



高等学校精品规划教材

C#程序设计

刘 兵 刘 冬 易 虹 王卫华 等编著



中国水利水电出版社
www.waterpub.com.cn

21 世纪高等学校精品规划教材

C#程序设计

刘 兵 刘 冬 易 虹 王卫华 等编著



中国水利水电出版社
www.waterpub.com.cn

内 容 提 要

本书介绍 C#程序设计,按照计算机及相关专业第一门高级程序设计语言课程的要求编写,全面细致地介绍了 C#面向对象编程的概念和方法。全书共分为 12 章,内容包括:程序设计语言与程序设计方法、C#程序设计基础、结构化程序设计、数组和字符串、类和对象、继承和多态、集合、事件与委托、接口、异常处理、文件操作和多线程等。本书每章均配有有一定数量的习题,以方便学生练习。本书的所有程序代码均已在 Visual Studio 2008 运行通过。

本书体系编排完整,内容结构合理,强调重要概念,各章节所选择的例题贴合重点、丰富适度。本书以培养程序设计、分析能力和计算机综合应用能力为目的,适合作为高等院校计算机及相关专业学习程序设计语言的教材,也可作为 C#程序设计培训教材,以及自学 C#程序设计的参考书。

本书配有免费电子教案,读者可以从中国水利水电出版社网站以及万水书苑下载,网址为: <http://www.waterpub.com.cn/softdown/>或 <http://www.wsbookshow.com>。

图书在版编目(CIP)数据

C#程序设计 / 刘兵等编著. — 北京:中国水利水电出版社, 2011. 12
21世纪高等学校精品规划教材
ISBN 978-7-5084-8890-5

I. ①C… II. ①刘… III. ①C语言—程序设计—高等学校—教材 IV. ①TP312

中国版本图书馆CIP数据核字(2011)第167370号

策划编辑:雷顺加 责任编辑:宋俊娥 加工编辑:毕露云 封面设计:李 佳

书 名	21 世纪高等学校精品规划教材 C#程序设计
作 者	刘 兵 刘 冬 易 虹 王卫华 等编著
出版发行	中国水利水电出版社 (北京市海淀区玉渊潭南路 1 号 D 座 100038) 网址: www.waterpub.com.cn E-mail: mchannel@263.net (万水) sales@waterpub.com.cn 电话: (010) 68367658 (发行部)、82562819 (万水)
经 售	北京科水图书销售中心(零售) 电话: (010) 88383994、63202643、68545874 全国各地新华书店和相关出版物销售网点
排 版	北京万水电子信息有限公司
印 刷	三河市鑫金马印装有限公司
规 格	184mm×260mm 16 开本 19.5 印张 479 千字
版 次	2011 年 12 月第 1 版 2011 年 12 月第 1 次印刷
印 数	0001—4000 册
定 价	35.00 元

凡购买我社图书,如有缺页、倒页、脱页的,本社发行部负责调换
版权所有·侵权必究

前 言

C#（读作 C Sharp）是微软公司基于.NET 平台推出的一种全新的、面向对象的高级程序设计语言，并充分吸收了 C/C++ 的优点，继承了 Visual Basic 的高效性和 C++ 的强大功能，基于.NET Framework 的有力支撑，提供了实现跨平台应用开发的强有力的集成开发工具和方法。用微软公司的话来说，“C#是从 C 和 C++ 派生出来的一种简单、现代、面向对象和类型安全的编程语言”。

C# 语言作为.NET 的核心编程语言，具有语法简洁、类型安全、面向对象、灵活性与兼容性强等特点，因此，它已经成为企业解决方案的首选开发语言。如何使教材简洁、通俗、先进、实用，保持介绍知识的连贯性、系统性与先进性，得到师生的认可与好评，一直是我们努力的方向，我们力争做到：选材恰当、说理严谨、深入浅出。

本书是作者总结多年教学经验并依据应用实践编写而成的，全面细致地介绍了 C# 面向对象编程的概念和方法，内容包括程序设计语言与程序设计方法、C# 程序设计基础、结构化程序设计、数组和字符串、类和对象、继承和多态、集合、事件与委托、接口、异常处理、文件操作和多线程等。本书每章均配有一定数量的习题，以方便学生练习。本书的所有程序代码均已在 Visual Studio 2008 集成环境中运行通过。

本书主要是面向应用型本科院校、大专院校计算机及相关专业的学生的，也适用于 C# 爱好者、初学者，还可以作为有关培训机构的培训教材。本书所配电子教案，可以从中国水利水电出版社和万水书苑网站免费下载，网址为：<http://www.waterpub.com.cn/softdown/> 及 <http://www.wsbookshow.com>。

本书由刘兵负责全书统稿及定稿工作，其中刘兵编写了第 1、2、3、4 章，易虹编写了第 5、6、7 章，刘冬编写了第 8、9、10 章，王卫华编写了第 11、12 章。武汉工业学院电气信息工程系的李禹生教授认真地审阅了全书，并提出了很多宝贵意见。丰洪才、管庶安等参与了本书大纲的讨论。同时要感谢在程序案例设计和调试方面给予大力帮助的贾瑜、蒋丽华、左爱群、易逵、徐军利、孙平等。在本书的编写过程中，得到了武汉工业学院计算机与信息工程系的领导的关心和支持，在此表示感谢。另外，感谢江小丽女士对全书文字资料的录入及校排工作给予的帮助。

由于作者水平所限，书中难免存在一些疏漏及不妥之处，敬请读者批评指正。作者的电子邮件地址为：lbliubing@sina.com。

作者
2011 年 10 月

目 录

19.25 印张

前言

第 1 章 C#概述.....	1	3.2.1 while 语句.....	44
1.1 C#基本概念.....	1	3.2.2 do...while 语句.....	46
1.1.1 什么是 C#.....	1	3.2.3 for 语句.....	48
1.1.2 .NET Framework.....	3	3.2.4 foreach 语句.....	51
1.2 C#程序开发环境.....	6	3.3 跳转语句.....	52
1.2.1 安装 C#开发环境.....	6	3.3.1 break 语句.....	52
1.2.2 Visual Studio 的集成开发环境.....	9	3.3.2 continue 语句.....	54
1.2.3 C#的控制台应用程序.....	13	3.3.3 goto 语句.....	55
本章小结.....	16	3.3.4 return 语句.....	57
习题一.....	17	本章小结.....	58
第 2 章 C#程序设计基础.....	19	习题三.....	58
2.1 数据类型.....	19	第 4 章 数组与字符串.....	63
2.1.1 值类型.....	19	4.1 函数.....	63
2.1.2 引用类型.....	24	4.1.1 函数的基本概念.....	63
2.1.3 数据类型转换.....	25	4.1.2 函数的返回值.....	65
2.2 变量和常量.....	27	4.1.3 函数的参数.....	66
2.2.1 变量.....	27	4.2 数组.....	70
2.2.2 常量.....	29	4.2.1 一维数组.....	70
2.3 操作符和表达式.....	29	4.2.2 数组的基本操作.....	72
2.3.1 操作符.....	29	4.2.3 多维数组.....	76
2.3.2 算术操作符.....	30	4.2.4 交错数组.....	78
2.3.3 赋值操作符.....	31	4.2.5 将数组作为参数传递.....	79
2.3.4 关系操作符.....	32	4.2.6 params 关键字的应用.....	81
2.3.5 逻辑操作符.....	32	4.3 字符串.....	83
2.3.6 位运算符.....	32	4.3.1 字符串类型定义.....	83
本章小结.....	35	4.3.2 字符串的基本操作.....	84
习题二.....	35	本章小结.....	90
第 3 章 结构化程序设计.....	38	习题四.....	90
3.1 选择语句.....	38	第 5 章 类和对象.....	94
3.1.1 if 语句.....	38	5.1 面向对象的基本概念.....	94
3.1.2 switch 语句.....	41	5.2 类的声明和对象的创建.....	95
3.2 循环语句.....	44	5.2.1 类的定义.....	96

5.2.2 类的成员	97	7.2.2 泛型类的成员	173
5.2.3 对象的声明	99	7.2.3 泛型类中的方法	176
5.3 成员函数	100	7.2.4 泛型约束	178
5.3.1 构造函数	100	7.3 泛型集合	181
5.3.2 析构函数	104	7.3.1 泛型集合的建立	181
5.4 属性	105	7.3.2 泛型集合的排序与搜索	182
5.5 方法与重载	108	本章小结	187
5.5.1 方法的定义	108	习题七	187
5.5.2 静态和非静态的方法	110	第 8 章 委托与事件	192
5.5.3 方法重载	110	8.1 委托	192
5.6 索引函数	114	8.1.1 声明委托	192
5.7 this 关键字	117	8.1.2 实例化委托	194
5.8 运算符重载	119	8.1.3 多重委托	196
本章小结	120	8.1.4 委托调用	197
习题五	121	8.2 事件	199
第 6 章 继承与多态	127	8.2.1 事件的基本概念	199
6.1 继承	127	8.2.2 事件定义与触发	200
6.1.1 基类和派生类	127	本章小结	204
6.1.2 派生类的继承	130	习题八	204
6.1.3 抽象类和抽象方法	133	第 9 章 接口	210
6.2 多态性	136	9.1 接口的基本概念	210
6.2.1 虚拟方法和重载方法	136	9.1.1 定义接口	210
6.2.2 密封类和密封方法	139	9.1.2 定义接口成员	213
6.3 值类型和引用类型	141	9.1.3 接口成员的访问	217
6.3.1 System.Object 对象	141	9.2 接口的实现	219
6.3.2 内存的组织	145	9.2.1 类对接口的实现	219
6.3.3 装箱与拆箱	147	9.2.2 接口的继承	220
本章小结	150	本章小结	222
习题六	150	习题九	222
第 7 章 集合与泛型	155	第 10 章 异常处理	225
7.1 预定义的集合	155	10.1 错误与异常	225
7.1.1 数组列表	155	10.1.1 程序错误	225
7.1.2 BitArray 集合	159	10.1.2 异常	227
7.1.3 Hashtable 集合	162	10.1.3 抛出异常	230
7.1.4 Queue 集合	165	10.2 异常类	231
7.1.5 Stack 集合	166	10.2.1 Exception 类	231
7.1.6 SortedList 集合	168	10.2.2 自定义异常	233
7.2 泛型	171	10.2.3 Checked 和 unchecked 语句	235
7.2.1 泛型概述	171	10.3 正则表达式	237

10.3.1 正则表达式的基本概念	237	第 12 章 线程	282
10.3.2 正则表达式类	241	12.1 多线程的概念	282
本章小结	247	12.1.1 多线程的概念	282
习题十	248	12.1.2 C#中的线程实现方法	284
第 11 章 文件操作	253	12.2 多线程的程序设计	284
11.1 文件操作概述	253	12.2.1 创建线程	284
11.1.1 文件系统的基本概念	253	12.2.2 线程的并行性	287
11.1.2 用于输入和输出的类	254	12.2.3 多线程的优先级别	290
11.2 文件存取	256	12.2.4 线程的后端与前端运行	291
11.2.1 文本文件的存取	256	12.2.5 线程的方法和状态	292
11.2.2 二进制文件的存取	260	12.3 线程同步	294
11.2.3 对文件的加密和解密	261	12.3.1 线程同步的基本概念	294
11.3 文件管理	264	12.3.2 锁定机制	295
11.3.1 目录管理	265	12.3.3 通知	298
11.3.2 文件管理	269	本章小结	301
11.3.3 路径管理	277	习题十二	301
本章小结	279	参考文献	303
习题十一	279		

第 1 章 C#概述



本章学习目标

C#.NET 是微软公司推出的一种基于 .NET 平台应用的现代编程语言，是一种由 C 和 C++ 衍生出来的面向对象的编程语言。学习本章以后，读者将对 .NET 和 C# 有一个初步的认识，了解 C# 的开发环境，通过编写第一个 C# 程序，了解 C# 应用程序的基本框架和开发步骤。通过对本章的学习，读者应该掌握以下主要内容：

- .NET Framework 的功能
- Visual Studio .NET 开发工具包的使用
- 认识 C# 控制台应用程序的结构

1.1 C#基本概念

1.1.1 什么是 C#

C#（读作 C Sharp）是微软公司在 2000 年 7 月发布的一种编程语言，是为生成在 .NET Framework 上运行的各种应用程序而设计的。微软对 C# 的定义是：“一种类型安全、现代、简单，由 C 和 C++ 衍生而来的面向对象的编程语言”。

C# 吸收了 C++、Visual Basic、Delphi、Java 等编程语言的优点，体现了当今最新的程序设计技术的功能和精华。C# 继承了 C 语言的语法风格，同时又继承了 C++ 的面向对象的特性。不同的是，C# 的对象模型已经面向 Internet 进行了重新设计，使用的是 .NET 框架的类库；C# 不再提供对指针类型的支持，使得程序不能随便访问内存地址空间，从而更加健壮；C# 不再支持多重继承，避免了以往类层次结构中由于多重继承带来的一些不可预知的错误。.NET 框架为 C# 提供了一个强大的、易用的、逻辑结构一致的程序设计环境，而公共语言运行时 CLR（Common Language Runtime）为 C# 程序语言提供了一个托管的运行环境，使程序比以往更加稳定和安全。

C# 语法表现力强，而且简单易学。C# 的大括号语法使任何熟悉 C、C++ 或 Java 的开发人员都可以立即上手。了解上述任何一种语言的开发人员都可以在很短的时间内使用 C# 高效地进行工作。C# 语法简化了 C++ 的诸多复杂性，并添加了很多强大的功能，例如，C# 支持泛型方法和类型，从而提供了更出色的类型安全和性能。C# 还提供了迭代器，允许集合类的实施者定义迭代行为，以便容易被客户端代码使用。

1. 简洁的语法

在默认情况下，Visual C# 的代码在 .NET Framework 环境中是不允许直接对内存进行操作的。这与 C++ 不同，C++ 中会出现大量的 “->”、“::” 操作符，而这些在 Visual C# 中已经

不再出现了，Visual C#只支持“.”操作符，对于读者来说，现在需要理解的一切仅仅是名字嵌套而已。

2. 真正的面向对象设计

Visual C#语言具有面向对象语言所应有的一切特性：封装、继承、多态。在 C#的类型系统中，每种类型都可以看作是一个对象，C#提供了一个叫做装箱（boxing）与拆箱（unboxing）的机制来完成这种操作，而不给使用者带来麻烦。

Visual C#中只允许单继承，即每个类只允许有一个父类（也称基类），从而避免了类型定义的混乱。同时，Visual C#不存在全局函数和全局变量，也不存在全局常数。所有的东西都必须封装在类中，这样做的好处是：代码具有更好的可读性，并且也不会有重命名冲突问题。

3. 与 Web 的紧密结合

Web 是现今编程的一大趋势与潮流，然而由于历史原因，现在的一些开发工具不能与 Web 紧密地结合。而在 .NET 中新增的程序开发模型越来越多的解决方案需要与 Web 标准相结合、相统一。Visual C#对简单对象访问协议 SOAP（Simple Object Access Protocol）的使用使得大规模深层次的分布式开发从此成为可能。

有了 Web 服务框架的帮助，对于程序员来说，网络服务就像是 C#的本地对象。程序员们能够方便地开发 Web 服务，并允许通过 Internet 被运行在操作系统上的任何语言调用。例如，XML（Extensible Markup Language，可扩展标记语言）已经成为网络中数据结构传送的标准，为了提高效率，Visual C#允许直接将 XML 数据映射为结构，这样就可以有效地处理各种数据。

4. 完全的安全性与错误处理

语言的安全性与错误处理能力，是衡量一种语言是否优秀的重要依据。任何人都会犯错误，即使是有丰富经验的程序员也不例外（例如，忘记变量的初始化，对不属于自己管理范围的内存空间进行修改等，这些错误常常会产生难以预见的后果）。一旦这样的软件被投入使用，寻找并改正这些简单错误的代价是非常巨大的。Visual C#的先进设计思想可以消除软件开发中的一些常见错误，并提供包括类型安全在内的完整的安全性能。为了减少开发中的错误，Visual C#会帮助开发人员通过使用更少的代码来完成相同的功能，这不但减轻了编程人员的工作量，同时也有效地避免了错误的发生。

在 Visual C#中，不能使用未初始化的变量，对象的成员变量由编译器负责将其置为 0，当局部变量未经初始化而被使用时，编译器将给出提醒；Visual C#不支持不安全的指向，因此，它提供了边界检查与溢出检查功能。

5. 灵活的版本处理技术

Visual C#提供了内置的版本支持来减少开发费用，使用 Visual C#将会使开发人员更加轻松地开发和维护各种商业应用程序。

升级软件系统中的组件（模块）是一项非常容易产生错误的工作。在代码修改过程中，可能会对现存的软件产生影响，很可能导致程序的崩溃。为了帮助开发人员处理这些问题，Visual C#内置了版本控制功能。例如，函数重载必须被显式地声明，而不会像在 C++或者 Java 中那样隐式地进行，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对接口继承的支持，这些特性可以保证复杂的软件能够被方便地开发和升级。

6. 灵活性和兼容性

在简化语法的同时, Visual C#并没有失去灵活性。尽管不是一种无限制的语言,例如,不能用它来开发硬件驱动程序,在默认状态下没有指针等,但是,在学习过程中,读者会发现,它仍然是相当灵巧的。

如果需要, Visual C#允许用户将某些类或者类的某些方法声明为非安全的,这样,就能够使用指针,并且调用这些非安全的代码不会带来任何其他问题。此外, C#还提供了委托(Delegate)来模拟指针的功能。例如, Visual C#不能支持类的多重继承,但是可以通过对多个接口的继承,来实现这一功能。

正是由于其灵活性, C#允许与 C 风格的需要传递指针型参数的 API 进行交互操作,动态链接库 DLL (Dynamic Link Library) 的任何入口点都可以在程序中进行访问。Visual C#遵守.NET 公共语言规范 CLS (Common Language Specification), 从而保证了 Visual C#与其他语言组件之间的互操作性。

1.1.2 .NET Framework

.NET Framework 是支持生成和运行下一代应用程序和 XML Web Services 的内部 Windows 组件。.NET Framework 的主要目的是实现下列目标:

(1) 提供一个一致的面向对象的编程环境,而无论对象代码是在本地存储和执行,还是在本地执行但在 Internet 上分布,或者是在远程执行的。

(2) 提供一个将软件部署和版本控制冲突最小化的代码执行环境。

(3) 提供一个可提高代码(包括由未知的或不完全受信任的第三方创建的代码)执行安全性的代码执行环境。

(4) 提供一个可消除脚本环境或解释环境的性能问题的代码执行环境。

(5) 使开发人员的经验在面对类型完全不同的应用程序(例如基于 Windows 的应用程序和基于 Web 的应用程序)时保持一致。

(6) 按照工业标准生成所有通信,以确保基于 .NET Framework 的代码可与任何其他代码集成。

图 1-1 给出了.NET Framework 的体系框架。.NET 框架的体系结构包括以下五大部分。

(1) 程序设计语言及公共语言规范(CLS)。

(2) 应用程序平台(ASP.NET 及 Windows 应用程序等)。

(3) ADO.NET 及类库。

(4) 公共语言运行库(CLR)。

(5) 程序开发环境(Visual Studio.NET)。

下面重点介绍.NET 框架平台使用的语言、平台的作用以及采用的通信协议三方面的问题。

1. .NET 框架使用的语言

在.NET Framework 上可以运行多种语言,这是.NET 的一大优点。.NET Framework 中的公共语言规范 CLS (Common Language Specification) 实际上是一种语言规范。由于.NET 框架支持多种语言,并且要在不同语言对象之间进行交互,因此就要求这些语言必须遵守一些共同的规则,CLS 定义了这些语言的共同规范,包括数据类型、语言构造等,同时 CLS 又

被设计得足够小。

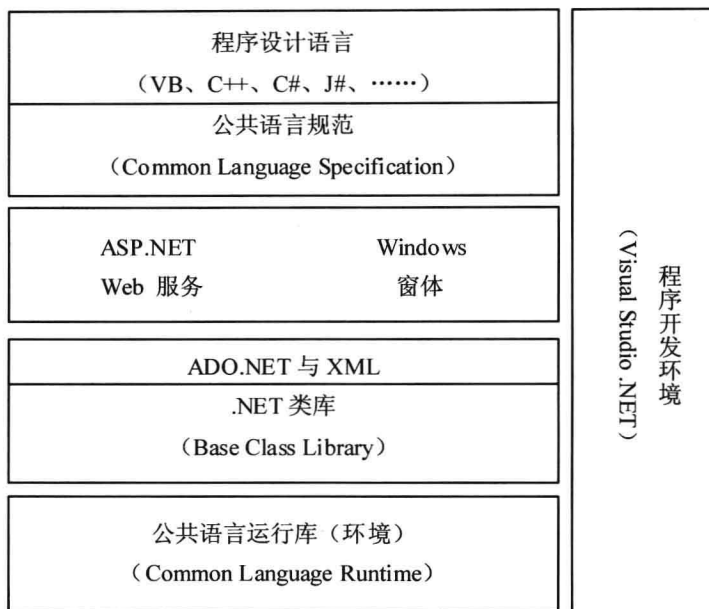


图 1-1 .NET Framework 的体系框架

凡是符合 CLS 规范的语言都可以在 .NET 框架上运行。目前已经有 C#.NET、VB .NET、C++.NET、J#.NET、JScript.NET 等 (VBScript 已不再使用)，预计还将有二十多种语言可以运行在 .NET 框架上。目前，有些公司还在创建符合 CLS 规范的自己的语言。

由于多种语言都运行在 .NET 框架之中，因此功能基本上都是相同的，只是语法有区别。程序开发者可以选择自己习惯或爱好的语言进行开发。而 Visual C#.NET 是专门针对 .NET 框架开发出来的语言，非常简练和安全，最适合在 .NET 框架中使用。

各种语言经过编译后，并不直接产生中央处理器 CPU 可执行的代码，而是先转变为一种中间语言 (Intermediate Language, IL 或 MSIL)，执行时再由公共语言运行库载入内存，通过实时解释将其转换为 CPU 可执行代码。转换的过程如图 1-2 所示。

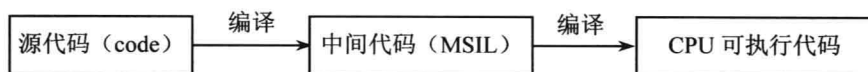


图 1-2 从源代码到 CPU 可执行代码的转换过程

设置中间语言的目的是为了满足不同平台的需要。源程序经过编译转换为中间语言，各类平台只要装上不同的转换引擎，就可以将其转换为指定 CPU 需要的代码。由于中间语言类似于汇编语言，与二进制代码非常接近，因此实时解释的速度也很快。转换的过程如图 1-3 所示。

2. 基础类库

.NET 框架的另一个主要组成部分是类库，包括数千个可重用的“类”。各种不同的开发语言都可以用类库来开发传统的命令程序和图形用户界面 GUI (Graphical User Interface)

应用程序。

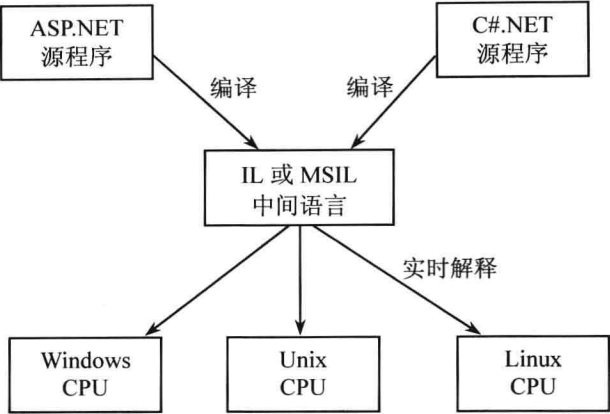


图 1-3 利用中间语言转换的过程

.NET 框架中的类被拆分为命名空间（Namespace），而命名空间是类库的逻辑分区。类库所采用的命名空间是采用分层次的结构，即命名空间下面又可以再分成子命名空间。每个命名空间都包含一组按照功能划分的相关类。这样，一个大型的.NET 库就变得易于理解和便于使用了。其分类如下：

- 所有微软公司提供的类都以 System 或 Microsoft 命名空间开头。
- 有关网络协议和简单编程接口的类放在 System.Net 命名空间中。
- 有关文件 I/O、内存 I/O、独立存储的类放在 System.IO 命名空间中。
- 基于 Windows 应用程序的用户界面的类放在 System.Windows.Forms 命名空间中。
- 有关 Web 服务器与浏览器交互，以及 Web 服务的类都放入 System.Web 及其子命名空间中。
- 所有用于处理 XML、XML 架构、XSL/T 转换、Xpath 表达式的类都放入 System.Xml 及其子命名空间中。

3. 公共语言运行库

公共语言运行库 CLR（Common Language Runtime），也称公共语言运行环境，相当于 Java 体系中的“虚拟机”，是 .NET 框架的核心，并提供了程序运行时的内存管理、垃圾自动回收、线程管理和远程处理以及其他系统服务。同时，还能监视程序的运行，进行严格的安全检查和维护工作，以确保程序运行的安全性、可靠性以及其他形式代码的准确性。

目前 .NET Framework 的最新版本是 3.5 版，是以 .NET Framework 2.0 版和 .NET Framework 3.0 版为基础的，主要包括如下组件：

- .NET Framework 2.0。
- .NET Framework 2.0 Service Pack 1，它更新包含 .NET Framework 2.0 中的程序集。
- .NET Framework 3.0，它使用 .NET Framework 2.0 或 .NET Framework 2.0 SP1（如果已安装）中的程序集，并且包含 .NET Framework 3.0 引入的技术所必需的程序集。例如，Windows Presentation Foundation（WPF）所必需的 Presentation Framework.dll 和 PresentationCore.dll 就随 .NET Framework 3.0 一起安装。

- .NET Framework 3.0 Service Pack 1, 它更新包含 .NET Framework 3.0 中的程序集。
- 一些新程序集, 它们为 .NET Framework 2.0 和 3.0 提供附加功能, 同时还提供 .NET Framework 3.5 中新采用的技术。

如果安装 .NET Framework 3.5 时缺少上述任何组件, 则系统会自动安装。应用程序无论针对的是 .NET Framework 2.0、3.0 还是 3.5 版, 都使用相同的程序集。例如, 对于使用 WPF 并针对 .NET Framework 3.0 的应用程序, 其所使用的 mscorlib 程序集与使用 Windows 窗体并针对 .NET Framework 2.0 的应用程序是相同的。如果 .NET Framework 2.0 SP1 已安装在计算机上, 则 mscorlib.dll 会自动更新, 并且两个应用程序都将使用 mscorlib.dll 的更新版本。

.NET Framework 3.5 为 2.0 和 3.0 中的技术引入了新功能, 并以新程序集的形式引入了其他技术。随 .NET Framework 3.5 引入的技术主要有:

(1) 语言集成查询。语言集成查询 LINQ (Language INtegrate Query) 是 Visual Studio 2008 和 .NET Framework 3.5 中的新功能。LINQ 将强大的查询功能扩展到 C# 和 Visual Basic 语言的语法中, 并采用标准的、易于学习的查询模式。

(2) 外接程序和扩展性。.NET Framework 3.5 中的 System.AddIn.dll 程序集为可扩展应用程序提供了强大而灵活的支持, 并引入了新的结构和模型, 可帮助开发人员完成向应用程序添加扩展性的初始工作, 并确保开发人员的扩展在宿主应用程序发生更改时仍可继续工作。

(3) WPF。WPF (Windows Presentation Foundation) 是微软推出的, 提供了统一的编程模型、语言和框架, 真正做到了分离界面设计人员与开发人员的工作; 同时提供了全新的多媒体交互用户图形界面。WPF 包括许多功能, 例如可扩展应用程序标记语言 XAML (eXtensible Application Markup Language)、控件、数据绑定、布局、二维和三维图形、动画、样式、模板、文档、媒体、文本和版式等。在 .NET Framework 3.5 中, WPF 包含多个方面的更改和改进, 其中包括版本控制、应用程序模型、数据绑定、控件、文档、批注和三维 UI (User Interface) 元素。

(4) WCF。WCF (Windows Communication Foundation) 是支持面向服务的应用程序。WCF 通过 SOAP 提供强大的交互通信支持, 这是现代计算机设备的基本要素。WCF 是一个运行库和一组 API, 用于创建在服务端与客户端之间发送消息的系统。同样的基础结构和 API 还可用于创建一些应用程序, 这些应用程序可与同一计算机系统上或通过 Internet 访问的系统上的其他应用程序进行通信。

1.2 C#程序开发环境

1.2.1 安装 C#开发环境

本书的 Visual C# 应用程序的开发环境使用的是 Visual Studio 2008, 其安装过程具体操作步骤如下:

(1) 将 Visual Studio 2008 安装光盘放入光驱中自动运行或双击 Setup 文件, 打开 Visual Studio 2008 安装程序界面, 如图 1-4 所示。



图 1-4 “Visual Studio 2008 安装程序”界面

(2) 单击图 1-4 安装提示中的第一项“安装 Visual Studio 2008”，进入安装向导界面，加载 Visual Studio 2008 的安装组件，如图 1-5 所示。



图 1-5 加载 Visual Studio 2008 的安装组件

(3) 在图 1-5 中检测已经安装的组件和需要安装的组件后，直接单击“下一步”按钮，进入如图 1-6 所示的界面，在该界面中选择“我已阅读并接受许可条款”单选按钮，并输入产品密钥，单击“下一步”按钮，进入安装方式选择界面，如图 1-7 所示。

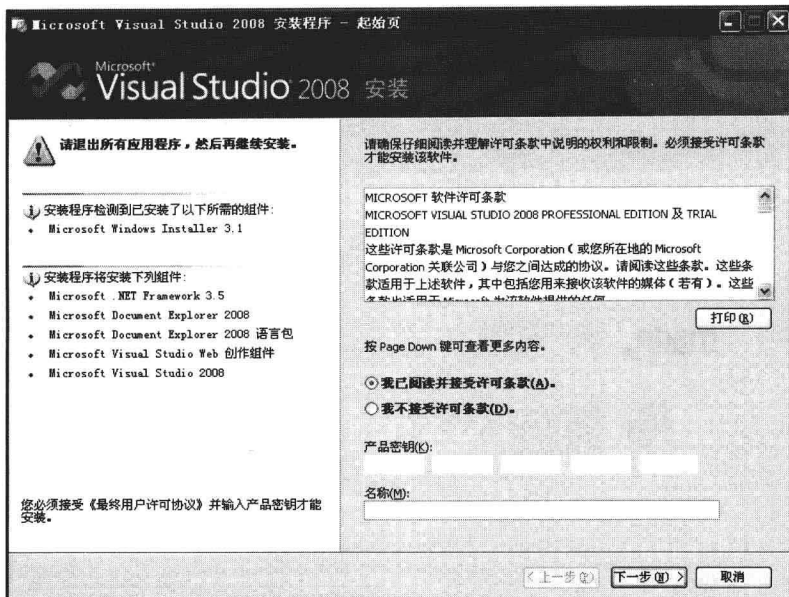


图 1-6 许可协议



图 1-7 安装方式选择界面

(4) 在图 1-7 中选择安装 Visual Studio 2008 的安装目录，并对安装的主要模块进行选择，如果没什么特殊要求，直接选择图 1-7 左边的“默认值”单选按钮，然后单击图 1-7 右下角的“安装”按钮进行安装，进入 Visual Studio 2008 安装程序文件复制界面，如图 1-8 所示。

(5) 安装完成后，进入“Visual Studio 安装完成”界面（如图 1-9 所示），然后单击“完成”按钮即可。



图 1-8 Visual Studio 2008 安装程序文件复制界面



图 1-9 Visual Studio 2008 安装完成界面

1.2.2 Visual Studio 的集成开发环境

Visual Studio 2008 集成开发环境 (Integrated Development Environment, IDE) 是用于开发和维护托管的、本机的和混合模式的应用程序，并提供了用于创建不同类别应用程序的多种项目模板，这些模板包括 Microsoft Windows 窗体、控制台、ASP.NET 网站、ASP.NET Web 服务、SmartPhone 2003、Windows CE 5.0 以及其他类型的应用程序。此外，还提供了针对特定设备（如移动设备）的模板。而且，开发人员还可以根据需要进行不同的编程语言，包括 C#、Microsoft Visual Basic .NET 和 C++ 等。

1. 起始页

起始页是进入 Visual Studio 2008 的入口，如图 1-10 所示。起始页对于新安装的 Visual

Studio 是其呈现的第一个窗口,即 Visual Studio 通常是从起始页开始的。如果起始页不可见,可以从菜单中依次选择“视图→其他窗口→起始页”命令。

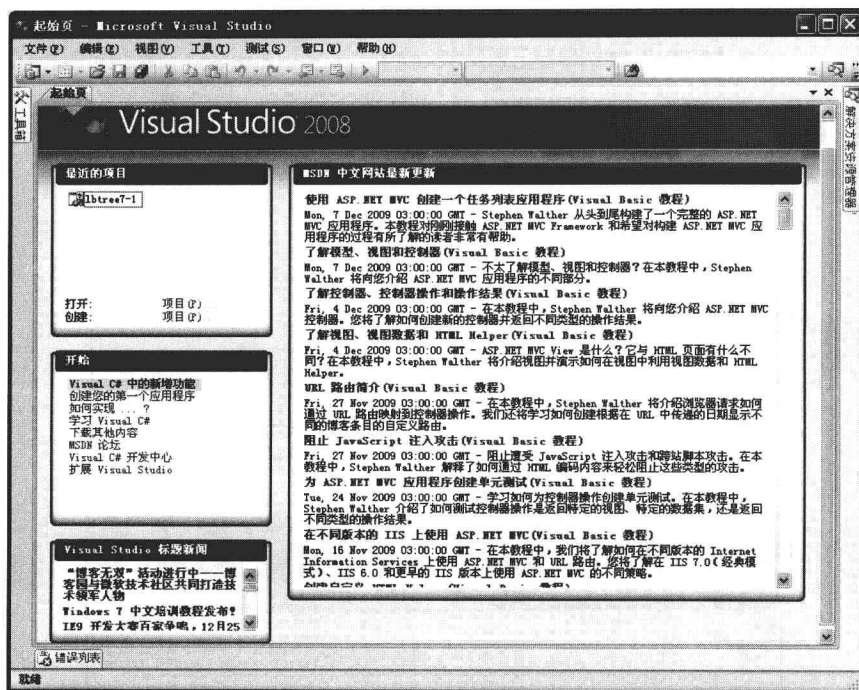


图 1-10 Visual Studio 2008 起始页

Visual Studio 2008 起始页包括以下 4 个窗口:

- (1) 最近的项目: 该窗口列出最近打开过的项目, 在项目列表中进行选择, 可以打开相应的项目。该窗口底部的“打开”和“创建”按钮, 分别用于打开和创建一个新项目。
- (2) MSDN 中文网站: 该窗口包含关于 Visual Studio 和 C# 的最新新闻的链接。每个主题都是以预览的方式显示的, 单击相关链接就可以观看完整的文章。
- (3) 开始: 该窗口包含对 Visual Studio 新手很有用的链接。
- (4) Visual Studio 标题新闻: 该窗口允许开发人员向 Microsoft 直接提交反馈。

2. 创建一个项目

在 Visual Studio 中开发一个应用程序通常是从创建一个新项目开始的。项目是 Visual Studio 的基本组织构件, 文件、资源、引用以及其他应用程序构件在这里被组织起来。创建一个项目的方法是从图 1-10 的菜单上依次选择“文件→新建→项目”命令(或者从图 1-10 上的“最近的项目”窗口中单击创建后的项目), 打开如图 1-11 所示的“新建项目”对话框。

由于本书讲解的是 C# 程序设计, 所以在图 1-11 左边的“项目类型”中选择“Visual C#”, 在右边的“模板”中选择相应的模板。由于本书重点讲解 C# 的主要编程思想, 所以本例中选择的模板是“控制台应用程序”, 并在图 1-11 下方的项目名称中输入“helloworld”, 选择该项目的存储位置, 然后单击“确定”按钮建立一个新的项目。当项目建立成功之后, 打开如图 1-12 所示的界面。