

高等院校信息技术规划教材

C#程序开发与应用教程

高凯 周万珍 高国江 编著

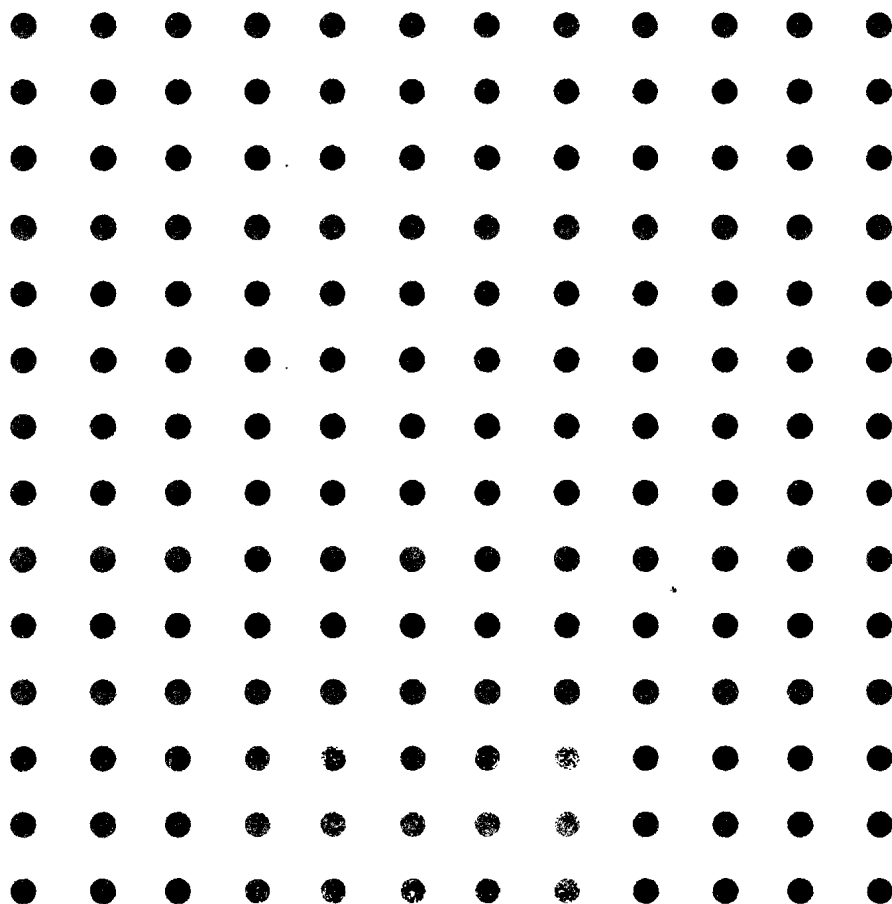


清华大学出版社

高等院校信息技术规划教材

C#程序开发与应用教程

高凯 周万珍 高国江 编著



清华大学出版社
北京

内 容 简 介

本书以基础知识为先导,以案例说明和工程实践为特色,以实用性为原则,以提高工程实践水平和动手能力为目的,深入浅出地讲解 Visual Studio 2010 开发平台下的 C# 程序设计方法、技巧等,内容涵盖了面向对象程序设计基础、异常处理与代码调试、集合索引器与泛型、字符处理和正则表达式、窗体应用程序设计、流与文件操作、基于 ADO.NET 的数据库及 XML 应用程序开发、多线程技术等。书中所有练习均在 Visual Studio 2010 环境下进行过验证。本书既可帮助初学者快速上手,也能为有一定基础的工程技术人员提供参考。

本书可作为高等学校的教材,并可供从事软件开发与维护的工程技术人员参考。

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# 程序开发与应用教程/高凯,周万珍,高国江编著. --北京:清华大学出版社,2012.7

(高等院校信息技术规划教材)

ISBN 978-7-302-28434-5

I. ①C… II. ①高… ②周… ③高… III. ①C 语言—程序设计—高等职业教育—教材
IV. ①TP312

中国版本图书馆 CIP 数据核字(2012)第 060497 号

责任编辑:焦虹 顾冰

封面设计:常雪影

责任校对:时翠兰

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

课 件 下 载: <http://www.tup.com.cn>, 010-62795954

印 刷 者:北京密云胶印厂

装 订 者:北京市密云县京文制本装订厂

经 销:全国新华书店

开 本:185mm×260mm

印 张:14.75

字 数:341 千字

版 次:2012 年 7 月第 1 版

印 次:2012 年 7 月第 1 次印刷

印 数:1~3000

定 价:25.00 元

产品编号:045050-01

前言

foreword

本书的编写指导思想是以基础知识为先导,以案例说明和工程实践为特色,以实用性为原则,以提高工程实践水平和动手能力为目的,基于 .NET Framework 4.0 和 Visual Studio C# 2010,深入浅出地讲解 C# 程序设计的方法、技巧,通过对众多实例以及处理代码的分析,讲述有关面向对象程序设计、异常处理与代码调试、集合索引器与泛型、字符处理和正则表达式、窗体应用程序设计、流与文件操作、基于 ADO.NET 的数据库及 XML 应用程序的开发、多线程技术等知识。全书主要章节及内容安排如下:第 1 章概述 Microsoft .NET 和 IDE 的使用,并从历史角度分析 .NET Framework 给程序设计带来的革命性的变化,简述了几种不同类型 C# 程序的建立与执行方式。第 2 章简述 C# 程序设计语言基础,并对程序的基本构成、命名空间、数据类型、流程控制、类型转换、函数等进行了说明。第 3 章介绍面向对象程序设计基础,内容包括类及其构成、构造函数和析构函数、类的数据成员、类的方法成员、方法和运算符的重载、类的属性成员等。第 4 章是异常处理与代码调试部分,并简介了 NUnit 的使用。第 5 章介绍集合、索引器与泛型,并对 .NET 中常见的集合类、索引器、泛型类和泛型方法等概念及其操作方法进行了说明。第 6 章是在信息处理中能发挥重要作用的有关字符处理和正则表达式的内容,主要对静态串 string 及其相关方法的使用,以及字符比较、定位、格式化、连接、分割、插入、删除、复制、编码等操作进行说明,简介了动态串 StringBuilder,给出了正则表达式的相关内容和应用实例。随着 Windows 和 Web 应用的迅速普及,开发基于窗体的应用程序显得尤为必要。第 7 章中对常见窗体相关控件的使用进行说明,并给出了建立菜单的方法以及常用对话框控件的使用技巧。第 8 章讲述 C# 中对流和文件进行操作的方法,内容涉及读写流、获取文件夹和文件信息及对其进行管理的一般方法,并简介 Directory 和 DirectoryInfo 类、File 和 FileInfo 类、StringReader 和 StreamWriter 类、FileStream 类、Path 类等的使用。第 9 章是基于 ADO.NET 的数据库及 XML 应用程序的开发,以



SQL Server 数据源为例,介绍基于 ADO.NET 的数据库应用程序开发方法,内容涉及连接数据库、Command 对象执行 SQL 命令,使用 DataReader 对象从数据库中检索数据,通过 DataAdapter 访问并操纵数据库中的数据,并介绍了常见的数据绑定、导航、表格展示控件等的使用,以及文件和数据库间的协同操作方法。第 10 章介绍有关多线程的编程方法,内容涉及创建线程、控制线程和线程同步等,并简介了 System.Threading 命名空间、System.Threading.Thread 类及其中常用的属性和方法。

本书注重对读者实践能力的培养,可为读者进一步的程序设计与软件开发奠定坚实的基础。无论是刚开始接触 .NET Framework 和 C# 的新手,还是有一定基础的 VC++、VB 程序员,都可以从本书中吸取到营养。本书既可帮助初学者快速上手,也能为有一定基础的工程技术人员提供参考。本书理论联系实际,案例有实用性,每章均提供了针对性较强的习题。由清华大学出版社出版的与本书配套的《C# 习题解析与工程实践》提供所有习题解答以及相关的工程实践开发指导。本书可以作为高等学校相关专业的教材,也可供工程技术人员参考。

本书提供源代码下载和课件下载,相关源代码和课件资源均可在清华大学出版社网站下载(www.tup.com.cn)。各章相应工程的完整代码可以在相应章节中的对应工程中找到,方便读者自学和实践,另外,提供针对每章内容的学习课件,方便教学。书中的所有练习均在 Visual Studio 2010 环境下进行过验证。

全书由高凯、周万珍、高国江合作编著。高凯提出写作大纲,周万珍编写了第 1、2 章,高国江编写了第 3、4 章,其余章节由高凯编写,最后由高凯完成了全书的统稿与审校工作。有关正则表达式与多线程部分的代码由周二亮协助编写,王伟、周二亮、沈琳、刘邵博、王亚歌等协助完成了书中部分代码的编写和测试工作。在本书的写作与相关科研课题的研究工作中,我们得到了多方面的支持与帮助。有关 C# 开发的相关资料和网站——如 MSDN、CSDN 等——亦为本书的编写提供了良好的基础。我们也参考了相关文献和互联网上众多热心网友提供的素材,在此谨向这些文献的作者、热心网友以及为本书提供帮助的老师,特别是那些由于篇幅所限未及在参考文献中提到的相关文献的作者和网站,致以诚挚的谢意和崇高的敬意!

由于我们的学识、水平有限,书中不妥之处在所难免,恳请广大读者批评指正。

高 凯 周万珍 高国江

2012 年 5 月

目录

Contents

第 1 章 Microsoft .NET 概述	1
1.1 .NET 及其之前的程序开发流程	2
1.1.1 C 及 Win32 API	2
1.1.2 MFC、VC++、VB 及 COM 技术	2
1.2 Microsoft .NET 及 .NET Framework 简介	3
1.3 Visual Studio .NET 2010 简介	5
1.3.1 Professional	6
1.3.2 Premium	6
1.3.3 Ultimate	7
1.3.4 Test Professional	7
1.4 C# 概述	8
1.4.1 C# 的发展历程	8
1.4.2 C# 的特点	8
1.5 安装 Visual Studio 2010	9
1.6 Visual Studio 2010 集成开发环境 IDE 简介	9
1.6.1 项目与解决方案资源管理器	10
1.6.2 代码编辑器和设计器	12
1.6.3 代码生成与调试	12
1.6.4 代码部署	12
1.6.5 常用视图和窗口简介	13
1.6.6 C# 的代码重构	15
1.6.7 可视化类设计器	16
1.6.8 个性化设置	17
1.6.9 重置布局	18
1.7 几种不同类型 C# 程序的建立与执行	18
1.7.1 控制台应用程序	19
1.7.2 Windows 应用程序	20



1.7.3 Web 程序	22
1.7.4 查找和修正错误	24
1.8 本章小结	24
习题	24

第2章 C# 程序设计语言基础

25

2.1 C# 程序的基本构成	26
2.1.1 C# 中的数据类型	27
2.1.2 用类搭建程序	27
2.1.3 使用命名空间	29
2.1.4 Main 函数：程序的入口	33
2.1.5 格式化输出与输入	34
2.2 常量、变量、常用运算符和表达式	35
2.2.1 常量和变量的定义与作用域	35
2.2.2 表达式与常用运算符	38
2.3 值类型和引用类型	39
2.3.1 值类型	40
2.3.2 引用类型	41
2.3.3 值类型和引用类型的关系	41
2.4 类型转换	41
2.4.1 隐式类型转换	41
2.4.2 显式类型转换	42
2.4.3 命名规范	44
2.5 流程控制	44
2.5.1 if 语句	44
2.5.2 switch-case 语句	45
2.5.3 while 语句	46
2.5.4 do-while 语句	46
2.5.5 for 语句	47
2.5.6 跳转语句	48
2.6 函数及其重载	48
2.7 常见的预处理指令	49
2.7.1 #define、#undef	50
2.7.2 #if、#elif、#else、#endif	50
2.7.3 #region、#endregion	51
2.7.4 #warning、#error	51
2.8 本章小结	51
习题	52

第 3 章 面向对象程序设计基础	53
3.1 类	53
3.1.1 类、对象及类成员类型	54
3.1.2 类的定义	54
3.2 静态类和非静态类	55
3.3 抽象类的定义及其覆盖	57
3.4 类成员及方法	58
3.5 类方法的调用及重载	59
3.6 封装、继承、多态	60
3.7 类的构造函数与析构函数	61
3.7.1 构造函数及其重载	61
3.7.2 在构造函数中使用 this 访问	63
3.7.3 析构函数	63
3.8 基类和派生类	64
3.9 属性及 get/set 方法	66
3.10 接口	68
3.11 本章小结	74
习题	74
第 4 章 异常处理与代码调试	76
4.1 异常处理机制概述	76
4.1.1 异常处理的特点	77
4.1.2 异常的抛出与捕获	77
4.1.3 具有异常处理机制的程序组成	78
4.2 系统的异常类及其使用	78
4.3 自定义异常类	80
4.4 try-catch 语句块	81
4.5 代码调试	84
4.5.1 断点设置与程序运行	85
4.5.2 程序调试的基本窗口	86
4.6 基于 NUnit 的单元测试	87
4.7 本章小结	91
习题	91
第 5 章 集合、索引器与泛型	93
5.1 集合与泛型概述	93



5.2	数组 Array	94
5.2.1	遍历数组元素	96
5.2.2	复制、排序数组元素	96
5.3	动态数组 ArrayList	98
5.3.1	ArrayList 的常用方法	98
5.3.2	ArrayList 和 string 间的转换	99
5.4	队列 Queue	100
5.5	栈 Stack	101
5.6	可排序字典 SortedList 和哈希表 Hashtable	101
5.6.1	可排序字典 SortedList	102
5.6.2	哈希表 Hashtable	102
5.7	索引器	103
5.8	泛型	105
5.9	本章小结	106
	习题	107
第 6 章	字符处理和正则表达式	108
6.1	静态字符串构造函数及部分常用方法简介	108
6.1.1	静态字符串构造函数的使用	108
6.1.2	常用方法	109
6.2	动态字符串 StringBuilder	114
6.3	字符编码简介	115
6.4	正则表达式	116
6.4.1	正则表达式概述	116
6.4.2	构造正则表达式	118
6.4.3	利用正则表达式分割提纯字符串	118
6.5	正则表达式的常见应用	119
6.5.1	用正则表达式验证电话号码的有效性	119
6.5.2	用正则表达式验证 E-mail 地址的合法性	120
6.5.3	正则表达式在字词查找中的应用	121
6.6	本章小结	121
	习题	122
第 7 章	窗体应用程序设计	123
7.1	窗体与控件概述	124
7.1.1	窗体的常用属性	124
7.1.2	窗体的常用方法和事件	125

7.1.3 主要的窗体控件概述	126
7.2 部分常见控件功能及使用简介	126
7.2.1 按钮与标签控件	126
7.2.2 文本控件	128
7.2.3 选择控件	130
7.2.4 列表控件	131
7.2.5 时间控件	136
7.2.6 容器控件	137
7.2.7 进度条控件 ProgressBar	138
7.2.8 设置控件的 Tab 顺序	138
7.2.9 创建工具提示	139
7.2.10 帮助文档组件 HelpProvider	139
7.3 菜单、工具栏和状态栏	140
7.3.1 菜单	140
7.3.2 工具栏	142
7.3.3 状态栏	143
7.4 对话框控件及其设计	144
7.4.1 模态对话框与非模态对话框	144
7.4.2 消息对话框	144
7.4.3 通用对话框	146
7.4.4 OpenFileDialog 控件	147
7.4.5 SaveFileDialog 控件	148
7.4.6 FolderBrowseDialog 控件	148
7.5 综合案例讲解	149
7.6 本章小结	151
习题	151
第 8 章 流与文件操作	152
8.1 概述	152
8.2 静态目录类 Directory 及非静态目录类 DirectoryInfo	154
8.2.1 Directory 类	154
8.2.2 DirectoryInfo 类	155
8.3 静态文件类 File 及非静态文件类 FileInfo	157
8.3.1 File 类	157
8.3.2 FileInfo 类	158
8.4 驱动器 DriveInfo 类	159
8.5 流及文件操作	159
8.5.1 概述	159

8.5.2 操作流的类	160
8.6 案例分析:统计自然语言相应词汇个数并排序	165
8.7 本章小结	168
习题	169
第9章 基于 ADO.NET 的数据库及 XML 应用程序的开发	170
9.1 数据库管理系统及常用 SQL 语句概述	171
9.1.1 SQL 语句完成数据查询	171
9.1.2 SQL 语句完成插入数据	172
9.1.3 SQL 语句完成删除数据	172
9.1.4 SQL 语句完成修改数据	173
9.2 ADO.NET 概述	173
9.2.1 主要数据访问对象简介	174
9.2.2 获取数据库记录的主要方式	180
9.2.3 DataProvider 对应的命名空间及 Connection 对象简介	181
9.3 建立数据库	182
9.4 连接数据库	183
9.4.1 连接 SQL Server 数据库	183
9.4.2 连接 Oracle 数据库	186
9.4.3 连接 Access 数据库	186
9.4.4 使用 config 配置文件增加数据库连接的灵活性	187
9.5 操纵数据库	188
9.5.1 利用 Command 对象直接操纵数据库	188
9.5.2 利用 SqlDataReader 对象及 SQL 语句操纵数据	189
9.5.3 利用 SqlDataReader 对象及存储过程操纵数据	190
9.5.4 使用 DataAdapter+DataSet 对象操纵数据	191
9.5.5 断开方式中以 DataReader 方式操纵数据	198
9.5.6 表格展示控件、数据源绑定和导航功能的使用	199
9.6 案例分析 SQL Server 表和文本文件间的数据转换	201
9.7 本章小结	203
习题	204
第10章 多线程技术	205
10.1 概述	205
10.2 System.Threading 命名空间和 System.Threading.Thread 类	206
10.3 创建自定义线程类	207
10.4 创建自定义线程调度类	209

10.5 多线程间的同步	210
10.5.1 使用 lock 关键字实现同步	212
10.5.2 使用 System.Threading.Monitor 类实现同步	213
10.6 利用多线程技术增强用户界面的响应性能	214
10.7 案例分析	216
10.8 本章小结	220
习题	220
参考文献	222
后记	223

Microsoft .NET 概述

.NET represents a major change in the way you'll think about programming. It is a new development platform designed to facilitate object-oriented Internet development. The programming language of choice for this object-oriented Internet-centric platform is C# which builds on the lessons learned from C (high performance), C++ (object-oriented structure), Java (security), and Visual Basic (rapid development) to create a new language ideally suited for developing component-based n-tier distributed web applications.

——Jesse Liberty

微软公司的 .NET 平台自从 2000 年 7 月推出后,已经逐步获得了广大程序开发人员的认同,成为事实上的主流开发平台之一。.NET 平台包含了与微软软件开发相关的绝大多数产品,主要有用于加载和运行应用程序的软件基础架构(.NET Framework 和 ASP.NET)以及开发工具 C# 等。Visual Studio 2010 的 IDE 提供了一个极好的程序开发环境。对程序开发者来说,用 C# 语言(或其他相关语言,如 VC++、VB 等)将 .NET Framework 类库中的组件加以有效组织,可以方便地实现应用程序要求的业务逻辑。

本章将简介 .NET Framework 4.0 与编程开发软件 C#、Microsoft Visual Studio 2010 的 IDE 环境等内容。首先介绍在 .NET 出现之前的程序开发流程,使读者能更好地了解 .NET 带来的编程上的革命性变化;之后简介 .NET Framework 4.0 组成,介绍 C# 编程语言;最后对 Visual Studio 2010 编程开发环境 IDE 进行介绍,以便读者能更有效地利用它进行程序开发。

本章学习要求如下:

- 了解 .NET Framework 与 C# 语言。
- 了解 Visual Studio 2010 IDE 的使用。
- 掌握控制台应用程序、窗体应用程序、Web 应用程序的建立与运行方法。



1.1 .NET 及其之前的程序开发流程

作为众多高级语言(如 VB、VC++、J# 等)和应用程序(如 ASP、Web Services、XML 应用等)的基石,微软公司的 Visual Studio 一直占据着重要的地位,而 .NET Framework 则在其中占据着核心位置。在 .NET 提供的强大功能面前,传统的编程语言诸如 C、C++、VB 语言等,以及传统的架构如 MFC、ATL、STL、COM、CORBA 等都显得有些不足。如纯 Win32 API 不是面向对象的;面向对象的 MFC 比较复杂;COM 虽概念简单,但实际代码亦较复杂。虽然 .NET 及在此基础上的 C# 也不是尽善尽美,但相比之前的开发工具,它拥有明显的优势。本节简介 .NET 及其之前的程序开发流程,以便读者能更好地理解 .NET 带来的革命性变化。

1.1.1 C 及 Win32 API

传统上,针对 Windows 环境的软件开发,一般会联合使用 C 语言和 Windows 的 API (Application Programming Interface),但这往往是一项艰巨的工作,因为开发者需要用微软公司的 SDK(Software Development Kit)直接调用 Windows 的 API 函数,这种程序开发模式主要有如下缺点:

- (1) 手动管理内存,需操纵复杂的指针运算。
- (2) 非面向对象编程,不具备面向对象程序开发的诸多优点。
- (3) Win32 API 有众多的全局函数和数据类型,较难掌握。

1.1.2 MFC、VC++、VB 及 COM 技术

除 C 语言外,VB 6.0、VC++ 6.0、微软基础类库 MFC 以及组建对象模型 COM 等,均为编程提供了便利,使得面向对象的程序开发的优点逐渐显现出来,为编程带来了诸多方便。例如,MFC 为程序开发提供了一些重要的类,它把原始 Windows API 的子集包装成类、宏等,封装了许多函数和控件(例如颜色选择、文件选择),方便编程及调用,简化了 Windows 程序开发。一般来说,基于 MFC 的应用程序运行大致需要如下几个步骤:

- (1) 产生应用程序对象(如 theApp),配置内存,设定初值。
- (2) 调用 AfxWinMain()函数,执行 AfxWinInit()函数,调用 AfxInitThread()函数,加大消息队列。
- (3) AfxWinMain()执行 InitApplication()。
- (4) 从 CWinApp 派生新类作为该应用程序的程序类,一般要重载用于程序初始化的 InitInstance()函数。CWinApp 构造函数将自动调用 InitInstance()函数。InitInstance()函数创建一个诸如 CMyFrameWnd 的对象或其他相关对象。
- (5) CMyFrameWnd 等相关的构造函数调用 Create()函数产生主窗口,可创建该窗口的构造函数、析构函数等。

(6) 回到 `InitInstance()` 中继续执行 `ShowWindows()` 显示主窗口。

(7) 执行 `UpdateWindow()` 函数, 发出 `WM_PAINT` 消息。

(8) 回到 `AfxWinMain()` 函数, 执行 `Run()` 函数, 进入消息循环。

从上面的分析可见, 基于 MFC 的程序仍然比较烦琐, 程序入口不清晰, 类间调用复杂, 给初学者带来一些不便。

在其他开发程序方面, VB 6.0 不是一个完全的面向对象程序, 只是支持对象, 且不便开发多线程的应用程序, 但是它比 VC++ 简单, 适合初学者; Java 来源于 C++, 它为程序员提供了许多预定义的各种“包”, 方便引用, 但它基本不支持与其他语言的集成。

对组件技术而言, 利用 COM 能创建一个可复用的二进制模块, 能以语言无关的方式被访问(如 C++ 创建的 COM 类可被 VB6 调用, Delphi 可调用由 C 创建的 COM 类), 但其脆弱的组件注册问题和部署的烦琐, 使其应用受限。为了简化 COM 开发的复杂性, 已经出现了相当数量的支持 COM 的框架, 如活动模板库 ATL 提供了一组简化创建 COM 类型的类、模板和宏等, 但这尚不足以隐藏 COM 的复杂性。

1.2 Microsoft .NET 及 .NET Framework 简介

.NET 平台为创建分布式的、基于 Web 的应用程序奠定了良好的开发基础。它支持标准的 Internet 协议(包括 HTTP、XML、SOAP 等), 从而实现了异构系统间的集成和通信。和 COM 不同的是, .NET 支持跨语言集成、跨语言异常处理和跨语言调试, 并不关注 COM 底层的操作, 简化处理过程(不需要将二进制单元注册到系统注册表中)。它主要包括基础架构 .NET Framework (包括 IIS、ASP.NET、分布式事物处理协调器等)、开发语言 C#、企业级服务器(进行企业集成和管理所有基于 Web 的各种服务器应用系列产品, 如 SQL Server、BizTalk Server 等)、Web Service 构件(提供一系列高度分布、可编程的公用性网络服务)、集成开发环境 IDE 等。更难能可贵的是, .NET 可以在非微软操作系统(如 Linux、FreeBSD、Mac OS X)上开发和执行, 详情可参阅相关文献, 在此不再赘述。

其中, .NET Framework 是 .NET 的核心部分, 它包括了开发人员编写、生成、测试和部署 .NET 应用程序所需要的核心服务, 是用于生成、部署和运行 XML Web Services 和应用程序的必备基础框架。从 Windows 2003 开始, .NET Framework 成为了 Windows 的一个必要组件。.NET Framework 提供了一个多语言组件开发和执行的环境, 对已有代码具有完全的互操作性、完全的语言集成, 且不关注 COM 底层操作, 是真正简化的部署模型。概括来说, .NET Framework 主要由三部分组成^[1-6]:

(1) 公共语言运行时(Common Language Runtime, CLR)环境。它提供了一个可靠而完善的多语言运行环境, 简化了应用程序的开发、部署和管理, 实现了组件能在多语言环境下的跨平台工作。程序运行时, CLR 除负责满足此组件在其他组件上可能具有的依赖项外, 还负责管理内存分配、异常处理、启动和停止线程、安全策略等, 而原先这些任务是需要编程人员借助语言来实现的。图 1.1 的中部表示了 CLR 的基本功能。从 CLR 的功能上可以看出, .NET 的垃圾回收器会自动释放所分配的内存, 因此开发者的负担减



轻了。从运行时的角度看,.NET 是基于 Windows 操作系统的跨设备的统一运行平台,凡是符合公共语言规范(Common Language Specification,CLS)的程序语言所开发的程序都可以在任何有 CLR 的系统上执行,具有部分的与平台无关特性。从这一点上看,CLR 和 Java 虚拟机有相似之处。

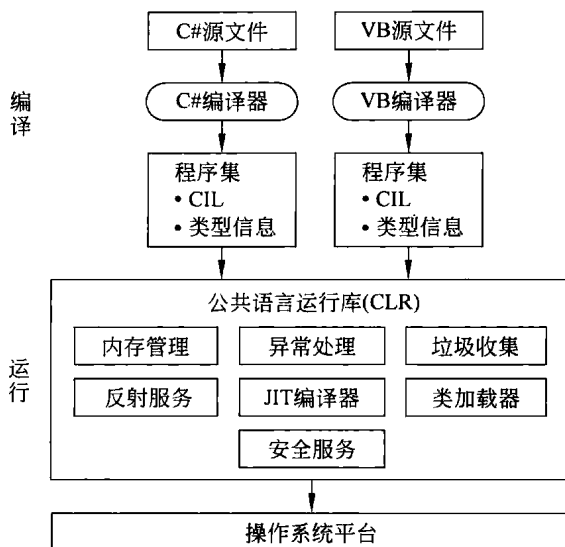


图 1.1 CLR 在程序运行中的基本功能

(2) .NET 的基础类库(Basic Class Library, BCL)或称统一编程类。它是 .NET 框架使用的一个大的类库,提供了几乎所有应用程序都需要的公共代码,解决了由于诸如 C++ 和 Java 等开发语言的差异而造成了代码共享和迁移问题。BCL 隐藏了原始 Windows API 的复杂性,提供了被所有支持 .NET 语言所使用的一致对象类型,通过创建跨编程语言的公共 API 集,使跨语言继承、错误处理和调试成为可能。由于它提供了几乎所有应用程序都需要的公共代码,为开发人员提供了统一的、面向对象的、分层的和可扩展的类库集,因此,它就好似“世界语”一样,“统一”了编程语言。

(3) ADO.NET、ASP.NET 以及 XML Web Services 等构件。

① ADO.NET 是 .NET Framework 提供的新一代的面向对象的数据处理技术,利用它可以快捷地开发数据库应用程序。

② ASP.NET 提供了一个 Web 应用程序模型并且包含使生成 ASP Web 应用程序变得简单的控件集和结构。ASP.NET 包含封装公共 HTML 用户界面元素(如文本框和下拉菜单)的控件集,并以 HTML 的形式将用户界面推送到浏览器。在服务器端,这些控件公开一个面向对象的编程模型,为 Web 开发人员提供了面向对象的方法。ASP.NET 还提供会话状态管理和进程回收等,进一步减少了开发人员必须编写的代码量并提高了应用程序的可靠性。而在 .NET 出现前,传统的 Web 应用程序需要通过 IIS 来管理以便远程客户端通过浏览器来进行访问。.NET 在开发过程中不需要 IIS,而是直接启动进程来执行 Web 应用程序,大大方便了 Web 程序调试。

③ XML Web Service 是一种能够在 Internet 环境中直接调用的组件。.NET

Framework 提供了创建、测试和部署 XML Web Service 的工具和类。使用 XML Web Service, ASP. NET 开发人员可编写自己的业务逻辑并使用 ASP. NET 通过 SOAP 交付该服务。

图 1.2 显示了 .NET Framework 的基本结构,其中下层为上层提供支持和服务。

较之以前的编程环境,.NET Framework 大大改进了编程环境,主要优点有^[7]:

- 拥有面向对象的开发环境。
- 类型安全性好。
- 拥有自动垃圾回收的功能。
- 用不同的 .NET 语言编写的软件可以无缝连接。
- 允许 .NET 的代码调用并使用非 .NET 的、但通过标准 Win 32 DLL 导出的纯 C 函数的代码,如 Windows API;并且允许与 COM 相互操作,即 .NET 软件能调用 COM 组件,COM 也能调用 .NET 组件。
- 在部署时,.NET 程序不需要使用注册表注册即可完成部署。
- 允许一个 DLL 的不同版本在同一台机器上存在。

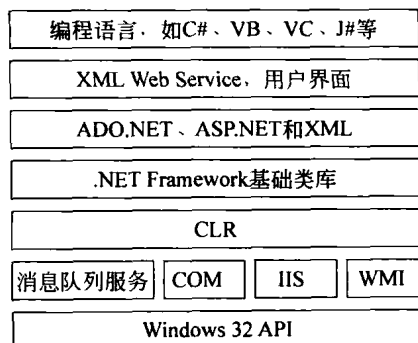


图 1.2 .NET Framework 的结构

1.3 Visual Studio .NET 2010 简介

Visual Studio 是一套完整的开发环境,用于生成基于 ASP. NET 的网络应用程序、XML Web Services、桌面应用程序和移动应用程序等,并用来创建符合 .Net Framework 的程序。它还为建立基于 .NET Framework 应用程序提供了一个可视化集成开发环境 IDE,为大多数主流编程语言提供了一个简单且统一的编码环境,包括代码编辑器、XML 编译器、HTML 编译器、SQL Server 接口、图形化的设计服务器端构件的设计器、监控远程机器的 Server Explorer、团队合作等。不论是 Visual Basic .NET、Visual C++ .NET,还是 Visual C# .NET 和 Visual J# .NET,都使用相同的集成开发环境 IDE,该环境允许共享工具,这有助于创建混合语言解决方案。Visual Studio 的特点有:

- Visual Studio 提供了一组项目模板,可加快开发应用程序的速度。
- 通过 Visual Studio 的扩展性,可以编写自己的项目模板和加载项。
- Visual Studio 社区是一个活动社区,可在其中分享经验和代码以及获得问题的解答。
- Visual Studio Team System 2010 打造一个功能平等、共同分担的平台以用于组织应用程序生命周期管理过程,提供应用程序生命周期管理。客户将在不同规模的组织内创建应用。VS2010 将确保在宽泛的范围内应用开发都得到支持。

随着 Visual Studio 2010 的发布,.NET Framework 4 主要为用户提供了以下新功能和改进: