

C#

实践教程

(第2版)

- 构思独特，所有案例来自一线实战场景；
- 实用性强，将抽象的理论结合到实战案例上；
- 内容全面，结构清晰，体例丰富；
- 视频教学，专业教学视频帮助读者快速上手。

◎ 李乃文 刘好增 编著



清华大学出版社



C#

实践教程

(第2版)

◎ 李乃文 刘好增 编著

清华大学出版社
北京

内 容 简 介

C#在编程语言排行中,始终位于领先地位。新版本 4.5 运用新的架构和模块,使 C#的编写更加灵活和智能化。本书讲述 C#的理论和应用,内容包括:C#开发工具及框架的介绍、C#程序的调试与运行、基础语法、控制语句、数组、类和类的高级应用、C#常用类介绍、委托和事件、窗体和控件、MDI 技术、数据库技术、文件和 IO 流技术以及综合的项目案例。各章节中,基础知识和动手练习结合起来,形象地介绍了各部分知识的使用技巧。

本书可作为在校大学生学习使用 C#进行课程设计的参考资料,也可作为非计算机专业学生学习 C#语言的参考书。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。
版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

C#实践教程/李乃文,刘好增编著. —2 版. —北京:清华大学出版社,2016
(清华电脑学堂)

ISBN 978-7-302-41857-3

I. ①C… II. ①李… ②刘… III. ①C 语言-程序设计-教材 IV. ①TP312

中国版本图书馆 CIP 数据核字(2015)第 252091 号

责任编辑:夏兆彦
封面设计:张 阳
责任校对:胡伟民
责任印制:沈 露

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦 A 座 邮 编:100084

社总机:010-62770175 邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质量反馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

印装者:北京鑫海金澳胶印有限公司

经 销:全国新华书店

开 本:185mm×260mm 印 张:28.25 字 数:710 千字

版 次:2009 年 5 月第 1 版 2016 年 6 月第 2 版 印 次:2016 年 6 月第 1 次印刷

印 数:1~2000

定 价:59.00 元

产品编号:055136-01

C#是微软公司为 Visual Studio 开发平台推出的一种简洁、类型安全的面向对象的编程语言，开发人员通过它可以编写在 .NET Framework 上运行的各种安全可靠的应用程序。C#面世以来，以其易学易用、功能强大的优势被广泛应用，而 Visual Studio 开发平台则凭借其强大的可视化用户界面设计，让程序员从复杂的界面设计中解脱出来，使编程成为一种享受。C#不但可以开发数据库管理系统，而且也可以开发集声音、动画、视频为一体的多媒体应用程序和网络应用程序，这使得它正在成为程序开发人员使用的主流编程语言。

本书内容

本书以目前主流的 C# 4.5 及 Visual Studio 2012 为例进行介绍。全书共分 15 章，主要内容如下：

第 1 章 C#与 Microsoft Visual Studio。本章详细介绍 C#的基础入门知识，包括 C#的优势和特点，以及 .NET Framework 和程序集的相关内容。同时，详细介绍如何安装 Visual Studio 2012，以及在 Visual Studio 2012 中简单创建和开发小程序。

第 2 章 C#基础语法。本章详细介绍 C#的基础语法，包括变量、常量、数据类型、运算符和表达式、数据类型转换、装箱和拆箱等相关内容；同时针对应用程序，介绍对代码的注释和调试，方便读者找出程序错误。

第 3 章 控制语句。本章主要介绍 C#提供的流程控制语句，包括空语句、语句块、if 语句、switch 语句、do 语句、for 语句、break 语句以及异常处理语句等。

第 4 章 数组。本章主要介绍 C#中一维数组的定义、遍历、排序、插入和删除，同时介绍了二维数组、多维数组、交错数组、静态数组以及动态数组的应用。

第 5 章 面向对象编程基础。本章首先介绍了面向对象基础知识和类的概念，然后介绍类的作用、构成和使用。

第 6 章 类的高级应用。本章主要介绍面向对象编程时类高级特性的实现方式，如类的封装、密封类、继承和抽象以及重写等。此外，介绍了与类作用和结构很相似的接口的相关知识和应用。

第 7 章 字符串。字符串的相关处理是程序中较为常用的，本章主要介绍 C#内置类对字符串的处理，包括 String 类和 StringBuilder 类的相关知识和应用。

第 8 章 其他常用类。C#的内置类有多种，除了字符串处理以外，还有数学运算类、日期类和随机数类等，本章介绍 C#中其他所有常用的类。

第 9 章 枚举、结构和集合。枚举、结构和集合都用于保存和处理数据，本章介绍这三个对象的相关知识和应用，将数据处理综合在一起，方便读者掌握。

第 10 章 委托和事件。委托和事件是面向对象编程所特有的，对于 C#这种高级编

程语言来说,掌握委托和事件的知识,能够更好地使用面向对象的思想。

第11章 Windows窗体控件。窗体和控件是用户与计算机交互的基础,面向对象编程通过可视化用户界面设计,让程序员从复杂的界面设计中解脱出来,使编程成为一种享受。本章介绍的窗体和空间即为可视化用户界面。

第12章 MDI应用程序。本章介绍多窗体应用程序的构建方法,包括父窗体和子窗体的设置和应用、常用于多窗体中的控件(ToolStrip控件、StatusStrip控件、MenuStrip控件)、对话框的使用以及窗体间的数据传递等。

第13章 数据库编程。本章主要介绍应用程序对数据库的访问技术,主要介绍ADO.NET技术,包括ADO.NET结构、使用ADO.NET系统对象对数据进行操作等。此外,还将介绍C#中提供数据显示控件DataGridView和TreeView等。

第14章 文件和IO流。本章详细介绍文件和目录的操作,包括System.IO命名空间类层次结构、流的分类、内存流和文件流、操作文件和目录,以及读取和写入文件等。

第15章 职工签到系统。本章主要介绍使用C#结合SQL Server数据库实现用户签到系统的过程,主要功能包括职员登录、注册、管理用户密码、查看签到信息和签到等。

本书特色

本书使用浅显易懂的练习,结合基础知识使用,因此即使没有任何计算机基础的初学者,也能够快速入门。

本书内容详尽、实例丰富、知识面广,全面地讲解了C#编程技术,因此可供有开发经验的人员作为知识手册使用,巩固和提升开发技术。

与已经出版的图书相比,本书的最大特点体现在如下几个方面:

□ 理论和实例结合

实例丰富而典型,案例涵盖主流应用。作为一本入门类型的图书,理论和实例很好地结合起来讲解,最易让读者快速掌握。从前面的编排体例就可以看出来,本书在这个方面下足了功夫。

本书中几乎每个技术点或者语法点都会列举典型实例进行讲解,案例的数量远远多于同类图书。

□ 随书光盘

本书配备了视频教学文件,包括每个章节所涉及的源代码、开发环境的安装演示等。读者可以通过视频文件更加直观地学习C#的相关知识。

□ 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.ztydata.com.cn与我们联系,作者会在第一时间给予帮助。

读者对象

本书适合作为软件开发入门者的自学用书,也适合作为高等院校相关专业的教学参考书,也可供开发人员查阅、参考。

□ 软件开发入门者

□ C#初学者以及各大中专院校的在校学生和相关授课老师

除了封面署名人员之外,参与本书编写的人员还有李海庆、王咏梅、康显丽、王黎、汤莉、倪宝童、赵俊昌、方宁、郭晓俊、杨宁宁、王健、连彩霞、丁国庆、牛红惠、石磊、王慧、李卫平、张丽莉、王丹花、王超英、王新伟等。在编写过程中难免会有漏洞,欢迎读者通过清华大学出版社网站 www.tup.tsinghua.edu.cn 与我们联系,帮助我们改正提高。

编 者

第 1 章 C#与 Microsoft Visual Studio 1

1.1 C#与 Microsoft Visual Studio 1

1.1.1 C#简介 1

1.1.2 Visual Studio 简介 3

1.2 .NET Framework 4

1.2.1 .NET Framework 简介 4

1.2.2 .NET Framework 与
Visual Studio 6

1.3 程序集 7

1.3.1 程序集概述 7

1.3.2 程序集优点 8

1.3.3 程序集内容 9

1.3.4 程序集清单 9

1.3.5 全局程序集缓存 10

1.3.6 程序集安全注意事项 11

1.3.7 程序集版本控制 12

1.4 配置.NET Framework 环境 13

1.4.1 安装 Visual Studio 2012 13

1.4.2 认识 Visual Studio 2012 16

1.5 实验指导 1-1: 创建控制台
应用程序 17

1.6 命名空间 20

1.6.1 命名空间基础 20

1.6.2 命名空间的使用 21

1.7 实验指导 1-2: 创建窗体应用程序 23

1.8 思考与练习 24

第 2 章 C#基础语法 26

2.1 数据类型 26

2.1.1 常用数据类型简介 26

2.1.2 值类型 29

2.1.3 引用类型 30

2.2 数据进制与格式 32

2.2.1 十进制、二进制与十六

进制 32

2.2.2 字符串类型格式化 33

2.3 变量 35

2.3.1 变量的声明和初始化 35

2.3.2 变量的修饰 36

2.3.3 变量的命名规则 37

2.3.4 C#中的关键字 37

2.3.5 变量的作用域和生命周期 38

2.4 常量 39

2.5 类型转化 40

2.5.1 隐式类型转换 41

2.5.2 显式类型转换 42

2.5.3 字符串类型的转换 43

2.6 装箱和拆箱 44

2.6.1 装箱 44

2.6.2 拆箱 44

2.7 运算符与表达式 45

2.7.1 运算符分类 45

2.7.2 运算符的优先级 48

2.8 实验指导 2-1: 面积比较 49

2.9 注释与调试 49

2.9.1 注释 50

2.9.2 调试技巧 51

2.10 预处理命令 52

2.11 实验指导 2-2: 变量运算 54

2.12 思考与练习 55

第 3 章 控制语句 57

3.1 语句概述 57

3.1.1 语句分类 57

3.1.2 基本语句 58

3.1.3 语句块 58

3.2 选择语句.....	59	4.6 静态数组.....	101
3.2.1 if 语句.....	59	4.7 动态数组.....	103
3.2.2 if else 语句.....	60	4.7.1 动态数组的声明.....	104
3.2.3 if else if 语句.....	61	4.7.2 属性和方法.....	104
3.2.4 switch 语句.....	63	4.7.3 动态数组应用.....	105
3.3 循环语句.....	66	4.8 实验指导 4-1: 求矩阵外环和.....	107
3.3.1 for 语句.....	66	4.9 思考与练习.....	109
3.3.2 do while 语句.....	68		
3.3.3 while 语句.....	69		
3.3.4 foreach in 语句.....	70		
3.4 嵌套语句.....	71	第5章 面向对象编程基础.....	111
3.4.1 选择语句嵌套.....	72	5.1 类简介.....	111
3.4.2 循环语句嵌套.....	73	5.1.1 类概述.....	111
3.4.3 混合语句嵌套.....	73	5.1.2 类的成员.....	112
3.5 跳转语句.....	74	5.1.3 访问修饰符.....	113
3.5.1 break 语句.....	74	5.2 字段和属性.....	114
3.5.2 continue 语句.....	75	5.2.1 字段.....	115
3.5.3 return 语句.....	76	5.2.2 属性.....	115
3.5.4 goto 语句.....	77	5.3 方法.....	117
3.6 异常处理语句.....	79	5.3.1 方法概述.....	117
3.6.1 throw.....	79	5.3.2 返回值.....	118
3.6.2 try catch.....	79	5.3.3 方法的定义.....	119
3.6.3 try catch finally.....	82	5.4 类和对象.....	119
3.7 实验指导 3-1: 输出等腰梯形.....	84	5.4.1 对象.....	120
3.8 思考与练习.....	86	5.4.2 对象应用.....	120
		5.4.3 参数传递.....	121
		5.4.4 静态类和类成员.....	123
		5.4.5 静态方法的调用.....	124
第4章 数组.....	88	5.5 构造函数.....	125
4.1 数组概述.....	88	5.6 析构函数.....	126
4.2 一维数组.....	88	5.7 实验指导 5-1: 创建数据统计类.....	127
4.2.1 一维数组简介.....	89	5.8 思考与练习.....	129
4.2.2 数组遍历.....	90		
4.2.3 数组排序.....	91		
4.2.4 插入数组元素.....	93	第6章 类的高级应用.....	131
4.2.5 删除数组元素.....	95	6.1 封装.....	131
4.3 二维数组.....	97	6.1.1 封装概述.....	131
4.3.1 二维数组简介.....	97	6.1.2 封装的使用.....	132
4.3.2 二维数组遍历.....	98	6.2 继承.....	133
4.4 多维数组.....	99	6.2.1 继承简述.....	133
4.5 交错数组.....	100	6.2.2 抽象类及类成员.....	135

6.2.3 密封类	137	7.3.4 字符串替换	180
6.3 多态	139	7.3.5 字符串拆分	181
6.3.1 重载	139	7.3.6 获取匹配项	182
6.3.2 重写	141	7.4 实验指导 7-1: 用户注册	184
6.3.3 虚函数	142	7.5 思考与练习	187
6.4 实验指导 6-1: 虚函数与抽象类	144		
6.5 接口	145	第 8 章 其他常用类	189
6.5.1 接口简介	145	8.1 时间和日期	189
6.5.2 接口定义	146	8.1.1 TimeSpan 结构	189
6.5.3 实现接口	148	8.1.2 DateTime 结构	193
6.5.4 IComparable 接口	150	8.1.3 DateTime 结构应用	195
6.5.5 IComparer 接口	151	8.1.4 格式化	199
6.6 实验指导 6-2: 实现简单 数学运算	151	8.1.5 计算时间差	200
6.7 思考与练习	154	8.2 数学类	201
		8.2.1 Math 类简介	201
第 7 章 字符串	156	8.2.2 Math 类的应用	202
7.1 String 类字符串	156	8.3 随机数	203
7.1.1 创建字符串	156	8.3.1 Random 类简介	203
7.1.2 大小写转换	158	8.3.2 随机数的应用	204
7.1.3 去除指定字符	159	8.4 实验指导 8-1: 时间与随机数的 综合运算	205
7.1.4 合并字符串	160	8.5 思考与练习	206
7.1.5 替换字符	161		
7.1.6 比较字符串	161	第 9 章 枚举、结构和集合	208
7.1.7 查找字符串	165	9.1 枚举	208
7.1.8 分隔字符串	166	9.1.1 枚举简介	208
7.1.9 截取字符串	167	9.1.2 声明枚举	209
7.1.10 移除字符串	168	9.1.3 使用枚举	210
7.2 StringBuilder 类字符串	169	9.1.4 转换枚举类型	211
7.2.1 创建字符串	169	9.1.5 使用 Enum 类型	212
7.2.2 插入字符串	170	9.2 结构	215
7.2.3 追加字符串	172	9.2.1 结构简介	215
7.2.4 移除和替换	174	9.2.2 声明结构	216
7.2.5 StringBuilder 类的 其他常用成员	175	9.2.3 使用结构	217
7.3 正则表达式	176	9.3 集合	218
7.3.1 基本语法	176	9.3.1 C#内置集合概述	218
7.3.2 正则表达式常用类	179	9.3.2 常见的几种集合类	219
7.3.3 字符串验证	179	9.3.3 ArrayList 集合类	220
		9.3.4 Stack 集合类	221

9.3.5	Queue 集合类	223	11.2.7	MaskedTextBox 控件	279
9.3.6	BitArray 集合类	225	11.3	选择类型控件	281
9.3.7	SortedList 集合类	230	11.3.1	RadioButton 控件	282
9.3.8	Hashtable 集合类	233	11.3.2	CheckBox 控件	284
9.4	自定义集合类	236	11.4	图像显示类型控件	286
9.5	泛型	239	11.4.1	ImageList 控件	286
9.5.1	泛型概述	239	11.4.2	PictureBox 控件	288
9.5.2	泛型类	240	11.5	列表类型控件	291
9.5.3	泛型方法	243	11.5.1	ComboBox 控件	291
9.5.4	泛型参数	244	11.5.2	ListView 控件	295
9.5.5	类型参数的约束	246	11.5.3	ListBox 控件	299
9.6	实验指导 9-1: 手机信息管理	248	11.5.4	CheckedListBox 控件	300
9.7	思考与练习	250	11.6	容器类型控件	302
第 10 章	委托和事件	252	11.6.1	TabControl 控件	303
10.1	委托	252	11.6.2	GoupBox 控件	305
10.1.1	委托简介	252	11.6.3	Panel 控件	305
10.1.2	委托的应用	253	11.7	其他常用类型控件	306
10.1.3	方法绑定	255	11.7.1	DateTimePicker 控件	306
10.2	事件	256	11.7.2	Timer 组件	307
10.2.1	事件简介	256	11.7.3	NotifyIcon 组件	308
10.2.2	事件的应用	257	11.8	实验指导 11-1: 会员信息登记	309
10.3	实验指导 10-1: 多方法执行	259	11.9	思考与练习	312
10.4	实验指导 10-2: 信息修改	260	第 12 章	MDI 应用程序	314
10.5	思考与练习	263	12.1	MDI 应用程序	314
第 11 章	Windows 窗体控件	265	12.1.1	MDI 概述	314
11.1	Windows 窗体概述	265	12.1.2	创建 MDI 父窗体	316
11.1.1	窗体概述	265	12.1.3	创建 MDI 子窗体	316
11.1.2	窗体控件的公有属性	268	12.1.4	排列 MDI 子窗体	317
11.1.3	窗体控件的公有事件	269	12.1.5	模式窗体和无模式窗体	318
11.2	基本类型控件	271	12.2	高级控件	319
11.2.1	基本控件	271	12.2.1	MenuStrip 控件	319
11.2.2	Label 控件	271	12.2.2	ContextMenuStrip 控件	323
11.2.3	LinkLabel 控件	273	12.2.3	ToolStrip 控件	324
11.2.4	Button 控件	274	12.2.4	StatusStrip 控件	325
11.2.5	TextBox 控件	275	12.3	常用对话框	327
11.2.6	RichTextBox 控件	276	12.3.1	消息对话框	327
			12.3.2	字体对话框	330

12.3.3 颜色对话框	332	13.6.2 在设计器中操作 DataGridView View 控件	370
12.3.4 浏览目录对话框	333	13.6.3 DataGridView 控件的 使用	372
12.3.5 打开文件对话框	334	13.7 实验指导 13-2: 用户注册	374
12.3.6 保存文件对话框	336	13.8 思考与练习	377
12.4 实验指导 12-1: 窗体间的 数据传递	338		
12.5 思考与练习	340		
第 13 章 数据库编程	342	第 14 章 文件和 IO 流	379
13.1 数据库开发基础	342	14.1 认识流	379
13.1.1 数据库开发技术简介	342	14.1.1 System.IO 命名空间	379
13.1.2 ADO.NET 概述	343	14.1.2 流抽象类	380
13.1.3 .NET Framework 数据 提供程序	344	14.1.3 编码	381
13.2 数据库连接	345	14.1.4 内存流	382
13.2.1 数据库连接技术	345	14.1.5 文件流	384
13.2.2 SQL Server 连接对象	346	14.2 获取文件系统信息	387
13.2.3 连接 SQL Server	346	14.2.1 文件信息 FileInfo 类	387
13.3 数据操作	349	14.2.2 目录信息 DirectoryInfo 类	390
13.3.1 数据操作对象	349	14.2.3 驱动器信息 DriveInfo 类	391
13.3.2 SqlCommand 对象	350	14.3 操作目录	393
13.3.3 操作 SQL Server 数据	351	14.3.1 Directory 类简介	393
13.3.4 SqlParameter 对象	352	14.3.2 创建目录	394
13.3.5 含参数的数据操作	354	14.3.3 移动和重命名目录	394
13.3.6 SqlDataReader 对象	355	14.3.4 删除目录	395
13.4 实验指导 13-1: 数据显示	356	14.3.5 遍历目录	396
13.5 数据集对象	358	14.4 操作文件	398
13.5.1 数据集对象简介	358	14.4.1 File 类简介	398
13.5.2 SqlDataAdapter 对象	358	14.4.2 创建文件	399
13.5.3 DataSet 对象	359	14.4.3 移动和重命名文件	400
13.5.4 DataTable 对象	362	14.4.4 文件复制和删除	400
13.5.5 数据显示	364	14.5 读取和写入文件	401
13.5.6 DataView 对象	367	14.5.1 读取文件	401
13.5.7 SqlDataReader 对象与 DataSet 对象的区别	368	14.5.2 写入文件	402
13.6 数据显示控件——DataGridView 控件	369	14.6 读写二进制文件	404
13.6.1 DataGridView 控件的 常用属性和事件	369	14.7 树形目录控件	406
		14.7.1 TreeView 控件的常用 属性和事件	406
		14.7.2 TreeView 的使用	408
		14.8 实验指导 14-1: 文件管理	409
		14.9 思考与练习	411

第 15 章 职工签到系统	413	15.3.2 添加引用	419
15.1 系统概述	413	15.3.3 程序入口	420
15.1.1 需求分析	413	15.3.4 数据库信息处理类	420
15.1.2 功能分析	414	15.4 登录模块	423
15.2 数据库设计	415	15.5 注册	425
15.2.1 表的设计	415	15.6 签到	427
15.2.2 存储过程设计	416		
15.3 准备工作	418	附录 思考与练习答案	434
15.3.1 搭建项目	418		

第1章 C#与 Microsoft Visual Studio

C#与 Microsoft Visual Studio 是不可分割的。C#是当前最为流行的高级编程语言之一，是由 Microsoft 推出的一种简洁、类型安全的面向对象的编程语言，该语言是在 Microsoft Visual Studio 平台基础上推出的。

本章详细介绍 C#、Microsoft Visual Studio 和 .NET Framework 的概念及它们之间的关系，Visual Studio 环境的配置以及简单的 C#小程序。

本章学习目标：

- ☐ 了解 C#与其他语言的差异。
- ☐ 掌握 C#的特点。
- ☐ 了解 Visual Studio 2012 的新特性。
- ☐ 了解 .NET Framework 与 C#的关系。
- ☐ 掌握公共语言运行时和 .NET Framework 类库。
- ☐ 了解程序集的概念和优点。
- ☐ 掌握程序集的内容和清单。
- ☐ 熟悉全局程序集缓存、安全注意事项以及版本控制。
- ☐ 掌握如何安装 Visual Studio 2012。
- ☐ 了解 Visual Studio 2012 的使用。

1.1 C#与 Microsoft Visual Studio

C#是微软公司为 Visual Studio 开发平台推出的一种简洁、类型安全的面向对象的编程语言，开发人员通过它可以编写在 .NET Framework 上运行的各种安全可靠的应用程序。本节介绍 C#与 Microsoft Visual Studio 的概括及联系。

1.1.1 C#简介

C#是运行于 .NET Framework 之上的面向对象高级程序设计语言。Visual Studio 开发平台凭借 C#强大的可视化用户界面设计，让程序员从复杂的界面设计中解脱出来，使编程成为一种享受。

C#是面向对象的编程语言。它使得程序员可以快速地编写各种基于 Microsoft .NET 平台的应用程序，Microsoft .NET 提供了一系列的工具和服务来最大程度地开发利用计算与通信领域。

C#不但可以开发数据库管理系统，而且也可以开发集声音、动画、视频为一体的多媒体应用程序和网络应用程序，这使得它正在成为程序开发人员使用的主流编程语言。

正是由于 C#面向对象的卓越设计，使它成为构建各类组件的理想之选——无论是高级的商业对象还是系统级的应用程序。使用简单的 C#语言结构，这些组件可以方便地转化为 XML 网络服务，从而使它们可以由任何语言在任何操作系统上通过 Internet 进行调用。

作为一种面向对象语言，C#支持封装、继承和多态以及所有的变量和方法，如包括应用程序入口点的 Main()方法。另外，C#还通过几种创新的语言结构加快了软件组件的开发。主要包括：

- ❑ 委托 即封装的方法签名，它实现了类型安全的事件通知。
- ❑ 属性（Property） 充当私有成员变量的访问器。
- ❑ 属性（Attribute） 提供关于运行时类型的声明性元数据。
- ❑ 内联 XML 的文档注释。

1. C#与 Java

C#与 Java 都有着单一继承、接口、几乎同样的语法和编译成中间代码再运行的过程。

但是 C#与 Java 有着明显的不同，C#是第一个面向组件的编程语言，其源码会编译成 MSIL 再运行。C#借鉴了 Delphi 的一个特点，与 COM（组件对象模型）是直接集成的，而且它是微软公司 .NETWindows 网络框架的主角。在细节方面，二者的区别如下所示。

- ❑ C#的 Main()方法的首字母要大写。
- ❑ C#在值类型和引用类型之间进行装箱和拆箱操作，无须创建包装类型。
- ❑ Java 中的最终类在 C#中是密封的。
- ❑ 在默认情况下，C#中的方法是非虚拟方法。
- ❑ 为了包括编辑器的其他信息，C#支持属性操作。

2. C#与 C/C++和 VB

C#是一种安全的、稳定的、简单的、优雅的编程语言，由 C 和 C++衍生出来。它在继承 C 和 C++强大功能的同时去掉了一些它们的复杂特性。

C#可调用由 C/C++ 编写的本机原生函数，因此拥有 C/C++原有的强大的功能。由于 C#与 C/C++具有极大的相似性，熟悉类似语言的开发者可以很快地转向 C#。在细节方面，C#与 C/C++的区别如下所示。

- ❑ 在 C#程序中，类定义中右大括号后不必使用分号。
- ❑ C#的 Main()方法的首字母大写，而且是静态类的成员，该方法的返回类型为 int 或 void。
- ❑ C#每个主程序中都必须包含 Main()方法，否则该程序不能编译。
- ❑ C#内存直接使用垃圾收集系统来管理。
- ❑ C#条件必须为 Boolean 类型。
- ❑ C#中的 switch 语句和 break 语句不是可选的。
- ❑ C#默认值由编译器分配（引用类型为 null，值类型为 0）。

C#综合了 VB 简单的可视化操作和 C++的高运行效率，以其强大的操作能力、优雅

的语法风格、创新的语言特性和便捷的面向组件编程的支持成为.NET 开发的首选语言。其与 VB 的区别如下所示。

- ☐ C#使用分号而不是分行符。
- ☐ C#区分大小写，例如 Main()方法的首字母大写。
- ☐ C#条件必须为 Boolean 类型。

3. C#的编译执行

C#开发的程序源代码并不是编译成能够直接在操作系统上执行的二进制本地代码。与 Java 类似，它被编译成为中间代码，然后通过.NET Framework 的虚拟机（通用语言运行时 CLR）执行。

所有的.NET 编程语言都被编译成这种被称为 MSIL（MicroSoft Intermediate Language）的中间代码，最终的程序具有“.exe”后缀名。

如果计算机上没有安装.NET Framework，那么这些程序将不能够被执行。在程序执行时，.NET Framework 将中间代码翻译成为二进制机器码，从而使它得到正确的运行。最终的二进制代码被存储在一个缓冲区中。

一旦程序使用了相同的代码，那么将会调用缓冲区中的版本。这样如果一个.NET 程序第二次被运行，那么这种翻译不需要进行第二次，速度明显加快。

1.1.2 Visual Studio 简介

Microsoft Visual Studio 简称 Visual Studio 或 VS，是目前最流行的 Windows 平台应用程序开发环境。

Microsoft Visual Studio 可以用来编写创建 Windows 平台下的 Windows 应用程序和网络应用程序，也可以用来创建网络服务、智能设备应用程序和 Office 插件。

任何一种高级编程语言，都需要有相应的编程环境。而 Visual Studio 支持多种编程语言，如 Visual BASIC、Visual C# 和 Visual C++等。

Visual Studio 是一套完整的开发工具，用于生成 ASP.NET Web 应用程序、XML Web Services、桌面应用程序和移动应用程序。

由于 Visual BASIC、Visual C# 和 Visual C++ 都使用相同的集成开发环境 (IDE)，这样就能够进行工具共享，并能够轻松地创建混合语言解决方案。另外，这些语言使用 .NET Framework 的功能，它提供了可简化 ASP Web 应用程序和 XML Web Services 开发的关键技术。

使用 Visual Studio 可以生成 Windows 应用商店应用程序、桌面应用程序、移动应用程序、ASP.NET Web 应用程序和 XML Web Services。还可以在 Visual BASIC、Visual C#、Visual C++、Visual F# 和 JavaScript 中编写代码，且可以更轻松地创建混合语言解决方案。

Visual Studio 当前最新版本为 Visual Studio 2012 版本，基于.NET Framework 4.5。与之前的版本相比，Visual Studio 2012 版本有着六大新特性，如下所示。

- ☐ VS 2012 和 VS 2010 相比，VS 2012 支持 Windows 8 Metro 开发。Metro 简洁、

数字化、内容优于形式、强调交互的设计已经成为未来的趋势。

- ❑ VS 2012 在界面上更容易使用,彩色的图标和按照开发、运行、调试等环境区分的颜色方案使用起来方便显眼。
- ❑ VS 2012 集成了 ASP.NET MVC 4,全面支持移动和 HTML 5,WF 4.5 相比 WF 4,更加成熟,它的设计器已经支持 C#表达式。
- ❑ VS 2012 支持 .NET 4.5,和 .NET 4.0 相比,.NET 4.5 更多的是完善和改进。.NET 4.5 是 Windows RT 被提出来的首个框架库,.NET 获得了和 Windows API 同等的待遇。
- ❑ VS 2012 和 TFS 2012 实现了更好的生命周期管理,使 VS 2012 不仅是开发工具,也是团队的管理信息系统。
- ❑ VS 2012 对系统资源的消耗并不大,但需要 Windows 7/8 的支持。

1.2 .NET Framework

.NET Framework 由微软开发,致力于敏捷软件开发(Agile Software Development)、快速应用开发(Rapid Application Development)、平台无关性和网络透明化的软件开发平台。

Visual Studio 与 .NET Framework 不可分割,本节介绍 .NET Framework,以及它与 Visual Studio 之间的关系。

1.2.1 .NET Framework 简介

.NET Framework 是 Microsoft 推出的一套类库,它被称为 .NET 框架,支持多种开发语言(如 C#、VB、C++、Python 等)。

.NET Framework 类库包括 ADO.NET、ASP.NET、Windows 窗体和 Windows Presentation Foundation(WPF)和 Windows Workflow Foundation(WF)。

.NET Framework 是一种采用系统虚拟机运行的编程平台,以通用语言运行时(Common Language Runtime)为基础,用于 Windows 的新托管代码编程模型。.NET Framework 的主要特点和功能如下所示。

- ❑ NET 为应用程序接口(API)提供了新功能和开发工具。这些革新使得程序设计员可以同时进行 Windows 应用软件和网络应用软件以及组件和服务(Web 服务)的开发。
- ❑ .NET 提供了一个新的反射性的且面向对象程序设计编程接口。其通用化使许多不同高级语言都得以被汇集。.NET Framework 中的所有语言都提供基类库(BCL)。
- ❑ 微软的 Windows 作为操作系统运行于 .NET 框架之下,在 Windows XP SP2/Windows Server 2003/Windows Vista 系统中内置 .NET 框架。
- ❑ .NET 框架作为 .NET 开发平台的核心组件为 Web 服务及其他应用提供构建、移植和运行的环境。

- ❑ .NET 组件是用于创建网络和 Windows 应用程序的,这些应用程序使一个应用程序所需的功能可以显示在外部。
- ❑ .NET 平台还包含 Web 表单, Web 表单是可从网上下载的标准接口。一个 Web 表单包含供使用者输入数据资料的文本框,然后使用者可以将表单提交给接收器。
- ❑ .NET 平台至关重要的一部分就是网络服务器。网络服务器查询协议和标准的合集。应用程序可以使用网络服务器通过计算机网络交换数据资料。例如在线上订购火车票。
- ❑ .NET Framework 安全解决方案基于管理代码的概念,以及由通用语言运行时 (CLR) 加强的安全规则。
- ❑ .NET Framework 提供了一个特殊的功能——隔离存储,用于存储数据,甚至是当不允许对文件进行访问时。
- ❑ .NET Framework 引入了基于证据的安全的概念。
- ❑ 一些活动,如读写文件,显示对话框,读写环境变量,可以通过包含在框架安全构架中的 .NET Framework 方法实现。
- ❑ .NET Framework 提供了一组加密对象,它们支持加密算法、数字签名、散列、生成随机数。同时还支持在 IETF 和 W3C 开发的 XML 数字签名规范。
- ❑ .NET Framework 使用加密对象支持内部服务。这些对象还作为管理代码提供给需要加密支持的开发人员。

当前 .NET Framework 的最新版本为 .NET Framework 4.5 版本,该版本不支持 Windows 2000、Windows XP。

.NET Framework 4.5 发行于 2012 年 8 月 16 日,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,.NET Framework 的关键组件为通用语言运行时(CLR)和 .NET Framework 类库。与先前版本相比,其更新内容如下所示。

- ❑ 在部署期间,能够通过检测和关闭 .NET Framework 4 应用程序来减少系统重启。
- ❑ 为大于 20GB 在 64 位平台上(GB)的数组支持。此功能可在应用程序配置文件中启用。
- ❑ 通过服务器的后台垃圾回收改进性能。当您使用服务器垃圾回收在 .NET Framework 4.5 中时,后台垃圾回收自动启用。
- ❑ 背景实时(JIT)生成,可以选择用在多核处理器改进应用程序性能。
- ❑ 在它超时之前,能够限制正则表达式引擎要多久能尝试解决正则表达式。
- ❑ 能够定义应用程序域的默认区域性。
- ❑ Unicode(UTF-16)编码的控制台支持。
- ❑ 为版本控制区域性字符串排序和比较数据支持。
- ❑ 在检索资源时,请改进性能。请参见打包和部署桌面应用程序中的资源。
- ❑ Zip 压缩改进可减少压缩文件的大小。
- ❑ 通过 CustomReflectionContext 类,能够自定义反射上下文来重写默认反射行为。
- ❑ 对于国际化域名的 2008 版在应用程序(IDNA)标准的支持,当 System.Globalization.IdnMapping 选件类在 Windows 8 使用时。