

王秀梅 王德军 杨磊 等编著

C#

从入门到精通

Cong RuMen Dao JingTong

中国标准出版社

C#从入门到精通

王秀梅 王德军 杨磊 等编著

中国标准出版社

图书在版编目 (CIP) 数据

C# 从入门到精通/王秀梅编. —北京: 中国标准出版社, 2002. 5

ISBN 7-5066-2775-2

I. C… II. 王… III. C 语言-程序设计
IV. TP312

中国版本图书馆 CIP 数据核字 (2002) 第 031168 号

中国标准出版社出版

北京复兴门外三里河北街 16 号

邮政编码: 100045

电话: 68523946 68517548

中国标准出版社秦皇岛印刷厂印刷

新华书店北京发行所发行 各地新华书店经售

*

开本 787×1092 1/16 印张 17¼ 字数 416 千字

2002 年 7 月第一版 2002 年 7 月第一次印刷

*

印数 1—3 000 定价 25.00 元

网址 www.bzchs.com

版权专有 侵权必究

举报电话: (010) 68533533

前 言

.Net 是微软推出的面向 21 世纪的软件开发平台。微软希望，在第三代因特网中它所发挥的作用就像 Windows 在第二代个人电脑中发挥的作用一样，拥有更友好的界面，更容易地将应用程序组合在一起，可以在各种设备上运行，无论对于个人电脑还是非个人电脑设备，都能得到相同的体验。.Net 作为第三代 Internet 产品，其思想与 Java 有着极大的区别，用户数据生存于网络，而不只是生存于不同平台，.Net 允许不同应用程序之间能互相传递信息，而 Java 的 beans 是不能和其他语言建立的构件共享数据的。.Net 是一种中立的语言，它基于 XML (Extensible Markup Language) 和 SOAP (Simple Object Access Protocol)，允许开发者使用各自不同语言，使用它所有的功能，所以可以肯定，在网络飞速发展的今天，.Net 将很快成为未来的主流。

C# 是一种现代的，面向对象的语言，它使开发人员能够在微软新的 .Net 平台上快速建立广泛的应用，其提供的工具和服务能充分发掘系统的计算和通讯能力。因为其优良的面向对象设计，在构建从高级业务对象到系统级应用的各种不同组件时，C# 是一个首要的选择。使用简易的 C# 语言构造，组件可以被转换为 Web 服务，从而允许从运行在任何操作系统上的任何语言中跨越 Internet 调用它们。不仅如此，C# 的设计为 C++ 程序员带来了快速的开发能力，而不用牺牲 C++ 已有的功能和控制能力。通过这种继承，C# 高度地保持了与 C 和 C++ 的一致。开发者只要熟悉 C 和 C++ 语言就可以快速地掌握 C# 并写出更多的 C# 应用程序。

本书给出大量的实例解释 C# 语言在 .Net 环境下的使用方法。使读者朋友们真实地感受 C# 究竟给我们带来了什么。全书分为三个部分，由浅入深地对 C# 做了详尽的介绍。

第一部分——入门篇：包括第 1 章至第 3 章，介绍了 .Net 平台及 C# 语言的基本状况，并给出第一个 C# 程序“Hello World!”；

第二部分——提高篇：包括第 4 章至第 8 章，介绍 C# 的数据类型、变量、类型转换、表达式和语句等语法知识，中间适当穿插较多

的程序源代码；

第三部分——精通篇：包括第 10 章至第 18 章，通过大量的实例介绍 C# 类、结构、数组、接口、组件、条件编译、XML 文档和跨平台调用、与 COM 的交互访问、输入输出等内容，最后给出一个综合实例——五子棋程序，以帮助读者更好地理解 C# 的强大功能。

附录分别对 C# 内幕、C# 与 Java 的比较、C# 与 VB 的不同之处以及 .Net 名字空间成员等作了介绍。

学习和掌握一门语言的最好方法是不断的实践。因此，书中提供了大量的示例源代码，相信它们对读者的学习会大有裨益。

本书由王秀梅、王德军、杨磊、谢松县等编著；郭建龙、丁鲲、周松君、董波、胡志、钱理全、吴光中、沈志光、王艳杰、于世佳、周建等参与书稿的讨论和部分章节的写作；韩旭、杨帆、朱文献调试了全书的程序。在此表示感谢！

由于作者水平有限，再加上时间紧迫，书中难免有所疏漏或不当之处，恳请读者批评指正。

编 者

2002 年 2 月

目 录

第 1 章	.Net 简介	1
1.1	一项具有革命性的事业	1
1.2	Microsoft .Net: 超越浏览·超越网站	2
1.3	Microsoft .Net 平台建造了新一代因特网	3
1.4	Microsoft.Net 能带给人类什么	6
第 2 章	.Net 编程语言 C#	7
2.1	NGWS runtime 技术基础	7
2.2	C# 语言的特点	11
2.3	如何安装 C# 编译器	14
2.4	如何用 Visual studio 编辑器编辑 C# 程序	14
第 3 章	Hello World!	16
3.1	Hello World! 程序	16
3.2	代码分析	16
3.3	运行程序	19
3.4	添加注释	20
第 4 章	数据类型	22
4.1	值类型	22
4.2	引用类型	28
4.3	装箱和拆箱	32
第 5 章	变量	33
5.1	变量的种类	33
5.2	缺省值	35
5.3	明确赋值	35
第 6 章	类型转换	38
6.1	隐式类型转换	38
6.2	显式转换	42

6.3	标准转换	44
第 7 章	表达式	45
7.1	表达式的分类	45
7.2	操作符	46
7.3	算术操作符和算术表达式	48
7.4	赋值操作符和赋值表达式	53
7.5	关系操作符和关系表达式	56
7.6	逻辑操作符	59
7.7	条件逻辑操作符	60
7.8	条件操作符	60
7.9	其他操作符	61
第 8 章	语句	66
8.1	结束点和可达性	66
8.2	块	67
8.3	空语句	68
8.4	标号语句	68
8.5	声明语句	69
8.6	表达式语句	70
8.7	选择语句	71
8.8	循环语句	76
8.9	跳转语句	80
8.10	try 语句	83
8.11	checked 和 unchecked 语句	85
8.12	lock 语句	85
第 9 章	类	87
9.1	类的声明	87
9.2	类的成员	90
9.3	常量	93
9.4	域	94
9.5	方法	99
9.6	属性	109
9.7	事件	116
9.8	索引	119
9.9	操作符	122
9.10	实例构造器	125

9.11	析构器.....	129
9.12	静态构造器.....	130
第 10 章	结构	133
10.1	结构声明.....	133
10.2	结构成员.....	134
10.3	结构示例.....	134
第 11 章	数组	138
11.1	数组类型.....	138
11.2	数组的创建.....	139
11.3	数组元素访问.....	140
11.4	数组成员.....	140
11.5	数组协变 (array covariance)	140
11.6	数组初始化器.....	141
第 12 章	接口	143
12.1	接口声明.....	143
12.2	接口成员.....	144
12.3	全权接口成员名称.....	148
12.4	接口实现.....	149
第 13 章	用 C# 写组件	160
13.1	第一个组件.....	160
13.2	使用名字空间工作.....	164
第 14 章	条件编译	169
14.1	预处理用法.....	169
14.2	条件编译.....	172
第 15 章	XML 文档和跨平台调用	175
15.1	XML 文档	175
15.2	跨平台调用.....	178
第 16 章	与 COM 的交互访问	183
16.1	产生一个非受管 COM 类容器	183
16.2	在 C# 代码中声明一个非受管 COM 组件对象类	184
16.3	在 C# 代码中产生一个非受管 COM 对象	185

16.4	在 C# 中声明一个非受管 COM 接口	185
16.5	在 C# 中查询接口 Query Interface	188
16.6	把所有的特性放在一起	188
第 17 章	C# 中的输入输出	194
17.1	文件和流	194
17.2	基本文件 I/O	195
17.3	复合流	200
17.4	创建一个 Writer	202
17.5	异步文件 I/O	202
第 18 章	综合实例：五子棋程序	209
附录 A	C# 内幕	216
附录 B	C# 与 Java 的比较	223
附录 C	C# 与 VB 的不同之处	228
附录 D	.Net 名字空间成员	233

第 1 章

.Net 简介

今天的软件产品仅仅是一张光盘，用户购买软件，亲自安装、管理和维护。但是软件服务是来自因特网的服务，它替用户安装、更新和跟踪这些软件，并让它们和用户一起在不同的机器间漫游。它为用户存储自己的信息和参考资料。

——鲍尔默

本章内容要点

Microsoft.Net 的概念；Microsoft.Net 能带给人类什么。

1.1 一项具有革命性的事业

对于计算机工业，革命就是一种“生活方式”。20 年以前，还是大型机的时代。那时只有极少数人能够使用计算机，而且只能通过邻近的信息产业机构来使用。个人电脑和图像化用户界面的出现却改变了一切，将计算机普及到了千家万户，并使它真正成为一种可以大工业生产的产品。企业界意识到，由个人电脑联结的网络和基于个人电脑的服务器可能改变他们的商务模式，而个人电脑对消费者来说也迅速地成为新兴的娱乐媒介。后来，因特网接踵而至，它革命性地改变了人们的交流方式，创造了丰富而新颖的信息和娱乐资源，并且在“商务”的前面加上了一个代表“电子”的字母“e”。今天，全球有将近三亿人口正在使用因特网。国际数据集团提供的资料显示，2001 年全球的网上交易金额将超过 250 亿美元。

面对这些网络神话，人们仍然发现存在着巨大的改进空间。今天的因特网在很大程度上还在模仿旧式大型计算机的工作方式。尽管有充足的带宽资源，大量的信息还是被“锁”在了中央数据库里，并由“保安人员”看守。用户必须依靠网络服务器来完成所有的上网操作，这酷似老式的分时复用系统。网站好像一个个孤立的小岛，并不能按照用户的指令在它们之间进行有意义的交流。今天的网络似乎只能通过单个的网站向单个用户提供有限的服务——因为大多数的网页只能呈现 HTML 格式的数据“图片”，而非信息本身（对大多数网页来说，在现有技术条件下要两者兼顾是非常困难的）。非但如此，浏览器本身在很多方面都只是一个被美化了的“哑巴只读终端”，用户可以轻易地浏览信息，但是

很难进行编辑、分析和复制（实际上也就是知识工作者需要做的所有工作）。“个性化”只意味着重复地进入网站，并不断地将个人隐私泄露给用户所访问的每一个网站。人们必须适应科技，而不是科技反过来适应人的要求。

当用户同时使用多台个人电脑或便携设备时，上面提到的问题就会成倍地出现。为了查阅在线信息、电子邮件、离线资料和其他数据，用户不得不与多个用户界面（有时甚至相互冲突）、多变的数据入口分级和时断时续的信息同步作“斗争”（就是当用户通过电脑使设备和网络处于物理连接时）。在线数据以一种不完善的预定格式呈现，极大地降低了信息的可用性。一个符合用户要求的风格化“个人信息空间”概念仍然是个梦想。

网络开发商希望建立、测试和发布有吸引力的网站，但是可用的工具真是少之又少。其中的许多人更多地注重网站的视觉吸引力，而不是实用性。没有任何开发商以一种持续而有效的方式去关注软件的整个生命周期——从构想、开发、投放到维护。还没有任何系统，可以让开发商为个人电脑编写的程序能够同时运用在其他的设备上。

公司用户还要面对更多的挑战。多如牛毛的小型服务器群的出现降低了单点失误，而在整体上提高了计算的可靠程度，但同时系统的管理也变得越来越复杂。性能测试、能力规划和运营管理正在对今天的多层级、多功能网站提出挑战。新的电子商务系统很少能够周密地计划：如何与现存的商务系统协调工作，新型系统应该使你的商务伙伴和消费者可以安全地跨越防火墙、高效地与用户做生意，但是类似的系统在设计上如此之难，以至于许多公司只好采用昂贵的双重系统。

难道一切发展都到尽头了吗？其实每个人都相信，网络总是会朝前发展。但是，为了这个能真正提高开发商、企业和消费者能力的新进展，人们必须具有超前的想象力。而一个优秀公司的职责，就在于为人类大胆设想，并把它变成现实。

1.2 Microsoft .Net：超越浏览，超越网站

Microsoft .Net 将创造出真正的分布式网络服务，它能够整合并协调为数众多的辅助程序，为消费者提供独到的服务，而这正是今天的网站所梦寐以求的。Microsoft .Net 将推动新一代因特网的发展。它将真正地让人们可以在任何时间、任何地点、通过任何设备得到信息。

Microsoft .Net 背后的基本理念就是：不再关注单个的网站和与因特网连接的单个设备，而是要让所有的计算机群、相关设备和服务商协同工作，提供更加广泛和丰富的解决方案。人们将能够控制何种信息、在何时、以何种方式传送给自已。计算机群、相关设备和服务商将能够相互协作，提供更加多样的服务，而不是像现在这样的一座座孤岛，由用户来提供仅有的整合。消费者将能够把企业所提供的产品与服务严丝合缝地嵌入自己的电子设备构架中。20 世纪 80 年代个人电脑的出现第一次向人们提供了信息自主的权利，而 Microsoft .Net 是另一种前景，是这种自主权利的又一次延伸。

Microsoft .Net 将推动因特网的一次变革，HTML 格式的表述将升格为 XML 格式的可编程信息。XML 是由万维网联盟（W3C）制定，得到业界广泛支持的标准，这个联盟也制定了网络浏览器的标准。在开发过程中，微软公司给予了全面的支持，但这项技术并非微软专有。XML 是将真实的数据从它的视觉表象中分离出来的一种方法。它是新一代因特网

的一项关键技术，是把信息“解放”出来的一种方法，让信息可以重新组织、编程和编辑；它同时也是一种新思路，将数据以更有用的形式传送给多种数字设备；它让网站之间相互协作，提供一系列互动式的新型网络服务。

Microsoft.Net 包含以下组成部分。

(1) Microsoft.Net 平台——包含 .Net 基础结构和工具，以运行新一代服务程序；.Net 用户体验支持更加丰富的客户端；.Net 构造模块，新一代高度分布式超服务；.Net 设备软件，以支持新型智能化因特网设备。

(2) Microsoft.Net 产品和服务——包含带有核心构造模块服务的 Windows.Net；MSN.Net；个人订阅服务；Office .Net，Visual Studio.Net 和 bCentral for .Net。

(3) 第三方 .Net 服务——众多微软的商务伙伴和第三方开发商将有机会制造出基于 .Net 平台的企业软件和垂直型服务程序。

Microsoft.Net 将把计算和通信带入一个丰富、合作和互动的环境中，远远胜过今天的单向网络。由新型高级软件支持的 Microsoft.Net 将利用一系列的应用程序、服务程序和相关设备来创造一种个性化的数字体验，它能够不断地按照用户本人、用户的家人和公司的需要进行自动的调整。它意味着用户将能够通过一整套新型软件构成的整体服务程序来管理自己在因特网时代的生活和工作。

(4) 对于消费者：Microsoft.Net 意味着简单化的整体服务；统一的信息浏览、编辑和授权；查看用户的资料、工作和在线与离线媒体；一种整体的系统方案；随时随地的个性化；绝对的自由。例如，对于用户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。

(5) 对于知识工作者和企业：Microsoft.Net 意味着统一的信息浏览、编辑和授权；丰富的同步传播；密切的移动通信联系；得力的信息管理和电子商务工具，在基于内部网和因特网的服务程序之间灵活地切换，为动态商务伙伴关系的时代提供支持。

(6) 对于独立软件开发商：Microsoft.Net 意味着他们将得到更多的机会，为因特网时代创造更多的新型高级服务——它们可以自动从本地或远程取得和利用所需信息，并通过任何语言 and 任何设备工作，无须为不同的工作环境重新编写程序。因特网上的一切都变成这种新一代服务程序的潜在构造模块，而每一种应用程序都可以在网上使用。

Microsoft.Net 的前景意味着消费者、企业、软件开发商和整个工业的一次解放。它意味着因特网全部潜在能力的释放，以及用户可以在网络上随心所欲的遨游。

1.3 Microsoft.Net 平台建造了新一代因特网

建立在 XML 和因特网协议的标准整合构造上的 Microsoft.Net 平台为开发新型高级软件提供了一个革命性的模型。在此之前，软件开发模式一向侧重于单个系统，甚至试图掩饰与其他系统的互动，使它们看起来像本地系统内部的互动。Microsoft.Net 的设计意图十分明确，要将网上所有可用资源整合成为一个解决方案，而对现有的技术来说，这种整合是极其复杂和昂贵的。Microsoft.Net 将使这一点成为所有软件开发活动的内在本质。

Microsoft.Net 基于 XML 的松散型编程模式引入了一个新概念，就是创造基于 XML 的

网络服务。鉴于今天的网站几乎用手工打造，如果没有突破性的改进，就不能与其他网站协调工作，Microsoft.Net 编程模式提供了一个内在机制，令所有网站和服务商都可以结成同盟，协调工作，就像通配型零部件的引入加速了工业革命的进程一样。Microsoft.Net 承诺：将促进新一代因特网的成长。

如果没有许多商业伙伴和数以万计的独立开发商和企业开发商所共同努力建立起来的计算机工业，所有这一切都是不可能的。正如 Alexander Graham Bell 曾经说过的，“伟大的发现和进步总是众多天才的合作成果”。DOS 在个人电脑上的推广为新一代独立开发商提供了大量的机会，成立公司，开发基于 DOS 的应用程序。Windows 将这些机会带向了一个更高的层次。Microsoft.Net 为独立开发商所提供的机会将是前所未有的。

微软公司将为开发商创造一整套全新的 Microsoft.Net 网络开发工具，横跨客户、服务器和服务软件。这些工具将使开发商能够把今天只能提供静态信息表述的网络改造成一个擅长互动服务的网络。由 Visual Basic 开发系统首创的“拖放式速用开发”为业界树立了典范。微软公司具有突破性的新一代 Visual Studio 工具套装通过这一典范使网络服务程序的开发变得自动化，而且其软件产品在任何接受 XML 标准的平台上均可使用。Visual Studio 甚至可以自动产生 XML 程序编码。微软即将发布一套新的 BizTalk 编程工具，它可以通过组合服务程序，使商业流程变得直观化，让商业分析家也能像开发商一样创造自己想要的解决方案。

Microsoft.Net 编程模型使独立软件开发商可以较少关注所开发的应用软件应该怎样和在何处使用，而更多地关注软件的功能——这才是价值所在。Microsoft.Net 致力于解决开发商面临的一些最主要的挑战，因为他们今天仍然在忙于在软件的功能性和易管理性之间权衡得失。它把 ASP 和程序集成带到了一个新的水平上：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；可以在离线状态下运行它们。

另外，开发商将能够对 Microsoft.Net 核心构造模块服务进行个性化再设计，然后应用到他们自己的软件产品中，从而降低研制新产品的难度。这些 Microsoft.Net 核心构造模块服务着重体现了微软公司所擅长的功能性，并且能够给广大开发商带来更大的价值。微软公司正从很多方面努力，将开发商的构造模块服务统一在 Windows 操作系统中，形成可以在网上实现的类似功能，它使高度分布式可编程服务软件能够在单独的设备、企业数据中心和因特网之间的传送更加容易。

有了订购这些 Microsoft.Net 核心程序的选择，开发商就可以根据各自研发资源的具体情况自由地决定“购买还是自建”。有人也许选择独立研发服务程序，但大多数人将转向购买具有强大开发工具的成品方案，就像他们在开发以 Windows 为平台的应用软件时并不自己编写打印机驱动程序或窗口系统，而将精力放在如何使他们的高级产品更有特色。

Microsoft.Net 核心构造模块服务将由下列部件组成：

(1) 身份认证——它建立在微软 Passport 和 Windows 鉴定技术的基础上，提供从口令、钱包到智能卡，生理数据装置等多级身份鉴定。它使得开发商的产品设计在个性化和保密性方面得到保证，消费者无论使用何种设备时都能享受更高级的安全服务。首先发布的代

号为“Whistler”的 Windows.Net 将支持这一功能。

(2) 通知和消息——它将即时消息、电子邮件、传真、声音邮件和其他形式的通知和消息功能整合在一体化的机构里，供个人电脑和其他智能设备使用。它建立在基于网络的 Hotmail (R) 邮件服务、Exchange 和 Instant Messenger 等软件的基础上。

(3) 个性化——让用户通过制定相关的规则和优先次序，明白地或含蓄地指定以何种方式处理通知和消息、处理希望分享用户个人数据的请求和个人的多个设备之间如何协调（也就是说总是将用户的笔记本电脑和用户的 Microsoft.Net 里储存的所有服务程序内容保持同步），它将使用户向新电脑里输入数据不费吹灰之力。

(4) XML 数据空间——它用一种通用的语言（XML）和协议（SOAP）对数据进行描述，使数据在多个网站和用户之间的传送始终保持整体性。其结果是网站变成可以互动的灵巧服务，交换并利用彼此的数据。Microsoft.Net 同时在网上提供了可以指定地址的安全空间用来储存个人的数据，而且用户的所有设备都能访问这些数据，并以最佳的方式复制数据以提高效率或供离线使用。其他设备在用户的同意下可以访问用户的数据空间。它综合了 NTFS, SQL Server, Exchange 和 MSN Communities 的服务。

(5) 日程安排——时间是实现用户控制的一个重要尺度：人们何时接受访客？何时乐于独处？随着人们花越来越多的时间通过越来越多的设备进行网上沟通，而且用户和服务程序之间的互动越来越紧密，时间安排这个问题日益突出。Microsoft.Net 提供了一个基础构架，在安全和确保隐私的前提下综合用户的工作、社交和在家的时间安排，用户的所有设备和经过用户允许的其他程序和个人都可以得到其中的数据。它建立在 Outlook (R) 讯息和客户协作以及 Hotmail (R) Calendar 的基础上。

(6) 目录和检索——Microsoft.Net 帮你找到与之互动的服务商和个人。Microsoft.Net 的目录不仅是搜索引擎或者“黄页”，它们能以既定的程序与服务商互动，为已经列出的问题寻找答案，了解那些服务商的能力。它们同时可以被其他服务程序集合并风格化，然后与自己的程序捆绑在一起。

(7) 动态传送——它使微软和其他开发商以动态的方式按照用户的要求，传送更多的软件功能和可靠的自动升级补丁，而不必麻烦用户动手安装和配置。Microsoft.Net 能按照用户的要求预先改编，适应用户的任何设备。它倒转了依靠人工安装的传统操作模式。用户将来会通过多个设备享受服务，而这种倒转确实是必须的。

Microsoft.Net 的分布式服务将通过在线和离线两种方式发布。服务程序将通过企业内部服务器或因特网群落调入未与因特网连接的分立设备。在不同的场合下，用户可以通过一个称作“邦联”的步骤进行合作或信息交换，机构用户可以自己决定是运行它们自己的基础设施，还是引用外部资源、同时不放弃对因特网接入的控制。例如，一个企业目录服务可以与某个因特网群落内的服务商结成联盟。这种基于 Microsoft.Net 的服务与基于因特网的服务不可同日而语。

Microsoft.Net 构造模块服务可以在任何支持 XML 标准的平台上运用。Windows 将提供创造和传送网络服务的最佳环境，同时，基于 Windows 的用户将能够以最佳方式将网络服务分布到各种设备上。微软的 Windows DNA 2000 已经提供了第一套支持 XML 的基础体系，

以建立和操作网络服务程序。

1.4 Microsoft.Net 能带给人类什么

1.4.1 .Net 与 .Com 的区别和联系

大家都知道 .Com 是每一个具体的网站，如果说因特网是一个信息海洋，那么，.Com 就是信息海洋中的“数字孤岛”，将这些数字孤岛连接起来，打破不同的上网设备、不同的操作系统、不同的网站以及各大机构和工业界的网络障碍，从而实现因特网的全部潜能，这就是微软公司的 .Net 的核心内容，也是 .Com 与 .Net 的区别和联系。

1.4.2 .Net 战略的核心内容

.Net 的核心内容之一就是要搭建第三代因特网平台，这个网络平台将解决网站之间的协同合作，从而最大限度地获取信息。在 .Net 平台上，不同网站之间通过相关的协定联系在一起，网站之间形成自动交流，协同工作，提供最全面的服务。

举个例子：不久前，微软公司在美国佛罗里达州奥兰多召开全球销售会议，微软中国有限公司的许多人出差去奥兰多参加会议，由于一些同事有比较长的休假，他们决定前去加勒比海乘大型游船去旅游度假。他们先上网去查找相关的旅游网站，然后去找到相关的旅游公司及船运情况介绍，再通过旅行社办理各种手续，最后得以成行。如果在 .Net 完成后，旅游者只需要告诉其目的及相关条件，其余的工作通过 .Net 就自动实现了，这样节省了大量的时间，上网的人也不需要太多的网络技巧，就可以得到相关的服务。

回到现实生活中，如果我们通过 .Net 将每个网上的购买者和每个商家真正匹配起来，这种网络之间的协同配合，就会成为未来商业的关键所在 .Net 将实现这一功能，从而完成历史性的网络平台的搭建。

1.4.3 开发 .Net 对人类的生活和工作将带来什么变化

今天，大家都在谈论电子商务，谈论网络泡沫。有一点很显然，如果中国仅有的 1600 万上网人群都是计算机爱好者和学生，那么中国的网络经济只能是泡沫经济，中国的网络热潮，只能是网络商人们的一厢情愿。

.Net 的出现，将改变这一状况。如果说使用安装 DOS 系统的计算机工作，还是一种技术工作，那么装有 Windows 的计算机就只能算一种工具，从大型计算机到装有 DOS 的 PC，使用电脑的人呈代数级数上升，从 DOS 到 Windows，使用电脑的人呈几何级数上升，而从 Windows 到 .Net，使用因特网的人将以更高的数量级上升。在 .Net 网络平台上，人们可以使用键盘、鼠标、手写、语言等各种方式的人机交流模式，这样将极大降低人们进入网络的门槛，降低用户使用网络的难度，从而为构建全真虚拟网上社会打下基础。

.Net 编程语言 C

本章内容要点

NGWS runtime 技术基础；C # 的特点；如何安装 C # 编译器；如何用 Visual Studio 编辑器编辑 C # 程序。

2.1 NGWS runtime 技术基础

C # 依靠由 NGWS 提供的运行时。因此，有必要知道运行时是如何工作的，以及它背后所蕴含的概念。所以，这一节大致分为两部分，它们是所有的概念和使用的基础。两部分的内容虽然有些重叠，但有助于加深理解正在学习的概念。

2.1.1 NGWS runtime

NGWS 和 NGWS runtime 为用户提供了一种运行时环境。该运行时管理执行代码，并提供了使编程更容易的服务。只要用户的编译器支持这种运行时，用户就会从这种受管理的执行环境中得益。

C # 编译器不仅支持 NGWS runtime，对 VB 和 C ++ 也支持。这些为支持运行时所创建的代码被称作“受管代码”（managed code）。以下是你的应用程序可以从 NGWS runtime 得到的利益：

- 交叉语言集成（通过通用语言规范）；
- 增强安全（包括类型安全）；
- 自动内存管理（垃圾收集）；
- 版本支持（“DLL 地狱”终结者）；
- 交叉语言异常处理（统一展开）；
- 组件交互简化模式。

为了使 NGWS runtime 能够提供所有的这些好处，编译器必须把文件和受管代码一起发出。文件描述代码中的类型，它和用户的代码存在一起。

正如你从很多种交叉语言功能所看到的一样，NGWS runtime 主要是关于高度集成交叉多异编程语言（tight integration across multiple different programming languages）。这种支持甚至到允许用户从一个 VB 对象派生出一个 C # 类的程度。

C # 程序员将会喜欢的一个功能是，他们不必担心内存管理——也就是说不必担心内存泄漏。NGWS runtime 提供了内存管理，当对象和变量的生命期结束（不再被引用）时，垃圾收集器释放它们。

应该鼓励配置一个管理应用程序或者组件。因为管理应用程序含有数据文件，NGWS

runtime 可以利用这些信息，以确保用户的应用程序具有它所需的各种规定版本。所产生的明显效果为，由于用户的代码没有相互之间的依赖，很少可能出现中断。

接下来将讨论下面两方面的内容：

- 中间语言 (Intermediate Language, IL) 和数据；
- 即时编译器 (just-in-time compilers, JITers)。

1. 中间语言和元数据

由 C# 编译器生成的受管代码并不是原始代码，但它是中间语言 (IL) 代码。这种 IL 代码自身变成了 NGWS runtime 的受管执行进程的入口。IL 代码明显的优势在于它是与 CPU 无关的，这也意味着，用户要用目标机器上的一个编译器才能把 IL 代码转换成原始代码。

尽管 IL 代码由编译器产生，但它并不是编译器提供给运行时的仅有的东西。编译器同样产生有关该代码的元数据，它告诉运行时有关该代码的更多的东西，例如各种类型的定义、各种类型成员的签名以及其他数据。基本上，元数据是类型库、注册表内容和其他用于 COM 的信息。尽管如此，元数据还是直接和执行代码合并在一起，并不处在隔离的位置。

IL 和元数据存放于扩展了 PE 格式的文件中 (PE 格式用于 .exe 和 .dll 文件)。当这样的 PE 文件被装载时，运行时从文件中定位和分离出元数据和 IL。

在进一步说明之前，这里先给出已有的 IL 指令的简短目录。尽管它不是一个完整的清单，也不需要你熟记和理解，但是它列出了读者所必需的、C# 程序所基于的知识基础。

- 算术和逻辑操作符
- 控制流
- 直接内存访问
- 堆栈操作
- 参数和局部变量
- 堆栈分配
- 对象模式
- 实例类型值
- 临界区
- 数组
- 分型位置
- 即时编译器

2. 即时编译器

由 C# 或其他能产生受管代码的编译器所生成的受管代码就是 IL 代码。虽然 IL 代码被包装在一个有效的 PE 文件中，但是用户还是不能执行它，除非它被转换成受管原始代码。这就是 NGWS runtime 即时编译器 (也称作 JITers) 要做的工作。

可能用户会对即时编译代码感到厌烦，为什么不把整个 IL PE 文件编译成原始代码？答案是时间——需要把 IL 代码编译成 CPU 规格的代码的时间。这种即时编译将更加有效率，因为有一些程序段从来就没有被执行过。

从技术上说，全部的处理过程如下：当一个类型被装载时，装载器创建一个存根 (stub)，并使它连接每一个类型的方法。当一个方法第一次被调用时，存根把控制交给 JIT。JIT 把 IL 编译为原始代码，且把存根指针指向缓冲了的原始代码。接着的调用将执行原始码。在某些位置上，所有的 IL 都被转换成为原始代码，而 JITers 处于空闲状态。