

主编 曹瑞宁

程序员成长之路

# C#

# 编程入门与提高

## 本书内容

- ▶ .NET与C#/数据类型
- ▶ 变量与表达式/C#基本语法
- ▶ 类和方法/数组和属性
- ▶ 结构和索引/C#的高级特性
- ▶ 初识Visual Studio.NET
- ▶ 认识Visual C#.NET环境
- ▶ Windows桌面应用程序开发
- ▶ ASP.NET:Web Form
- ▶ ASP.NET:Web Services
- ▶ 创建分布式应用程序
- ▶ 利用C#开发网络程序
- ▶ C#数据库开发



Microsoft®

## Visual Studio.net

上海科学普及出版社

TP312

C135

程序员成长之路——

# C#编程入门与提高

主 编 曹瑞宁



上海科学普及出版社

674562

**图书在版编目 (CIP) 数据**

程序员成长之路: C#编程入门与提高 / 曹瑞宁主编.  
上海: 上海科学普及出版社, 2004. 2

ISBN 7-5427-2726-5

I. 程… II. 曹… III. C 语言—程序设计  
IV. TP312

中国版本图书馆 CIP 数据核字 (2003) 第 123869 号

策 划 铭 政  
责任编辑 徐丽萍

**C#编程入门与提高**

曹瑞宁 编著

上海科学普及出版社出版发行

(上海中山北路 832 号 邮政编码 200070)

---

|                  |                   |
|------------------|-------------------|
| 各地新华书店经销         | 北京市燕山印刷厂印刷        |
| 开本 787×1092 1/16 | 印张 22.5 字数 602000 |
| 2004 年 2 月第 1 版  | 2004 年 2 月第 1 次印刷 |

---

ISBN 7-5427-2726-5 / TP · 540

定价: 28.00 元

## 内 容 提 要

本书详细介绍用 C# 语言进行编程的方法与技巧, 本书从结构上可分为前后两个部分: 前半部分主要侧重于 C# 的基本知识及一些重要的概念, 包括 C# 的类型系统、控制语言与运算符、C# 的类、异常处理、线程以及 C# 的一些高级编程概念等; 后半部分介绍 C# 在各个方面的实际应用, 如组件技术、VS.NET、桌面程序、Asp.NET、数据库编程以及 Web Service 等。

本书语言轻松活泼, 讲解深入透彻, 结构安排合理, 是学习 C# 语言的最佳参考书, 也可作为各类大专院校和电脑培训班的学习教材。

# 前言

相看两不厌  
只有敬亭山  
——李白

根据某条不成文的定律，程序开发者每过十年就会遭遇一种全新的编程技术：在上世纪 80 年代他们使用的是 UNIX 和 C 语言，到了 90 年代则是 Windows 与 C++ 大行其道，而现在他们又面临着 .NET 与 C# 的挑战。如果单纯将后者作为一种编程语言来看的话，你会发现，随着“+”号的增加（你可以将“#”看做是四个“+”号），我们使用的语言变得越来越简单，并且与平台之间的结合也越来越紧密。

当然，没有人会称呼 C# 为 C++++。实际上，C# 正确的发音应该是 C Sharp，中文可以意译为“锋利的 C”，Sharp 这个词除了“锋利”的意思外，还兼有“轮廓分明的、敏锐的、机警的”等意思。但是我听到很多人把它叫做“C 井”，坦白地说，我很讨厌这种读法，就像不喜欢听到他们把 Dot NET 念成“点 NET”一样。还有一些人坚持把 C# 读为“C 升”，这是因为“#”这个符号在曲谱中代表了在原有音调上升半调，很明显这也是一种愚蠢的读法。

## 关于本书

本书向你介绍的是 C# 语言，但是在学习此种语言之前，我们首先将会对 .NET 进行一番讨论，因为 C# 根植于 .NET 这片土壤之上，失去了 .NET 平台的支持，C# 只会变成一株无根之木。许多（精彩的）C++ 语言的入门书都喜欢在第一页就向读者展示用 C++ 写成的第一个程序，但是 C# 却并不适宜这么做：当我回首自己的 C# 学习之路时，我发现我是在了解 .NET 好几个月之后才真正见识到第一个 C# 程序的。每个人的学习方法都不尽相同——有人希望打开一本书就能即刻在他的计算机上写出代码，有人却希望在学习之前先明了语言的背景。但愿你会喜欢我的这种方法。

本书在结构上大体分两部分。第一部分主要侧重于 C# 的基本知识及一些重要的概念，包括 C# 的类型系统、控制语言与运算符、C# 的类、异常处理、线程以及 C# 的一些高级编程概念等。在学习编程语言时我常常将它们与自然语言进行类比：如果把类型看做字母，把类看作单词，那么写一份 C# 程序多像写一篇文章啊！不同的是，写文章时你从来不用考虑类的封装、继承或者多态什么的。作为一种现代的编程语言，C# 当然是完全面向对象的。也许你对面向对象还不太了解，但是请相信我，它是如此地重要，以至于几乎会出现在你的整个编程行为之中。由于面向对象技术是如此重要，所以这一部分将会有两章内容，并特别地结合着 C# 语法介绍这一技术。实际上，在我看来，面向对象更像是一种思想而不是一种技术，它应该保存在每一位程序员的潜意识之中，随时影响你的编程行为。所以，除了单独使用一章进行介绍外，本书的其他部分也会时时提醒你如何运用面向对象的思想去设计。

从自然语言的角度来看，仅仅知道单词和语法是写不出优美文章的，你还需要了解大量的习惯用法才不会犯类似 Programmer's status is high 这样的错误（这句话看起来并没有错，

但不符合我国国情)。所以在学完基础知识后,本书的第二部分将向你介绍 C#在各方面的实际应用,如组件技术、VS.NET、桌面程序、ASP.NET、数据库编程、Web Service 等。这部分内容更侧重于实用性,使用的工具也将从.NET Frameworks SDK 升级到 VS.NET。如同一个习武之人需要时常与人过招一样,编写程序也是一种操作性很强的技术,需要你每天都操练一下才不会忘记,需要你进行实际的应用才会更深刻地领会。

除了.NET 与 C#之外,读完这本书你会发现,这其中还包括一些故事在内,那是笔者在学习 C#过程中所经历的一些事情,笔者把它们加上自己的评论包含在每章的引言和小结中贯穿全书。这一点略有别于其他的技术性书籍。

需要说明的是,这些故事中所提到的人和事,大部分都是真实的。如有雷同,纯属不幸。

## 关于 C#

在阅读本书之前你一定也听到过不少有关 C#的传说,或者还能说出“C#是一种简单的、现代的、面向对象的、类型安全的、版本控制的、兼容的、灵活的、基于组件开发的编程语言”这样的话。这句话出自 C#的开发者微软公司之口,一开始我还不能全部体会这句话的意思,但是已经能将它说得很熟练,直到随着对 C#学习的进一步深入,才逐渐明白这其中每一个形容词的含义。可以说,本书上部很大部分都是向你解释这句话的。虽然读完第 1 章内容可能你还是有些不明白,但是别急,随着学习的深入,你会慢慢体会出 C#的优雅之美。

在我看来,C#是一种优雅的语言,当今世上,堪与之媲美的惟有 SUN 公司出品的 Java 语言。C#刚刚推出的时候,在国内论坛上引来议论无数,有谩骂的,有赞赏的,还有更多人保持着怀疑的态度。事实上,我国人士在技术的选择方面,总是显得信心不足,新技术一来,就会产生出“红旗到底还能扛多久”之类的动摇念头。所以 C#一经推出,很多人就发出“Java Killer”、“C++ Killer”等惊呼,其实 C#并非要取代哪种语言,更不能取代所有的语言,否则微软也就不会在同一时期推出 VB.NET 和 VC++.NET 这些语言了。另外还有一些人(在看过 C#的文档后)将其称为“Java Clone”,这种说法有一定的道理,但是却未免失之偏颇。C#看起来很像 Java,那是因为它们都源自同一语言(C++),并且在核心部分都使用了虚拟机技术。另外我想,恐怕连微软自己都不得不承认,Java 实在是一种先进的语言。

“我是一个 Java 程序员,也是一个 Java 迷。Java 是一门最严格、最规范的完全面向对象的跨平台的语言。C#实在太像 Java 了。这一点也不奇怪。Java 当初是吸取 C++的精华而诞生的,因此 Java 内在的优秀性加上为数众多的 C++程序员能很快地向 Java 移植,使得全世界特别以硅谷为中心形成了一股 Java 热。JSP 模仿 ASP 而诞生,但它比 ASP 更优秀。而今,C#模仿 Java 而诞生,先不管其是否能超越 Java,其学习精神本身便是可取的。事实上,世界上现在语言众多,缺少的便是统一和规范。微软向来以霸道和自以为是著称,这里面当然是有其雄厚的财力作后盾,以至于其推出的无论结构性,面向对象性,还是稳定性都相当差的 VB 直至今今天还有人对它热衷不已,说什么‘微软的东西,准没错!’。而今,微软能放下架子,虚心吸取他人之长,我个人先给它评四星。当然,微软出此下策也是不得已而为之,眼睁睁看着大片市场被 Java 占领(当今正狂热的电子商务几乎都以 Java 为解决方案),微软怎能不放在心上。微软的官方人员早就指出:.NET 的主要竞争对手是 Sun、Oracle、AOL 等公司。不管其争斗的结果如何,能促进行业的统一、规范 and 发展的话,作为一名中国程序员就感到很高兴了。”

——网友 CharlyOnline 的评论



在刚开始学习 C# 的时候,我看到太多的此类文字,主题就是讨论某某语言与某某语言孰优孰劣,大家往往各执己见,争得不亦乐乎。相比之下,上述的评语还算是比较温和的。在相互的讨论与争吵中,我了解了 C# (以及其他许多种语言) 各个方面的特点,但同时我又很害怕跟别人讨论某种语言的优劣,因为这就好像讨论一位大侠使用什么样的兵刃更厉害些一样,剑有其轻盈灵巧之美,刀有其势重雄浑之利,所以重要的不是技术,而是施术之人。另外,只对一些语言了解一二的人也很难让我相信,所以,不必关心别人怎么说,选择自己喜欢的语言去学习,去使用,去研究,你就会发现其实所有的编程语言都是为处理数据而生的,只要掌握了其中的思想,没有什么不同。等上升到哲学这个高度,你就可以以大师自居了。

争论语言的优劣是没有任何价值的,不过在这里我倒想谈一谈 C# 与 Java 之间的异同,如果你曾经用过 Java 的话,这将会是一个很好的 C# 入门。C# 与 Java 有着太多的相似之处,所以 Java 用户转向学习 C# 非常容易。但是,毕竟 C# 比 Java 晚出世了近十年之久,所以在很多方面 C# 针对 Java 的缺陷进行了调整和改进。例如就语言本身而言,C# 就做了以下改进:

(1) C# 提供了枚举类型,并且允许通过 Boxing/Unboxing 机制对值类型与引用类型进行转换;

(2) C# 支持运算符重载;

(3) C# 使用代理(一种类型安全的指针)来代替 C++ 中的指针;

(4) C# 允许使用不安全代码。

另外,在编译及运行时的底层机制上二者也有很大的不同,这一点在本书后面的一些章节中将会深入地谈到。

像 Java 一样,C# 同样来源于 C++: C# 与 C++ 拥有同样的代码风格,并进行了更进一步的简化。但是 C# 毕竟是一种全新的语言,在很多方面都与 C++ 不尽相同,所以 C++ 用户在学习 C# 之前,应该注意以下问题:

首先,C# 不支持多重继承,或者说,C# 不支持类的多重继承,但是它为用户提供了更为有用的接口来实现多重继承;其次,C# 的类型系统有值类型与引用类型之分,值类型在栈上保存值,而引用类型变量本身位于栈上,但它们所指向的对象却都位于堆上,当被传递给方法时,值类型是传递值,而引用类型则仅仅传递了引用。并且在 C# 中结构是值类型而非引用类型,这与 C++ 中结构的概念有了很大的不同。最后,还有需要注意的一点:C# 不允许把布尔值转换为整数值,这意味着如下语句在 C# 中是错误的:

```
bool istrue = 0;
```

这是 C# 与 C++ 显著不同的几点,在细节方面 C# 与 C++ 还有着诸多的不同之处,所以就像 C 语言用户要把 C++ 当作一个全新的语言来学习一样,C++ 用户也应将 C# 视为一种全新的语言来学习,以避免造成概念上的混淆。

.NET 使 C# 与 C++ 这对原本应该很相似的语言具有了很大的不同,但也使原本与 C/C++/C# 语系风马牛不相及的 VB 具有了某种诡异的特性。新版本的 VB—VB.NET 与它的前辈相比截然不同:前者顺应时代潮流,加入了面向对象的特性,在许多功能表现上较之 C# 有过之而无不及,犹如 C++ 较之与 C,VB.NET 较之于原来的 VB 除了语言风格外均发生了翻天覆地的变化。除了引入面向对象的编程思想外,VB.NET 还被赋予了更多的 .NET 语言的特性,例如 VB 运行时被 .NET 通用语言运行时 (CLR) 所取代,数据类型被重新设计以兼容 .NET 的公共类型系统 (CTS),增加了结构化异常处理、多线程支持、命名空间以及自动垃圾收集等功能,使 VB.NET 与其他的 .NET 语言完全一致。

所以，讨论应选择 VB.NET 还是 C# 是毫无意义的，事实上，几乎任何被 .NET 平台所支持的语言（预计有 20 种或更多）都会具有上述的特性，我们选择语言惟一的动机可能会只剩下出于对某种风格的热爱：有人喜欢 C/C++/C# 语系的那种严谨、简洁、灵活的风格，有人喜欢 VB 语言的那种不区分大小写、傻瓜式（这没什么不好）的行文风格，还有人喜欢把代码缩进都变成语法的一部分（Python）。这都不要紧，因为在 .NET 大家庭中所有的语言都没有功能上的优劣之分，只有格调上的高低之别。由于 .NET 语言对象级的互操作性以及跨语言的高度集成能力，你再也不用担心由于所使用的语言和团队其他成员不兼容，而使自己身处尴尬境地；并且以后的招聘要求也不会强调你必须擅长何种语言，只要注明面向 .NET 平台编程即可。

前面我说过以后将是 .NET 平台的天下，Java 的爱好者肯定会持不同的意见，现在我收回这句话：以后的几年内将是 .NET 与 Java 两个虚拟平台之间的对抗阶段。所以选择什么语言已变得不重要，重要的是找准自己以后的舞台。面对 .NET，你将可以有无数种选择，而现在，C# 无疑将是其中最激动人心的选择。

前面我还提到程序员每过十年就要面对一种新的技术体系，你一定会对十年以后将面临的变化很感兴趣，说实话，我也很想知道，不过我想一定会有一种更先进的新语言出现并取代现在的 Java 和 C#。这我倒不怕，接着再学习就成了，反正程序员都是活到老学到老的；我最担心的就是将来有一天这个新的编程语言变成了自然语言：即在某个超级平台上，任何人都可以使用自己的母语来进行编程活动。不少人曾担心过程序员会抢走他们的饭碗，这下好了，是计算机抢走了程序员的饭碗。

## 关于学习

现在，我的拍档（就坐在我旁边）正在看电视，他一只手（有时候是两只手）拿着遥控器，过几分钟就换一个台（他总是相信另一个频道有更精彩的节目），一晚上总要将 40 多个频道轮换上好几遍，而我则只会在 10 点半的时候看上 40 分钟的《甜心俏佳人》（顺便说一句，这部电视剧很好看，郑重向你推荐）。家兄曾经提醒过我，除去愚蠢之外，浮躁将是阻碍我们进步的最大障碍。当然学习不同于看电视，但是，如果你总是能发现比自己正在学习的还要好的语言，那么我相信，你已经开始拿起遥控器了……

很早以前我就知道，如果所选择的路线正确的话，那么只要坚持下去，就一定能到达目的地。我想说的是，如果你将 C# 作为你现在的选择，那么请一定坚持学习下去。C# 是一个概念清晰、内涵丰富的语言，虽然融合了多种语言的优点，但是在学习 C# 之前那些语言的知识都不是必需的。不过在学习此种语言时一定要集中精力将语言的各个基本特性学好，不但要理解其含义，知道其用法，还应多想想语言中为何要包括这个特性，并弄清楚它们与其他特性之间的关系。

无论学习何种语言，开始的时候都应该只专注于语法本身，然后再来深入理解面向对象技术（如果你还没掌握的话），最后，在各种应用中学习 .NET 的类库并且逐渐去领会语言的本质。当你在实际工作中使用得多了，自然就会慢慢地理解一些更高级的知识，如 Design Patterns（设计模式）等。一般到这个阶段，你已经是一位技术高手了，再进一步发展，你将会开始研究 .NET 的源码，自己也开始动手为 .NET 平台添砖加瓦，到这时候，在江湖上你就已经是属于大师一级的人物了。

学习是一个慢慢体会的过程，同时也是一个逐渐发现的过程，一个人能够学习、积累、



总结和思考，这将是幸福的。

## 关于笔者

和大多数人一样，笔者是一个程序员，目前仍在从事编程工作。这是份很有意思的工作，程序员也是一种很有意思的职业。我记得有位朋友曾说过，程序员是天下最幸福的人：做着自己最喜欢的事，还有人给发工资……

作为一个程序员，我有许多业内的朋友，除了讨论技术之外，我们也讨论其他的一些问题，我将这些讨论集结在书内，希望你未来的程序人生能有所启发。

总之，这是一本特别的书，因为本书的笔者是一个特别的人，并且你会在读完这本书后认识他。

## 关于其他

笔者鼓励初学者手工输入代码进行练习，所以将不会在任何地方提供源代码下载。

另外值得一提的是，由于笔者采用了一种特别的讲述方式，所以不必为某个概念或新名词的出现而迷惑不解，致使学习的进度徘徊不前。耐心地读下去，也许你就会慢慢地了解一切。

在此书的编辑与出版过程中，得到了许多专家和友人的大力支持与帮助，他们是崔慧勇、任立功、吴闯、叶勇、秦志敏、芦淑珍和李建慧，在此向他们表示由衷地感谢。当然，本书也许还有某些疏漏与不足之处，欢迎广大读者批评指正。

联系网址：<http://www.china-ebooks.com>。

编 者

2004 年 1 月

# 目 录

|                               |                             |
|-------------------------------|-----------------------------|
| <b>第 1 章 .NET 与 C#</b> .....1 | 3.2.1 运算符.....52            |
| 1.1 理解.NET.....1              | 3.2.2 常用运算符.....53          |
| 1.1.1 微软简史.....2              | 3.2.3 其他运算符.....55          |
| 1.1.2 .NET 战略.....5           | 3.2.4 运算符优先级.....57         |
| 1.1.3 .NET 平台.....8           |                             |
| 1.2 认识 C#.....11              | <b>第 4 章 C#基本语法</b> .....58 |
| 1.2.1 语言简史.....11             | 4.1 结构化程序设计的基本流程.....58     |
| 1.2.2 C#语言的特点.....13          | 4.2 条件语句.....59             |
| 1.2.3 第一个 C#程序.....16         | 4.2.1 if 语句.....59          |
| 1.3 C#程序在.NET 平台中的执行.....18   | 4.2.2 switch 语句.....61      |
| <b>第 2 章 数据类型</b> .....24     | 4.3 循环语句.....63             |
| 2.1 值类型.....26                | 4.3.1 while 语句.....63       |
| 2.1.1 简单类型.....27             | 4.3.2 do-while 语句.....65    |
| 2.1.2 结构类型与枚举类型.....30        | 4.3.3 for 语句.....66         |
| 2.2 引用类型.....31               | 4.3.4 foreach 语句.....67     |
| 2.2.1 类.....32                | 4.4 跳转语句.....68             |
| 2.2.2 接口和委派.....33            | 4.4.1 break 语句.....68       |
| 2.2.3 数组.....34               | 4.4.2 continue 语句.....69    |
| 2.3 值类型与引用类型的关系.....34        | 4.4.3 return 语句.....70      |
| 2.3.1 值类型与引用类型的比较.....34      | 4.4.4 goto 语句.....71        |
| 2.3.2 装箱和拆箱.....36            | 4.5 组织应用程序.....72           |
| 2.4 C#类型系统的特点.....38          | 4.5.1 动态链接库.....72          |
| 2.4.1 一切皆对象.....38            | 4.5.2 编译单元.....73           |
| 2.4.2 缺省值及类型转换.....39         | 4.5.3 装配与名字空间.....73        |
| 2.4.3 公共类型系统.....40           | 4.5.4 名字空间声明.....74         |
| <b>第 3 章 变量与表达式</b> .....44   | 4.5.5 名字空间成员.....75         |
| 3.1 变 量.....45                | 4.5.6 Using 指示符.....75      |
| 3.1.1 变量命名.....46             | 4.5.7 名字空间的应用.....80        |
| 3.1.2 静态变量和实例变量.....47        |                             |
| 3.1.3 数组变量.....48             | <b>第 5 章 类和方法</b> .....83   |
| 3.1.4 参数变量.....48             | 5.1 面向对象编程.....84           |
| 3.1.5 局部变量.....52             | 5.2 类.....85                |
| 3.2 表达式.....52                | 5.2.1 类的声明.....85           |
|                               | 5.2.2 类的成员.....87           |
|                               | 5.2.3 构造函数.....90           |
|                               | 5.2.4 析构函数.....93           |





|                                    |     |                                   |     |
|------------------------------------|-----|-----------------------------------|-----|
| 5.3 方 法.....                       | 95  | 9.3.2 创建新项目实例 .....               | 158 |
| 5.3.1 方法的声明 .....                  | 95  | 9.4 保存与关闭解决方案 .....               | 161 |
| 5.3.2 方法中的参数 .....                 | 96  | 第 10 章 认识 Visual C# .NET 环境 ..... | 162 |
| 5.3.3 静态方法与实例方法 .....              | 99  | 10.1 Visual C# .NET 特色 .....      | 162 |
| 5.3.4 方法的重载和覆盖 .....               | 100 | 10.1.1 易用的窗口管理 .....              | 163 |
| 第 6 章 数组和属性 .....                  | 106 | 10.1.2 强大的源代码编辑功能 .....           | 164 |
| 6.1 数 组 .....                      | 106 | 10.2 菜单 .....                     | 165 |
| 6.1.1 数组的基本概念 .....                | 106 | 10.2.1 【文件】菜单 .....               | 165 |
| 6.1.2 数组声明 .....                   | 107 | 10.2.2 【编辑】菜单 .....               | 168 |
| 6.1.3 数组的初始化 .....                 | 108 | 10.2.3 【视图】菜单 .....               | 170 |
| 6.1.4 数组的应用 .....                  | 109 | 10.2.4 【项目】菜单 .....               | 171 |
| 6.2 属 性 .....                      | 114 | 10.2.5 【生成】菜单 .....               | 173 |
| 6.2.1 属性的声明 .....                  | 114 | 10.2.6 【调试】菜单 .....               | 174 |
| 6.2.2 属性的应用 .....                  | 116 | 10.2.7 【工具】菜单 .....               | 174 |
| 第 7 章 结构和索引 .....                  | 120 | 10.2.8 【窗口】菜单 .....               | 177 |
| 7.1 结 构 .....                      | 120 | 10.2.9 【帮助】菜单 .....               | 178 |
| 7.1.1 结构声明 .....                   | 120 | 10.3 工具箱 .....                    | 179 |
| 7.1.2 结构的应用 .....                  | 121 | 10.3.1 【数据】标签 .....               | 179 |
| 7.2 索 引 .....                      | 124 | 10.3.2 【XML 架构】工具箱 .....          | 180 |
| 7.2.1 索引的声明 .....                  | 125 | 10.3.3 【Web 窗体】标签 .....           | 181 |
| 7.2.2 索引的应用 .....                  | 127 | 10.3.4 【组件】标签 .....               | 182 |
| 7.2.3 属性和索引的比较 .....               | 130 | 10.3.5 【Windows 窗体】标签 .....       | 183 |
| 第 8 章 C# 的高级特性 .....               | 132 | 10.3.6 HTML 标签 .....              | 184 |
| 8.1 委派和事件 .....                    | 132 | 10.3.7 【剪贴板循环】工具箱 .....           | 187 |
| 8.2 异常处理 .....                     | 136 | 10.3.8 【常规】标签 .....               | 187 |
| 8.2.1 checked 和 unchecked 语句 ..... | 136 | 10.4 【代码编辑器】窗口 .....              | 188 |
| 8.2.2 异常处理语句 .....                 | 137 | 10.4.1 打开【代码编辑器】窗口 .....          | 188 |
| 8.2.3 用户自定义异常类 .....               | 140 | 10.4.2 【代码编辑器】简介 .....            | 189 |
| 8.3 特征和反射 .....                    | 141 | 10.5 【解决方案资源管理器】<br>对话框 .....     | 190 |
| 8.4 与非托管代码交互操作 .....               | 144 | 10.6 【属性】对话框 .....                | 191 |
| 8.4.1 使用原来的 DLL .....              | 145 | 10.7 【服务器资源管理器】对话框 .....          | 192 |
| 8.4.2 非安全代码 .....                  | 147 | 10.8 【对象浏览器】对话框 .....             | 194 |
| 8.4.3 与 COM 进行交互 .....             | 150 | 10.9 【宏资源管理器】对话框 .....            | 195 |
| 第 9 章 初识 VS.NET .....              | 155 | 第 11 章 Windows 桌面应用<br>程序开发 ..... | 198 |
| 9.1 Visual Studio .NET 简介 .....    | 155 | 11.1 桌面应用程序开发一般步骤 .....           | 199 |
| 9.2 启动 Visual Studio .NET .....    | 156 | 11.1.1 第一步: 新建项目 .....            | 199 |
| 9.3 创建新项目 .....                    | 158 | 11.1.2 第二步: 界面设计 .....            | 200 |
| 9.3.1 解决方案与项目 .....                | 158 |                                   |     |





|                                                      |            |                                                      |            |
|------------------------------------------------------|------------|------------------------------------------------------|------------|
| 11.1.3 第三步: 组件属性设置 .....                             | 200        | 13.2.3 使用 Visual Studio 部署<br>XML Web services ..... | 269        |
| 11.1.4 第四步: 程序设计 .....                               | 201        | 13.2.4 WebService 目录 .....                           | 274        |
| 11.2 表单 .....                                        | 201        | <b>第 14 章 创建分布式应用程序 .....</b>                        | <b>279</b> |
| 11.2.1 属性 .....                                      | 201        | 14.1 将创建的分布式应用程序<br>的结构 .....                        | 279        |
| 11.2.2 表单的基本方法 .....                                 | 206        | 14.2 分布式应用程序的创建过程 .....                              | 280        |
| 11.2.3 表单响应的主要事件 .....                               | 208        | 14.2.1 创建中间层业务对象 .....                               | 280        |
| 11.2.4 表单应用的例子 .....                                 | 209        | 14.2.2 创建用户界面 .....                                  | 283        |
| <b>第 12 章 ASP.NET: Web Form .....</b>                | <b>212</b> | 14.2.3 部署解决方案 .....                                  | 288        |
| 12.1 ASP.NET 简介 .....                                | 212        | <b>第 15 章 利用 C# 开发网络程序 .....</b>                     | <b>290</b> |
| 12.1.1 ASP.NET 中的概念 .....                            | 214        | 15.1 网络基础知识 .....                                    | 290        |
| 12.1.2 ASP.NET 与 ASP 的比较 .....                       | 216        | 15.1.1 简述 .....                                      | 290        |
| 12.1.3 ASP.NET 中的文件类型 .....                          | 217        | 15.1.2 计算机网络发展历史 .....                               | 291        |
| 12.2 基于 Web 表单的开发 .....                              | 218        | 15.1.3 计算机网络知识 .....                                 | 293        |
| 12.2.1 Web Form 介绍 .....                             | 218        | 15.2 Socket 类开发 .....                                | 301        |
| 12.2.2 第一个 Web Form .....                            | 219        | 15.2.1 说明 .....                                      | 301        |
| 12.2.3 页面处理过程 .....                                  | 221        | 15.2.2 开发实例 .....                                    | 305        |
| 12.3 使用 ASP.NET 开发应用程序 .....                         | 223        | 15.3 NET 类开发 .....                                   | 307        |
| 12.3.1 Global.asax 文件 .....                          | 224        | 15.3.1 程序设计的主要思路<br>及实现方法 .....                      | 308        |
| 12.3.2 Web.config .....                              | 225        | 15.3.2 制作过程 .....                                    | 309        |
| 12.4 服务器控件 .....                                     | 227        | <b>第 16 章 C# 数据库开发 .....</b>                         | <b>311</b> |
| 12.4.1 HTML 控件与 Web 控件 .....                         | 227        | 16.1 数据库结构简介 .....                                   | 311        |
| 12.4.2 文本输入控件 .....                                  | 230        | 16.1.1 数据库基础知识 .....                                 | 311        |
| 12.4.3 按钮控件 .....                                    | 234        | 16.1.2 ODBC 与 ADO .....                              | 319        |
| 12.4.4 复选控件 .....                                    | 235        | 16.1.3 ADO.NET 简介 .....                              | 320        |
| 12.4.5 单选控件 .....                                    | 236        | 16.1.4 ADO.NET 的特点 .....                             | 321        |
| 12.4.6 列表框控件 .....                                   | 238        | 16.1.5 ADO.NET 与 ADO 的比较 .....                       | 325        |
| 12.4.7 panel 控件 .....                                | 239        | 16.2 连接数据源与数据读取 .....                                | 326        |
| 12.4.8 选择控件 .....                                    | 241        | 16.2.1 连接数据源 .....                                   | 327        |
| 12.4.9 ImageButton 控件 .....                          | 244        | 16.2.2 数据适配器 .....                                   | 330        |
| 12.4.10 列表控件 .....                                   | 245        | 16.3 数据集 .....                                       | 332        |
| 12.4.11 Repeater 控件 .....                            | 247        | 16.3.1 数据集简介 .....                                   | 332        |
| 12.4.12 数据列表控件 .....                                 | 250        | 16.3.2 数据表与数据视图 .....                                | 335        |
| 12.4.13 数据表格 DataGrid .....                          | 254        | 16.3.3 DataGrid .....                                | 337        |
| <b>第 13 章 ASP.NET: Web Services .....</b>            | <b>262</b> | 16.4 C# 数据库开发实例 .....                                | 339        |
| 13.1 Web Services 简介 .....                           | 262        | 16.4.1 说明 .....                                      | 339        |
| 13.2 VS.NET 中的 XML<br>Web services .....             | 265        | 16.4.2 开发步骤 .....                                    | 339        |
| 13.2.1 XML Web services 简介 .....                     | 266        |                                                      |            |
| 13.2.2 使用 Visual Studio 创建<br>XML Web services ..... | 266        |                                                      |            |



# 第1章 .NET与C#

昔我往矣，杨柳依依  
今我来思，雨雪霏霏  
——《诗经》

这一章将要谈到的是.NET与C#，这可以算得上是当下最热门的两个单词了。关于.NET你可能知道它有很多种解释，而对于C#你或许也知道它是一种语言，并且是有史以来第一个面向组件编程的语言。如前所述，C#有很多高级特性都是建立在.NET平台之上的，所以要学习C#语言就必须先了解有关.NET的知识。而对于.NET，不同的人可能会对它有不同理解，本章将会谈到.NET计划、.NET战略、.NET平台，这将帮助你从不同的角度来认识.NET。根据我个人的经验，首先站在非程序员的角度，将会更容易理解.NET究竟是个什么东东。对于C#，它是一种“简单的、现代的、面向对象的、类型安全的、版本控制的、兼容的、灵活的、基于组件开发的编程语言”，要完全地理解这句话也颇为不易，本章将会站在历史的角度来讨论有关C#的这些特点。另外，本章还实现了C#的第一个程序，并对C#程序在.NET平台上运行的情况进行了一番描述。

2000年6月我大学毕业，从北京回到石家庄，正式开始了我的职业生涯。一开始我使用的语言是ASP，我一直认为这不能称之为编程，因为ASP不是一种编程语言，把它叫做动态网页实现技术可能更好。另外，ASP很简单，并且，简单就是它全部的特点——这使得它很容易就能学会（在后来的工作中，我接触到许多应聘的学生，他们都告诉我自己精通ASP语言）。ASP虽然学习起来很简单，但是使用起来却不得不多费点劲儿：我还能记得自己晚上一个人在办公室用VI一步一步调试某一个页面的情景。那时候ASP在国内拥有大量的用户，人们已经开始认识到组件的重要性。

就在我回到石家庄的同一时间（美国西部时间2000年6月22日上午），微软公司在位于美国西雅图郊外的总部邀请新闻记者、新闻分析家等约400人，举行了新闻发布会Forum2000，宣布正式推出.NET计划。这个计划中包括了新的网络计算平台（.NET Framework）、新的语言（C#）、新的开发工具（Visual Studio.NET）以及ASP的下一个版本ASP.NET（最开始被称为ASP+）。那时候我学习的主要兴趣就在ASP.NET上，并且通过这个窗口开始了解Microsoft.NET的各个方面。

## 1.1 理解.NET

有关Microsoft.NET，不同的人可能会对它有不同理解。很多人曾经问我：.NET究竟是个什么东西？我的回答总是随着时间的推进而改变。最早我告诉他们这是一个新的平台，





后来告诉他们这是 Microsoft 的一个新战略,但是现在我想我会这样回答: .NET 是一个概念,是一种构想,或者是微软的一个梦想。只有站在各种不同的角度,或许才能更深入地理解.NET 的要领。

对于 Microsoft .NET, 微软有如下正式的描述:

“ .NET 是 Microsoft 用以创建 XML Web 服务 (下一代软件) 的平台, 该平台将信息、设备和人以一种统一的、个性化的方式联系起来。”

“借助于 .NET 平台, 可以创建和使用基于 XML 的应用程序、进程和 Web 站点以及服务。它们之间可以按照设计, 在任何平台或智能设备上共享和组合信息与功能, 以向单位和个人提供定制好的解决方案。”

“ .NET 是一个全面的产品家族, 它建立在行业标准和 Internet 标准之上, 提供开发 (工具)、管理 (服务器)、使用 (构造块服务和智能客户端) 以及 XML Web 服务体验 (丰富的用户体验)。 .NET 将成为您今天正在使用的 Microsoft 应用程序、工具和服务器的一部分, 同时, 新产品不断扩展 XML Web 的服务能力, 以满足您的所有业务需求。”

这个美国时间 2000 年 6 月 22 日上午面世的新概念一经发布, 就引起了全球 IT 界的关注, 因为发布它的微软公司向以改变业界的游戏规则而闻名, 该公司另一个出名的地方就是精通“拿来主义”, 善于继承别人的思想再经过封装变成自己的产品。在 .NET 的新产品中, VS.NET 有 Delphi 的影子, C# 更宛若 Java 的一个 clone, 甚至 .NET 的“网络计算”的概念也曾经在很早以前由 SUN 提出过。但是这并不能妨碍 .NET 成为去年最热门的技术词汇, 随着 VS.NET 和 Windows.NET 的发布, 它还会成为今年人们谈论最多的话题之一。

和国内很多同行一样, 起初我对这家全球最大的软件公司充满了景仰之情。有一次 (我正在上大), 一批美国教师到我们学校交流访问, 在一个谈话节目中, 我的一位同学就曾经问其中一位美丽的女士: 在贵国普通大众中间, 是微软的老板比尔·盖茨的知名度高, 还是 (当时) 公牛队的飞人乔丹更有名些? (此位同学极端崇拜盖茨先生, 发誓将来要当中国的比尔)。但是后来, 由于受到来自某种力量的蛊惑, 我又开始对其抱有极端反对的态度, 最后, 随着进一步的认识, 才开始对微软, 对比尔, 以及对其竞争对手如 SUN、AOL 保持了公正的态度, 也就是说, 不再只是站在技术的角度来看待这些公司, 开始对其抱有一颗平常心了。

### 1.1.1 微软简史

意大利文艺美学家克罗齐有一句名言: “一切历史都是当代史”。也许回忆一下历史会使我们更容易理解微软的 .NET 计划。

1975 年, 罗伯茨因其发明了第一台微电脑“牛郎星”大获成功。当时由于他经营的 MITS 公司陷入困境, 情急之下才发明了这台微机。《大众电子》曾在宣传中说, “牛郎星”将面向普通用户, 采用最简便的 BASIC 语言编程。然而, 它的内存只有 4KB, 专家们都认为它难以支持 BASIC, 罗伯茨手中根本没有这种软件。

那一日, 罗伯茨突然接到一个电话, 对方声称自己是西雅图市的交通资料公司, 可以为阿尔泰计算机开发出 BASIC 语言, 并询问他是否感兴趣。

罗伯茨岂止是感兴趣, 如果有谁真能做成这件事, 那阿尔泰就不再是件玩具, 它将成为一台名符其实的个人计算机。一个星期后, 罗伯茨又接到一封正式的信函, 信中说他们已经





做成了软件。罗伯茨当即按信里提供的联系方式打去电话，可对方答复说那儿是西雅图市的湖边中学。罗伯茨摇摇头，心想这一定是哪个调皮学生的恶作剧。正准备作罢时，这个子虚乌有的公司又打来长途。这一回，罗伯茨不客气了，他恼怒地大声嚷道：“如果你们真有BASIC软件，那就劳驾送到MITS，我保证在阿尔伯克基机场恭候！”他本想激出对方的真情，哪知电话传来肯定的答复：“完全可以，三周后，请您到机场接站。”

三周后，罗伯茨从飞机上接到的竟然是位乳臭未干的毛孩子。来者说他叫保罗·艾伦，全权代表他的伙伴比尔·盖茨。而此时，比尔·盖茨正在远隔千里的波士顿哈佛大学校园里，忐忑不安地等待着消息。

这时的比尔·盖茨，戴着大镜片的眼镜，一副娃娃脸，和现在没什么两样，不同的只是那时候他只是一个不到19岁的大学生。他生于西雅图一位律师和一位教师组成的家庭，家教良好，但绝无与计算机有关的遗传因子。约在1969年，他就读的湖边中学引入一套PDP-10小型计算机终端，用电线连接着电传打字机，并按使用时间付费。比尔就在这台电传机上做起了他的“软件梦”，同时也结识了比他高两年级的保罗·艾伦。两人共同在计算机上写出一个“井字棋”游戏，玩得好不开心。小家伙们终日迷恋计算机的后果，是学校付不起昂贵的费用，不得已对他们的上机次数进行了严格限制。无计算机可玩，比尔和保罗实在技痒难忍，翻箱倒柜凑齐360元钱，捧回一块英特尔生产的微处理器8008芯片，设想做一台记录分析交通情况的小装置，就这样，西雅图出现了一家由两个中学生组成的“交通资料公司”。这种灰色盒子里缠满电线的小玩意，不知为何被好几个州的交通部门看中，为此他们真的净赚了近2万元。直到后来，比尔·盖茨考上了哈佛大学法律专业，仍对计算机情有独钟。

还是1974年12月的一个寒冷的傍晚，满身是雪的艾伦举着《大众电子》杂志，冲进比尔·盖茨的房间。“千载难逢的机会来了，”他兴奋地嚷着，“阿尔泰微电脑诞生，该是我们大显身手的时候了！”比尔受到感染，灵机一动想出了好主意，他对艾伦说：“我们可以用学校的PDP-10机开发一种模拟阿尔泰的程序，然后向BASIC冲击！”接下来，两人侃出了软件框架并初步分了工：艾伦负责编写模拟程序，比尔则主攻阿尔泰的BASIC解释程序。

两个不知天高地厚的少年，当他们通过电话向罗伯茨拍胸保证三周交货的时候，所谓BASIC软件还只是凭空想像而已。好在无论是模拟还是解释程序，对两人来说都不陌生。艾伦的工作必须先行，他不到两周就干完了模拟程序。比尔·盖茨面对的难题更多，他整整写了8000行的机器语言程序，又千方百计地精简压缩，设法使它能装进阿尔泰微电脑狭小的“身躯”。然而，越是临近完工，他们就越是紧张，毕竟没有一台真正的阿尔泰，编写的程序能不能在微型计算机上运行，谁也不敢保证。直到最后把艾伦送上了飞机，比尔·盖茨的心仍不踏实。

显然，跟随罗伯茨来到MITS的开发实验室的艾伦更加紧张。当他终于第一次看到了真实的阿尔泰时，立即就把带来的程序纸带送进计算机的读带装置，此时，他的心已经提到了喉咙口。时间一分一秒地过去，读带机还在“咔咔”地“吃”进纸带，这几分钟就像是过了好几年。突然，电传打字机轻轻地动作了一下，打印纸上印出了“READY”，标志计算机做好了准备。艾伦猛地转过身，双手抱住罗伯茨的肩膀：“阿尔泰现在有了BASIC！”他马上写了一段模拟阿波罗飞船登月的程序，交给罗伯茨打孔输入，程序果然十分准确地运行并计算出结果。这大概是世界上第一台微电脑的第一次真正的应用。

1975年2月，大功告成，艾伦亲赴MITS演示，十分成功。这年春天，艾伦进入MITS，担任软件部经理。念完二年级课程，盖茨也飞往MITS，加入艾伦从事的工作。那时他们已





有创业的念头，但要等到 BASIC 被广大用户接受才可，此前他们是不会离开罗伯茨的，他们还有待羽翼渐丰。

罗伯茨开着他那辆蓝色货车巡回美国各大城市，为阿尔泰计算机摇旗呐喊。如今，在他的手里有了一件最能吸引大众的法宝——买一部“牛郎星”计算机，可配上一位“织女”——BASIC 软件，这引来各界人士纷纷解囊。到 1980 年为止，这种软件竟卖出了 100 万套。

在 BASIC 软件成功的鼓舞下，比尔·盖茨毅然从哈佛大学退学，于 1975 年 7 月在阿尔伯克基竖起了微软公司的大旗，公司简称 MS，连同保罗和他自己，一共 6 员大将，专门从事微型计算机的软件开发。比尔·盖茨为公司制定了目标：“每一个家庭每一张桌上都有一部微型计算机运行着我们的程序！”这个时候，微软还只能算是一个编程语言公司（软件公司）。1978 年，微软公司搬到了太平洋西北岸，并继续为另外一些微型计算机编写程序。从此，微软迅速发展起来，1978 年，微软的收益仅为 100 万美元，到 1984 年，微软收益已经突破 230 亿美元。

但是，这时候微软对行业的影响力还远没有现在那么大，直到有一天，IBM 公司找上门来，要求与盖茨联合开发新的操作系统，于是，就产生了后来对 IT 行业影响深远的 MS DOS。

1985 年 6 月，微软和 IBM 达成协议，联合开发 OS/2 操作系统。根据协议，IBM 在自己的计算机上可随意安装，几乎分文不取，而允许微软向其他计算机厂商收取 OS/2 的使用费。当时 IBM 在 PC 市场拥有绝对优势，兼容机份额极低，当时 IBM 负责此项目的洛伊几乎是不假思索地同意了。到了 1989 年，兼容机市场已达到 80% 的份额，使得微软在操作系统的许可费上，短短几年就赢利 20 亿美元。

当然，双方在 OS/2 上的合作未能持续下去。微软从 1981 年就开始开发后来称之为 Windows 的操作系统。Windows 最初版本与 OS/2 有很大关系。1985 年，Windows 1.0 的问世，才真正使其成为一个产品，而 1995 年 8 月，Windows 95 的发布，则正式把微软推向计算机业的巅峰，至此，微软帝国雏形初现。随后微软四处出击，向各个领域伸出触角，在文字处理、电子表格、IDE、网络产品等各个方面逐渐确立了自己的领先地位，完成了由一个小小的程序语言开发公司到操作系统公司再到应用产品集成公司的转变。

但是，随着互联网的出现，一切都发生了变化。

第一个挑战来自 Netscape，1994 年 10 月 13 日，Marc Andreessen 发布了名字叫做 Netscape Navigator 的新一代浏览器，并发展了 HTML 标准。1995 年 Navigator 2.0 发布时，Netscape 已经有了几百名员工，超过 1 000 万的用户和快速增长的产品线，包括代理服务器 Proxy Server、网络开发工具还有商业应用程序 I-Apps，并支持在网络上销售产品或提供服务。1995 年，Netscape 的年收入达到令人惊异的 8 000 万美元，成为历史上增长最快的软件公司。这给微软很大刺激，于是在当年，Microsoft 发布了人们期望已久的新操作系统 Windows 95，同时还有 WWW 浏览器 Internet Explorer 1.0。但是此时再行动显然已经慢了一步，很快，微软在 IM 软件、搜索引擎、流媒体、服务器等方面都受到了不同程度的挑战。尤其是在与 Netscape 的竞争中，不但耗资巨大，而且还受到了使用不正当竞争手段的控告。很明显，微软已经感觉到了未来互联网的力量，比尔·盖茨在 IE 的发布会上，将 Internet 的崛起称为“IBM 生产个人电脑 10 年以来最重要的发展”。虽然 Windows 95 取得了巨大的成功，但这并不能说明微软已经就此拥有了未来，比尔·盖茨显然已经意识到，随着互联网的发展，他们用以征服天下的武器——操作系统的作用将会越来越小，也许有一天，人们再不需要安装操作系统来管理计算机和运行各种各样的软件，只需要使用浏览器就可以完成现在所有的工作了



(现在 B/S 三层结构的应用系统大行其道就是很好的一个证明), 到那个时候, 软件业“日不落帝国”的称号就有可能落到 Netscape 的头上。

于是微软凭借其操作系统的优势, 正式对 Netscape 宣战, 虽然付出的代价不小, 不过最后总算取得了胜利。

此次激战使微软意识到, 只有成为未来的设计者, 才会在未来立于不败之地。有鉴于此, 比尔·盖茨开始密谋 .NET 计划, 意图为人们的未来生活进行全方位的设计。

### 1.1.2 .NET 战略

何谓 .NET? 根据微软官方的解释, .NET 就是 Microsoft 面向 XML Web Service 的平台, 这种下一代的软件将使用一种统一、个性化的方式, 将我们的信息、设备和人员紧密联系在一起。但你可能又会问道, “准确地说, 什么是 XML Web 服务呢?” 简单地说, 它就是未来的计算。正如比尔·盖茨所言, 发布 .NET 之后, “网络将由呈现式的平台转为完全的平台”。说得形象一点就是, 在未来, 网络不但能提供给我们内容, 还能提供给我们服务。在微软推广 .NET 的时候, 还提出了一个叫做 .NET 体验的概念。所谓 .NET 体验, 就是用户个性化和集成的计算体验, 这种体验使用连接在一起的 XML Web 服务, 而服务则通过智能设备提供。举个简单的例子, 现在我们要从互联网获得帮助, 一般情况下都是通过电脑获取信息, 再经过自己的分析判断, 才(有可能)借助到网络的力量。这也是当下最常见的一种方式: 我们由许多个网站获得各种各样的信息, 然后再打电话或发传真与他们进行联系。也就是说, 大量的工作是需要在网下完成的, 互联网所提供给我们的帮助, 不过是一些孤立的信息罢了, 这无疑给我们的生活和工作带来很大的不便, 也远远没有发挥出互联网的威力。 .NET 正是为解决这个问题而来的, 它通过 Web Service 将各个网站(现在被称为信息孤岛)联结起来, 共同为使用者提供服务而不再仅仅是内容。我曾经跟母亲谈到过未来的网络, 在将来, 我们的冰箱会自己检测食物是不是还够大伙吃, 如果它发现缺少了某种食物, 就会自动地通过 Web Service 联系到离家最近的那间超市, 向超市定购食品, 再通过银行提供的 Web Service 付账。这一切都是完全自动完成的。

在介绍 .NET 战略之前, 还是让我们先来了解一下微软的 .NET 计划: 什么是 .NET, 为何我们需要 .NET, .NET 能为我们带来什么?

#### 1.1 .NET 是什么

.NET 计划是微软公司于美国时间 2000 年 6 月 22 日发布的下一代计算计划, 这个计划的主要目的是为了让网络由呈现式的平台转为完全的平台。具体来讲, 就是通过 .NET 计划, 可以让我们在任何时间、任何地点、使用任何设备获取信息并且得到服务。

简单地说, .NET 就是 Microsoft 的 XML Web 服务平台。不论操作系统或编程语言有何差别, XML Web 服务都能使应用程序在 Internet 上传输和共享数据。

Microsoft .NET 平台包含广泛的产品系列, 它们都是基于 XML 和 Internet 行业标准构建的, 并提供从开发、管理、使用到体验 XML Web 服务的每一方面。XML Web 服务将成为我们今天正在使用的 Microsoft 的应用程序、工具和服务的一部分, 并且还将打造出全新的产品, 以满足我们所有的业务需求。

现在, 微软正在从五个方面创建 .NET 平台, 即工具、服务器、XML Web 服务、客户端

