

轻松学

# C#编程

祝红涛 王伟平 郝相林 等编著



DVD-ROM

## 1780分钟多媒体教学视频

- **结构独特**: 每个知识点按照“概念→语法描述→实例描述→实例应用→运行结果→示例分析”的模式进行讲解
- **形式新颖**: 用准确的语言总结概念, 用直观的图示演示过程, 用详细的注释解释代码, 用形象的比喻帮助记忆
- **内容丰富**: 重要知识点覆盖全面, 实例丰富多彩
- **附赠光盘**: 在随书附赠的光盘中配备了完整的视频教学讲座和本书所使用到的全部项目文件与代码
- **技术支持**: 读者可直接登录[www.itzcn.com](http://www.itzcn.com)与作者取得联系, 会在第一时间内得到帮助



化学工业出版社

轻松学  
编程

轻松学

# C#编程

祝红涛 王伟平 郝相林 等编著



化学工业出版社

· 北京 ·

C#是随 Visual Studio.NET 发布的一种新语言,主要以组件为基础进行软件开发。在本书中首先介绍搭建、配置开发环境等基础知识;接着介绍 C#语言中的基础语法知识,例如基本数据类型、变量和常量、运算符、流程控制语句以及数组等;再重点介绍面向对象编程技术,例如类、对象、方法、接口、抽象类、封装、继承以及多态等;然后介绍一些常用的扩展技术,例如集合、异常处理等;之后介绍字符串和正则表达式、Windows 窗体编程、MDI 程序设计、使用 ADO.NET 访问数据库以及文件和 IO 流;最后介绍了一个综合案例——图书管理系统。

本书具有知识全面、实例精彩、指导性强的特点,力求以全面的知识及丰富的实例来指导读者透彻地学习 C#各方面的知识。本书可以作为 C#基础入门学习书籍,也可以帮助中级读者提高编程技能。

## 图书在版编目(CIP)数据

轻松学 C#编程 / 祝红涛, 王伟平, 郝相林等编著. —北京:  
化学工业出版社, 2012. 4

(轻松学编程)

ISBN 978-7-122-13421-9

ISBN 978-7-89472-586-8 (光盘)

I. 轻… II. ①祝… ②王… ③郝… III. C 语言-程序设计  
IV. TP312

中国版本图书馆 CIP 数据核字 (2012) 第 019264 号

责任编辑: 瞿 微

装帧设计: 王晓宇

出版发行: 化学工业出版社 (北京市东城区青年湖南街 13 号 邮政编码 100011)

印 装: 化学工业出版社印刷厂

787mm×1092mm 1/16 印张 22 字数 563 千字 2012 年 4 月北京第 1 版第 1 次印刷

购书咨询: 010-64518888 (传真: 010-64519686)

售后服务: 010-64518899

网 址: <http://www.cip.com.cn>

凡购买本书, 如有缺损质量问题, 本社销售中心负责调换。

定 价: 45.00 元 (含 1DVD-ROM)

版权所有 违者必究

C#是一种面向对象的编程语言，主要强调以组件为基础的软件开发。该语言自从由 Microsoft 公司推出以来，就快速流行，进而成为使用 .NET Framework 的 Windows 和 Web 开发人员无可争议的首选语言。C#语言既提供了 Visual Basic 的简单易用性，同时也提供了 Java 和 C++语言的灵活性和强大功能。与其他 .NET 语言一样，C#语言也是在 .NET Framework 平台上开发的，该语言在 .NET Framework 强有力的支持下，使开发人员能够方便快捷地编写各类应用程序及组件。

## ● 本书内容

第 1 章：创建 C#开发环境。介绍了 C#语言的特点和 .NET Framework 的概述，.NET Framework 的环境配置，C#程序的开发步骤。

第 2 章：C#编程基础。介绍了 C#语言中的基本数据类型，变量和常量以及运算符。

第 3 章：控制语句。介绍了 C#语言中的选择语句、循环语句和跳转语句。

第 4 章：数组。主要介绍了一维数组的创建和使用，复合数组，以及数组元素的复制和排序等操作。

第 5 章：面向对象的基础。介绍了面向对象编程，类、对象、方法、构造函数和析构函数的创建和使用，以及 C#语言中常用的访问修饰符。

第 6 章：面向对象的关键技术。主要介绍了面向对象的封装、继承、多态三大特征的应用，抽象类与接口的使用以及 base 关键字和方法重写。

第 7 章：集合。主要介绍了 C#语言中的内置集合和自定义集合。

第 8 章：字符串和正则表达式。介绍了字符串的基础知识，String 类和 StringBuilder 类中方法的使用，以及正则表达式的使用以及字符串的格式化。

第 9 章：异常处理。主要介绍了异常的类型，异常处理的机制以及异常类型的种类，以及如何抛出异常、声明异常以及自定义异常。

第 10 章：创建 Windows 窗体应用程序。主要介绍了窗体常用的一些控件。

第 11 章：MDI 程序设计。主要介绍 MDI 应用程序的创建和 MDI 应用程序中菜单的使用。

第 12 章：使用 ADO.NET 数据库访问技术。主要介绍了 ADO.NET 一些相关的知识，如何连接到数据库，以及使用一些控件将数据显示并对这些数据进行操作。

第 13 章：文件和 IO 流。主要介绍 System.IO 的相关知识，如何操作目录、文件以及文件对话框的使用，如何读写文件。

第 14 章：图书管理系统。本章使用 C#语言开发了一个窗体应用程序——图书管理系统，通过该系统，读者可以了解 C#项目的开发流程。

## ● 本书特色

本书中引用了大量的动手实践来讲解 C#程序开发知识，使读者更容易掌握 C#应用开发。

本书难度适中，内容由浅入深，实用性强，覆盖面广，条理清晰。

- 结构独特

通过“概念、语法描述、动手实践”的模式将每个知识与实际应用中的问题相结合。

- 形式新颖

用准确的语言总结概念，用直观的图示演示过程，用形象的比喻帮助记忆。

- 内容丰富

涵盖了实际开发中所遇到的语法基础、控制语句、数组、字符串、WinForm、ADO.NET 以及 MDI 等方面的热点问题。

- 随书光盘

本书为实例配备了视频教学文件，读者可以通过视频文件更加直观地学习 C# 的使用知识。

- 网站技术支持

读者在学习或者工作的过程中，如果遇到实际问题，可以直接登录 [www.itzcn.com](http://www.itzcn.com) 与我们取得联系，编者会在第一时间给予帮助。

- 贴心的提示

为了便于读者阅读，全书还穿插着一些技巧、提示等小贴士，体例约定如下。

**提示：**通常是一些贴心的提醒，让读者加深印象，提供建议或者解决问题的方法。

**注意：**提出学习过程中需要特别注意的一些知识点和内容等相关信息。

**技巧：**通过简短的文字，指出知识点在应用时的一些小窍门。

## ● 读者对象

本书具有知识全面、实例精彩、指导性强的特点，力求以全面的知识及丰富的实例来指导读者透彻地学习 C# 各方面的知识。本书可以作为 C# 基础入门学习书籍，也可以帮助中级读者提高编程技能。

本书适合以下人员阅读和学习。

- 没有编程基础的人员。
- C# 初学者以及在校学生。
- 各大中专院校的在校学生和相关授课老师。
- 其他 C# 从业人员。

本书主要由祝红涛、王伟平、郝相林编写，其他参与编写、资料整理、程序开发的人员还有张小素、朱海洋、李海燕、陈军红、刘珍霞、王俊伟、韩林林等。

由于编者水平有限，书中难免存在不足和疏漏之处，恳请读者批评指正。

编者  
2012 年 1 月

<b>第 1 章 创建 C#开发环境</b> .....	1
1.1 C#语言简介 .....	2
1.2 .NET Framework 概述 .....	2
1.3 程序集 .....	4
1.4 配置.NET Framework 环境 .....	4
1.4.1 安装 Visual Studio 2010.....	5
1.4.2 熟悉 Visual Studio 2010.....	7
1.5 用 C#创建.NET 应用程序 .....	8
1.5.1 设置应用程序的属性 .....	8
1.5.2 使用 C#创建控制台程序 .....	8
1.5.3 使用 C#创建 Windows 窗体程序 .....	11
1.6 命名空间 .....	14
1.6.1 命名空间的结构 .....	15
1.6.2 定义和引用命名空间 .....	16
1.6.3 引用命名空间 .....	17
1.7 实践案例：使用控制台程序模拟登录系统 .....	18
1.8 动手练一练 .....	18
1.8.1 练习题 .....	18
1.8.2 上机实践 .....	19
<b>第 2 章 C#编程基础</b> .....	20
2.1 数据类型 .....	21
2.1.1 值类型 .....	21
2.1.2 引用类型 .....	24
2.1.3 数据类型转换 .....	25
2.2 变量和常量 .....	26
2.2.1 变量 .....	26
2.2.2 常量 .....	28
2.3 运算符 .....	28
2.3.1 算术运算符 .....	29
2.3.2 赋值运算符 .....	30
2.3.3 关系运算符 .....	32
2.3.4 逻辑运算符 .....	33
2.3.5 特殊运算符 .....	33
2.3.6 条件运算符 .....	34
2.3.7 运算符的优先级 .....	34

# Contents

2.4	装箱和拆箱 .....	35
2.4.1	装箱 .....	35
2.4.2	拆箱 .....	36
2.5	C#关键字 .....	36
2.6	实践案例:逻辑运算符的真值表 .....	37
2.7	动手练一练 .....	39
2.7.1	练习题 .....	39
2.7.2	上机实践 .....	40
<b>第3章</b>	<b>控制语句 .....</b>	<b>41</b>
3.1	条件语句 .....	42
3.1.1	if 语句 .....	42
3.1.2	if else 语句 .....	43
3.1.3	if else if else 语句 .....	44
3.1.4	嵌套 if 语句 .....	45
3.1.5	switch 语句 .....	47
3.2	循环语句 .....	49
3.2.1	for 循环语句 .....	49
3.2.2	while 循环语句 .....	50
3.2.3	do while 循环语句 .....	51
3.2.4	foreach 语句 .....	52
3.3	跳转语句 .....	53
3.3.1	break 语句 .....	53
3.3.2	continue 语句 .....	54
3.3.3	return 语句 .....	55
3.3.4	goto 语句 .....	55
3.4	实践案例:编写简单的帮助系统 .....	57
3.5	动手练一练 .....	59
3.5.1	练习题 .....	59
3.5.2	上机实践 .....	61
<b>第4章</b>	<b>数组 .....</b>	<b>62</b>
4.1	一维数组 .....	63
4.1.1	声明数组 .....	63
4.1.2	初始化数组 .....	63
4.1.3	访问数组 .....	64
4.1.4	遍历数组 .....	65
4.2	复合数组 .....	66
4.2.1	二维数组 .....	66

4.2.2	多维数组 .....	67
4.2.3	锯齿数组 .....	68
4.3	数组操作 .....	69
4.3.1	Array 类概述 .....	69
4.3.2	Array 类的简单操作 .....	69
4.3.3	复制数组 .....	71
4.3.4	对数组排序 .....	73
4.4	实践案例：拆分锯齿数组 .....	76
4.5	动手练一练 .....	77
4.5.1	练习题 .....	77
4.5.2	上机实践 .....	78
<b>第 5 章</b>	<b>面向对象的基础 .....</b>	<b>79</b>
5.1	面向对象编程 .....	80
5.2	类 .....	81
5.2.1	定义类 .....	81
5.2.2	类的字段 .....	82
5.2.3	类的属性 .....	83
5.3	对象 .....	84
5.3.1	创建对象 .....	84
5.3.2	对象的使用 .....	85
5.3.3	对象的生存周期 .....	86
5.3.4	对象的清理 .....	87
5.4	方法 .....	87
5.4.1	无参方法 .....	87
5.4.2	有参方法 .....	88
5.4.3	方法的重载 .....	89
5.4.4	静态方法 .....	90
5.5	构造函数 .....	92
5.6	析构函数 .....	94
5.7	访问修饰符 .....	96
5.8	实践案例：制作简单的计算器 .....	98
5.9	动手练一练 .....	100
5.9.1	练习题 .....	100
5.9.2	上机实践 .....	101
<b>第 6 章</b>	<b>面向对象的关键技术 .....</b>	<b>102</b>
6.1	封装 .....	103
6.2	继承 .....	103



6.2.1	继承的意义 .....	103
6.2.2	如何定义派生类 .....	104
6.2.3	作用域和继承 .....	106
6.2.4	覆盖基类成员的方法 .....	107
6.2.5	this 关键字的使用 .....	108
6.3	base 关键字和方法重写 .....	109
6.3.1	方法重写 .....	109
6.3.2	base 关键字 .....	111
6.4	抽象类和密封类 .....	112
6.4.1	抽象类 .....	112
6.4.2	密封类 .....	113
6.5	多态 .....	114
6.5.1	什么是多态 .....	114
6.5.2	多态的实现 .....	115
6.6	接口 .....	117
6.6.1	接口的定义和实现 .....	117
6.6.2	接口与继承 .....	118
6.7	实践案例：实现购物系统中的用户信息 .....	120
6.8	动手练一练 .....	122
6.8.1	练习题 .....	122
6.8.2	上机实践 .....	123
<b>第 7 章</b>	<b>集合 .....</b>	<b>124</b>
7.1	C#内置集合 .....	125
7.1.1	ArrayList 集合 .....	125
7.1.2	使用堆栈 (Stack 集合) .....	127
7.1.3	使用队列 (Queue 集合) .....	128
7.1.4	使用 Hashtable 集合 .....	129
7.1.5	使用 BitArray 集合 .....	131
7.1.6	使用 SortedList 集合 .....	133
7.2	自定义集合 .....	134
7.2.1	声明集合 .....	134
7.2.2	为集合添加对象 .....	137
7.2.3	删除集合中的元素 .....	139
7.2.4	在集合中检索元素 .....	140
7.3	实践案例：模拟树形菜单 .....	141
7.4	动手练一练 .....	145
7.4.1	练习题 .....	145
7.4.2	上机实践 .....	146

<b>第 8 章 字符串和正则表达式</b> .....	147
8.1 字符串概述 .....	148
8.2 String 类 .....	148
8.2.1 String 类的构造函数 .....	148
8.2.2 String 类的属性和索引器 .....	150
8.2.3 比较字符串 .....	150
8.2.4 定位与查找字符串 .....	153
8.2.5 提取字符串 .....	154
8.2.6 拆分字符串 .....	155
8.2.7 String 类中的其他常用方法 .....	156
8.3 StringBuilder 类 .....	159
8.3.1 追加字符串 .....	159
8.3.2 使用 AppendFormat 对字符串进行处理 .....	160
8.3.3 插入字符串 .....	161
8.3.4 替换字符串 .....	162
8.4 正则表达式 .....	163
8.4.1 正则表达式的语法规则 .....	163
8.4.2 基本类 .....	165
8.4.3 匹配、组合和捕获 .....	167
8.5 格式化字符 .....	168
8.6 实践案例：字符串操作系统 .....	171
8.7 动手练一练 .....	175
8.7.1 练习题 .....	175
8.7.2 上机实践 .....	177
<b>第 9 章 异常处理</b> .....	178
9.1 异常概述 .....	179
9.2 异常处理 .....	179
9.3 异常类 .....	182
9.3.1 基于类型的筛选异常 .....	183
9.3.2 System.Exception 类 .....	184
9.3.3 内部异常类 .....	186
9.4 自定义的异常类 .....	190
9.5 实践案例：模拟邮箱注册 .....	191
9.6 动手练一练 .....	194
9.6.1 练习题 .....	194
9.6.2 上机实践 .....	195
<b>第 10 章 创建 Windows 窗体应用程序</b> .....	196

# Contents

10.1	Windows 窗体控件概述 .....	197
10.2	基本控件 .....	197
10.2.1	TextBox 控件 .....	197
10.2.2	MaskedTextBox 控件 .....	199
10.3	显示信息的控件 .....	200
10.3.1	Label 控件 .....	200
10.3.2	LinkLabel 控件 .....	201
10.4	图形和图像类控件 .....	203
10.4.1	ImageList 控件 .....	203
10.4.2	PictureBox 控件 .....	205
10.5	按钮类控件 .....	206
10.5.1	Button 控件 .....	206
10.5.2	RadioButton 控件 .....	207
10.5.3	CheckBox 控件 .....	209
10.6	列表类控件 .....	210
10.6.1	ListBox 控件 .....	210
10.6.2	ComboBox 控件 .....	213
10.6.3	ListView 控件 .....	216
10.7	容器类控件 .....	220
10.7.1	Panel 控件 .....	220
10.7.2	TabControl 控件 .....	221
10.7.3	GroupBox 控件 .....	223
10.8	实践案例：员工管理系统 .....	223
10.9	动手练一练 .....	226
10.9.1	练习题 .....	226
10.9.2	上机实践 .....	227
<b>第 11 章</b>	<b>MDI 程序设计 .....</b>	<b>228</b>
11.1	MDI 概述 .....	229
11.2	MDI 窗体 .....	229
11.2.1	创建 MDI 应用程序 .....	229
11.2.2	标准窗体 .....	231
11.3	MDI 应用程序中的菜单 .....	232
11.3.1	MDI 程序菜单的特征 .....	233
11.3.2	创建 MDI 菜单 .....	233
11.3.3	合并菜单 .....	235
11.3.4	上下文菜单 .....	237
11.4	MDI 窗体事件 .....	239
11.5	管理子窗体 .....	242

11.5.1	排列子窗体 .....	242
11.5.2	使用菜单选择 MDI 子窗体 .....	244
11.6	RichTextBox 控件 .....	245
11.6.1	使用 RichTextBox 控件读写文件 .....	246
11.6.2	格式化富文本框 .....	248
11.7	实践案例：制作简单的记事本 .....	251
11.8	动手练一练 .....	255
11.8.1	练习题 .....	255
11.8.2	上机实践 .....	256
<b>第 12 章</b>	<b>使用 ADO.NET 数据库访问技术 .....</b>	<b>257</b>
12.1	ADO.NET 概述 .....	258
12.1.1	ADO.NET 命名空间 .....	258
12.1.2	数据库特定的类（提供者对象） .....	259
12.2	ADO.NET 连接数据库 .....	260
12.2.1	定义一个连接字符串 .....	261
12.2.2	存储连接字符串 .....	263
12.2.3	读取连接字符串 .....	264
12.2.4	测试连接 .....	265
12.3	操作数据库 .....	267
12.3.1	使用 DataAdapter 填充 DataSet 对象 .....	267
12.3.2	使用 Command 对象对数据进行操作 .....	269
12.3.3	DataReader 的使用 .....	271
12.3.4	DataTable 和 DataView 的使用 .....	272
12.3.5	使用 Command Builder 生成 SQL 语句 .....	274
12.4	DataSet 概述 .....	276
12.4.1	更新 DataSet .....	276
12.4.2	向 DataSet 中添加数据 .....	277
12.4.3	为 DataSet 筛选和排序 .....	279
12.5	DataGridView 控件 .....	281
12.5.1	认识 DataGridView 控件 .....	281
12.5.2	使用 DataGridView 显示数据 .....	281
12.6	实践案例：学生信息管理系统 .....	283
12.7	动手练一练 .....	286
12.7.1	练习题 .....	286
12.7.2	上机实践 .....	287
<b>第 13 章</b>	<b>文件和 IO 流 .....</b>	<b>288</b>
13.1	System.IO 简介 .....	289

13.2	目录	289
13.3	文件	293
13.4	文件对话框	294
13.4.1	OpenFileDialog 类的使用	294
13.4.2	SaveFileDialog 类的使用	297
13.5	读写文件	299
13.5.1	使用 StreamWriter 类写入顺序文件	299
13.5.2	使用 StreamReader 类读取顺序文件	301
13.5.3	使用 BinaryWriter 和 BinaryReader 类读写二进制文件	302
13.6	实践案例: 制作文本文件读写器	305
13.7	动手练一练	307
13.7.1	练习题	307
13.7.2	上机实践	308
<b>第 14 章</b>	<b>图书管理系统</b>	<b>309</b>
14.1	系统概述	310
14.2	系统功能模块	310
14.2.1	登录模块	310
14.2.2	人员模块	311
14.2.3	图书模块	311
14.3	数据库设计	312
14.3.1	数据库需求分析	312
14.3.2	数据库表设计	312
14.4	公用模块	314
14.4.1	定义程序入口	314
14.4.2	连接数据库	314
14.4.3	主界面模块	315
14.5	登录模块	317
14.6	人员模块	319
14.6.1	客户管理	319
14.6.2	员工管理	326
14.7	图书模块	328
14.7.1	库存管理	328
14.7.2	租借图书	331
14.7.3	历史记录	334
14.7.4	归还图书	335
14.8	总结	336
	<b>练习题参考答案</b>	<b>337</b>

# 第 1 章

## 创建 C# 开发环境

C#是微软（Microsoft）为.NET Framework 量身订做的程序语言，任何程序语言都需要一个开发环境来支持它的运行，而 C#语言需要的是.NET Framework 环境。在本章中将会详细介绍 C#语言、.NET Framework 以及程序集的概念，如何配置.NET Framework 环境，通过案例来说明如何使用 C#来创建应用程序，并针对应用程序的命名空间做了进一步的讲解。

### 本章学习要点：

- 了解 C#语言
- 熟悉.NET Framework 的概念和 Visual Studio 的关系
- 了解什么是程序集
- 熟练掌握.NET Framework 环境的配置
- 了解如何设置应用程序的属性
- 了解如何使用 C#创建控制台程序
- 熟悉如何使用 C#创建窗体程序
- 掌握命名空间的结构和使用

## 1.1 C#语言简介

C#是用于创建运行在.NET 公共语言运行库上的应用程序的语言之一，它从 C 语言和 C++语言演化而来，是 Microsoft 专门为使用.NET 平台而创建的，并且考虑了其他语言的许多优点，例如像 Visual Basic 的易用性。

使用 C#可以创建以下几种常见的应用程序。

- **Windows 应用程序：**如 Microsoft Office，它具有我们很熟悉的 Windows 外观和操作方式。.NET Framework 的 Windows Form 模块是一个控件，使用它可以建立 Windows 用户界面（UI）。
- **Web 应用程序：**这些是 Web 页，可以通过任何 Web 浏览器查看。.NET Framework 包括一个动态生成 Web 内容的强大系统，允许个性化、实现安全性等，就是我们所说的 ASP.NET。我们可以使用 C#通过 Web Form 创建 ASP.NET 应用程序。
- **Web 服务：**这是创建各种分布式应用程序的新方式，使用 Web 服务可以通过 Internet 虚拟交换数据，无论使用什么语言创建 Web，也无论 Web 服务驻留在什么系统上，都使用一样简单的语法。

为了确保 C#程序和库能够以兼容的方式逐步演进，在 C#的设计中充分强调了版本控制。C#的设计在某些方面直接考虑到了版本控制的需要，其中包括单独使用的 virtual 和 override 修改、方法重载决定规则以及对显式接口成员声明的支持。

总之，C#是一个易于使用的、能够开发出功能强大、安全、稳定的应用程序的语言，在本书的其余部分将详细描述 C#的这些特性。

## 1.2 .NET Framework 概述

Microsoft 发布的.NET Framework 简称为.NET，是 Visual Studio.NET 应用程序开发环境的核心，同样也是支持生成和运行下一代应用程序和 Web 服务的内部 Windows 组件，它提供了托管执行环境、简化的开发和部署以及与各种编程语言的集成。

总而言之，如果想要开发和运行.NET 应用程序，就必须安装.NET Framework，而.NET Framework 在安装 Visual Studio.NET 时自动安装。已经安装好的.NET Framework 能够实现下列目标。

- 提供一个一致的、面向对象的编程环境，而无论对象代码在本地存储和执行，还是在本地执行但在 Internet 上分布，或者在远程执行。
- 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。
- 提供一个可提高代码（包括由未知的或不完全受任的第三方创建的代码）执行安全性的代码执行环境。
- 提供一个可消除脚本环境或解释环境性能问题的代码执行环境。
- 使开发人员的经验在面对类型大不相同的应用程序（例如基于 Windows 的应用程序

和基于 Web 的应用程序)时保持一致。

- 按照工业标准生成所有通信,以确保基于 .NET Framework 的代码可与其他任何代码集成。

.NET Framework 包括两个组件,分别是:公共语言运行时(Common Language Runtime, CLR)和 .NET Framework 类库(FCL)。其中,公共语言运行时提供内存管理、线程管理和远程处理等核心服务。.NET Framework 类库是一个综合性的、面向对象的、可重用类型的集合。

.NET Framework 在安装 Visual Studio.NET 时自动安装,那么它们之间有什么关系呢?

Visual Studio.NET 包含了创建窗体和管理 XML 文档的可视化设计器,以及管理大型多文件工程并把这些工程编译为可执行文件的工具。Visual Studio.NET 还包含部署应用程序以及把 .NET Framework 部署到其他计算机的工具。尽管不使用 Visual Studio.NET 我们仍能开发出功能强大的应用程序,但是使用 Visual Studio.NET 无疑会更高效。Visual Studio.NET 与 .NET Framework 之间的关系如图 1-1 所示。

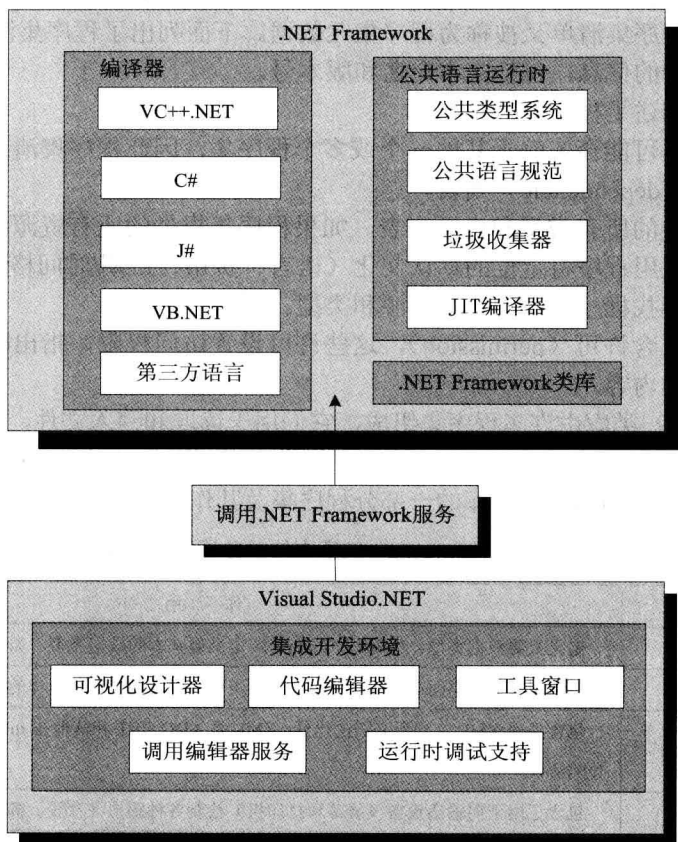


图 1-1 Visual Studio.NET 与 .NET Framework 之间的关系

从图 1-1 中可以看出, Visual Studio.NET 依赖于 .NET Framework 提供的服务,这些服务包括 Microsoft 公司或者第三方提供的语言编译器。这些语言编译器是 .NET Framework 自身的组成部分,而不属于 Visual Studio.NET。Visual Studio.NET 提供了大量的工具来调用某一种安装的编译器。



## 1.3 程序集

程序集 (assembly) 是 .NET Framework 应用程序的构造块, 我们创建的每一个应用程序便是一个程序集, 它构成了部署、版本控制、重复使用、激活范围控制和安全权限的基本单元。程序集是为协同工作而生成的类型和资源的集合, 这些类型和资源就构成了一个逻辑功能单元。程序集具有以下特点。

- 名称必须是全局唯一: 当一个程序集以共享程序集部署时, 开发者必须确保程序集的名称不会与其他程序集名称发生冲突。
- 将程序集发布者的身份保存在程序集中: 开发者必须确保一个程序集来自其实际的发布者, 欺骗是不可能的。

每一个程序集都包含一个程序集清单 (assembly manifest), 它描述了程序集以及组成程序集的各个模块。程序集清单又被称为程序集元数据。下面列出了程序集清单的基本元素。

- 程序集身份的信息, 包括文本命名和版本号。
- 程序清单描述了程序集的内容。
- 一个程序集可能会依赖于其他一个或多个程序集, 因此程序集清单中还包含了一个依赖关系 (dependencie) 列表。
- 程序集提供的所有类型和资源列表。如果程序集提供的所有资源都定位了, 则清单也含有该应用程序将定位的默认文化 (语言、货币、日期/时间格式等)。CLR 使用这项信息查找程序集内的特定资源和类型。
- 清单中还包含许可 (permission), 这些许可设置访问权限, 指出哪些程序才可以访问程序集的内容。

.NET Framework 类库由许多程序集组成, 它们用于读取和写入文件, 从数据库保存和检索信息以及提供窗体的功能, 不但如此, 第三方的供应商还提供了扩展 .NET Framework 类库附加的程序集。 .NET Framework 类库的一部分程序集及其作用如表 1-1 所示。

表 1-1 .NET Framework 类库中的部分程序集及其作用

程序集	作用
System.dll	定义主要数据类型, 如 Int 和 Long, 还定义最基本的数据类型, 如 System.Object
System.Windows.Forms.dll	包含用来实现桌面应用程序使用的窗体组件, 以及创建这些窗体的控件
System.XML.dll	包含处理 XML 文档所必需的组件。XML 是 ADO.NET 和其他与 Internet 相关服务的主要的传输协议
System.Drawing.dll	包含了用于向输出设备 (屏幕和打印机) 绘制各种图形 (直线、椭圆等) 的组件
System.Data.dll	定义组成 ADO.NET 的组件。ADO.NET 提供了 Visual Studio.NET 使用的数据库处理程序

## 1.4 配置 .NET Framework 环境

通过上面对 .NET Framework 环境的介绍, 相信读者已经对 .NET Framework 有了更多的