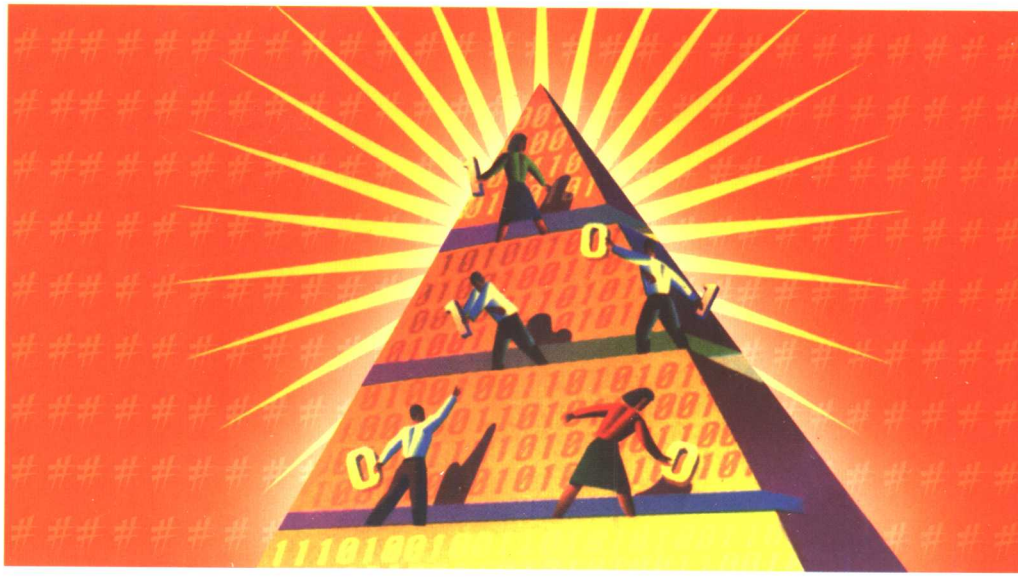
C#.NET

程序设计教程

郭 胜　秦 岸　马 丽　编著



清华大学出版社

C#.NET 程序设计教程

郭胜　秦岸　马丽　编著

清 华 大 学 出 版 社

（京）新登字 158 号

内 容 简 介

C#是一种类型安全的、现代的、简单的，由 C 和 C++衍生出来的面向对象的编程语言，它牢牢根植于 C 和 C++语言中，并可以很快被 C 和 C++的使用者所熟悉。

本书分为 8 章：第 1 章讲述了 C#的产生背景及其主要特点，第 2 章讲述了 C#的基础知识，第 3 章讲述了 C#的异常处理，第 4 章讲述了使用 C#开发应用程序，第 5 章讲述了使用 C#开发 Windows 应用程序，第 6 章讲述了 C#的数据库运用，第 7 章讲述了 C# Internet 的高级编程，第 8 章讲述了 C#与 XML 的运用。

如果读者对 C++有一定程度的了解，那么学习 C#会很容易。读者可以将本书作为参考手册，随时查阅。如果读者熟悉其他面向对象语言，例如 Java、Delphi 等，熟悉了面向对象的思想，就只需要学习如何使用 C#实现运用开发；对于初步涉及到该领域的读者，本书可以作为系统学习 C#语言的教材，从而帮助读者掌握扎实的 C#语言基础知识，建立面向对象的编程思想。

图书在版编目（CIP）数据

C#.NET 程序设计教程/郭胜，秦岸，马丽编著.—北京：清华大学出版社，2002.9

ISBN 7-302-05690-0

Ⅰ.C… Ⅱ.①郭… ②秦… ③马… Ⅲ.C 语言-程序设计 Ⅳ.TP312

中国版本图书馆 CIP 数据核字（2002）第 050055 号

出 版 者：清华大学出版社（北京清华大学学研大厦，邮政编码：100084）
http://www.tup.tsinghua.edu.cn
责任编辑：林庆嘉
印 刷 者：世界知识印刷厂
发 行 者：新华书店总店北京发行所
开　　本：787×1092 1/16　**印张**：20.25　**字数**：501 千字
版　　次：2002 年 9 月第 1 版　2002 年 9 月第 1 次印刷
书　　号：ISBN 7-302-05690-0/TP・3352
印　　数：0001~4000
定　　价：28.00 元

前　言

在过去的 20 年中，C 和 C++已经是开发商用软件和企业软件时使用最为广泛的编程语言之一。这两种语言为开发者提供了大量细致灵活的控制，这种灵活性是以生产的成本为代价的。同 VB、VC 相比较，C 和 C++应用程序相对来说需要较长的开发时间。由于复杂性和较长的开发周期，许多 C 和 C++程序员都已经在寻找一种能在功能和生产力之间提供更好均衡的编程语言。有几种语言是通过牺牲 C 和 C++程序员常常需要的灵活性来提高生产力的。这样的解决方案对开发者的约束太多，并且通用性很差。它们与先前存在的系统很难相互操作，并且它们与当前的 Web 设计方法不能很好地吻合。

对于 C 和 C++程序员来说，理想的解决方案是快速的开发与访问所有潜在平台的能力相结合。开发环境应该完全与新的 Web 标准同步并容易与现在的应用系统集成。同时，C 和 C++程序员还希望能够在必要的时候在底层编写代码。

微软公司为了解决上述问题，提出了一种叫 C#的语言（英文中读作：C sharp）。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公司新的.NET 平台上快速建立广泛的应用程序。其提供的工具和服务能充分发挥系统的计算和通信能力。

C#和 C、C++的差别在于 C#有更简单的语法、紧密集成的组件和函数、类型安全等。它将比 Java 更适合融入 Microsoft 的技术。C#在增加生产力和可靠性功能方面仿照了 Java 的做法，比如为防止在各种变量之间错误地搭配而采取的强大的类型转换，以及采集无用信息来自动地进行内存分配。

本书分为 8 章，第 1 章介绍了 C#语言的新特性及其与其他程序设计语言，特别是 C 和 C++的比较，同时也介绍了 Microsoft.NET Framework SDK 中的编译工具和 Visual Studio 7.0 开发软件包；第 2 章介绍了 C#的语法基础知识，简单扼要地使用实例对 C#语言基础知识进行了示范，程序设计语言的掌握水平往往是在大量实例的练习中不断提高的；第 3 章介绍了 C#语言中的异常处理，通过实例程序来分析怎样捕获异常，怎样处理异常；第 4 章介绍了开发应用程序，主要介绍的是 C#的多线程处理，以及进程的运用；第 5 章介绍了开发 Windows 应用程序，通过一个综合实例讲述了使用 C#来设计 Windows 应用程序以及界面设计，通过该实例读者会体会到使用 C#也能够开发功能强大的应用程序；第 6 章介绍了 C#的数据库运用，每种程序设计语言对数据库的处理都是比较重要的部分，在本章中详细介绍 C#对数据库的访问，对远程数据库、本地数据库的访问，也讲述了 ADO.NET 和 ASP.NET 结合 C#对数据库的运用；第 7 章介绍了 Internet 的高级编程，讲述了 NET 基类，网络高级编程的运用。第 8 章讲述了在 Microsoft.NET 中对 XML 提供了广泛的支持，无论是程序员编写的代码，或是应用程序的发布，都离不开 XML。Microsoft 实际上把 XML 作为了.NET 框架的基础和核心。

全书在讲述 C#的知识时，提供了大量的实例程序。读者可以结合实例来学习 C#的基本概念和实际运用，在此基础上设计自己的应用程序。

本书的编写人员为郭胜、秦岸、马丽等，冯明茏、曾雨苓、蒋静、李秋菊、宋玉霞、缪军、杨治国、王巨、晏国英、严英怀、肖庆、付膺豪、刘吉香、涂正伟、江仕亮、郭光通等也参与了本书的部分章节的写作、插图、录入和校对工作。另外感谢那些在此书编写过程中给予大力支持的各位专家教授及各位朋友。

编　者

2002 年 1 月

目　录

第 1 章　C#简介 1
1.1　.NET 和 C# 2
1.1.1　.NET 的未来 2
1.1.2　为什么我们要使用 C# 2
1.1.3　C#和 C++的主要不同点 3
1.1.4　C#语言的特点 3
1.2　C#的运行环境 5
1.2.1　.NET SDK 的安装 5
1.2.2　Microsoft Visual Studio 7.0 的安装 6

第 2 章　C#语言基础 7
2.1　“Welcome”程序 8
2.2　类型 10
2.2.1　值类型 10
2.2.2　引用类型 14
2.2.3　类型转换 15
2.3　变量 17
2.4　表达式 18
2.4.1　表达式类型 18
2.4.2　简单表达式 19
2.4.3　操作符 20
2.4.4　函数成员 24
2.4.5　语句 26
2.5　数组 36
2.5.1　抽象基类 System.Array 36
2.5.2　数组的初始化 36
2.6　枚举 40
2.6.1　System.Enum 基类 41
2.6.2　枚举的声明 42
2.6.3　枚举成员 42
2.7　代表 44
2.7.1　System.Delegate 类 44
2.7.2　代表的声明 45

2.7.3 代表与事件.....47
2.8 接口.....50
2.8.1 接口声明.....50
2.8.2 接口成员.....51
2.8.3 接口实现.....55
2.8.4 显式接口成员实现.....57
2.8.5 接口映射.....59
2.8.6 接口重实现.....61
2.8.7 抽象类和接口.....63
2.9 结构.....64
2.9.1 结构声明.....64
2.9.2 结构的构造函数和继承.....66
2.9.3 结构和特性.....66
2.10 类.....68
2.10.1 类声明.....68
2.10.2 类成员.....69
2.10.3 常量.....71
2.10.4 字段.....73
2.10.5 方法.....75
2.10.6 属性（Properties）.....86
2.10.7 事件（Events）.....91
2.10.8 索引（Indexers）.....93
2.10.9 操作符.....95
2.10.10 实例构造函数（Instance Constructors）.....96
2.10.11 静态构造函数（Static Constructors）.....97
2.10.12 析构函数（Destructors）.....100
2.10.13 .NET 的基类.....100
2.10.14 基类查看工具 WinCV.....100

第 3 章 异常处理.....102
3.1 什么是异常.....103
3.2 捕获异常.....103
3.2.1 校验(checked)和非校验(unchecked)语句.....103
3.2.2 给溢出校验设置编译器.....104
3.2.3 在语句中设置溢出检查.....104
3.3 异常处理语句.....105
3.3.1 使用 try 和 catch 捕获异常.....106
3.3.2 使用 try 和 finally 清除异常.....108
3.3.3 使用 try-catch-finally 处理所有异常.....110

3.4 Runtime 提供的标准异常 111

第 4 章 应用程序开发 112
4.1 线程 113
4.1.1 Thread 类 113
4.1.2 线程的使用 113
4.1.3 线程的同步 117
4.1.4 线程池 120
4.1.5 C#中特有的线程功能 123
4.2 进程 125
4.2.1 启动、停止进程 125
4.2.2 获取进程信息 126
4.3 创建组件 129
4.3.1 新建一个动态链接库（DLL） 129
4.3.2 调用一个动态链接库（DLL） 130
4.3.3 调用外部 DLL 库 131

第 5 章 C#设计 Windows 程序 133
5.1 创建一个新项目 134
5.2 常用控件 135
5.2.1 使用菜单 135
5.2.2 使用工具栏 139
5.2.3 文本编辑框 143
5.2.4 状态栏 144
5.2.5 对话框 146
5.3 综合实例 150

第 6 章 数据库运用 161
6.1 数据库控件 162
6.1.1 DataSet 控件 162
6.1.2 DataView 控件 162
6.1.3 OleDbConnection 控件 162
6.1.4 OleDbCommand 控件 162
6.1.5 OleDbDataAdapter 控件 163
6.1.6 SqlConnection、SqlCommand、SqlDataAdapter 控件 163
6.2 数据操作 163
6.2.1 数据源连接 163
6.2.2 数据存取 165
6.2.3 数据显示 176

6.2.4 修改数据库中的记录 180
6.2.5 输入数据的合法性验证 188
6.3 ADO.NET 编程基础 188
6.3.1 ADO.NET 特点 188
6.3.2 ADO.NET 的常用对象 190
6.3.3 ADO.NET 编程实例 191
6.4 数据库综合实例 194
6.4.1 自定义字段标题及查询功能 194
6.4.2 添加数据功能 200
6.4.3 删除功能 201
6.4.4 修改数据功能 204
6.4.5 程序优化和异常处理 214

第 7 章 Internet 高级编程 224
7.1 Net 类 225
7.1.1 请求/响应层 225
7.1.2 应用协议层 227
7.1.3 运输层 229
7.1.4 应用举例 229
7.2 套接字编程 234
7.3 创建 Web 留言本 240
7.4 发送邮件 242

第 8 章 C#与 XML 247
8.1 XML 快速入门 248
8.1.1 XML 声明 248
8.1.2 XML 注释 249
8.1.3 XML 元素 249
8.1.4 XML 浏览器 250
8.1.5 XML 样式单 251
8.2 利用 C#产生 XML 程序文档 253
8.2.1 生成程序文档的 C#编译命令 253
8.2.2 C#中的预定义 XML 元素 253
8.2.3 编译器对 XML 文档的处理 266
8.3 .NET 框架中的 XML 编程 269
8.3.1 DOM 模式 269
8.3.2 Push 模式 271
8.3.3 Pull 模式 271
8.4 XmlDocument 类 272

8.4.1 使用 XmlDocument 类创建 XML 文档.....272
8.4.2 使用 XmlDocument 类修改 XML 文档.....274
8.4.3 使用 XmlDocument 类遍历 XML 文档.....275
8.5 XmlTextReader 类.....276
8.6 XmlTextWriter 类.....279
8.7 XslTransform 类.....282
8.8 综合实例——XML 聊天室.....287
8.8.1 聊天室主要功能模块和数据文件.....287
8.8.2 实现站点访问量的统计 global.asax.....289
8.8.3 创建聊天室首页 chathome.aspx.....291
8.8.4 创建新用户注册页面 registernew.aspx.....296
8.8.5 创建用户聊天页面.....299
8.8.6 处理异常和错误.....312

第1章

C#简介

.NET 和 C#
C#语言的特点
C#的运行环境

这一章将把你引进 C#.NET 的天地。我们首先谈论.NET 的未来，为什么我们要使用新一代的编程语言——C#，C#语言的特性是什么，C#与 C++主要有何不同，为什么 C#使开发更容易而且还使我们感到很有趣，最后将讲述如何安装在.NET 平台上使用 C#语言开发.NET 应用程序的开发环境。

1.1 .NET 和 C#

1.1.1 .NET 的未来

.NET 平台终于和大家见面了，我们先来讨论一下.NET 的未来。当一个开发者开始开发一个软件的时候，他所要求或需要的是适当的编程语言知识来将商业逻辑写成程序，那么他将考虑下面的问题：

- 数据库知识分析商业数据。
- 在分布式环境中将商业逻辑连接到数据库的工具。
- 懂得如何将应用程序打包发布。
- 如果需要在客户端提供用户透明性则需要设计 Web 解决方案。
- 希望得到一些组件来复用其他人辛勤劳动的成果或者他人高效的解决方案。
- 希望能够通过脚本或服务器端的逻辑来完成更新客户端的任务。
- 希望能够授权一个团体来重用你的组件以便促进你的业务或工作。
- 希望提供一个简洁友好的用户界面。

但上面提到的几点都是所有开发者需要认真考虑的问题。所以从技术上讲，在.NET 到来以前，我们知道对这些问题可以通过以下方法来解决：

- 使用适当的编程语言知识来将商业逻辑写成程序如：C、C++、Delphi 和 Java 等。
- 使用数据库如：ORACLE、SOL Server、DB 等。
- 使用在分布式环境中将商业逻辑连接到数据库组件的工具如：CORBA、DCOM 等。
- 使用为分发应用程序打包发布的安装程序，二进制文件以及最终注册组件。
- 提供在线解决方案如：HTML、DHTML、CSS 和 XML 等。
- 希望使用软件重用机制：Java Beans、COM 等。
- 使用客户端更新脚本如：Java Script、JScript、VBScript、Applets 等和服务器端商业逻辑如：Perl、ASP、PHP 等。
- 组件发售：进行物理拷贝和注册组件或完全通过服务器访问。

但是对于.NET，微软公司在一个软件包里依靠几种拥有明显界线的技术集成了几乎所有的东西，这意味着你拥有了所有的东西但不会把它们混淆起来。因此在开始学习.NET 时，你不是要学习所有的东西，而是要学习很具有代表性的主要技术如：C#、ASP.NET、XML 等。

1.1.2 为什么我们要使用 C#

为什么我们要选择 C#语言作为编程语言呢？其实市场就是最好的回答：“在企业计算

机领域中，C#将变成为用于编写‘下一代窗口服务’（Next Generation Windows Services，简写为 NGWS）应用程序的主要语言。”

C#可以用来编写客户端更新的脚本，创建商业逻辑，做服务器端编程，编写 Windows 应用程序，编写控制台程序，做组件设计。而且它还可以将 XML 作为数据、元数据（自描述的数据）、商业逻辑和数据库之间的连接工具，但同时你需要 ADO.NET 协助 C#实现上述功能。更进一步的是如果要将内容发布到网上，你需要在服务器端安装 ASP.NET。如果要创建可重用的 Web 组件，你需要 Web Services 的概念，就像 VC++等应用软件开发的 Windows Services 一样。但是 C#却没有物理上的局限，所以我们从程序员和开发者的角点来看选择 C#作为开发语言，这是一个高度的成功。

1.1.3　C#和 C++的主要不同点

C#语言自 C/C++演变而来。但是它现代、简单、完全面向对象和类型安全。如果你对 C/C++语言比较熟悉，那么学习 C#的曲线将会很平坦。许多 C#语句直接借用 C/C++语言的特点，包括表达式和操作符等。假如不仔细看，简直会把它当成是 C++语言。

对于 C#最重要的一点是：它是现代的编程语言。它简化和现代化了 C++的类、名称空间、方法重载和异常处理等领域。屏蔽了 C++的复杂性，使它更易用、更方便、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。特别是对于企业开发者来说，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），等等。所有这些功能的目标都是瞄准了开发面向组件的软件。

C#具有而 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实用。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。工作在受管理的代码中，在那里不允许如直接存取内存等不安全的操作。在 C++中，如有“::”、“.”、“→”等操作符，它们将用于名称空间、成员和引用。对于新手来说，这些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。在 C#中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，屏蔽了 C++多变的类型系统。这种系统允许你把各种类型作为一个对象查看，它是一个原始类型也是一个 Full-Blown 类。和其他编程语言相比，由于加框（Boxing）和消框（Unboxing）的机制，把简单类型当做对象处理并不能获得性能的改善。在以后的章节将详细解释加框和消框，但基本上仅当需要时才使用对象访问简单类型这种技术。

1.1.4　C#语言的特点

1. 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的。你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#中不过是一部分很基本的功能而已。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。你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，

根据你应用程序的特殊需求，可以自己很容易地创建出新的适合企业本身的一种数据类型。这为企业开发程序带来了很大的方便。指针不再是你编程武器的一部分，因为全面的内存管理已经不是你的任务，而是在运行时 NGWS 提供了一个垃圾收集器，专门负责 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，异常处理的已经不是新的东西，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，你必须处理异常，但现在由于使用了基于异常的健壮出错处理，这一切都已经结束。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。

2. 面向对象

你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（Virtual Object System，简写为 VOS）的上层。对象模式只是基础的一部分，不再是编程语言的一部分。在其他语言中我们比较关心的是全局函数、变量或者常量。C#把这些东西都封装在类中，包括事例成员（通过类的事例——对象可以访问）或静态成员（通过数据类型访问）。这些使 C#代码更加易读，并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），这样可消除由于偶尔改写方法而导致另外一些原码出错。如果要改写方法，必须具有显式的虚拟标志。这种方法不但缩减了虚拟函数表，而且还确保了正确版本的控制。在使用 C++编写类时，你可以使用访问权限（access modifiers）给类成员设置不同的访问等级。在 C#语言中同样支持 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“类”中说明。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。

3. 类型安全

在 C++中拥有一个指针，你能自由地把它强制转换成为任何类型，包括可以诸如把一个 int*（整型指针）强制转换成一个 double *（双精度指针）这样的傻事。只要内存支持这种操作，它就“可以”。这并不是你所想象的企业级编程语言的类型安全。C#实施最严格的类型安全，以保护自己及垃圾收集器（garbage collector）。所以在使用时必须遵守 C#中一些相关变量的规则：你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，则必须由你负责清零。当你使用一个没有初始化的变量时，编译器会教你怎么做。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，而使你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，派生类真的是从上往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，你却超额地使用了 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，当溢出发生时产生一个异常。在 C#中，被传递的引用参数是类型安全的。

4. 兼容

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 Language Specification，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，你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 自动化是很特殊的，任何一个使用 C++创建 OLE 自动化项目的人已经喜欢上各种各样的自动化数据类型。有个好消息就是 C#支持它们，而没有繁琐的细节。

C#允许你用 C 原型的 API 进行内部操作，可以从你的应用程序访问任何 DLL 中的入口点（有 C 的原型）。用于访问原始 API 的功能称做平台调用服务（Platform Invocation Services）。

5. 灵活

尽管 C#代码的默认状态是类型安全的，但是你可以声明一些类或者仅声明类的方法是非安全类型的。这样的声明允许你使用指针、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，这样暗示着当从安全码调用非安全码时不会陷入列集（marshaling）。

1.2　C#的运行环境

开发.NET 程序需要.NET SDK，这个开发包里有命令行编译器 CSC.EXE，用它来编译 C#源程序；或者是使用微软开发的 Microsoft Visual Studio.NET 7.0，它提供了对.NET 应用程序开发的全面支持。C#是 Microsoft Visual Studio.NET 7.0 中的一个组成部分，在 Microsoft Visual Studio.NET 7.0 中还包括几种用来开发下一代窗口服务平台的编程语言如：Visual Basic、Visual C++、VBScript 等。Microsoft Visual Studio.NET 7.0 提供了一个普通的执行引擎和一个丰富的类库。

C#同 C++程序一样，可以使用任何文本编辑器来编写 C#源程序，然后在有.NET SDK 库支持的环境下调试和编译。如果你只是简单地调试 C#程序，就直接使用 .NET SDK 提供的编译器 CSC.EXE 就可以了，但是如果要开发 Windows 程序最好就使用 Visual Studio 7.0 的集成开发环境来调试和编译程序。

1.2.1　.NET SDK 的安装

如果只是学习 C#语言，就没有必要安装庞大的 Microsoft Visual Studio 7.0，而直接使用.NET SDK 来调试 C#程序，在 .NET SDK 中自带了一个命令行编译器和两个调试工具。微软公司提供了.NET SDK 的下载和免费安装，可以直接到微软网站（http://download.Microsoft.com/download/VisualStudionet /Install/2204/NT5/ENUS/setup.exe）下载安装，可以运行在 Windows 2000、Windows NT 4.0、Windows 98 和 Windows Me 操作系统上，在安装完后就可以在命令行下编译 C#源程序。

1.2.2 Microsoft Visual Studio 7.0 的安装

与.NET SDK 自带的命令行调试工具相比，Microsoft Visual Studio 7.0 的集成开发环境的功能要强大得多，但是它对系统的要求要高得多。

1. Microsoft Visual Studio 7.0 的系统要求

安装 Microsoft Visual Studio 7.0 系统的计算机硬件配置要求有以下几个方面。

- 处理器：至少 Pentium II 450 MHz（推荐：PentiumIII 733MHz 或以上）。
- 内存：至少 128MB（推荐：256MB）。
- 可用磁盘空间：至少 3GB。
- 显示器：800×600，256 色（推荐：16 位真彩色或以上）。
- 操作系统：微软公司的 Windows 2000、Windows NT 4.0、Windows Me 或 Windows 98。

2. Microsoft Visual Studio 7.0 的安装过程

首先安装 Microsoft Visual Studio 7.0 的支持环境，安装方法如下：

(1) 插入安装盘的第一张，安装向导文件将自动检测当前的系统是否安装了 Windows 2000 SP2 或 Windows 98 Second Editon SP2 和 IE 6.0。如果系统没有安装 SP2 或 IE 6.0 则安装向导文件将提示要求用户安装，此时要求用户插入第四张安装盘，安装向导文件将按要求自动安装。

(2) 在安装完支持环境以后，将安装 Microsoft Visual Studio 7.0 了，插入第一张安装盘，系统将自动启动安装文件。因为我们使用的 Microsoft Visual Studio 7.0 是测试版本，因此产品序列号都是自动输入的，只需选择“接受”选项后，进入 Microsoft Visual Studio 7.0 的组件安装画面，然后按照提示进行安装即可。

第2章

C#语言基础

“Welcome”程序
类型
变量
数组
枚举
接口