万水计算机实用编程技术系列

C#实用编程技术

钱昆　等编著

中国水利水电出版社
www.waterpub.com.cn

万水计算机实用编程技术系列

C#实用编程技术

钱　昆　等编著

中国水利水电出版社

内 容 提 要

本书比较全面地介绍了 C#程序设计的基本要素，从最为基础的数据类型，到面向对象的程序设计，到线程和例外处理，本书在介绍这些内容的同时，还编写了大量的例子来对关键的技术要点加以演示。在介绍 C#的程序设计的基础上，本书还介绍了基于 Windows 的应用程序的界面构造及其功能实现，以及 C#在网络和数据库领域内的应用。

本书适合那些对新技术有强烈兴趣的读者朋友。对于那些对 C/C++有一定了解的读者，阅读本书将更为容易。本书非常适合大中专在校学生，各级各类的计算机培训班的学员学习 C#语言，以掌握最新的程序设计技术。

图书在版编目（CIP）数据

C#实用编程技术/钱昆等编著. —北京：中国水利水电出版社，2001.8
（万水计算机实用编程技术系列）
ISBN 7-5084-0769-5

I. C… II. 钱… III. C 语言－程序设计 IV. TP312
中国版本图书馆 CIP 数据核字（2001）第 055126 号

书　　名	C#实用编程技术
作　　者	钱昆　等编著
出版、发行	中国水利水电出版社（北京市三里河路 6 号　100044） 网址：www.waterpub.com.cn E-mail：mchannel@public3.bta.net.cn（万水） sale@waterpub.com.cn 电话：（010）68359286（万水）、63202266（总机）、68331835（发行部）
经　　售	全国各地新华书店
排　　版	中科辅龙
印　　刷	北京市天竺颖华印刷厂
规　　格	787×1092 毫米　16 开本　24.75 印张　539 千字
版　　次	2001 年 9 月第一版　2001 年 9 月北京第一次印刷
印　　数	0001—5000 册
定　　价	38.00 元

前　言

Microsoft .Net 的推出再一次证明了微软公司在软件世界的霸主地位。.Net 曾经一度被微软在内部称作“下一代视窗服务”（NGWS），最终确定为.Net 名称不仅仅是微软惯用市场化手段为了朗朗上口和便于用户的识别，而且是把这个包含创新性的概念转变成为一个集全新商业模型、全新技术模型于一体的统一框架。

.Net 的雄伟构想确实让人心潮澎湃，但是要实现这一目标，一个必要的条件就是能够有一款有效的开发工具，它在整合现有资源的同时能够很好地适应未来发展的需要。纵观现有的开发工具，无论是 Microsoft 的 Visual C++，还是 Sun 公司的 Java，还是 Inprise 公司的 Delphi，都难以担此重任；历史选择了 C#——一种由 Microsoft 公司开发的完全面向对象的现代高速开发语言。开发一种全新的程序设计语言很早就被计算机业界提上了议事日程并取得了很大的进步。早在 5 年前，作为 Microsoft 的竞争对手的 Sun 公司就推出了其 Java 语言，在业界获得了一致的好评并迅速在世界范围内流行开来。

尽管我们称 C#为一种新的程序设计语言，但是它也吸收了很多已有语言的优点，尤其是它继承了 C++的大量语法要素和编程模式。一名熟练的 C++程序员可以很容易地使用 C#进行程序设计。但是 C#在 C/C++的基础上也作了比较大的革新，这些革新措施使得 C#更为简洁、更为安全，由于采用了完全的面向对象的编程思想，使得 C#的代码的复用性更高。同时 C#通过和 Microsoft 同期推出的 CLI（Common Language Infrastructure，公共语言基础平台）的整合，能够实现最大程度上的语言之间的互操作，解决了多年以来困扰软件产业的一大障碍。最后，C#还是一个开放的体系，它可以适应从桌面应用到网页浏览，从数据库到多媒体等多种领域。

本书比较全面地介绍了 C#程序设计的基本要素，从最为基础的数据类型，到面向对象的程序设计，到线程和例外处理，本书在介绍这些内容的时候，还编写了大量的例子来对关键的技术要点加以演示。在介绍 C#的程序设计的基础上，本书还介绍基于 Windows 的应用程序的界面构造及其功能实现，以及 C#在网络和数据库领域内的应用。

本书适合那些对新技术有强烈兴趣的读者朋友。对于那些对 C/C++有一定了解的读者，阅读本书将更为容易。本书非常适合大中专在校学生，各级各类的计算机培训班的学员学习 C#语言，以掌握最新的程序设计技术。

参加本书编写的有钱昆、王宏、靳莹、李好、邹杰、王艳燕、郭美山、石利文、李炎、阎高峰、刘小华、李琪、刘晓刚、徐平、张晨、苏扬等。

由于时间仓促、作者能力有限，错误和遗漏在所难免，希望广大读者朋友们批评指正。

编者

2001 年 6 月于北京

目　　录

第 1 章　C#概述

C#准确地讲应该是 C#.Net，它是 Microsoft 推出的.Net 开发平台的一部分，本书为了简化起见，后续内容都以 C#代替 C#.Net。为了让广大读者能够对 C#有一个全面的认识，这里先就.Net 的基本情况作一个简要的介绍，然后再对 C#的特点以及它和 C++之间的异同作一个简要的介绍。

1.1　.Net 概述

家喻户晓的微软公司，是一个让无数竞争对手心惊胆战的软件帝国。Microsoft .Net 的推出再一次证明了微软公司在软件世界的霸主地位。在互联网商业狂热的几年中，微软一方面巩固自己的阵地，另一方面也潜心学习和研究未来网络的特征、计算模式和商业模型。微软在 1999 年重新定义了公司的远景，“用先进的软件让人们随时随地通过任何设备获得强大能量”。这个新的表述不但为微软重新设计了一个能够展望十几年乃至几十年的远景，更明确了微软将开始一个新的发展阶段。在 1998 年至 2000 年初的一段时间里，微软公司一直蓄势待发，精心部署研发力量和管理团队，准备在新世纪进行全面转折。果然，2000 年 6 月 22 日，微软公司正式对外宣布其.Net（音 DOT NET）战略，并确定每年为这个新的战略直接投入 40 亿美元的研发费用。.Net 这个曾经模糊不定的庞大体系迅速成为业界瞩目的焦点。事实上，此次宣布在微软内部被称为第二次重要转折，是从 MS-DOS 向视窗转化后的一次全公司上下策略的大变换。

1.1.1　.Net 的技术体系

.Net 曾经一度被微软在内部称作“下一代视窗服务”（NGWS），最终确定为.Net 名称不仅仅是微软惯用市场化手段为了朗朗上口和便于用户的识别，而且是把这个包含创新性的概念转变成为一个集全新商业模型、全新技术模型于一体的统一框架。.Net 的核心概念就是“把软件当作服务”，也就是把软件应用产品与商业、内容、信息服务合并成一种事物，使之成为可以在网络上订阅使用的服务形式。人们设计、构造、实施、运作、集成和使用软件的方式都将透过网络完成，所以也就要按照使用这些服务的不同方式支付相应的费用。

.Net 彻底地把计算模式从单机、客户机/服务器和 Web 网站的方式转向分布式计算（Distributed Computing）。我们知道，Corba 和 COM 是今天比较流行的部件对象计算模型。但是它们都存在着“局部计算”的局限性。也就是说，这些模型都仅仅是本地计算或本网计算的模式，而不能把整个互联网当作是一个计算资源体系来加以利用。.Net 则通过一种称作“网

络服务”（Web Service，这是.Net 的核心概念）的技术把分布在互联网上的各种资源有效地通过编程手段整合在特定的应用界面中。

除了 Web Service 这种新型计算模式的出现，微软还在.Net 中增加了自己已有的研究成果，包括：自然语言的处理和识别等。这些技术能够让用户更加灵活地操纵各种计算设备，而不必依赖传统的鼠标、键盘。自然语言的处理还能够提高计算机自动处理各种 XML 信息的智能性。总之，.Net 包含了新一代的计算模式，即跨越全球的分布式计算。这种规则的制定者将有可能从与之配合的商业模型中大获其利。

.Net 还为微软带来了一种全新的商业模式。微软预见了“服务”是数字经济的核心商业模式，因此它将逐步转换今天依靠销售盒装软件的获利形式，利用.Net 把“软件作为服务”（Software-As-Service）。微软已经先行一步，把各种软件改造成为能够通过“订阅”方式使用的服务，就像我们今天使用电话业务、水电和订阅杂志的方式一样，按照“使用量”付费。为此，微软还在.Net 中构造了一系列辅助模块。这些模块包括用户认证、通知、网络存储等功能。一旦微软.Net 的各种基础构造模块开发完毕，微软将最先开始拿自己的产品做“服务”尝试，不远的将来，软件用户将可以体验到在网络上注册使用软件并根据使用时间、存储要求或使用的功能模块多少来付费的方式。此外，微软还会把各种商业服务、内容服务与软件服务并列考虑，这样用户在未来将不会感到“软件”与其他信息产品有任何不同，因为人们真正需要的是快速地完成自己的工作目标，而不是考虑“软件”等产品形式。

1.1.2 .Net 的深远影响

.Net 反映了微软的高层人士对数字经济的深刻把握，也体现了微软的整体决策能力。正因为.Net 包含的一系列商业思想、技术思想和经营哲学反映了未来的趋势，所以不管最终是否由微软公司来承担实现未来目标的重担，这个过程所带来的变化都将对社会经济、商业和个人带来深远的影响。

首先，人们将在未来几年能够借助无线、宽带的方式，随意使用各种设备访问网络服务。其次，人们获得的服务将能够相互集成，智能地为人们提供个性化的支持。再次，社会的效率和准确度会大大提高。更多的商业活动可以在设定的规则下自动完成，各种中间过程和繁文缛节会逐步遭到淘汰，取而代之的是更多有竞争力的服务供人们选择。这一切都将取决于类似.Net 这样的新一代网络软件和计算技术的发展。微软的策略是积极的，如果能够把握微软.Net 的思想精华，服务提供商、软件开发商、企业用户都将受益匪浅。

微软的开发工具系列产品将在.Net 中发挥重要作用，因为.Net 对微软软件技术的基础做了重大改变，微软的开发工具也将脱胎换骨，全面支持.Net 的 Web Service 开发。微软在近期推出的 Visual Studio.Net 测试版软件已经引起了全球开发者的极大兴趣。尽管 Visual Stuido.Net 在开发模式、开发工具的编程环境，以及语言的变化上都较从前的版本有很大跨越，先进的开发人员还是对一个可以进行全球资源访问的计算模型表现得异常激动。微软公司的 MSDN（微软开发者网络）也积极配合，不断推出支持传统开发人员向新的观念和工具转化。要知道，微

软公司操作系统成功的重要原因之一就是 API 的推动，今天.Net 提供的下一代网络 API 将对.Net 整个商业成功产生举足轻重的影响。

随着微软.Net 的全面铺开，Windows 95、Windows 98、Windows NT 等也都将逐步成为历史。然而就像人们今天还在回顾微软用 MS-DOS 创造第一次成功一样，这些产品的作用和功绩也都不可磨灭。还值得一提的是微软的 Windows Me 这个在千禧年借势推出的家庭操作系统在市场上并没有当年 Windows 95 那么热烈的反响。事实上微软也只是想在 Windows XP 之前测试一下市场的消费态度和需求。在美国，这种过渡产品的测试更为重要。尤其周围还有 Apple 等公司“缤纷多彩”的新产品的一些压力存在，微软一定要有所表示，才不会让消费者失望或转移视线。Windows Me 纯粹是一个市场营销式的产品，在微软的操作系统中甚至没有一个合法的地位。目前还不知道微软在 Windows Me 上究竟赚了多少钱，但是可以肯定是达到了目的，至少让微软获得了消费者最近阶段在购买行为和对操作界面上的反馈。

1.1.3　.Net 面临的竞争和挑战

在软件和技术方面，SUN 公司和 Oracle 公司自然是微软最大的竞争对手。自从微软推出.Net 策略后，SUN 公司和 Oracle 等公司都毫不示弱。SUN 在微软推出.Net 半年之后宣布了一个称作 ONE（Open Net Environment）的概念，在公众面前迎击微软的策略。Oracle 公司则以 Dynamic Web Service 的概念表示与微软抗衡。这些概念的技术取向都是大同小异的，但是从实现的可行性角度来看就不尽相同了。

但是正是因为未来的竞争格局一定还会发生变化，微软的竞争对手与合作伙伴也就会不断变化。也许有一天，今天的竞争对手可能会自然走到一起，成为志同道合的合作者，而谁又能保证他们最终不会成为一家人呢？从某种程度来看，以微软为代表的一些巨头将“控制”世界的数字命脉，这才是必须审慎对待和观察.Net 的真正原因。

1.1.4　.Net 和 C#

C#作为一种全新的语言，它的推出主要就是为了体现 Microsoft 的.Net 战略。Microsoft 的开发平台尽管拥有了 Visual Basic，Visual C++等在业界广泛应用的开发工具，但是由于这些开发工具最初是针对桌面应用提出的，C++的历史甚至可以追溯到 70 年代。勿庸置疑的是，这些开发工具对计算机软件产业的发展立下了汗马功劳。但是在计算机技术发展日新月异的今天，它们已经不能很好地满足所有的应用需要了。

开发一种全新的程序设计语言就提上了议事日程。事实上，早在 5 年前，作为 Microsoft 的竞争对手的 SUN 公司就推出了其 Java 语言，Java 语言的推出迅速在业界引起了轰动并很快成为一款主流的开发工具。可以说 SUN 公司的这一举措给 Microsoft 施加了巨大的压力并最终导致了 C#的产生。尽管我们称 C#为一种新的程序设计语言，但是它也吸收了很多已有语言的优点，尤其是它继承了 C++的大量语法要素和编程模式。一名熟练的 C++程序员可以很容易地使用 C#进行程序设计。

在 C#的白皮书中，Microsoft 这样描述 C#——“C# （pronounced C sharp） is a new programming language introduced in Visual Studio.NET. An evolution of C and C++, C# is simple, modern, type safe, and object oriented. C# was designed for building a wide range of enterprise applications that run on the .NET Platform.”。从中可以清楚看到，Microsoft 把 C#定位为 C++语言的一种革新版本，同时对 C#的应用背景也作了明确的界定，那就是主要是用于构筑基于.Net 平台的应用程序。

1.2 C#简介

C#语言自 C/C++演变而来。但是，它现代、简单、完全面向对象并且类型安全。如果是 C/C++程序员，学习曲线将会很平坦。许多 C#语句直接借用程序员所喜爱的语言，包括表达式和操作符。假如不仔细看，简直会把它当成 C++关于 C#最重要的一点：它是现代的编程语言。它简化和现代化了 C++在类、命名空间、方法重载和异常处理等领域的工作。摒弃了 C++的复杂性，使它更易用、更少出错。对 C#的易用有贡献的是减少了 C++的一些特性，不再有宏、模板和多重继承。特别对企业开发者来说，上述功能只会产生更多的麻烦而不是效益。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（garbage collect）等。所有的这些功能的目标都是瞄准了开发面向组件的软件。

1.2.1 简单

C#具有 C++所没有的一个优势就是学习简单。该语言首要的目标就是简单。很多功能（还不如说是缺少了 C++的一些功能）有助于 C#全方位的简单。在 C#中，没有 C++中流行的指针。默认地，所有的工作都放在受管理的代码中，在那里不允许如直接存取内存等不安全的操作。我想没有 C++程序员可以声称，从没有使用指针访问过不属于它们的内存。与指针“戏剧性”密切相关的是“愚蠢的”操作。在 C++中，有“::”、“.”、和“->”操作符，它们用于命名空间、成员和引用。对于新手来说，操作符至今仍是学习的一道难关。C#弃用其他操作符，仅使用单个操作符“.”。现在一个程序员所需要理解的就是嵌套名字的注解了。

不必记住基于不同处理器架构的隐含的类型，甚至各种整型的变化范围。C#使用统一的类型系统，摒弃了 C++多变的类型系统。这种系统允许程序员把各种类型作为一个对象查看，它是一个原始类型还是一个 full-blown 类。和其他编程语言相比，由于加框（boxing）和消框（unboxing）的机制，把简单类型当作对象处理并不能获得性能的改善。

首先，老练的程序员可能不喜欢它，但是整型和布尔型如今终归是两种完全不同的数据类型。这就意味着原来 if 语句中错误的赋值现在会被编译出错，因为 if 语句只接受布尔类型的值。再也不会出现误用赋值符为比较符这样的错误！C# 同时也解决了存在于 C++中已经有些年头的多余东西（redundancies）。这种多余包括常数预定义，不同字符类型等。鉴于多余表单已经从该语言中消失，故一般在 C#中都可以使用表单了。

1.2.2　现代

投入学习 C# 的努力是一笔大投资，因为 C# 是为编写 NGWS 应用程序的主要语言而设计。将会发现很多自己用 C++可以实现或者很费力实现的功能，在 C# 中不过是一部分基本的功能而已。对于企业级的编程语言来说，新增的金融数据类型很受欢迎。程序员用到了一种新的十进制数据类型，它专用于金融计算方面。如果不喜欢这种现成简单的类型，根据应用程序的特殊需求，可以很容易地创建出新的一种数据类型。

前面已经提到，指针不再是程序员编程武器的一部分。不要太惊讶，全面的内存管理已经不是程序员的任务。运行时 NGWS 提供了一个垃圾收集器，负责 C#程序中的内存管理。因内存和应用程序都受到管理，所以很有必要增强类型安全，以确保应用的稳定性。

对于 C++程序员，异常处理的确不是新的东西，但它是 C#的主要功能。C#的异常处理与 C++的不同点在于它是交叉语言（运行时的另一个功能）。在没有 C#之前，必须处理怪异的 HRESULTs，但现在由于使用了基于异常的强健的出错处理，这一切都结束了。

对于现代的应用程序，安全是首要的，C# 也不会例外。它提供了元数据语法，用于声明下述 NGWS 安全模式的能力和许可。

1.2.3　面向对象

不会预料一种新语言不支持面向对象的功能吧？C# 当然支持所有关键的面向对象的概念，如封装、继承和多态性。完整的 C#类模式构建在 NGWS 运行时的虚拟对象系统（VOS，Virtual Object System）的上层，VOS 将在下章描述。对象模式只是基础的一部分，不再是编程语言的一部分。

程序员一开始必须关注的事，就是不再有全局函数、变量或者是常量。所有的东西都封装在类中，包括事例成员（通过类的事例——对象可以访问）或者静态成员（通过数据类型）。这些使 C#代码更加易读且有助于减少潜在的命名冲突。

定义类中的方法默认是非虚拟的（它们不能被派生类改写）。主要论点是，这样会消除由于偶尔改写方法而导致另外一些原码出错。要改写方法，必须具有显式的虚拟标志。 这种行为不但缩减了虚拟函数表，而且还能确保正确版本的控制。

使用 C++编写类，程序员可以使用访问权限（access modifiers）给类成员设置不同的访问等级。C# 同样支持 private、protected 和 public 三种访问权限，而且还增加了第四种：internal。有关访问权限的详细情况将在第 5 章“继承与多态”中说明。

程序员曾经创建了多少个类是从多基类派生出来的。大多数情况，仅需从一个类派生出。多基类惹出的麻烦通常比它们解决的问题还多。那就是为什么 C#仅允许一个基类。如果程序员觉得需要多重继承，可以运用接口。

一个可能出现的问题：在 C# 中不存在指针，如何模仿它？这个问题的答案很有代表性，它提供了对 NGWS 运行时事件模式的支持。