

第2版

C#语言

Windows程序设计

于国防 李 剑 张玉杰 ● 编著

- 通俗易懂，深入浅出
- 取材广泛，注重实际应用
- 简洁、清晰、高效的示例代码
- 配套教学资源丰富



清华大学出版社

C#语言 Windows 程序设计

(第 2 版)

于国防 李 剑 张玉杰 编著

清华大学出版社

北 京

内 容 简 介

本书结合微软公司推出的 Visual Studio 2012 集成开发环境的应用,讲述如何使用 C#语言开发基于 .NET Framework 平台的 Windows 窗体应用程序。全书共分 3 篇 17 章,内容由浅入深、循序渐进,包含了变量、运算符与表达式、流程控制、窗体及控件、面向对象的程序设计、文件 I/O、GDI+、数据库访问以及网络通信等 C#编程最常用的要点知识。书中主要章节后新增扩展学习,便于学习者趁热打铁,拓宽知识;还增加应用篇的典型示例,便于学习者参考引用,增强编程实战技能。

本书适用于计算机科学以及电子与信息工程等相关专业大、中专院校师生的 C#教学用书,也可以作为 C#语言程序开发者的参考书。

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#语言 Windows 程序设计/于国防,李剑,张玉杰编著. —2 版. —北京:清华大学出版社,2015
ISBN 978-7-302-39605-5

I. ①C… II. ①于… ②李… ③张… III. ①C 语言-程序设计 IV. ①TP312

中国版本图书馆 CIP 数据核字(2015)第 080926 号

责任编辑:贾小红

封面设计:刘 超

版式设计:魏 远

责任校对:王 云

责任印制:王静怡

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:三河市少明印务有限公司

经 销:全国新华书店

开 本:185mm×260mm 印 张:16.25 字 数:385 千字

版 次:2010 年 9 月第 1 版 2015 年 8 月第 2 版 印 次:2015 年 8 月第 1 次印刷

印 数:1~3000

定 价:36.00 元

产品编号:060146-01

再版序

《C#语言 Windows 程序设计》(ISBN: 978-7-302-23375-6)自2010年9月出版以来,受到了广大读者和众多高校的欢迎和好评,近年来已经重印了4次。尽管如此,无论是为了采纳使用者的良好建议,还是为了跟上技术的最新发展,笔者都有义务及时对本书进行更新和完善,于是,根据出版社的要求,本书编写组再次投入大量时间和精力对第1版的内容进行修改和补充。

关于第2版的相应修改,具体说明如下。

◆ 章节编排更规范

在第1版的基础上,所有章节重新编排,使技术内容前后衔接、循序渐进。

◆ 内容编写改进

对第1版的部分重点、难点问题,字斟句酌,使之更加通俗易懂,并将代码进一步优化编写;尽可能地增加屏幕截图的尺寸,增强插图的易视性;尽可能地使所有表格不再跨页显示,提高内容的易读性。

◆ 采用新版开发工具

开发工具由第1版的 Visual Studio 2005 更新为 Visual Studio 2012,书中所有屏幕截图均在 Visual Studio 2012 运行环境中进行,所有示例源程序(包含实验)也都是更新版。

◆ 补充常用或新技术

通过3个章节,并结合示例程序,分别补充了常用的 Windows 打印组件的基本用法、创新性的 LINQ 技术应用知识以及实用的视频应用程序设计。

◆ 增设“扩展学习”

主要章节后新增一节“扩展学习”,技术难度略有提高,但可使读者对本章的技术掌握进一步提升,知识面进一步扩展。

◆ 规划“应用篇”

利用有限的篇幅,将增补的视频应用以及第1版的图像处理和邮件发送与接收这3个方面的程序设计规划为“应用篇”,引领读者实战入门。

另外,虽然第2版的知识含量较第1版有较多扩充,但全书篇幅并无明显增加。

编 者

前言

自 2000 年 6 月微软公司推出 Microsoft.NET 以来,.NET 技术及其相关产品在信息技术领域得到了广泛应用。而为 .NET Framework 量身定做的新一代面向对象的 Visual C# 语言,也随之成为业界主流的程序设计语言之一。C# 不仅功能强大,而且其程序设计简洁灵活、易学易用。

本书结合运用优秀的开发工具 Visual Studio.NET (简称 VS.NET),重点讲述功能强大、应用广泛的 C# 语言 Windows 应用程序设计。

全书共分 3 篇 17 章,章节内容前后衔接,技术要点逐层深入;示例分析图文并茂;代码注释详尽明晰;难点问题随时“说明”,重点知识及时“提示”,特别是每章的“扩展学习”,使技术提升更进一步。全书讲解注重“教与学面对面”的知识传授风格,使读者始终在一种轻松自然、自信好学中进步。各章内容概括如下。

第 1 章:有限的篇幅介绍了 .NET 和 C# 的基本知识,为后续相关技术的学习奠定了基础。

第 2 章:简要介绍 Visual Studio 的发展历史和优点,重点介绍 Visual Studio 2012 集成开发环境的安装方法,从而为后续的程序设计准备好了称手的兵器。

第 3 章:变量、数据类型、运算符以及流程控制等可谓任何编程语言之必备基础,C# 语言当然也不例外,但考虑到有的读者可能因为学习过其他编程语言,已经具备了一定的基础知识,所以,本章的内容讲解既重点突出,又简明扼要。

第 4 章:以一个简单的上机考试系统开发为目标,结合 Visual Studio 2012 的运用,介绍 Windows 程序的界面设计,其中,将窗体(含控件)、菜单以及工具栏等的创建有机地融为一体。

第 5 章:首先简要介绍面向对象编程的基本知识,并重点引入了 .NET 的命名空间概念及其用法说明,此乃 C# 程序设计之重要技术基础。而后,结合第 4 章设计的上机考试系统窗体,逐步完善其各项功能的程序设计。至此,轻松的设计、实效的结果,令初学者自信地跨入 C# 语言程序设计的大门!

第 6 章:Windows 窗体的显示模式与对话框简单、易用,本章以有限的篇幅简单介绍,并为后续的相关应用奠定技术基础。

第 7 章:进程和线程是实时及多任务应用程序开发的技术基础。结合形象、直观的示例程序,并采用类比式的分析方法,深入浅出地重点讲述多线程开发,并为后续的网络程序设计奠定相关的技术基础。

第 8 章:首先以精简的篇幅介绍 Access 和 SQL Server 数据库的应用和开发知识,然后结合一个简单、易理解的管理系统示例,逐步引导读者综合运用 ADO.NET 的相关技术以及 SQL 语句,开发 C# 应用程序。

第 9 章:LINQ 是 Visual Studio 2008 和 .Net Framework 3.5 版本开始具备的一项突出性

创新,它在对象领域与数据领域之间架起了一座桥梁,因其重要性和实用性,本章继续结合示例程序,介绍 LINQ 技术应用知识。

第 10 章:首先通过一个形象化的 GDI+绘图基本步骤类比,使读者在运用 GDI+之前就对其基本用法有一个清晰、直观的整体认识,然后在此基础上结合不同的示例,分别介绍 GDI+的图形绘制、文本呈现等基础知识。

第 11 章:结合示例,介绍常用的 Windows 打印组件的基本用法。

第 12 章:介绍最基本、最常用的文件读/写程序设计方法,并为后续的网络文件流操作奠定技术基础。

第 13 章:网络编程是程序设计的重点,也是难点。遵循由浅入深的原则,结合应用示例,分别详解了基于 TCP 协议和 UDP 协议的网络程序设计。行文中的具体分析、代码中的详细注释,都极大地增强了读者对本章知识的理解和掌握。

第 14 章:有限的篇幅、详细的步骤,介绍最基本的 Windows 程序的安装和部署方法。

第 15、16 和 17 章:为实用扩展篇,分别介绍了视频应用、图像处理基础以及邮件发送与接收的程序设计。使读者能够在此前章节知识学习的基础上,学以致用,开发一些简单、实用的 Windows 应用程序。

本书由于国防、李剑和张玉杰共同编著。于国防编写了第 3~11 章以及第 13 章,李剑编写了第 1、12、14 章以及第 15 章,张玉杰编写了第 2、16 章以及第 17 章。全书由于国防统稿。

作为本书的编者,我们不仅是从事软件开发项目的程序员,更是传道授业的高校教师,所以,我们更深知编书助教的责任之重,于是,在编写此书的过程中,我们虚心请教同行教师和专业人士,广泛征求学生建议,并参考大量相关教材和参考书,可谓竭尽全力、精益求精,尽管如此,我们仍然要诚恳地承认,由于水平所限,书中难免会有疏漏或不妥之处,殷切希望广大读者、同仁批评指正。

为了便于读者测试和学习书中示例,随书提供了书中所有示例(包括实验)的源程序;同时,为了便于教师进行多媒体课堂教学,随书还提供了相应的 PPT 课件。

编 者

目 录

第1部分 基础篇

第1章 .NET 与 C#简介	2	3.3.1 简单值类型	17
1.1 .NET 简介	2	3.3.2 结构类型	19
1.1.1 .NET 平台简介	2	3.3.3 枚举类型	20
1.1.2 .NET Framework 结构与功能	3	3.4 引用类型	20
1.1.3 .NET Framework 类库	4	3.4.1 类类型	20
1.1.4 公共语言运行库 (CLR) 简介	4	3.4.2 字符串类型	21
1.2 C#简介	6	3.4.3 数组类型	22
1.2.1 C#语言的由来	6	3.5 类型转换	23
1.2.2 C#语言的特点	6	3.5.1 隐式转换	24
习题	7	3.5.2 显式转换	24
第2章 Visual Studio 简介与安装	8	3.5.3 使用 Convert 类的方法进行 转换	24
2.1 Visual Studio 的发展历史和 优点	8	3.6 运算符与表达式	25
2.2 Visual Studio 2012 集成开发 环境安装	9	3.6.1 运算符	25
习题	12	3.6.2 表达式	27
第3章 C#语言基础	13	3.7 流程控制	27
3.1 标识符与关键字	13	3.7.1 条件语句	28
3.1.1 关键字	13	3.7.2 循环语句	29
3.1.2 标识符	14	3.7.3 跳转语句	31
3.2 常量和变量	14	3.8 异常处理	33
3.2.1 常量	14	3.8.1 异常简介	33
3.2.2 变量	15	3.8.2 异常处理语句	33
3.3 值类型	17	扩展学习: 数值除法应用技巧	34
		习题	36

第2部分 入门篇

第4章 Windows 窗体设计基础	38	4.2 项目与解决方案	39
4.1 Windows 窗体应用程序设计 流程	38	4.3 窗体与控件	39
		4.3.1 窗体简介	39

4.3.2 控件简介	40	5.4 上机考试系统程序设计	66
4.4 简单的上机考试系统界面设计	40	5.4.1 菜单栏程序设计	67
4.4.1 系统功能设计及项目创建	40	5.4.2 工具栏程序设计	68
4.4.2 Windows 窗体应用	42	5.4.3 考试评分程序设计	69
4.4.3 PictureBox 控件简介及其应用	43	5.4.4 考试计时程序设计	70
4.4.4 Label 控件简介及其应用	45	5.5 命名空间	71
4.4.5 添加新窗体	46	5.6 代码的书写规范及其注释用法	72
4.4.6 TextBox 控件简介及其应用	47	5.6.1 代码书写规范	73
4.4.7 RadioButton 与 Panel 控件简介及其应用	48	5.6.2 代码注释方法	73
4.4.8 ListBox 与 ComboBox 控件简介及其应用	50	扩展学习: 程序调试	74
4.4.9 CheckBox 控件简介及其应用	51	习题	76
4.4.10 Timer 控件简介及其应用	53	第6章 Windows 窗体的显示模式与对话框	77
4.4.11 Button 控件简介及其应用	54	6.1 Windows 窗体的显示模式简介	77
4.4.12 设置项目的启动窗体	55	6.2 Windows 通用对话框简介	78
4.4.13 MenuStrip 控件简介及其应用	56	6.3 MessageBox 消息对话框简介及其应用	78
4.4.14 ToolStrip 控件简介及其应用	57	扩展学习: 关于对话框简介及其应用	80
扩展学习: StatusStrip 控件简介及其应用	58	习题	82
习题	60	第7章 Windows 进程与线程程序设计	83
第5章 Windows 窗体应用程序设计	61	设计	83
5.1 面向对象编程简介	61	7.1 进程与线程简介	83
5.1.1 类和对象	61	7.2 进程程序设计	84
5.1.2 类的声明与实例化	61	7.3 线程程序设计基础知识	86
5.1.3 封装、继承与多态性	63	7.4 多线程程序设计	90
5.2 方法	63	7.4.1 多线程简介	90
5.2.1 方法的声明与调用	64	7.4.2 多线程互斥与同步简介	91
5.2.2 方法的参数简介	64	7.4.3 多线程互斥程序设计	91
5.2.3 方法的重载简介	66	7.4.4 多线程同步程序设计	101
5.3 事件	66	7.5 Windows 窗体控件的跨线程访问	104
		7.5.1 非线程安全的窗体控件访问简介	105

7.5.2 线程安全的窗体控件访问	
简介	105
扩展学习: 线程安全的窗体控件访问	
程序设计	105
习题	107
第 8 章 数据库访问程序设计	109
8.1 数据库应用基础	109
8.1.1 数据库概述	109
8.1.2 Access 数据库简介	110
8.1.3 SQL Server 数据库简介及其	
应用	110
8.1.4 SQL 简介	113
8.2 ADO.NET 简介	115
8.2.1 ADO.NET 功能简介	115
8.2.2 ADO.NET 数据提供程序	116
8.2.3 ADO.NET 数据提供程序的核心	
对象	116
8.3 ADO.NET 数据库访问	117
8.3.1 ADO.NET 数据库访问基础	
知识	118
8.3.2 利用 SqlConnection 类连接数	
据库	119
8.3.3 DataSet 数据集与 DataAdapter	
数据适配器简介及应用	120
8.3.4 利用 SqlCommand 类访问数	
据库	123
8.3.5 DataReader 类简介	125
8.3.6 利用 DataTable 类和 DataView	
类更新数据及控制视图	126
扩展学习: 将 DataGridView 控件的	
数据导出为 Excel	130
习题	132
第 9 章 LINQ 技术及其应用	133
9.1 LINQ 简介	133
9.2 LINQ to SQL 简介	134
9.3 LINQ to SQL 应用	134
9.3.1 利用 LINQ to SQL 查询	
SQL Server 数据	134
9.3.2 利用 LINQ to SQL 添加	
SQL Server 数据	138
9.3.3 利用 LINQ to SQL 修改	
SQL Server 数据	139
9.3.4 利用 LINQ to SQL 删除	
SQL Server 数据	140
扩展学习: 利用 LINQ to SQL 实现	
聚合查询	141
习题	142
第 10 章 GDI+图文绘制程序设计	143
10.1 GDI+简介	143
10.2 GDI+绘制图文的基本	
步骤	144
10.3 GDI+绘制图文的技术	
基础	144
10.3.1 .NET 图形图像处理命名	
空间	144
10.3.2 Graphics 类	145
10.3.3 颜色	145
10.3.4 坐标系统	146
10.3.5 画笔与画刷	147
10.4 绘制基本图形	148
10.4.1 绘制直线	148
10.4.2 绘制多边形	149
10.4.3 绘制曲线	150
10.4.4 绘制矩形	152
10.4.5 绘制椭圆 (或圆)	153
10.4.6 绘制圆弧	153
10.4.7 绘制扇形	154
10.5 绘制文本	155
10.5.1 Font 类简介	155
10.5.2 利用 Font 类绘制文本	156
扩展学习: 鼠标画线程序设计	157
习题	160

第 11 章 Windows 打印组件程序	
设计	161
11.1 PrintDocument、PrintPreviewDialog 及 PrintDialog 组件简介及其 应用	161
11.2 PrintPreviewControl 组件简介 及其应用	164
11.3 PageSetupDialog 组件简介及 其应用	166
扩展学习: 图片打印	168
习题	170
第 12 章 文件 I/O 操作程序设计	171
12.1 文件与流简介	171
12.2 目录类、文件类及路 径类	171
12.3 文件对话框类	172
12.3.1 OpenFileDialog 类	173
12.3.2 SaveFileDialog 类	173
12.4 流操作	173
12.4.1 文件流操作	173
12.4.2 网络流操作简介	179
扩展学习: 判断文件是否正在被 使用	180
习题	181
第 13 章 网络通信程序设计	182
13.1 网络协议简介	182
13.2 定义和获取主机消息	183
13.3 Socket 简介	184
13.4 套接字网络通信简介	185
13.5 TCP 网络通信程序设计	187
扩展学习: UDP 网络通信程序 设计	194
习题	198
第 14 章 Windows 程序的安装	
部署	199
14.1 Windows 程序的安装部署 简介	199
14.2 Windows 程序的安装部署 方法	199
习题	206
第 3 部分 应 用 篇	
第 15 章 视频应用程序设计	208
15.1 VFW 基础知识	208
15.2 基于 VFW 的视频应用程序 设计	209
习题	213
第 16 章 图像处理程序设计	214
16.1 GDI+图像处理简介	214
16.2 GDI+图像像素操作的 3 种 方法	214
16.3 GDI+图像处理基础程序 设计	218
16.3.1 图像灰度化	220
16.3.2 图像滤波	222
16.3.3 图像边缘检测	224
习题	228
第 17 章 邮件发送与接收程序	
设计	229
17.1 邮件发送	229
17.1.1 SMTP 协议简介	229
17.1.2 利用 System.Net.Mail 发送邮件 的程序设计	230
17.2 邮件接收	234
17.2.1 POP3 协议简介	234
17.2.2 利用 LumiSoft.Net 接收邮件的 程序设计	234
17.3 IMAP 协议简介	243
习题	243

参考文献.....	244	实验 4 LINQ 技术应用程序设计 ...	246
附录 A 实验参考	245	实验 5 GDI+绘图与显示程序 设计	247
实验 1 窗体及控件程序设计	245	实验 6 打印组件应用程序设计	247
实验 2 线程开发程序设计	245	实验 7 文件读/写程序设计	248
实验 3 数据库访问程序设计	246	实验 8 网络通信程序设计	248

▶▶ 第 1 部分

基础篇

- 第 1 章 .NET 与 C#简介
- 第 2 章 Visual Studio 简介与安装
- 第 3 章 C#语言基础

第 1 章 .NET 与 C#简介

学习要点

- 了解 .NET 平台、.NET Framework 及 .NET Framework 类库
- 了解 CLR 的基本功能
- 了解 C#语言的由来及其特点

1.1 .NET 简介

1.1.1 .NET 平台简介

微软公司总裁史蒂夫·鲍尔默对“什么是.NET”的解释是：.NET 代表了一个集成、一个环境、一个编程的基本结构，它可以作为一个平台来支持下一代的互联网，是一个可实现的环境。比尔·盖茨对.NET 的评价是：这是一个崭新的平台，它将影响我们今后编写的每一行应用程序代码，它将重新定义用户界面，屏幕内容以及人机交互方式都将大为改观，如同从 DOS 到 Windows 的转变一样，微软的所有产品都将受其影响。

如今.NET 已是一个可以支持下一代互联网服务和运营的平台。这个平台包含微软新一代的操作系统、大量的互联网服务软件、对各种设备（移动通信设备、机顶盒、信息家电等）的支持和应用软件开发套件——Visual Studio.NET 等。在这个平台下，从用户的角度看，只需发出请求，无论在什么设备上运行着什么操作系统，只要安装了 .NET Framework，就可以运行.NET 可执行程序，就能获得所需要的服务，而无须关心后台的复杂操作；从开发人员的角度看，平台对开发进行强有力的支持，方便创建各种应用软件（无论是传统的 Windows 应用软件还是 Web 站点），而且部署和发布应用程序更加简便。

.NET 平台结构如图 1-1 所示。它采用开放式的体系结构，集中体现了微软公司在软件设计领域的先进技术成就，其核心技术包括.NET Framework、.NET 企业服务器、模块服务以及 Visual Studio.NET。



图 1-1 .NET 平台结构

- ❑ .NET Framework: 是.NET 战略的核心, 它为.NET 平台下应用程序的开发和执行提供基本的环境架构。
- ❑ .NET 企业服务器: 是一系列的技术服务产品, 为用户提供数据通信、协作、交换、存储等服务, 包括 SQL Server、BizTalk Server、Office Communications Server、Host Integration Server 和 Exchange Server 等。
- ❑ 积木块服务: 主要是微软公司提供的 COM+组件服务和 XML Web 服务技术, 在应用程序中作为功能模块调用, 以便快速完成开发。

1.1.2 .NET Framework 结构与功能

.NET Framework (.Net 框架) 是支持生成和运行.NET 应用程序以及 XML Web Services 的内部 Windows 组件。为了创建和运行基于.NET 平台的应用程序, .NET Framework 提供了一个多语言组件开发、编译和运行的环境。图 1-2 是.NET Framework 的组织结构图。

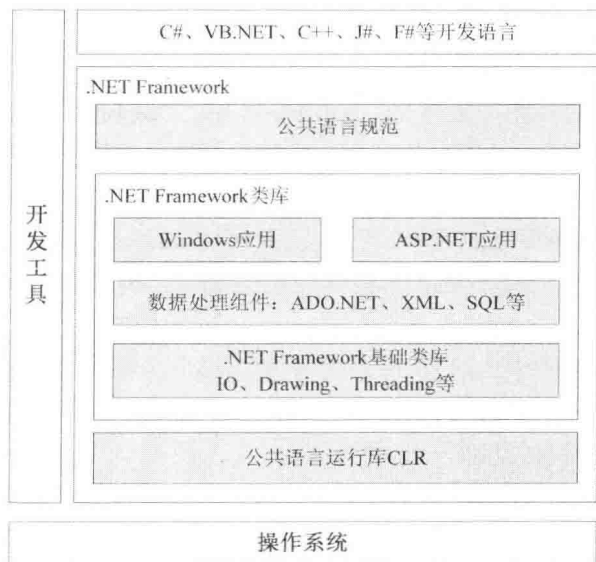


图 1-2 .NET Framework 的组织结构

.NET Framework 主要包括 3 个部分: 公共语言规范、.NET Framework 类库和公共语言运行库。其中, 公共语言运行库是.NET Framework 的基础, 其作用是托管和执行代码, 并向托管代码提供内存管理、线程管理等服务; .NET Framework 类库是一个综合性的面向对象的可用类型集合, 利用它可以开发多种应用程序; 公共语言规范是公共语言运行库支持的语言功能的子集, 包括几种面向对象的编程语言的通用功能。符合公共语言规范的组件和工具能够保证与其他符合公共语言规范的组件和工具交互操作。

.NET Framework 正式发布的版本从 1.0 开始, 逐步升级为 1.1、2.0、3.0、3.5、4.0 以及 4.5, 并且仍在不断升级中。.NET Framework 1.0、1.1、2.0 这 3 个版本是彼此完全独立的, 可以同时位于一台计算机上, 每个版本都有自己的公共语言运行库、类库和编译器等。

 **提示：**虽然 2.0 版本以上的 .NET Framework 具有向下兼容性，但是有的软件可能无法运行于高版本的 .NET Framework 中，所以，仍需要安装相应版本的 .NET Framework。

.NET Framework 实现的功能目标如下：

- ☐ 对于不同的编程语言，提供一个统一的面向对象的编程环境；对于同一种编程语言，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，其编程环境是基本一致的。
- ☐ 代码执行环境将软件部署和版本控制冲突最小化。
- ☐ 代码执行环境能够提高代码执行的安全性。
- ☐ 使开发人员能够轻松开发不同类型的程序（如传统的命令行、图形化的 Windows 窗体应用程序、ASP.NET 应用程序、XML Web Services 以及 Windows 服务）。
- ☐ 能够按照工业标准生成所有应用，以确保基于 .NET Framework 的代码可与其他任何代码集成。

1.1.3 .NET Framework 类库

.NET 提供了强大的类库，如同 VC 有 MFC 类库、Delphi 有 VCL 类库、Java 有 Swing 和 AWT 等类库一样，这些类库封装了系统底层的功能，并提供了很好的操作方式，可以使开发者轻松地构建应用程序。.NET Framework 类库是一个由 Microsoft .NET Framework 中包含的类、接口和值类型组成的库，它封装了对 Windows、网络、文件、多媒体的处理功能，是所有支持 .NET 的编程语言都能够使用的类库。类库的源代码保存在称为程序集的 DLL 文件中。

.NET Framework 类库中大约有 7000 多个类（每个类可能会有上百个方法或属性），这些类库被分类打包，功能相似的类库包含在同一个命名空间下，通过调用不同的命名空间来使用类库，完成不同的功能。

在 .NET 平台中进行开发的任何编程语言都使用同样的类库，由于统一了开发模式，开发者再也不必也没有必要掌握全面的开发技术（如 ATL、COM、MFC、ASP、VB 以及 C++ 等）。所以，在 .NET Framework 下，无论是开发桌面程序还是开发 Web 程序，因为使用的是同样的类库和几乎同样的技术，使得应用程序的开发因此变得非常简单。

 **提示：**学习类库时，不需要掌握其全部应用技术，了解类库的基本架构和实现原理，并能将其正确地运用于实际程序的开发中才是最重要的。

1.1.4 公共语言运行库（CLR）简介

公共语言运行库（Common Language Runtime, CLR）是 .NET Framework 的一大特色。CLR 是一个执行并管理代码的中枢，它主要提供内存管理、线程管理、远程处理、异常处理、类加载、自动垃圾收集、安全和认证、通过 BCL 得到广泛的编程功能（包括 Web 服务和数据服务等），并且 CLR 还强制实施严格的类型安全检查，实施提高安全性和可靠

性的其他形式的动作。

代码管理的概念是 CLR 的基本概念。以运行库为目标的代码称为托管代码，即在 CLR 下运行的代码都是托管代码，而不以运行库为目标的代码称为非托管代码。C#语言中代码也能使用非托管方式，但是考虑到代码的安全性，除非必要，不建议这么做。托管代码具有许多优点，如跨语言集成、跨语言异常处理、安全性增强、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。

CLR 的工作流程（托管代码的执行过程）如下。

（1）选择编译器

CLR 是一个多语言执行环境，支持各种数据类型和语言功能，开发人员可能使用不同的编程语言编写代码，因此必须选择针对该编程语言的编译器或第三方编译器。为了能够正确使用 CLR 的功能，开发人员使用的语法必须符合公共语言规范 CLS。

（2）将代码编译为 Microsoft 中间语言（MSIL）

当源代码编译为托管代码时，编译器将源代码编译为 Microsoft 中间语言（MSIL），这是一组可以有效地转换为本机代码且独立于 CPU 的指令，不是 CPU 能够直接执行的代码。MSIL 包括用于加载、存储和初始化对象以及对对象调用方法的指令，还包括用于算术和逻辑运算、控制流、直接内存访问、异常处理和其他操作的指令。

（3）将 MSIL 编译为本机代码

在执行时，实时（JIT）编译器将 MSIL 编译为本机代码。这种本机代码是针对特定 CPU 结构的 JIT 编译器生成的（CLR 为每种 CPU 结构都提供了 JIT 编译器）。JIT 编译器并不是把整个应用程序一次编译完（这样会有很长的启动时间），而是只编译它调用的那部分代码（这是其名称由来）。代码编译过一次后，就将得到的内部可执行代码存储起来，直到退出该应用程序为止，这样在下次运行这部分代码时就不需要重新编译了。Microsoft 认为这个过程要比一开始就编译整个应用程序代码的效率高得多，因为一个应用程序在运行过程中，其大部分代码实际上并不是在每次运行过程中都执行的。在此编译过程中，代码必须通过类型安全的验证。

（4）运行代码

CLR 提供能够使执行发生以及可在执行期间使用的各种服务的结构。这些服务涉及垃圾回收、安全性、与非托管代码的互操作性、跨语言调试支持、增强的部署以及版本控制支持等。

整个 CLR 执行的工作流程如图 1-3 所示。

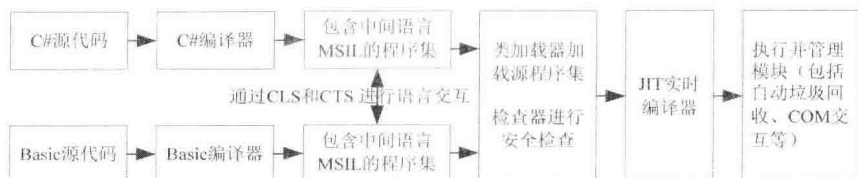


图 1-3 CLR 的工作流程

1.2 C#简介

C#(读作“C sharp”)是微软公司发布的一种安全、稳定、简单、运行于.NET Framework 之上面向对象的编程语言。

1.2.1 C#语言的由来

随着软件开发技术的不断发展,程序员们不但要解决各种技术平台上的组件兼容问题,还需要解决由不同编程语言开发的组件间的集成问题;同时 Web 应用已成为一种趋势,编程语言还应该能够快速地进行网络应用的开发。为了满足这些需求,2000 年微软公司发布了 C#编程语言,是由 Anders Hejlsberg 和 Scott Wiltamuth 领导的小组专门为.NET 平台设计的、运行于.NET Framework 之上的编程语言。这种语言作为微软公司.NET 战略的一部分,是.NET 平台应用的首选语言,其版本随着 Visual Studio.NET 版本的逐步升级而升级。

对 C#的由来有两种解释:从字面意义来解释,C#是 C 语言的开发利器;而微软公司给出的解释是:C#是 C++的升级语言,具有比 C++更优越的开发特性。C#在表达式、运算符和语句等方面沿用了 C/C++的许多特性,而在类型安全、错误处理、版本转换、事件和垃圾回收等方面做了很大的改进和创新。

1.2.2 C#语言的特点

相对于其他常用编程语言,C#的特点可概括如下。

1. 简洁的语法

C#语言相对 C++语言而言,简单易学、容易入门。C#语言淘汰了 C++语言中繁乱的表示符号和伪关键字,使用了有限的、统一的操作符、修饰符和运算符。另外,C#语言很少使用 C++语言中功能强大却难以掌握的指针,因为使用指针可能会带来内存泄漏以及管理漏洞等不安全因素,这将使开发和维护的难度大大增加。而 C#语言中操作的基本上是实例的对象,只有部分类的类型支持指针,而且不建议使用指针。

2. 面向对象的设计

C#语言支持面向对象的所有关键特性,如封装、继承和多态等,是真正纯粹的面向对象的编程语言。

C#语言以类和结构为基础构建所有的类型,每种类型都是一个对象。C#语言通过命名空间对代码进行层次化的管理,所有的常量、变量、属性、方法、事件等都被封装在类中,从而使程序具有更好的可读性,并且避免了发生命名冲突的可能。

3. 完备的安全性

默认情况下,代码工作在一种受托管的环境中,在托管环境下不允许进行类似直接存