■ 微 软 新 技 术 教 材

C#编程语言
程序设计与开发

陈 钟　刘 强　张 高　等编著

清华大学出版社

微软新技术教材

C#编程语言程序设计与开发

陈钟　刘强　张高　等 编著

清华大学出版社

北　京

内 容 简 介

C#是专在 . NET 平台上开发的新型编程语言，是从 C 和C++ 语言演化而来的，并充分考虑了已有编程语言的优点，是一种简单而功能强大的编程语言。本书从先理论后实践的角度出发，以读者不具备面向对象概念以及缺乏编程知识为前提，按照难易程度编排内容。首先介绍 C#语言特性，然后分别从编程语言基础知识、C#初级特性和高级特性三个方面对 C#进行阐述，充分突出了 C#的面向对象思想以及 C#所具有的新特性；最后结合 . NET 平台介绍 C#的 Windows 应用程序、数据访问、Web 编程的程序开发，并提供了相关的开发案例以供参考。

本书言简意赅，易懂，结构清晰，内容比较全面而且容易掌握，有利于初学者阅读和理解。本书篇幅不大，适合于 . NET 与 C#的初学者使用。

图书在版编目(CIP)数据

C#编程语言程序设计与开发/陈钟等编著 . —北京：清华大学出版社，2003.9
(微软新技术丛书)
ISBN 7-302-07081-4

Ⅰ. C… Ⅱ. 陈… Ⅲ. C 语言－程序设计 Ⅳ. TP312

中国版本图书馆 CIP 数据核字(2003)第 071336 号

出 版 者：清华大学出版社　　**地　　址**：北京清华大学学研大厦
http://www.tup.com.cn　　**邮　　编**：100084
社总机：010-62770175　　**客户服务**：010-62776969
组稿编辑：徐培忠
文稿编辑：郑寅堃
封面设计：付剑飞
印 刷 者：清华大学印刷厂
发 行 者：新华书店总店北京发行所\清华大学出版社出版发行
开　　本：185×230　**印张**：19.5　**字数**：434 千字
版　　次：2003 年 9 月第 1 版　2003 年 9 月第 1 次印刷
书　　号：ISBN 7-302-07081-4/TP·5196
印　　数：1～3000
定　　价：30.00 元

序

计算机程序设计语言是用于编写计算机程序的人工语言。与人们日常使用的自然语言类似，程序设计语言的基础也是一组记号和规则(后面有时简称为语言)。语言的发展对软件的开发和应用推广起到了巨大的推动作用。早期的机器语言、汇编语言面向特定的计算机体系结构(计算机指令集)，现在的高级程序设计语言以描述待解决问题为核心，更接近于数学语言和自然语言。其间，被市场广泛接受的程序设计语言大都突出地代表了发展历史阶段的时代特征。例如，20 世纪 80 年代以 C 语言为核心，代表了过程式程序设计的时代特征；20 世纪 90 年代，随着图形化用户界面(GUI)的广泛应用，进入了以C++ 语言、Java 语言为代表的面向对象技术时代。进入 21 世纪，Web 和 Internet 已经广泛渗透到信息处理的方方面面，标志着面向构件的时代已经到来。C#是为适应新的面向构件的时代特征而发布的计算机程序设计语言。

微软公司 2000 年 7 月首次推出 C#和. NET 框架，2001 年 12 月信息通信技术国际标准化组织 ECMA 批准了 C#标准 ECMA-334，并向 ISO 提交了该标准；2003 年 4 月，ISO 批准了该标准，编号为 ISO/IEC 23270，这为 C#的发展和推广奠定了基础。

C#语言是微软公司的. NET 体系中的一个关键环节。作为语言，C#引入了属性/属性信息和装箱/拆箱等构件开发和组装所需要的概念，同时通过与运行环境、开发环境的集成使得构件的部署变得非常简单；为基于 Internet 的应用与 Web 服务开发提供了有效的支持。可以预期，如果. NET 可以真正实现独立于操作系统的目标，C#可望成为跨平台分布式网络环境中应用程序开发的主流语言之一。

国家教育部和国家发展计划委员会批准的示范性软件学院是面向产业培养高层次、实用型、复合型、国际化人才的人才培养基地。希望这本由微软亚洲研究院、北京大学软件学院、清华大学软件学院等共同组织出版的书籍能够为软件人才的培养起到积极的作用。

杨芙清

2003 年 9 月 9 日

前　言

随着新一轮的跨平台软件开发浪潮的到来，Microsoft 提出了 .NET 战略，并于 2000 年 6 月 22 日正式推出了其下一代的计算计划——Microsoft .NET(简称 .NET)。它将使微软现有的软件在当前的 Web 时代不仅适用于传统 PC，而且也能够满足目前不断出现的新设备，如移动电话以及 PDA 的需要。

什么是 .NET？来自微软官员的声音："……因特网的革命……从微软的角度来说，就是要建设一个平台来创建并且支持新一代的应用。……我们必须有一套通用系统服务来支持这样的操作。这种观点说明，我们还有下一个层次的发展，也就是说因特网下一步的发展，将使因特网的作用远远超越展现一个网站。"

.NET 是一个面向网络、支持各种用户终端的开发平台环境，在这个平台上，可以开发出运行在 Windows 和其他平台上的几乎所有的应用程序。.NET 平台提供了大量的工具和服务，能够最大限度地发掘和使用计算及通信能力。

配合 .NET，微软推出了一种新的程序语言——C#。它是从 C 和C++ 语言演化而来的，它吸取了以前的教训，考虑其他语言的优点，并解决了其中存在的问题。C#是一种现代的面向对象的程序开发语言，充分体现了构件化的程序设计思想。C#为程序员提供了快捷的程序开发方式，具有强大的控制能力。在 .NET 运行库的支持下，.NET 的各种优点在 C#中表现得淋漓尽致，C#与 .NET 得到了完美结合。

本书作为面向 C#及 .NET 初学者(主要面向学生)的教材，从最基础的编程语言知识开始，逐步介绍 C#的初级、高级特性，并结合 .NET 提供的类库介绍使用 C#开发各种应用程序。通过本书的学习，读者将对 C#有一个比较全面的认识和了解，并能应用 C#开发应用程序。

本书主要内容

第 1 章简要地介绍了编程语言的发展历史，使读者了解当前编程语言的概况。

第 2 章中介绍 .NET 运行时(Runtime)环境。首先简述了 .NET Framework，对其特性作了简要概述；然后对其开发工具 Visual Studio .NET 进行介绍。

从第 3 章到第 6 章主要讲述了 C#编程语言。

第 4 章对 C#语言基础进行了介绍，主要内容包括 C#基本类型、变量和常量、数组、表达式、语句、操作符以及命名空间等编程语言的基本要素。

在第 4 章中，首先简述了面向对象的相关概念，如对象、继承、封装及多态等；然后介绍 C#面向对象的初级特性，包括如何用 C#定义类、创建对象、销毁对象，C#方法的调

用，C#的静态成员，性质等。

第5章给出了C#的高级特性。首先介绍在C#中实现类的继承和多态以及操作符重载，其中类的多态包括有abstract类、虚函数、seal类等内容；接下来对C#结构和接口、类型之间的转换、索引器和集合、异常处理进行了介绍；然后介绍了C#在C和C++的基础上引入的一些新的特性，包括有委托、程序集、属性等，这些新的特性有利于更轻松、更方便地开发应用程序。

第6章主要介绍如何使用文件，包括文件和目录操作的类、文件的创建、文件的读写以及异步访问文件。

第7章到第10章的内容为如何使用C#开发应用程序。

在第7章中，讨论了如何创建传统的Windows应用程序，同时介绍了Windows Forms的各种可用资源，如控件、菜单和对话框等。

第8章主要探讨了如何使用ADO.NET进行数据访问。介绍了ADO.NET中的DataSet、Data Provider等对象以及如何利用这些对象访问数据。

第9章首先概述了Windows Service，然后介绍如何创建Windows Service，如何对Windows Service进行监视和控制。

第10章主要讲述了用C#进行Web编程。.NET Framework提供了ASP.NET用于开发Web应用程序以及Web服务。通过ASP.NET可以应用Web Forms进行Web编程。

本书读者对象

本书适合作为学生的教材，主要是面向那些希望学习C#、没有面向对象概念而且缺乏程序开发经验的学生及初学者。

在本书的编写过程中，得到了微软亚洲研究院的大力支持，在此表示感谢。此外，北京大学的刘鹏、张锦懋、武勇以及清华大学的谌卫军、张宏官、王武君、余正洋等人也参与了本书的具体编写和校对工作，付出了很多努力，没有他们的努力工作，就不会有本书的最终面世。

最后感谢清华大学出版社责任编辑郑寅堃先生的辛勤工作。

编　者

2003年4月25日

目　录

第1章　编程语言的发展

一种编程语言包含一套完整的语法和语义规则，我们用它来描述在计算机上运行的程序。自从有了计算机以来，计算机的应用领域越来越广泛，对计算机程序的需求也越来越多，因此就推动了人们去构造各种适合于不同用途的编程语言。在本章中将简要介绍编程语言的发展历史，使得读者了解当前编程语言的概况，更好地学习 C#语言。

1.1　机器语言与编程语言

在计算机内部，是由计算机的机器代码来控制计算机的算术、逻辑、输入输出等操作的。这种机器代码实质上是计算机本身的语言，计算机可以对这种代码直接响应。但是这种语言与人们的思考方法和描述问题解决的方式相距太远，因此人们设计了所谓的编程语言，通常它们是高级而且通用的，即不依赖于具体的机器而且可以适用于解决各种领域的问题。在过去的 50 多年中，人们设计并创建了 1000 多种编程语言，尽管其中大部分语言我们并不熟悉甚至不知道，但是它们的出现为编程语言的发展都起到了积极的作用。从语言的级别来分，可以把语言分为低级语言和高级语言：低级语言指的是机器语言和类似于机器代码的汇编语言；高级语言指的是和机器无关的语言，例如 FORTRAN、Pascal 以及 C 等。

1.1.1　低级语言

用机器语言编写的程序必须考虑到用于其运行的具体机器的各种细节，它层次最低，而且仅由许多 0 和 1 执行。虽然它可以为机器识别且运行效率高，但是可读性太差，因此就需要另一种语言来编写程序，它应该是符号式的、有助于记忆的。

汇编语言是机器语言的一种变形，是面向机器的一种编程语言，可以说是机器语言的符号化描述。汇编语言用名字和符号来取代各种机器操作、值和存储位置，使得各条指令具有了相当的可读性。但是由于它也是面向机器的，受到机器的束缚，对机器的体系结构有一定的依赖性，对硬件需要有更深入的了解，而且编程效率不高。

1.1.2 高级语言

由于汇编语言依赖于硬件，助记符量大且难记，于是人们又设计了更加易用的所谓高级编程语言。在这种语言下，其语法和结构更类似普通英文，且由于远离对硬件的直接操作，使得一般人经过学习之后都可以很容易地开始编写程序。

从历史上看，自从 1951 年 H. Rutishauser 提出了高级编程语言及其程序的思想，1956 年在 J. Backus 的领导下实现了 FORTRAN 语言以后，编程人员就从使用机器语言编写程序的烦琐工作中解脱出来，应用高级语言开发程序使得编程人员的生产率有了显著提高。经过 50 多年的发展，高级语言已经在绝大部分应用领域中取代了机器语言和汇编语言。与低级语言相比，它主要具有以下优点：

- 良好的可读性　由于高级语言的语法和结构类似于普通英文，人们很容易读懂用它编写的程序，易于对这些程序的理解、修改和扩展；
- 可移植性　由于高级语言和具体机器无关，因此用高级语言编写的程序能够不做修改或者只需做很少的修改，就可以在不同机器上运行。

此外，它还有高可用性的程序库、实现过程中的错误检查等优点。

正因为第一个高级语言 FORTRAN 具备了以上的优点，它一经推出，便广受欢迎。可以说从 FORTRAN 开始，编写程序逐渐进入了一个高级语言时代。

1.2 程序设计模式

程序设计模式是指编写计算机程序或软件的模式，也可以说是如何进行编程的方式。在这么多年高级语言的发展中，人们不断引入新的程序设计模式，设计新的语言以适应不断拓展的计算机应用领域。目前主要有四类程序设计模式：命令式程序设计、函数式程序设计、面向对象的程序设计以及逻辑程序设计。

1.2.1 命令式程序设计(Imperative Programming)

命令式程序设计是一种通过程序状态和用于修改程序状态的表达式来描述计算过程的编程方式。命令式程序就是一个计算机要完成的命令序列。目前大部分编程语言都属于命令式语言。最早的命令式语言应该说是机器语言，因为用机器语言编写的程序实际上就是直接让计算机进行操作的指令集合。

命令式高级语言最早起源于 FORTRAN，它主要用于编写科学计算程序。尽管它当时主要是为 IBM 704 这种主机设计，而且 Backus 也没有考虑把它移植到其他类型的主机上，

但是它所使用的数学记号使得 FORTRAN 基本上和机器无关。经过几十年的发展，FORTRAN 先后经历了 FORTRAN66、FORTRAN77、FORTRAN 90 以及 FORTRAN95 等若干个版本。在接下来的 20 年当中，出现了许多命令式的高级语言。

Algol60 主导着 20 世纪 60 年代编程语言的发展，以至于当时人们称那时的命令式语言族为 Algol 语言族。它引入了块结构、复合表达式、递归过程调用、循环表达式、嵌套的 if 语句，可以说是 Pascal 和 C 的祖先。60 年代还出现了 COBOL 和 BASIC 等命令式语言。

Niklaus Wirth 在 1970 年，Dennis Ritchie 在 1972 年先后推出了 Pascal 和 C 语言。当时 Pascal 的设计目的是用于教学，因此具有严谨的语法结构，它是基于 Algol 创建的。而 C 语言当时是为了支持可移植操作系统而设计的，它提供了一套丰富的运算符集合和比较紧凑的语法形式，UNIX 在 1973 年用 C 进行了重写。尽管 Pascal 和 C 的最初设计目的都不是通用编程语言，但现在它们已经成为了非常流行的编程语言，应用在广泛的领域中。

在 20 世纪 70 年代，已经有很多种编程语言在使用。美国国防部为了其嵌入式系统的开发和维护，1974 年开始设计 Ada，并于 1983 年推出了一个 ANSI 标准，即 Ada83；在 1995 年经过了一次修正，形成了 Ada95，它修正了 Ada83 的缺陷，并拓展了其功能。

1.2.2　函数式程序设计(Functional Programming)

函数式程序设计着重于函数表达式的求值而不仅仅是命令的执行，这里的函数主要是指数学函数。最早的函数式语言是 1958 年设计的 Lisp，它当时是为了人工智能应用而设计的。Lisp 的作者 McCarthy 对它的解释是：Lisp 的初始设计是为了做符号数据处理。它被用于做各种符号演算、微分和积分演算、电子电路理论、数理逻辑、游戏推演以及人工智能的其他领域。

从 20 世纪 60 年代到 90 年代，人们在 Lisp 基础上发展出了很多函数式语言，有 ML、Scheme、Haskell、Erlang 以及 Clean 等。Scheme 是 Guy L. Steele 和 Gerald Jay Sussman 在 1975 开发的，由于它清晰的设计，在研究和教学领域非常受欢迎。Lisp 也有多个版本，有强调性能和产品质量的 MacLisp，有引入了编程环境概念的 InterLisp。Lisp 的繁荣发展使人们在 1984 年创建了 Common Lisp，此后在 Common Lisp 中引入了面向对象的概念，建立了 Common Lisp Object System，简称 CLOS。

1.2.3　面向对象的程序设计(Object-Oriented Programming)

面向对象的程序设计既是一种程序设计模式，也是一种软件设计的方法学。它的主要思想是根据对象的类来构造程序，其中，对象是包含状态(数据)和行为(方法)的实体。面向对象的程序设计具有抽象、封装、多态以及继承等特点。

面向对象的程序设计最早出现在Simula语言中，它是由Ole-Johan Dahl和Kristen Nygaard在1961～1967年设计的，当时他们的设计目标是用于模拟仿真领域。Simula的推出改变了人们编程的思考方式，这其中最关键的概念就是上面提到的对象的类概念。

20世纪70年代Adele Goldberg、Dan Ingalls、Alan Kay、Ted Kaehler和其他人一起开发了Smalltalk，它对Simula中对象和类的概念进行了精炼，同时它的设计也受到了Lisp的影响。Smalltalk不仅仅是一种语言，它还是一个具有图形交互界面的完整的交互系统。Smalltalk对很多语言的发展都有很大影响，如Objective-C、Actor、Java以及Ruby等。90年代一些软件开发的思想，如设计模式，就来源于Smalktalk。尽管今天有了更多的语言供选择，Smalltalk仍然拥有它忠实的用户群。

到了80年代中期，面向对象的程序设计已经成为了主流，人们在已有的编程语言中加入新特征以使它们支持对象的概念，在20世纪的最后20年我们可以看到很多编程语言都在朝这个方向发展，其中的代表非C++莫属。它是由Bjarne Stroustrup创建的，它基于C语言，因此C++保持了C语言的高效率。它在C语言中引入了对象的概念，并受到了Smalltalk的影响。在1998年C++被采纳为标准(ISO/IEC 14882-1998)。在C++标准的基础上，一些软件公司扩展了C++并推出自己的开发工具。目前用户最多的两种C++版本分别是Microsoft的Visual C++和Borland公司的Borland C++。

除了C++以外，Ada，BASIC，Lisp及Pascal等编程语言都增加了面向对象的特性。例如Visual Basic已经成为了广受欢迎的编程语言之一；Pascal被扩展为Object Pascal；Borland公司把Object Pascal和可视化编程结合在一起，推出了Delphi，它也是非常流行的编程语言之一。一些80年代、90年代后期创建的语言，如Perl、Python，则面向对象与非面向对象的程序设计模式兼而有之。

1995年Sun公司推出了Java，这是一种“纯”面向对象的编程语言，它跨平台而且支持对Internet的编程，这使得Java在短短几年内便成为非常热门的编程语言。尽管用Java编写的程序运行速度较慢，但是它的跨平台特性使它和其他编程语言相比具有它的优越性。

2001年微软公司推出了它新设计的一种面向对象的编程语言——C#，它是在充分吸收了C++、Java、Delphi等语言的优点的基础上开发出来的。2001年12月，信息通信技术的国际标准组织ECMA批准了C#标准ECMA-334，并向ISO提交了该标准；2003年4月，ISO批准了该标准，标准编号为ISO/IEC 23270，这为C#的发展和推广奠定了坚实的基础。

1.2.4 逻辑程序设计(Logical Programming)

逻辑程序设计也叫约束程序设计(Constraint Programming)，它强调的是计算的逻辑属性而不是计算的步骤和过程。目前广泛使用的逻辑编程语言是Prolog，它来源于program-

mation en logique(在逻辑中做程序)，是 1973 年 Alain Colmerauer 与 Philippe Roussel 在法国为自然语言处理而开发的编程语言。它的语法和语义非常简洁明了，主要应用于解决人工智能问题。Mercury 是另一种逻辑编程语言，它是基于 Prolog 开发的，对于现实世界的编程问题更有用。

根据上面对四种程序设计模式的介绍，我们以图 1-1 来说明编程语言的发展过程。图中有一些编程语言上面没有提及，为了完整性我们也把它们加进图中。

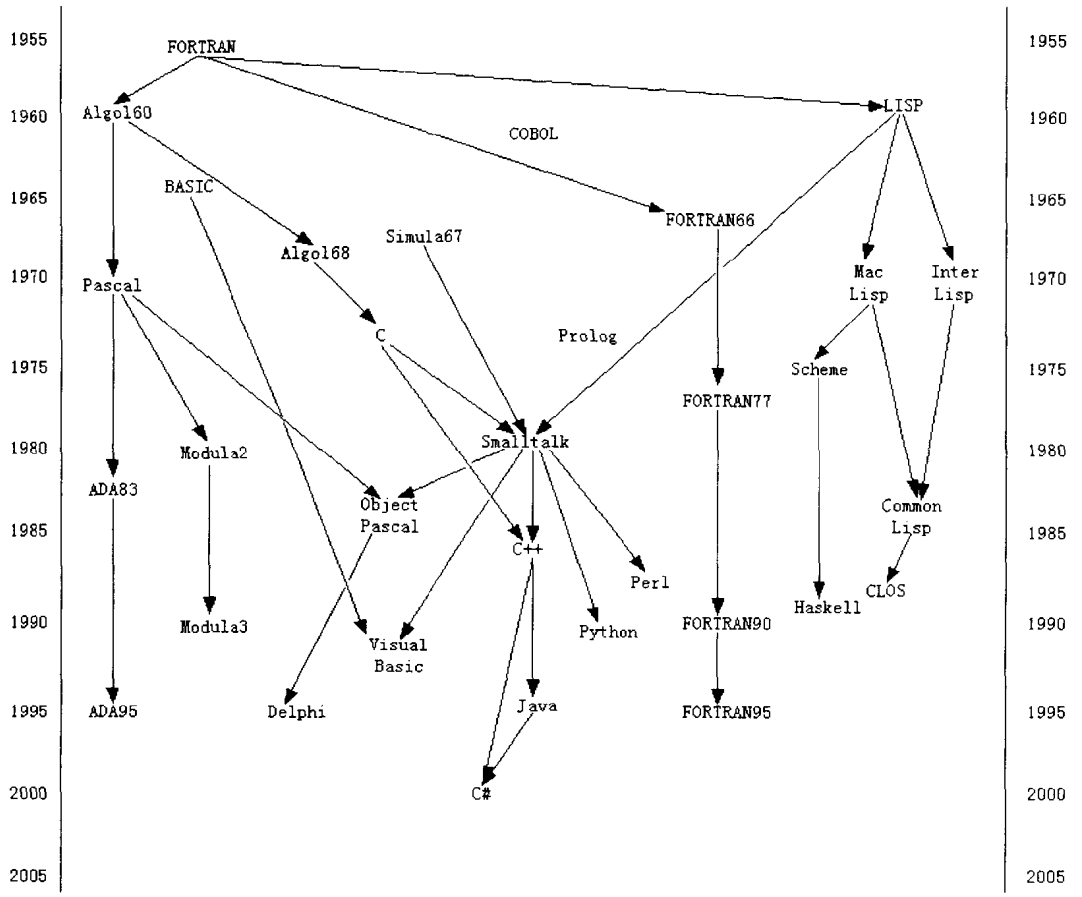

图 1-1　编程语言发展的示意图

当初人们设计高级编程语言是为了某种特殊用途，但是随着计算机技术的不断发展，计算机应用领域的日益广泛，人们对于如何进行程序设计不断地有新的看法和思想，对编程语言也就有了新的要求，这推动着编程语言的持续发展。从图 1-1 我们可以看到，编程语言的发展趋势是语言的通用化，新出现的编程语言不再是只解决某类特殊问题，它们的目标是可以适用于任何应用领域，同时降低程序设计的复杂度。