里面也有类似 C++ 语法的功劳。在操作系统、设备驱动程序、视频游戏等领域，C++ 在很长的时间内仍将占据主要地位，而在数量最大的应用软件的开发上，C# 很可能取代 C++ 的位置。首先，C# 和 Java 一样，简直就是照搬了 C++ 的部分语法，因此，对于数量众多的 C++ 程序员学习起来很容易上手，另外，对于新手来说，C# 比 C++ 要简单一些。其次，Windows 是目前占垄断地位的平台，所以，开发 Windows 应用程序，微软的声音是不能忽略的。最重要的是，相对于 C++，用 C# 开发应用软件可以大大缩短开发周期，同时可以利用原来除用户界面代码之外的 C++ 代码。所以，C# 语言将成为新世纪最流行的编程语言之一。

1.1 C# 特性概述

C# 语言自 C 和 C++ 演变而来，它是一门现代、简单、完全面向对象和类型安全的编程语言。有 C 和 C++ 语言基础的程序员能够轻松地掌握和精通这门语言。因为许多 C# 语句直接借用 C 和 C++ 语言的语法，包括表达式和操作符，都稳固地植入在 C# 语言中。但是在一些诸如名称空间、类、方法和异常处理等特定领域，它们之间还存在着巨大的差异。

C# 语言让软件开发人员更容易发展以全球信息网为导向的服务。C# 结合了 Microsoft 的 C++ 程序开发的威力及 Visual Basic 的简易性，同时，C# 为用户提供了垃圾收集、类型安全、版本控制等方便的功能，而其“代价”只是代码操作默认为类型安全，并且通常不允许使用指针。但是，在非保护环境外仍可以使用指针。

C# 语言中最重要的一点：它是现代的编程语言。它简化和提升了 C++ 在类、名称空间、方法重载和异常处理等方面的应用，舍弃了 C++ 的复杂性，使它更易用、更少出错。在 C# 语言中，使编程更方便的新功能是严格的类型安全、版本控制、垃圾收集等。所有这些功能的目标都是开发面向组件的软件。

C# 语言还是一种类似 Java 的软件编程语言，其设计目的是简化使用微软软件的网络服务构建。C# 语言面向对象的设计，可以用来构建高水平的商务目标以及体系标准应用程序等广泛的组件。使用 C# 编程语言，程序员能够迅速地建造提供开拓计算和通信工具和服务的新 Microsoft.NET 平台。C# 语言的组成部分使用简单的 C# 语言结构，该结构能被转换成 Web 服务，并允许在任何操作系统上运行的任何语言通过 Internet 调用。

C# 作为一种编程语言，它具有以下几个重要的特点：

- 快速应用开发功能

C#支持快速开发（Rapid Application Development, RAD），这也是目前开发语言最为重要的一大功能，也正是C和C++的致命伤。网络时代应用系统的开发必须按照网络时代的速度来进行，支持快速开发可以使得开发人员的开发效率倍增，从而使得他们可以从繁重的重复性劳动中解放出来。

C#的RAD功能主要表现在垃圾收集、代表等众多特性上。垃圾收集机制将减轻开发人员对内存的管理负担，而代表功能更是可以让开发者不经过内部类就可以调用函数。利用C#的这些功能，可以使开发者通过较少的代码来实现功能更强大的应用程序，并且能够更好地避免错误发生，从而缩短了应用系统的开发周期。

- 语言的自由性

用C#编写的程序能最大程度地与任何支持.NET的语言互相交换信息，这样的好处是节省了大量的工作，不必把COBOL等语言强行改成另一种语言，而使用C#语言就可以与它沟通，所以.NET让各种语言真正地互相交流了。

C#和其他.NET语言有着最好的协作，这对开发人员非常重要。全球从事编写软件的人当中，大约有50%的人以Visual Basic作为基本的编程工具，在跨入.NET编程时代的时候，这些人轻松地使用了.NET的VB开发Web上的应用程序。

- 强大的Web服务端组件

在C#的Web编程中，最让人兴奋的是Web服务的组件，它们不仅包括传统组件，还包括那些能够自动连接五花八门服务的可编程组件，这样可以用C#编写自己的服务器端组件，服务器端组件和标准的HTML提供的服务自然是不一样的，使用它们可以更自由、更容易地进行数据绑定。不久，那些视频服务器组件也很快会出现，有了这些强大的组件，就可以设计出功能更加强劲的企业级分布式应用系统。

- 支持跨平台

随着互联网应用程序的应用越来越广，人们逐渐意识到由于网络系统错综复杂，使用的硬件设备和软件系统各不相同，开发人员所设计的应用程序必须具有强大的跨平台性。而C#编写的应用程序就具有强大的跨平台性，这种跨平台性也包括了C#程序的客户端可以运行在不同类型的客户端上，比如PDA、手机等非PC设备上。

- 与XML的融合

由于XML技术真正融入到了.NET和C#之中，C#的编程变成了真正意义上的网络编程，甚至可以说.NET和C#是专为XML而设计，使用C#的程序员可以轻

松用 C# 内含的类来使用 XML 技术。就这方面而言, 目前 C# 提供给了程序员更多的自由和更好的性能来使用 XML。

- 对 C++ 的继承

C# 继承并保留了 C++ 强大的功能, 例如, C# 保留了类型安全的检测和重载功能, C# 还提供了一些新功能取代了一些原来的 ANSI C 预处理程序的功能, 提高了语言的类型安全等安全性。

1.2 C# 与 C++ 的比较

C# 与 C++ 有很多类似的风格, 如变量声明、参数传递和操作符等, 这种风格的类似使 C++ 程序员能很容易地使用 C# 进行编程, 但作为一个新版本的 C# 本身又带有许多新特性, 如简洁、现代、面向对象、类型安全等, 所以, 从 C++ 升级到 C# 过程中有更多的变化和区别, 下表列出了 C++ 和 C# 之间重要特性的比较。

表 1.1 C++ 与 C# 特性比较

特性	描述	相关主题
全局变量定义	C# 中不能在所有类之外定义全局变量, 只能通过通过类中定义 Public 和 Static 变量实现全局变量的定义	变量
数组	声明 C# 的数组与 C++ 中数组声明不一样, “[]” 在 C# 数组类型之后出现	数组
指针	C++ 中指针使用是灵活的, 但也是最容易产生错误的, 所以, 在 C# 中程序员不能直接使用指针操作 C# 中只能在非安全模式使用指针	无 unsafe
bool 类型	在 Bool 类型和其他类型间不能转换	布尔值
结构类型	类和结构在 C# 中是不同的, 结构是值的类型, 而类是引用类型	结构类
内存管理	C++ 中通过运算符 new 和 delete 分配和释放内存, 在 C# 只通过运算符 new 为对象分配内存, 同时, 也利用 new 对对象进行操作	运算符
Switch 语句	C++ 中该语句支持从一种标记转换为另一种, C# 中不支持	转换
delegate 类型	类似于 C++ 中指针的功能, 但是, delegate 类型是安全的	代表
数据类型可移植性	对于不同平台, C++ 中分配给 int、float 等数据类型的字节数不同, 导致代码的不可移植性, 而 C# 中给数据类型分配固定长度的字节数, 这就保证了平台的无关性	数据类型
新增的数据类型	C# 比 C++ 新增几种数据类型。sbyte (带符号 8 位整数), ushort (无符号 16 位整数), uint (无符号 32 位整数), ulong (无符号 64 位整数)	数据类型

续表

特性	描述	相关主题
数据类型转换	C++中可以对数据进行任意转换而由此引发不安全性, C#中, 系统对对象进行处理时进行类型兼容性检查	数据类型
头文件	C++中用头文件声明全局变量、库函数等, 在 C#中不支持头文件的使用	头文件
宏命令	C++中支持宏命令的使用, C#中不支持宏命令的使用, 只通过关键字 <code>final</code> 声明常量	
Base class	C#中从派生类中调用被覆盖的基本类	基本类
New	C#用 <code>new</code> 编辑器确定隐藏继承成员	<code>new</code>
override 关键字	C#用 <code>override</code> 关键字声明覆盖方法	<code>override</code> 关键字
异常处理	C#用 <code>finally</code> 语句进行异常处理	<code>try-finally</code> <code>try-catch-finally</code>
C#操作符	C#支持操作符	<code>&</code> 、 <code> </code> 和 <code>^</code> 操作符 <code>is</code> 、 <code>typeof</code>
extern 关键字	C#中使用 <code>extern</code> 关键字	<code>extern</code> 关键字
static 关键字	C#中使用 <code>static</code> 关键字	<code>static</code> 关键字
virtual 关键字	C#中类似于创建基本类的一种可选的 C++初始化列表方法	<code>virtual</code> 关键字
C#程序一般结构	名称空间、类、结构、代表和计数	C#程序的结构
Main 方法	C#中 <code>Main</code> 方法的声明不同于 C++中 <code>main</code> 函数	<code>Main</code> 方法
Method 参数	C#中支持 <code>ref</code> 和 <code>out</code> 参数, 它们通过引用在传递参数过程中代替指针	<code>Ref</code> <code>Out</code>
重载操作符	C#中超载操作符的执行与 C++中不同	C#操作符
字符串	C#中的字符串与 C++中不同	<code>string</code>

1.3 C#与 Java 的比较

很多人说 C#是微软用来和 Java 抗衡的武器, 因为二者在很大程度上有着惊人的相似, 尽管如此, 两者不同的地方也很多, 所谓“于细微处见差异”。

1.3.1 C#与Java的共同点

对于程序员来说, 了解一下二者的相同之处也是很有必要的。下面简单地列出 C#和 Java 的相似之处:

- 二者都编译成跨平台、跨语言的代码, 并且代码只能在一个受控制的环境中运行。