“九五”国家重点电子出版物规划项目·希望计算机知识普及系列

体验办公新工具丛书（3）

Microsoft C#

语言及程序设计

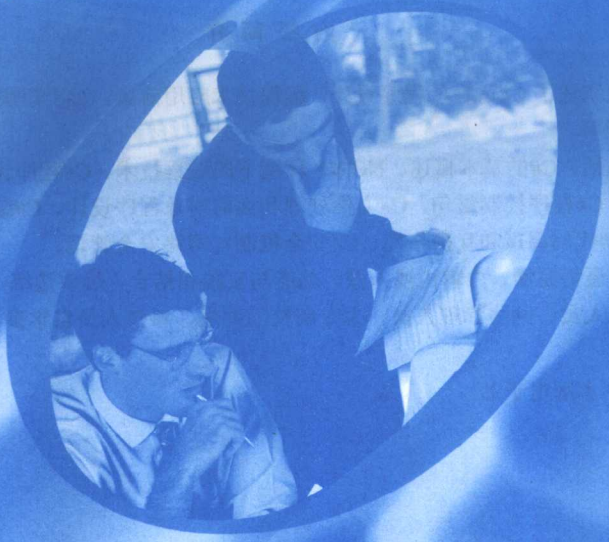
北京希望电子出版社　总策划
单银根　黎连业　编　写

本光盘内容是：本版电子书

北京希望电子出版社
Beijing Hope Electronic Press
www.bhp.com.cn

"九五"国家重点电子出版物规划项目·希望计算机知识普及系列

体验办公新工具丛书（3）



Microsoft

Visual C++ 语言及程序设计

北京希望电子出版社　总策划
单银根　黎连业　编　写

本光盘内容是：本版电子书

北京希望电子出版社
Beijing Hope Electronic Press
www.bhp.com.cn

内 容 简 介

本书深入浅出地叙述了C#语言及程序设计、编程技术应用前景。对于想快速掌握C#编程的软件设计者，本书是你的好帮手。

本书主要内容包括：C#的基本概述、NGWS环境下的编程技术、C#程序设计的基本过程、C#数据类型、C#中使用的类、C#程序控制语句、C#文件处理和面向对象程序设计、C#编写组件的方法、C#异常处理、C#应用配置、C#代码调试和互操作性、C#安全机制与网络安全管理。

本书范例丰富，通俗易懂，内容由浅入深，功能与实践相结合，边讲边练，讲练结合，实用性和操作性强，适合软件开发人员、网络管理人员、大专院校的师生、科技人员自学参考书，也可作为C#程序设计培训班教材。

本光盘内容包括本版电子书。

系列盘书名 ："九五"国家电子出版物规划项目　计算机知识普及系列 体验办公新工具丛书（3）
盘　书　名 ：C#语言及程序设计
总　策　划 ：北京希望电子出版社
文本著作者 ：单银根　黎连业
责任编辑 ：文华
CD制作者 ：希望多媒体开发中心
CD测试者 ：希望多媒体测试部
出版、发行者 ：北京希望电子出版社
地　　址 ：北京中关村大街26号，100080
网址: www.bhp.com.cn
E-mail: lwm@hope.com.cn
电话: 010-62562329,62541992,62637101,62637102,62633308,62633309（发行）
010-62613322-215（门市）　010-62547735（编辑部）
经　　销 ：各地新华书店、软件连锁店
排　　版 ：希望图书输出中心　全卫
CD生产者 ：北京中新联光盘有限责任公司
文本印刷者 ：北京媛明印刷厂
开本/规格 ：787毫米×1092毫米　1/16　16.25印张　296千字
版次/印次 ：2001年7月第1版　2001年7月第1次印刷
印　　数 ：1-8000册
本　版　号 ：ISBN 7-900071-75-X/TP・74
定　　价 ：25.00元（1CD，含配套书）

前　言

C#是当今最流行的程序设计语言之一，它将成为未来的主导程序设计语言，C#提供了现代、简化、完全面向对象、类型安全的下一代Windows和网络应用开发语言。本书通俗易懂、简明扼要地向读者介绍C#编程。书中所讲内容清晰、实用性强，并充分吸取了C#的核心技术和精华，能够在短时间内使读者成为一个经验丰富的C#编程人员。为了帮助读者更好地掌握C#的使用，我们结合国内许多读者了解掌握C、C++的实际使用情况，编写了《C#语言及程序设计》一书。

全书共分十二章，内容包括：C#的基本概述、NGWS运行环境技术基础、C#程序设计的基本过程、C#的数据类型、C#中使用的类、C#程序控制语句、C#结构、文件处理和面向对象程序设计、C#编写组件的方法、C#异常处理、C#应用配置、C#代码的调试和互操作性、C#安全机制与网络安全管理。

具有如下特点：

（1）对C#语言进行了全面的叙述；

（2）由浅入深，通俗易懂，便于自学；

（3）范例丰富，功能与实践相结合，边讲边练；

（4）覆盖面广，可供高等理工科院校的数学系、计算机系、自动化系的学生参考，也可供职工大学、夜大学等各类专业人员阅读；

（5）可供从事IT行业的软件编程人员、网络应用开发人员、科研人员阅读；

（6）可供从事软件教学人员参考。

本书在编写过程中得到许多同志的帮助和支持，尤其是南京大学网络信息中心、中科院计算所（二部）网络研究开发中心的天地写作组对本书的章节安排进行了统筹规划，并对本书提出了许多积极的建议，在此对天地写作组的全体同仁表示衷心的感谢。由于作者水平有限，不当和错误之处在所难免，恳请读者批评指正。

作者

2001.6

于中科院计算所

目　录

第一章　C#的基本概述

在企业开发领域，微软的 C#将会变成用于编写下一代窗口服务（Next Generation Windows Services，简称 NGWS）应用程序的主要语言。本章将介绍 C#的基本概况、C#的主要特性、安装 C#编译器的方法以及讨论了网友关心的.NET 与 C#的有关问题。

1.1　C#的定义

C#（读音为 C-sharp）是微软最新开发的一种新的软件设计程序语言，功能类似 Java。这种语言将作为开发套件 Visual Studio .Net 的关键组成部分。

C#作为一种先进的，面向对象的开发语言，并且能够方便快捷地在 MS 网络平台建立各种应用和建立能够在网络间相互调用的 Web 服务。从开发语言的角度来讲，C#可以更好地帮助开发人员避免错误，提高工作效率，而且同时具有 C/C++的强大功能。

C#是由 C 和 C++派生而来的一种“简单、流行、面向对象、类型安全”的程序设计语言，C#提供了 C 和 C++程序员开发飞速发展的 Web 应用程序所需的强大而灵活的功能。例如更强的安全功能，以及清理应用软件所占电脑存储器的“垃圾收集”功能。C#和 Java 的核心与 C++比较时有着相同的优势和局限，比起 C++，C#将更容易被理解，将来大量.NET 平台的应用将由 C#来开发。

微软表示，这种新程序设计语言是 C++语言的简易版本，用意在大幅简化并加速软件开发过程，以推助微软发布的新一代视窗服务（NGWS）策略。NGWS 的目标是让视窗更全面融入网络，使微软成为主要的网络服务供应者。归纳起来，有关 C#的主要定义如下：

C#是现代的、面向对象的语言，它使开发者快速、简单地为微软.NET 平台建立解决方案。它所提供的框架允许 C#组件变成 Internet 上运行在任何平台上的任何应用可获得的 Web 服务。

C#语言提高了开发者的生产力，同时，消除了可能导致开发成本增加的编程错误。

C#语言为 C/C++开发者提供了快速的 Web 开发环境，同时也保留了 C/C++程序员想要的功能和灵活性。

C#是提高生产性能的程序设计语言。开发者的目的是为了能够以更少的源代码完成更多的事情。因为可以灵活应用在 C 或 C++已经熟练了的技巧，削减开发成本并可缩短开发时间。

C#将在保持 C/C++灵活性的基础上为程序员带来更高效的 RAD 开发方式。它不仅能用于 Web 服务程序的开发，并且还能开发强大的系统级程序。

C#既保持了 C++中熟悉的语法，而且还包含了大量的高效代码和面向对象特性，是 C 和 C++的混合体。C 与 C++用于撰写视窗应用软件，广泛地受到软件开发人员欢迎。

1.2 为什么需要 C#

1.2.1 C#将成为.NET 平台的编程先锋

未来的世界将是一个以网络为中心的世界，这是无庸置疑的。那么，面对这个即将来临的网络世界，微软，这个软件巨人又有着怎样的思考呢？最近，微软公布了它的网络战略计划，计划中表明，微软将以网络为中心，彻底转换产品开发、发布方式，改变产品经营范围。

微软新一代平台的正式名称叫做“新一代 Windows 服务”（NGWS），其正式商标为——Microsoft.Net。在.Net 环境中，微软不仅仅是平台和产品开发者，并且还作为架构服务提供商，以及应用程序提供商开展全方位的 Internet 服务。谈及这个平台中使用的新技术，微软透露，它将在.Net 环境中提供更多新产品和一揽子的全套服务。

微软公司声称在.Net 环境中的突破性改进在于：使用统一的 Internet 标准（如：XML）将不同的系统对接；这是 Internet 上首个大规模的高度分布式应用服务架构；使用了一个名为“联盟”的管理程序，这个程序能全面管理平台中运行的服务程序，并且为它们提供强大的安全保护后台。

.NET 平台包括如下组件：用户数据访问技术。其中包括一个新的基于 XML 的、以浏览器为组件的混合信息架构，叫做“通用画板”；基于 Windows DNA 2000 的构建和开发工具；一系列模块化的服务，其中包括认证、信息传递、存储、搜索和软件传递功能；当然最后还包括一系列驱动客户设备的软件。

微软还特别指出，第三方软件服务供应商将会在.Net 平台的服务提供上扮演一个重要角色。不过在微软的发言中也隐约暗示着它也将亲自涉足这一前景广阔的架构服务市场，特别是：基于 Hotmail、Exchange 和 Instant Messenger 技术的讯息传递业务；包括认证、个性化信息和 XML 数据转换定义的在线数据存储业务；目录服务、日历服务和搜索引擎业务。

但同时，微软也强调，虽然它坚信 Windows 是发布 Web 服务的最佳环境，不过.Net 的各个平台模块将会与任何支持 XML 的平台兼容。这一点也暗示着微软会成为具有类似能力的第三方服务商的强大竞争者。

不过有一点可以确定的是，微软自己的终端用户产品以及服务将会按照他提供的架构服务标准化，他的架构服务将会以新一代 Windows 用户平台——Winodws.Net 为主导，计划在 2001 年的适当时候发布 1.0 版本，而在最早后年发布其升级版本。Windows.Net 和在线服务的紧密集成意味着 Web 浏览功能将会成为客户端应用的内建功能。

对于开发者来讲，Visual Studio.NET 将提供一个基于 XML 的编程模型以及快速开发环境。

1.2.2 C#是企业级程序设计的首选语言

在过去的二十年内，C 和 C++已经成为广泛应用在商用软件的开发中的开发语言。但是 C 和 C++都提供了一些容易使开发者产生错误的特性，也可以说 C 和 C++的灵活性是牺牲了开发效率。如果和其他的开发语言相比（比如说 VB），相同功能的 C/C++软件通常会需要更长的开发周期。正是由于 C/C++开发的复杂性和需要较长的开发周期，所以许多 C/C++开发人员都在寻找一种可以在功能和开发效率间提高更多平衡的开发语言。

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

一种合理的 C/C++替代语言应该是能够提供对现存和潜在的平台上的高效开发提供有效和有力的支持。并可以使 Web 开发可以非常方便的与现存的应用开发相结合。而且 C/C++开发人员都倾向于在必要的时候使用底层代码。

在这个问题上微软的解决方案是推出一种命名为 C#的开发语言。C#是一种先进，面向对象的语言，通过 C#可以让开发人员快速的建立大范围的基于管理系统的 MS 网络平台的应用，并且提供大量的开发工具的服务，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。即使通过简单的 C#构造也可以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开发语言所开发的应用调用。

即使抛开上面所提到的优点，C#也可以为 C/C++开发人员提供快速的开发手段而不需要牺牲任何 C/C++语言的优点。从继承角度来看，C#在更高层次上重新实现了 C/C++，熟悉 C/C++开发的人员可以很快的转变为 C#开发人员。

1.2.3 C#设计考虑了开发效率与安全性

目前的各种基于 Web 应用的软件开发向传统的商业应用软件开发提供了挑战，开发者被组织起来开发具有更短开发周期的各种应用，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多的代码完成相同的功能，并且能够更好的避免错误发生。其效率主要表现在三个方面。

1. 与 Web 开发相结合

新的开发模式意味着需要更好的利用现有的各种 Web 标准，例如 HTML，XML，SOAP（简单对象存取协议）。现存的开发工具是在 Internet 出现前或是未得到充分应用前出现的，所以都不能很好的适应目前 Web 技术的开发需要。

C#开发者可以方便的在 MS 网络平台上扩展自己的应用。C#可以将任何组件转变为 Web 服务，并且可以被运行于 Internet 上的任何平台的任何应用的调用，重要的是 C#对这

一特性提供了内置的支持。

更重要的一点，Web 服务框架可以让任何 Web 服务都看起来类似于 C#的内置对象，所以可以让开发人员在开发过程中继续使用他们已经具备的面向对象的开发方法和技巧。

此外 C#还拥有许多其他特性使自己成为最出色的 Internet 开发工具。例如，XML 目前已经成为网络中数据结构传送的标准，为了提高效率 C#允许直接将 XML 数据映射成为结构。这样的话可以有效地处理各种数据。

2. 减小开发中的错误

即使是优秀的 C/C++开发人员都难于避免在编码过程出现一些常见错误，比如错误的初始化一个变量，而这种错误将有可能导致各种不可以预知的错误，并且难于被发现。如果一旦错误在发现前被投入生产环境，排除这些错误将会付出昂贵的代价。而 C#的先进设计思想可以消除 C/C++开发中的许多常见错误，比如:

垃圾收集机制将减轻开发人员对内存的管理负担。

C#中的变量将自动根据环境被初始化。

变量是类型安全的。

使用 C#将会使开发人员更加轻易地开发和维护各种商业应用。

3. 提供内置的版本支持来减少开发费用

更新软件系统中的组件（模块）将会是一种容易产生错误的工作，在代码修改过程中可能对现存的软件产生影响。为了帮助开发人员处理这些问题，C#在语言中内置了版本控制功能。例如: 函数重载必须被显式的声明（这种情况在 C++和 JAVA 中时常发生），这可以防止代码级错误和保留版本化的特性。另一个相关的特性是接口和接口继承的支持。 这些特性可以保证复杂的软件可以被方便地开发和升级。

总结起来，这些特性可以帮助开发更强壮的软件后继版本和减轻开发费用。

1.2.4 C#具有功能强、易于表现和灵活性

1. 更好地结合商业应用中的流程与软件实现

为了更好实现公司的各种商业计划，在软件系统中必须在商业流程和软件实现有紧密的联系。但是大多数的开发语言都不能轻易的将各种应用逻辑与代码相联系。例如，开发人员会使用各种注释来标明各种类所代表抽象商业对象。C#允许使用在任何对象上使用预定义数据或是经过扩展的元数据。在系统结构中可以使用区域属性（类似 NT 的网络域结构），并且将这些属性添加到类、接口或者其他元素上。开发者可以独立的测试各种元素上的属性。这将会使得一些如同收集区域中对象属性，或者编写自动工具来保证的区域中的类，接口是否被正确定义的类似工作变得简单。

2. 可扩展的协作能力

虽然管理性强，透明型好，类型安全的开发环境对大多数的商业应用都适合，但现实的经验告诉我们一些应用出于执行效率或者与现存的应用接口 API 相结合的原因需要使用

原有的开发方式来进行编码。也正是如此，许多 C/C++开发人员宁愿放弃使用一些可以提高开发效率的开发工具。C#通过下面的方法来解决这些问题:

内置支持 COM 模型和 Windows 平台 API。

允许有限制的使用指针。

在 C#中任何对象都会自动成为 COM 对象，开发者不需要显式的实现 IUnknown 和其他一些 COM 接口，同时也可以方便而自然地使用现存的 COM 对象，而不需要关心这些 COM 对象是否使用 C#开发。

对于使用 C#的开发人员来讲，C#允许开发人员调用 OS 所提供的 API。在经过标记的代码区域内使用指针并手工管理内存分配。这可以让 C/C++开发人员更快地熟悉和转向 C#并且不需要放弃在以前开发中所形成的习惯，而且以前的 C/C++代码依然可以被重用。无论是对于 COM 的支持还是对于 API 调用的支持都是为了为开发人员提供足够的开发控制能力。

1.3 C#的各种特性

C#语言从 C 和 C++演变而来，它是给那些愿意牺牲 C++一点底层功能，以获得更方便和更产品化的企业开发人员而创造的。C#具有现代、简单、面向对象和类型安全的特性。尽管它借鉴了 C 和 C++的许多东西，但是在一些诸如名字空间、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#为用户提供了方便的功能，如垃圾收集、类型安全、版本控制等等。仅有的"代价"就是，代码操作默认是类型安全的，不允许指针。光是类型安全就可以了。但是，如果需要指针，仍可以通过非安全码使用它们，而且当调用非安全码时，不会引起混乱。

作为编程语言，C#具有以下各种至关重要的特性:

- 简单性
- 现代性
- 面向对象
- 类型安全性
- 版本控制性
- 兼容性
- 灵活性

1.3.1 简单性

C#具有比 C++简单易学的特点，这种语言的首要目标就是简单。通过增加必要的特性，同时舍弃了一些特性，对 C#的总体简单性做出了贡献。

在 C#中，没有 C++中流行的指针，默认情况下，可以使用一种可操控的（managed）代码进行工作。此时，不允许进行不安全的操作，如直接的内存操作。

与指针密切相关的是 madness 的操作。在 C++中，有::、.、和 ->这样的操作符，它们分别用于名字空间、成员和引用等操作。对于一个初学者来说，操作符是学习中的一道难关。

C#去掉了其它操作符，只使用“.”操作符。编程人员所需要理解的就是嵌套名字的注解。

C#使用统一的类型系统，去掉了 C++多变的类型系统，这样编程人员就不必去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。这种类型系统允许你把各种类型作为一个对象查看，不管它是一个原始类型还是一个 full-blown 类。与其它编程语言相比，把单个类型当作对象看待并不会增加执行上的难度，因为它提供了一个叫做加框（boxing）和消框（unboxing）的机制。这个技术仅在需要时才让一个对象去访问单个类型。

C#中整型和布尔型是两种完全不同的数据类型，这就意味着 if 判别式的结果只能是布尔数据类型，如果是别的类型编译会出错。

C#同时解决了 C++中冗余问题。这种冗余包括常数（const）和预定义（#define）、各种字符类型等等。

1.3.2 现代性

在学习 C#上所付出的努力是一项巨大的投资，因为 C#是为编写 NGWS 应用程序的主要语言而设计。你将会发现很多在用 C++中很费力实现的功能，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，新增的货币数据类型很受欢迎。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别，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，指针不再是编程武器的一部分。所以当得知不用再负责整个内存的管理时不应该感到惊讶，全面的内存管理已经不是您的任务。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的（运行时的另一个功能）。在没有 C#之前，必须处理怪异的 HRESULTs，现在由于使用了基于异常的出错处理，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。元数据是 NGWS 运行时的一个关键的概念。

1.3.3 面向对象

C#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 将在下面描述。

对象模式只是基础的一部分，不再是编程语言的一部分。

你可能从一开始必须关注的事，就是不再有全局函数、变量或者常量。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）或静态成员（通过数据类型）。这些使C#代码更加易读且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，而且还确保正确版本的控制。

使用C++编写类，您可以使用访问权限（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。

曾经创建了多少个类是从多基类派生出来的（ATL程序员，您的投票不计在内！）？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。如果觉得需要多重继承，可以运用接口。

1.3.4 类型安全性

类型安全可以选指针作为一个例子。在C++中拥有一个指针，您能自由地把它强制转换成为任何类型，包括把一个int*（整型指针）强制转换成一个double*（双精度指针）。只要内存支持这种操作就行。这并不是您所想象的企业级编程语言的类型安全性。

因为上述原因，C#实施最严格的类型安全，以保护自己及垃圾收集器（garbage collector）。所以必须遵守C#中一些相关变量的规则：不能使用没有初始化的变量。对于对象的成员变量，编译器负责清零。而局部变量，则由您负责清零。如果使用一个没有初始化的变量时，编译器提醒您怎么做。优点是能够避免由于使用不经初始化的变量计算结果而导致的错误，而您还不知道这些奇怪的结果是如何产生的。

C#取消了不安全的类型转换。不能把一个整型强制转换成一个引用类型（如对象），而当向下转换时，C#验证这种转换是正确的。（也就是说，派生类真的是从向下转换的那个类派生出来的。）

边界检查是C#的一部分。再也不会出现这种情况：当数组实际只定义了n-1个元素，却超额地使用了n个元素。

算术运算有可能溢出终值数据类型的范围。C#允许在语句级或应用程序级检测这些运算。在允许检测溢出的情况下，当溢出发生时将会抛出一个异常。

在C#中，被传递的引用参数是类型安全的。

1.3.5 版本控制技术

几乎所有的程序员都至少有一次不得不涉及到众所周知的“DLL 地狱”。该问题起因于多个应用程序都安装了相同DLL名字的不同版本。有时，老版本的应用程序可以很好地和新版本的DLL一起工作，但是更多的时候它们会中断运行。现在的版本问题真是令人头

痛。NGWS runtime 将对您所写的应用程序提供版本支持。C#可以最好地支持版本控制。尽管 C#不能确保正确的版本控制，但是它可以为程序员保证版本控制成为可能。有这种支持，一个开发人员就可以确保当他的类库升级时，仍保留着对已存在的客户应用程序的二进制兼容。

1.3.6 兼容性

C#并没有存在于一个封闭的世界中。它允许使用最先进的 NGWS 的通用语言规定（Common Languae Specifiation，简写为 CLS）访问不同的 API。CLS 规定了一个标准，用于符合这种标准的语言的内部之间的操作。为了加强 CLS 的编译，C#编译器检测所有的公共出口编译，并在通不过时列出错误。

当然，您也想能够访问旧一点的 COM 对象。NGWS 运行时提供对 COM 透明的访问。OLE 自动化是一种特殊的东西。任何一个使用 C++创建 OLE 自动化项目的的人已经喜欢上各种各样的自动化数据类型。C#支持这些数据类型。

另外，C#允许用 C 风格的 API 进行内部操作。可以从应用程序访问任何 DLL 中的入口点。用于访问原始 API 的功能称作平台调用服务（Plaform Invocation Services，缩写 PInovke）。

1.3.7 灵活性

程序员可能会问："难道就没有传递指针的 API 吗？"不是仅有少数的这种 API，而是很多（有点保守的估计）。这种对原始 WIN32 代码的访问有时导致对非安全类指定指针的使用（尽管它们中的一些由于受 COM 和 PInvoke 的支持可以解决）。

尽管 C#代码的缺省状态是类型安全的，但是可以声明一些类或者仅声明类的方法是非安全类型的。这样的声明允许使用指针、结构，静态地分配数组。安全码和非安全码都运行在同一个管理空间，这意味着当从安全码调用非安全码时不会有什么事情是不可能的。

1.3.8 C#的弱点

尽管 C#有许多至关重要的特性，但这个方案也暴露了微软对 C#和.NET 介绍中的一些有趣的漏洞。微软中间语言 MSIL 是.NET 提供的一个新特性，允许很流行的程序设计语言编译到一个单独的公用语言。(.NET 支持的语言种类是相当惊人的)。这些语言都要服从一种叫"通用语言规范（CLS）"的构架，微软称之为"CLS 兼容语言和类库之间可互操作的通用语言"。

编译所有的语言到一个单一的公用语言上，能让"继承"通过多重语言真正执行。这轻而易举的扫除了 C#可能遇到的错误概念。.NET 组件使用 COM 的 IDispatch，它只允许接口执行。它比先前 C#被评定为一种 OOP 语言的看法容易接受得多。它在程序设计上或许像 Java 一样是面向对象的。

但不幸的是 CLS 这种包括 MSIL 的共享语言基础，只让 RAD 开发者受益，而损害了

硬件的核心开发者，有人认为创造一种新程序设计语言的目的就是有能力充分运用它和服务于可微调的执行能力，这一点在 CLS 世界里是做不到的。加速充分利用从来不是许多语言的唯一目标。许多语言的唯一目标（最瞩目的是像 Visual Basic 和 Java 的 RAD 语言）是加速和美化开发和展开能力，而不仅仅是运行时刻的速度。

C#将把微软领向何方就一目了然了。因为所有项目编写会只依靠 MSIL 和 CLS JIT 编译程序。这样 C#或任何 MSIL 前端语言比 Java 任何时候都快。但很不幸，程序设计和编译程序级的优化不能在非微软的平台上充分利用，想在非 Windows 平台上展开.NET，再充分运用它们也是不现实的。

1.4 如何安装 C#编译器

C#的最小系统要求是 Microsoft Windows 2000， Microsoft Internet Explorer 5.5，但是 Windows 98 下也可以编译 C#文件，我们可以在 Windows 98 下安装 NGWS SDK for Windows 2000，你可以从下面连接下载 NGWS SDK:

http://download.microsoft.com/download/platformsdk/Trial/1812.10full/NT5/ENUS/Setup.exe

将下载好的 setup.exe 文件用 Winzip 打开。执行其中的 prejit.exe, comredist.msi, comsdk.msi 这三个文件即可，安装完毕后，你会找到一个叫 csc.exe 的文件，它是编译 C#程序的编译器。

csc.exe 编译器的操作命令如下:

```
    - OUTPUT FILES -
/out:<file>          Output file name (derived form first source file if not specified)
/target:exe  Build a console executable (default) (Short form: /t:exe)
/target:winexe     Build a windows executable (Short form: /t:winexe)
/target:library     Build a library (Short form: /t:library)
/target:module     Build a module that can be added to another assembly (Short form:
    /t:module)
/win32icon:<file>       Use this icon for the output
/nooutput [+|-]    Only check code for errors; do not emit executable.
/defint:<symbol list>   Define conditional compilation symbol (s) (Short form: /d)
/doc:<file> XML Documentation file to generate
    - INPUT FILES -
/recurse:<wildcard>    Incude all files in the current directory and subdirectories according
    to the wilcard specifications
/main:<type>     Specifies the type that contains the entry point (ingore all other
    possible entry points) (Short form: /m)
/reference:<file list>    Refernce metadata form the specified assembly files (Short form: /r)
```

```
/addmodule:<file list>  Link the specified modules into this assembly
      - RESOURCES -
/resource:<resinfo>     Embeds the specified resource (Short form: /res)
/linkresource:<resinfo>     Links the specified resource to this assembly (Short form: /linkres)
      - CODE GENERATION -
/debug [+|-]     Emit debugging information
/optimize[+|-]   Enable optomizations (Short form: /o)
/incremental [+|-]     Enable incremental compiation (Short form: /incr)
      - ERRORS AND WARNINGS -
/warnaserrror[+|-]     Treat warnings as errors
/warn:<n>  Set warning level (0-4) (Short form: /w)
/nowarn:<warning list>     Disable specific warning messages
      - LANGUAGE -
/checked [+|-]   Generate over/underflow checks
/checked [+|-]   Generate over/underflow checks
/unsafe [+|-]    Allow' unsafe' code
      - MISCELLANEOUS -
@<file>    Read response file for more options
/help Display this usage message (Short form: /?)
/nologo    Suppress compiler copyright message
      - ADVANCED -
/baseaddress:<address>     Base address for the library to be built
/Win32res:<file> Used to specify a WIN32 resource file for version and icon      information
/bugrpeort:<file> Create a'Bug Report' file
/codepage:<n>   Specifies the codepage to use when opening source files
/fullpath    Compiler generates fully qualified paths
/nostdlib[+|-]    Do not reference standard library (mscorlib.dll)
```

1.5 网友关心的有关.NET 和 C#问题

1. .NET 是运行库呢还是开发平台?

都是! 并且微软的宏伟目标是让 Microsoft.NET 彻底改变软件的开发方式、发行方式、使用方式等等，并且不止是针对微软一家，而是面向所有公司! ".NET 架构"是 Microsoft.NET 计划中首先问世的一部分，它包括了两方面的组件: ".NET 通用运行库"和 ".NET 类库"。最近传来好消息说这两个组件已经被打包到 ".NET 架构 SDK"中，放在微软的站上免费供大家下载，有兴趣的朋友一定要去试试看! 另外，这个 SDK 中还包括 C#、