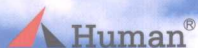


一群中国优秀程序员编程的亲身体会
一盏进入程序设计大门的指路灯
一套全新的多媒体教材
祝贺您的选择

作者：张俊 乔宇峰 周欣

洪恩软件



编程之道

C# 程序设计入门

专家提示：

先学光盘后学教材

多媒体教学，全程语音讲解，案例演示

教材重点：

实例与分析贯穿始终

全新理解网络编程

数据库设计思路

C#编程轻松掌握



吉林电子出版社
北京洪恩教育科技有限公司
北京洪恩教育培训学校

编程之道 系列

C# 程序设计入门

北京洪恩教育科技有限公司 开发制作

作者 张俊 乔宇峰 周欣

吉林电子出版社

内 容 提 要

.NET 技术使得程序员的生产效率能够得以大幅度提高, 从而以更低的成本开发出功能强大的程序。本教材是专门针对 C# 入门用户编写的, 既适合没有编程经验的读者, 也适合有 C、C++、Java 或其它编程语言基础的读者使用。

C# 是微软为 .NET 平台量身定做的编程语言, 它是一种现代面向对象程序设计语言, 使程序员能够快速地在 .NET 平台上开发种类丰富的应用程序, 它继承了 C++ 和 Java 的语法, 去掉了 C++ 中的许多复杂和容易引起问题的东西, 比起 Java, 它又多了一些非常简洁优美的特性。由于语法上的相似性, C++ 和 Java 程序员能够很容易地掌握 C#。

本教材共分为 14 章, 主体采用简介 - 实例 - 分析 - 深化的四步教学方法。

教材前半部分是用一个完整的小程序讲解了 C# 的基本语法、程序结构和面向对象知识等内容, 这样避免了初学者单纯学习基础知识而无法很好地进行实践的问题; 后半部分主要介绍了用 C# 开发应用程序的一些具体问题, 如 Windows 窗体程序、数据库应用程序、ASP .NET Web 应用程序、多媒体程序、网络程序等; 最后通过一个企业级的网上书店案例将 C# 的知识加以总结, 让读者在真实的项目中领会开发的真谛。

通俗、形象、生动的多媒体教学光盘更是为用户铺平了编程之道, 该光盘具有以下特点:

- 全程语音讲解
- 真实操作演示
- 重要代码分析
- 详细示例操作

可以辅助读者更快捷地掌握 C# 编程。

强烈建议读者先学习教材中的多媒体光盘, 通过光盘能在最短的时间内得到作者的倾力传授和多年的编程经验和技巧, 可以有效地降低学习的难度, 对读者来说是不可多得的快速入门的学习软件。教材中所有涉及到的例子源代码和相关的安装软件都可以在该光盘中找到。

版权所有 翻印必究

教材名: C# 程序设计入门
出版社: 吉林电子出版社出版
责任编辑: 陈 波
教材编著: 北京洪恩教育科技有限公司
CD 著作者: 北京洪恩教育科技有限公司
开 本: 787 × 1092 1/16 26.375 印张 466 千字
印刷厂: 山东新华印刷厂德州厂
印 次: 2005 年 2 月第 1 次印刷
本版号: ISBN 7-900393-66-8
定 价: 38.00 元 (1CD + 配套教材)

《编程之道》系列制作群

策 划: 李晓松
稿 件: 张 俊 乔宇峰 周 欣
制 作: 李晓松
CD 制作: 辛 建
编 校: 张晓宜
封面设计: 郭大卫
插 图: 郑小康

为普及计算机技术做贡献

原清华大学校长 张孝文 书赠

编者言

2000年6月22日，对于微软和全世界与微软产品打交道的开发者来说，都是一个值得纪念的日子，在这一天，微软正式推出了其下一代的计算机平台——Microsoft .NET，这不仅仅是一个新的产品，它表明微软将以网络为中心，彻底转换产品研发、发布的方式，改变产品和服务的范围。从此，微软的未来（以及所有微软产品开发者的未来）都将与.NET紧密地连结在一起。

一、关于.NET和C#

.NET是一个可以作为平台支持下一代Internet的可编程结构。其最终目的就是让用户在任何地方、任何时间，以及利用任何设备都能访问他们所需要的信息、文件和程序。

C#是微软为.NET平台量身定做的编程语言，它是一种现代面向对象程序设计语言，使程序员能够快速地在.NET平台上开发种类丰富的应用程序，它继承了C++和Java的语法，去掉了C++中的许多复杂和容易引起问题的东西，比起Java，它又多了一些非常简洁优美的特性。由于语法上的相似性，C++和Java程序员能够很容易地掌握C#。

由于其一流的面向对象设计，开发者使用C#能够更加容易地开发大型应用程序，尤其是分布式应用程序，例如基于Web的程序以及多层系统。C#与.NET框架紧密集成，拥有.NET框架提供的跨语言特征，不仅可以使使用其他语言开发的组件，而且通过交叉语言继承，也可以用C#派生用其他.NET语言（如Visual Basic .NET和Visual C++ .NET）编写的类。

二、多媒体光盘有些什么

专家提示：请先使用本教材中的多媒体教学软件光盘！

本教材与其他编程图书的最大区别在于：由洪恩众多电脑教学专家精心打造的多媒体教学光盘，用绘声绘色的讲解、生动直观的演示，让读者更轻松、更快捷地学习C#编程。当读者对教材中的一些知识点理解不透、或环境配置不正确时，又可以借助光盘中的多媒体讲解内容来迅速地找到解决问题的办法。

试想一下，对于技术性强、步骤繁杂的操作，错了一步就可能影响全局或者学习的兴趣，是停下来找资料、还是需要一位明师的指点呢？现在，这些问题都已经不再是问题，这张教学光盘就充当了名师的角色。尤其在配置服务器和数据库、代码讲解，以及案例剖析时，更能体现该软件的优越性。对于没有编程经验的读者，这张光盘就显得更为重要。读者可以跟着光盘入门，然后通过教材深入学习。这也是“先学光盘后学教材”

的来由。

本教材中的范例程序在光盘的 `Example\Code` 目录下，每个范例程序放在一个目录中，范例目录的名称是 `X_X` 的格式，第一个 `X` 表示第几章，第二个 `X` 表示这一章的第几个范例，这些名称和教材中提及的一一对应。比如：`04_05` 目录，表示这个目录中的范例是：第 4 章的第 5 个范例。

通过光盘主界面上的“源程序”按钮可以打开 `Example\index.htm` 文件，这个文件中是所有范例程序的列表，这样也可以很方便地打开范例程序。不过我们还是建议读者先自己尝试按照教材中的讲解，自己动手实践，然后再参看光盘上的实例。“眼过千遍，不如手过一遍”，在学编程的时候同样是有道理的。

三、本书的读者定位

本教材主要面向 C# 入门者，读者并不需要有 C++ 或 Java 的基础，当然，学习过 C++ 或 Java 的读者学习起本书来应该更容易一些。本教材不是一本高级编程手册，希望对面向对象思想和设计模式有更多了解的读者应该参看更高级的书籍；本教材也不是一本参考手册，对于庞大的 .NET 框架和 .NET 框架类库，都作了基本的介绍，如果希望对其使用有深刻的了解，读者应该参考 Microsoft Visual Studio.NET 文档。

本教材面向入门级的读者，通过阅读本教材，读者能够掌握使用 C# 和 Visual Studio .NET 2003 进行程序开发的基本技能，对于面向对象的思想也会有一定的了解。更重要的是，本教材希望让读者能够获得深入学习的能力，在阅读完本教材之后，读者应该可以比较容易地通过 .NET 文档进一步提高水平。

四、本书的结构

本教材共分十四章，介绍了用 C# 进行开发的各个方面。

第 1 章介绍了什么是 .NET，什么是公共语言运行库，Visual Studio .NET 2003 集成环境的使用等。引导读者开发第一个 C# 应用程序。

第 2 章主要讲解 C# 的基本语法，包括运算符、表达式、流程控制语句、数组的使用、函数和过程的编写以及如何操作字符串。

第 3 章在上一章的基础上深入，引入对象的概念，介绍如何利用对象来进行程序的开发。主要包括类的构造和析构、类方法的编写、方法的重载、类的属性、数组和字符串的介绍等等。

第 4 章介绍如何通过命名空间来组织代码以及程序集的概念。

第 5 章的内容是 C# 的面向对象特征以及如何使用 C# 进行面向对象开发。主要包括类的继承、装箱和拆箱、多态、抽象和封装、接口、委托、类的事件等。

第6章 I/O 操作和文件系统，讲述如何通过 .NET 框架提供的服务来与外界交互，包括如何进行命令行的输入输出、对文件系统的访问等等。

第7章是如何使用 .NET 框架提供的 Windows Forms 类来开发 Windows 窗体应用程序。

第8章的高级主题把一些在前几章不好分类，而在后面又会用到的主题，如集合类、对象的序列化等等归纳在一起逐一讲述。

数据库应用程序的开发可能是现阶段实际工作中最常用到的，我们在第9章中介绍了数据库应用程序开发，本章介绍了如何使用 ADO.NET 连接数据库、获取数据、操作数据等。对 ADO.NET 的本质进行了探讨。

.NET 的一大成就就是统一了 Windows 应用程序和 Web 应用程序的开发，使开发 Web 应用程序的变得像开发 Windows 应用程序那样的轻松。第10章中介绍了 ASP.NET、Web Forms 控件、HTML 控件以及最重要的：如何在 Web 应用程序中访问数据库。

第11章介绍了数据库和 Web 的结合运用。

第12章讲述如何用 C# 进行 Socket 编程。

在第13章移动设备编程中，对如何进行移动设备编程进行了一个初步介绍。

第14章引导读者进行了一个多层 Web 应用程序的分析、设计和开发，是对前面章节的综合应用和总结，同时也渗透了一些设计思想和原则。

五、有关本教材的说明

对于初学者，建议按顺序从头开始学习，先学习了前面的基础知识，再学习后面的应用案例就会倍感轻松了。

当遇到教材中的一些知识点、范例不易理解时，可以借助多媒体光盘来学习，光盘中包含了教材中大部分的实例演示和代码分析，旨在帮助读者更加直观、有效地学习。

六、结束语

凡事贵在坚持，编程更是如此。希望读者在决定加入到编程的队伍中以后，能够勤学不辍。笔者的书架上也有不少书仅仅是翻看了前面的几页，并不是因为内容缺乏吸引力，而是自己没能坚持下去。既然读者已经买回这本书，就请把它读完。很多持之以恒的道理我们都懂，只是需要不断加以提醒，所以读者不妨把这段文字作个明显标记，看到的时候就催促自己奋发前行！

最后，感谢各位读者使用本教材，也希望读者指出本教材中的瑕疵、错漏之处，以便我们在下一版中修正。

前言

如果选择了本教材，那么你一定想成为一名优秀的程序员或者一位计算机专家，最起码也是希望将来能借此谋生。我们推出《编程之道》系列教材的目的，就是要让像你这样的有志之士把愿望变成现实。

“编程之道，其道远兮”。了解一些计算机的人，大概都知道这个行业的技术发展日新月异，“恐怕自己多睡了几个懒觉，就要被这个行业的新技术所抛弃”。难道只有那些聪明绝顶的人才能在这个赚钱的行业里淘金吗？其实只要你掌握了其中之“道”，自然就可以大胆地去淘金了。

那么“道”在何处？让我们静下心来，澄清以下观点：

1. 编程只是实现工具

读一些计算机历史的文章，我们可以清楚地知道，计算机是数学家们的发明，他们并非个个都会编程。现今也会看到这样有趣的现象，不少大师级的计算机技术研究者并非谙熟编程。而各种媒体上的炒作和现实中优厚的待遇往往把编程神秘化了。

计算机技术包括了管理信息、多媒体、计算机网络、人工智能、辅助设计等方面。编程只是这些具体技术在理论研究或者实践中表达算法的过程。编程的人不一定对计算机技术就有一定很高深的了解，但要成为一名编程高手，就必须扎实地掌握丰富的计算机技术。

因此，首先要明白，编程只是实现想法的工具，而解决问题的方法和思想更重要。

2. 学习编程，莫忘基础

如果你想成为一名优秀的程序员，建议你除了学习编程语言、开发工具之外，莫忘扎实地学习如下一些课程：汇编语言、算法和数据结构、计算机体系结构、操作系统原理；还可以补充学习计算机网络、数据库原理等课程。掌握了这些基础知识，才能让你的“编程之道”更宽广、更平坦。当然这里只是列举了几门最基本的课程。

3. 切莫追风，静心修道

不要被那些流行的新技术、新名词所迷惑，不然就要犯狗熊掰玉米的毛病了。

.NET、XML 等等技术固然诱人，能在短时间内让人找到一种满足感，可是如果自己的基础不扎实，就会像在浓雾中行走一样，只能看到眼前，无法看到更远的地方。

这些“包装精美”的洋货掩盖了许多底层的原理，要想真正学技术还是走下云端，脚踏实地地把 C、C++ 以及前面提到的基础知识先学好。尽管这是一个艰苦的过程（不要梦想 21 天怎样……除非你生来就是搞计算机的）。有了这些基础，学起这些时髦的东西

也就是个把星期的事情了。

4. 编程不难！道在基础

比起学英语，学习编程的难度实在太小，毕竟一门编程语言的“单词”只有区区几十个，语法也不多。耗费时间的是学习那些基础的计算机知识，但“道”在其中，自然要学。

《编程之道》系列教材则是为修炼“编程技术之道”的众生设计的经文，希望你在这一系列教材时，可以同时学习前面提到的基础课程，或者提前学习它们。

《编程之道》系列教材分为两大类：入门级、提高级。每一级中都涉及一系列应用广泛的编程语言和开发工具。

入门级是为那些没有编程基础或稍有一点基础的人设计的，从最基本的语法到基本应用，照顾的方面较多。提高级是为有编程基础的人设计的，旨在提高某专业方向上的编程技能及开发工具的深入应用，涉及的内容有：数据库、网络、Internet、多媒体等方面。

凡事贵在坚持，编程更是如此。修“编程之道”在于能够勤学不辍，不断提醒自己，奋发前行！

我们的技术服务热线：(010) 58858203, E-mail: pcbook@goldhuman.com。

北京金洪恩电脑有限公司

董事长 池宇峰

目 录

第 1 章 .NET 与 C#

1.1 .NET 概述	1
1.2 C#语言	2
1.3 公共语言运行库	2
1.4 编写第一个 C#程序	3
1.5 Visual Studio .NET 2003 集成开发环境	5
1.5.1 Visual Studio .NET 2003 集成开发环境窗口介绍	6
1.5.2 Visual Studio .NET 2003 集成开发环境菜单介绍	11
1.5.3 【文件】菜单	11
1.5.4 【编辑】菜单	12
1.5.5 【视图】菜单	12
1.5.6 【工具】菜单	14
1.5.7 【窗口】菜单	16
1.6 第一个 C#程序	17
1.6.1 开发步骤	17
1.6.2 运行	19
1.6.3 点评	20
练 习	20

第 2 章 C#语言程序设计基础

2.1 简单数据类型	21
2.1.1 整数类型	22
2.1.2 布尔类型	23
2.1.3 浮点类型	24
2.1.4 decimal 类型	25
2.1.5 字符类型	26
2.2 值类型和引用类型	27
2.2.1 结构	27
2.2.2 类	29
2.2.3 值类型和引用类型的区别	30
2.3 变量和常量	31
2.3.1 变量	31
2.3.2 常量	32
2.4 表达式与运算符	32
2.4.1 最常用的运算符	33
2.4.2 算术运算符	34
2.4.3 关系运算符	36
2.4.4 逻辑运算符	36
2.4.5 移位运算符	38
2.4.6 运算符的优先级	39
2.5 条件语句 if...else	40

2.6	条件语句 switch...case.....	44
2.7	循环语句 while 和 do.....	47
2.8	循环语句 for.....	50
2.9	跳转语句 break 和 continue.....	53
2.10	错误和异常处理.....	54
2.11	综合实例：猜数字.....	57
	练 习.....	60
第 3 章 C#基于对象的程序设计		
3.1	对象的概念.....	61
3.1.1	生活中的对象概念.....	61
3.1.2	程序设计中的对象.....	62
3.2	类.....	62
3.3	类的字段.....	63
3.3.1	访问修饰符.....	64
3.3.2	静态字段.....	65
3.4	类的方法.....	66
3.4.1	方法的参数.....	68
3.4.2	方法的返回值.....	68
3.4.3	静态方法.....	70
3.5	类的构造函数.....	70
3.6	方法的重载.....	71
3.7	类的属性.....	73
3.8	数组.....	76
3.8.1	使用数组.....	76
3.8.2	多维数组.....	80
3.8.3	交错数组.....	81
3.8.4	foreach 语句.....	83
3.9	字符串.....	84
3.9.1	逐字字符串.....	84
3.9.2	创建字符串.....	86
3.9.3	剪裁字符串.....	89
3.9.4	字符串的比较.....	90
3.9.5	确定字符串的包含关系.....	92
3.9.6	控制大小写.....	94
3.10	类的索引器.....	95
	练 习.....	100
第 4 章 应用程序的组织		
4.1	命名空间.....	101
4.1.1	使用命名空间.....	101
4.1.2	声明命名空间.....	102
4.2	程序集.....	103
4.2.1	使用程序集.....	103
	练 习.....	106

第5章 C#面向对象的程序设计

5.1 类的继承.....	107
5.1.1 认识继承.....	107
5.1.2 Shape 类.....	110
5.1.3 从 Shape 类继承.....	112
5.2 所有类的基类: Object.....	118
5.3 装箱和取消装箱.....	120
5.4 封装.....	122
5.5 多态.....	123
5.6 继承性多态的另一个例子.....	127
5.7 接口.....	128
5.8 委托.....	133
5.9 类的事件.....	136
5.10 自行定义类的事件.....	139
5.11 程序设计演练.....	140
练习.....	144

第6章 IO 操作和文件系统

6.1 .NET 框架提供的 IO 类.....	145
6.1.1 System.Console 类.....	145
6.1.2 System.IO 命名空间.....	147
6.2 Stream 及其派生类.....	148
6.2.1 查询流的特性.....	148
6.2.2 读取流.....	150
6.2.3 写入流.....	153
6.2.4 查找流.....	154
6.3 TextReader/TextWriter 及其派生类.....	157
6.3.1 阅读器.....	157
6.3.2 编写器.....	159
6.4 BinaryReader/BinaryWriter 及其派生类.....	161
6.5 访问文件系统.....	163
6.5.1 文件.....	163
6.5.2 目录.....	165
练习.....	168

第7章 Windows 应用程序开发

7.1 手工编写第一个带有控件的 Windows 程序.....	169
7.1.1 带有控件的 Windows 程序.....	169
7.1.2 Windows Forms.....	171
7.2 使用 IDE 设计 Windows 应用程序.....	172
7.2.1 设计过程.....	172
7.2.2 分析 IDE 编写的代码.....	176
7.3 设计一个稍微复杂的 Windows 应用程序.....	180
7.3.1 界面设计.....	180
7.3.2 编写代码.....	182

7.3.3	运行计算器程序.....	185
7.3.4	一些设计思想.....	186
7.4	窗体的继承: 扩展计算器程序.....	189
7.4.1	继承一个窗体.....	189
7.4.2	添加记忆功能.....	192
7.4.3	对程序的解释.....	195
	练习.....	196
第 8 章 高级主题		
8.1	使用集合类.....	197
8.1.1	System.Collections 的整体设计.....	197
8.1.2	ArrayList.....	199
8.1.3	Queue 和 Stack.....	203
8.1.4	Hashtable.....	207
8.1.5	SortedList.....	211
8.2	更多了解集合类.....	214
8.3	控制垃圾回收器.....	218
8.4	序列化和反序列化.....	221
8.4.1	二进制序列化.....	222
8.4.2	XML 序列化.....	226
8.5	使用 Attribute.....	229
8.6	运行时类型信息.....	233
8.6.1	System.Type 类.....	235
8.6.2	反射.....	237
	练习.....	240
第 9 章 访问数据库		
9.1	Northwind 示例数据库.....	241
9.1.1	Northwind 数据库关系图.....	241
9.1.2	Northwind 数据库的结构.....	242
9.2	ADO.NET 体系结构.....	245
9.2.1	DataSet.....	246
9.2.2	Connection.....	247
9.2.3	Command.....	248
9.2.4	DataReader.....	249
9.2.5	DataAdapter.....	252
9.3	主-从关系数据.....	257
9.4	DataSet 详解.....	259
9.4.1	DataSet 到 XML.....	259
9.4.2	XML 到 DataSet.....	261
9.4.3	用代码建立 DataSet.....	266
9.4.4	类型化的 DataSet.....	268
	练习.....	272
第 10 章 Web 应用程序开发初步		
10.1	第一个 ASP.NET 程序.....	274

10.1.1	检查 IIS 的安装	274
10.1.2	配置 IIS	276
10.1.3	编写和运行	277
10.1.4	ASP.NET 的执行过程	278
10.2	服务器端代码块	279
10.3	分离页面和代码	281
10.4	ASP.NET 服务器控件	283
10.4.1	Label 控件	283
10.4.2	Button 控件	284
10.4.3	DropDownList 控件	286
10.4.4	对服务器控件使用后端代码	287
10.5	一个实例: 简易计算器	289
10.6	使用客户端验证	291
练 习		294
第 11 章 Web 应用程序与数据库		
11.1	在页面上显示数据	295
11.2	使用 DataSet	298
11.3	使用 Repeater	300
11.4	使用 DataList	303
11.5	将数据更改提交到数据库	309
练 习		314
第 12 章 图形和图像的绘制与编辑		
12.1	GDI+	315
12.1.1	Graphics 类	315
12.2	绘制简单图形	315
12.3	填充图形	320
12.4	显示文本	323
12.4.1	有关字体的基本知识	323
12.4.2	绘制文本	323
12.5	轨迹	325
12.6	显示图像	327
12.7	图像的缩放和裁剪	330
12.8	图像的扭曲和变形	333
练 习		336
第 13 章 网络编程		
13.1	Socket	337
13.2	可插接式协议	339
13.3	使用 TCP 和 UDP 通讯	342
13.3.1	使用 TCP 的 HelloServer	342
13.3.2	使用 UDP 的 HelloServer	345
13.4	使用 Socket 类	348
练 习		352

第 14 章 综合实例

14.1 Duwamish7 总体结构	353
14.2 Duwamish7 数据库	355
14.2.1 Books.....	355
14.2.2 Items.....	356
14.2.3 Orders.....	356
14.2.4 DailyPick.....	356
14.3 Duwamish7 各层结构	356
14.3.1 业务实体.....	357
14.3.2 数据访问.....	357
14.3.3 业务规则.....	358
14.3.4 业务外观.....	358
14.4 Duwamish7 业务实体分析	358
14.4.1 代码分析.....	358
14.4.2 可能的改进.....	362
14.5 Duwamish7 数据访问层分析	362
14.5.1 Dispose() 方法分析	362
14.5.2 Categories 类分析.....	364
14.5.3 Books 类分析.....	366
14.5.4 Orders 类分析.....	366
14.5.5 Customers 类分析.....	370
14.6 Duwamish7 业务规则层分析	371
14.7 Duwamish7 业务外观层分析	373
14.7.1 OrderSystem 类分析.....	373
14.7.2 ProductSystem 类分析.....	376
14.7.3 CustomerSystem 类分析	377
14.8 Duwamish7 Web 层情景分析	378
14.8.1 情景一：建立新帐户	379
14.8.2 情景二：购物.....	385
14.9 总结.....	396

附录 练习答案

第 1 章 .NET 与 C#	397
第 2 章 C#语言程序设计基础	397
第 3 章 C#基于对象的程序设计	398
第 4 章 应用程序的组织.....	399
第 5 章 C#面向对象的程序设计	400
第 6 章 IO 操作和文件系统.....	401
第 7 章 Windows 应用程序开发.....	404
第 8 章 高级主题.....	404
第 9 章 访问数据库.....	406
第 10 章 Web 应用程序开发初步.....	407
第 11 章 Web 应用程序与数据库.....	407
第 12 章 图形和图像的绘制与编辑.....	407
第 13 章 网络编程.....	408

第 1 章 .NET 与 C#

本章要点：

- 什么是.NET
- C# 语言和公共语言运行库
- 认识 Visual Studio .NET 2003

1.1 .NET 概述

.NET 恐怕是微软公司自从创立以来进行的最大一次改变，它是一整套平台，内容几乎涵盖程序开发的任何方面，使用微软技术建立的软件迟早有一天都要转化到.NET 平台上来。在发行之初，.NET 曾经被称为一次“豪赌”，而现在看来，微软在这次“豪赌”中又下对了赌注。

那么，.NET 究竟是什么东西呢？实际上，从程序员的眼光看来，.NET 就是一组用于建立 Web 服务器应用程序和 Windows 桌面应用程序的软件组件，用该平台创建的应用程序在公共语言运行库（Common Language Runtime, CLR）的控制下运行。公共语言运行库是一个软件引擎，用来加载应用程序，确认它们可以没有错误地执行，进行相应的安全许可验证，执行应用程序，然后在运行完成后将它们清除。类库集提供了使应用程序可以读写 XML 数据、在 Internet 上通信、访问数据库等的代码。所有的类库都建立在一个基础的类库之上，它提供管理使用最为频繁的数据类型（例如数值或文本字符串）的功能，以及诸如文件输入/输出等底层功能。

所以，.NET 框架的整体结构可以用图 1-1 来说明。

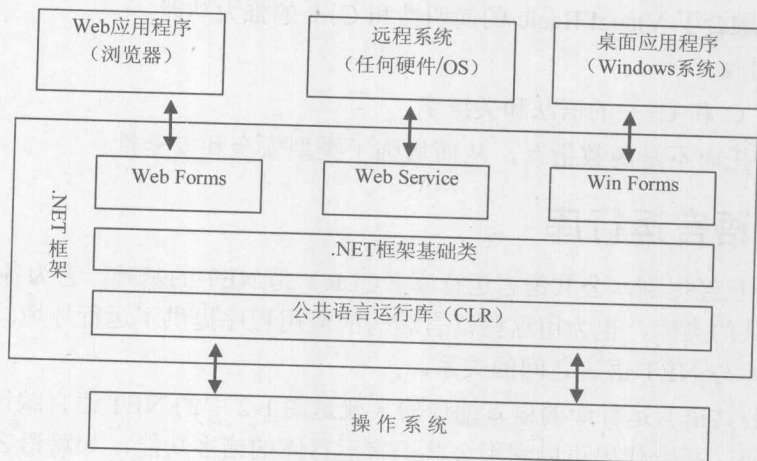


图 1-1 .NET 框架体系结构

从图中可以看出，.NET 框架在操作系统之上为程序员提供了一个环境，让程序员可以

方便地编写出 Windows 应用程序、Web 应用程序、移动设备应用程序等等，还可以通过 Web Service 来与任何系统平台实现互操作。

1.2 C# 语言

在很长的一段时间里，C 和 C++ 一直都是最有生命力的通用程序设计语言，它们都提供了丰富的功能、高度的灵活性和强大的底层控制能力。然而相应的，它们（尤其是 C++）也具有漫长的学习周期，而且它们对于组件、Web 开发等等的有限支持也显得不那么符合时代潮流了。因此，虽然在系统级的开发中 C 和 C++ 仍然是主流，但是在很多应用领域，如 Web 应用中，使用这类语言的人已经不多见了。

20 世纪 90 年代中期，Sun 微系统发布了 Java 语言，这是一种强大的通用程序设计语言，它以 C 和 C++ 为基础，加入了更多更纯粹的面向对象特性。用它所编写的程序运行在 Java 虚拟机上，因此只要在任何计算机或设备上安装 Java 虚拟机，就可以实现所谓的“一次编写，到处运行”。经过这么多年的发展，Java 已经相当成熟，成为了工业界的主流程序设计语言之一。正是因为这个原因，微软一直视 Java 为眼中钉，整个 .NET 平台就可以说是微软自己的 Java 虚拟机。但是 .NET 并不是一种语言，它需要一种能够与 Java 分庭抗礼的语言，为此，微软重金从 Borland 挖来了技术天才 Anders Hejlsberg（Delphi 之父）为 .NET 平台量身打造了 C# 语言。

C# 是一种现代的面向对象（微软也说是面向组件）程序设计语言，它能够最大限度地发挥 .NET 平台的威力，使程序员能够在 .NET 平台上快速开发种类丰富的应用程序。

C# 的语法基于 C/C++，在很多方面还很类似 Java，这样，数量庞大的 C/C++/Java 程序员都能够很快的熟悉 C# 语法，而它的简洁与优美对于这些程序员来说也具有巨大的吸引力。总的说来，C# 具有以下的特点：

- 1) 完全面向对象的设计
- 2) 非常强的类型安全
- 3) 很好地融合了 Visual Basic 的简明性和 C++ 的强大功能
- 4) 垃圾回收
- 5) 类似于 C 和 C++ 的语法和关键字
- 6) 使用委托而不是函数指针，从而增强了类型安全性和安全性

1.3 公共语言运行库

在 1.1 节中已经提到，公共语言运行库（CLR）是 .NET 的基础，它为各种各样的 .NET 语言提供了公共的类库，也为用这些语言编写的应用程序提供了运行环境。图 1-2 说明了公共语言运行库与 .NET 语言之间的关系。

使用针对公共语言运行库的语言编译器（就是图 1-2 中的 .NET 语言编译器）开发的代码称为托管代码。托管代码可以利用公共语言运行库的诸多功能，如跨语言集成、跨语言异常处理、增强的安全性、版本控制和部署支持、简化的组件交互模型、调试和分析服务等。

公共语言运行库的另外一大好处就是，用一种语言设计的对象，可以在另一种语言中