

高等学校计算机科学与技术教材

C#程序设计

COMPUTER Science and Technology

□ 陈 锐 张 蕾 李绍华 主 编

□ 郑耀杰 祁文献 夏敏捷 副主编

- 原理与技术的完美结合
- 教学与科研的最新成果
- 语言精练，实例丰富
- 可操作性强，实用性突出



YZLI0890118022

高等学校计算机科学与技术教材

C#程序设计

陈 锐 张 蕾 李绍华 主 编
郑耀杰 祁文献 夏敏捷 副主编



YZLI0890118022

清华大学出版社
北京交通大学出版社

· 北京 ·

内 容 简 介

C#是一种安全的、稳定的、简单易学的面向对象编程语言，它继承了C和C++的强大功能，同时去掉了一些复杂特性，成为目前使用最为广泛的开发语言，是目前大中专院校计算机专业的必修课程。本书内容包括：.NET 开发环境介绍、C#编程基础、面向对象编程基础、面向对象高级编程、目录与文件管理、图形图像与多媒体编程、ADO.NET 数据库开发、水晶报表、Web 应用程序设计基础、ASP.NET Web 服务器控件。

本书内容全面，结构清晰，语言通俗，重难点突出，例题丰富，所有程序无需修改都能够直接运行。本书可作为大中专院校的计算机及相关专业的C#程序设计教材，也可供广大C#开发用户参考。

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 13501256678 13801310933

图书在版编目 (CIP) 数据

C#程序设计 / 陈锐，张蕾，李绍华主编. — 北京：清华大学出版社；北京交通大学出版社，2011.12

(高等学校计算机科学与技术教材)

ISBN 978-7-5121-0819-6

I. ① C… II. ① 陈… ② 张… ③ 李… III. ① C语言-程序设计-高等学校-教材
IV. ① TP312

中国版本图书馆CIP数据核字(2011)第253171号

责任编辑：杨正泽 特邀编辑：刘 冀

出版发行：清华大学出版社 邮编：100084 电话：010-62776969 <http://www.tup.com.cn>
北京交通大学出版社 邮编：100044 电话：010-51686414 <http://press.bjtu.edu.cn>

印刷者：北京市德美印刷厂

经 销：全国新华书店

开 本：185×260 印张：29.25 字数：850千字

版 次：2012年1月第1版 2012年1月第1次印刷

书 号：ISBN 978-7-5121-0819-6/TP·673

印 数：1~3 000册 定价：42.00元

本书如有质量问题，请向北京交通大学出版社质监组反映。对您的意见和批评，我们表示欢迎和感谢。

投诉电话：010-51686043, 51686008；传真：010-62225406；E-mail: press@bjtu.edu.cn。

前言

微软推出的 C# 是 .NET 中最简单、方便和高效的程序设计语言，在继承 C++ 和 Java 等语言优点的基础上，不仅具有封装、继承和多态等特性，而且增加了不少创新元素，广泛应用于开发桌面系统、Web 应用、数据库应用、网络应用等，在不到 10 年的时间，C# 语言得到了快速发展和普及，已成为目前主流的开发工具。

随着 Visual Studio .NET 平台的不断升级，C# 的功能变得更强，应用开发变得更方便。为此，我们以当前最新的 Visual Studio .NET 2008 作为平台，结合近年来的教学和应用开发实践，编写了本书。

本书首先介绍 C# 的特点并大致介绍 .NET 开发平台 Visual Studio .NET 2008，通过简单实例了解控制台应用程序、Windows 应用程序、ASP.NET Web 应用程序的开发过程；然后介绍 Visual Studio .NET 2008 的程序结构和调试技巧；在此基础上，系统地介绍 C# 的编程基础、面向对象编程基础、面向对象高级编程、窗体和控件、目录和文件管理、图形图像和多媒体编程；最后介绍 ADO.NET 和数据操作、水晶报表、Web 应用程序开发、ASP.NET Web 服务器控件、Web 应用程序开发实例。一般来说，通过教程的学习、习题的练习，特别是认真的上机操作，在较短的时间内，读者便基本能够在 Visual Studio .NET 2008 环境下用 C# 解决一些小型应用问题。

本书作者多年来一直从事 C# 程序的开发、数据结构与算法、自然语言理解方面的研究工作，具有较为丰富的计算机理论基础与开发实战经验，本书是作者教学实践经验的结晶，非常适合计算机专业的学生作为教材与自学参考书。

本书的特点

1. 内容全面，讲解详细

为了方便学生学习，本书首先介绍了 C# 语言的开发环境，然后介绍了 C# 语言编程基础、常用控件、图形图像编程、ADO 技术、报表设计、Web 开发。本书内容全面，覆盖了几乎全部 C# 语言的知识点，对于每个知识点，都通过具体实例进行讲解，以便学生迅速掌握 C# 语言。

2. 结构清晰，内容合理

本书将 C# 语言分篇、章、节和小节划分知识点，每章都给出了教学提示与教学目标，给人一目了然的感觉。结构框架清晰，内容讲解合理，逻辑性强，循序渐进，由浅入深，最后通过例子强化知识点，这样的讲解方式使学生更容易理解和消化。

3. 语言通俗，叙述简单

针对每个知识点，使用比较通俗的语言讲解，避免使用晦涩难懂的语句，以便学生理解。这样，学生可以更加容易学习并掌握 C# 语言。

4. 配有习题，巩固知识

在每一章的最后，都配有一定数量的实践题目，在学习了每一章的内容之后，可以通过这些习题试着编写 C# 语言，以巩固本章的学习内容。

本书的内容

第 1 章：介绍了 Visual C#. NET 2008 开发环境及 C# 语言的程序结构，使学生对 C# 语言

有一个初步的了解。

第2章：介绍了C#语言的数据类型、常量与变量、运算符与表达式及C#的流程控制语句，使学生快速入门，能使用Visual Studio开发简单的应用程序。

第3章：主要介绍了常用的数据类型：数组、字符串、集合、堆栈等，使学生初步理解C#中的集合类，能熟练使用常见的类编写较为复杂的程序。

第4章：主要介绍了面向对象编程的一些基础知识：类、方法、属性，使学生掌握面向对象编程方法。

第5章：主要介绍了面向对象编程的一些比较高级的知识：继承、多态、接口、序列化、泛型。

第6章：主要介绍了常用控件，帮助学生掌握商业化应用软件的开发基本技术。

第7章：主要介绍了系统环境相关类，使学生理解目录管理方式，学会对文件进行读写操作。

第8章：主要介绍了图形图像与多媒体编程的基础知识，使学生了解.NET的图形类库，掌握基本的图形绘制方法。

第9章：主要介绍了ADO.NET数据访问方法，使学生熟练掌握数据库编程技术。

第10章：主要介绍了水晶报表的基本知识，并详细讲解了拉模式水晶报表和推模式水晶报表的设计步骤，使学生熟练掌握报表设计技术。

第11章：主要介绍了Web页面的设计基础、Web应用程序中的常用对象及ASP.NET中页面中的数据传递方式，使学生熟悉HTML、CSS和JavaScript的使用方法，熟练掌握Web应用开发技术。

第12章：主要介绍了ASP.NET Web服务器控件，使学生掌握控件属性的设置方法、事件的订阅及控件的使用。

第13章：通过一个具体的案例来介绍了Web应用程序开发，使学生掌握Web应用程序开发过程，深入理解三层架构的概念及三层架构的搭建过程，熟练掌握Web应用程序的前台和后台功能模块设计。

本书由陈锐（国家高级程序员）、张蕾（西北大学教授）、李绍华（大连外国语学院）任主编，郑耀杰（焦作市教育局信息中心）、祁文献（温县教育局）、夏敏捷（中原工学院）任副主编，王悦（西安文理学院）、李爱菊（北京联合大学）、扶晓（空军航空大学）、张西广（中原工学院）、徐飞（中原工学院）、吴向阳（温县教育局）、卢香清（南阳师范学院）参编。全书由陈锐负责统稿。

在本书的出版过程中，还要感谢张红军（安阳师范学院人文管理学院）、李学国（重庆科创职业学院）、李红（北京信息职业技术学院）、邱保志（郑州大学）、冯志祥（聊城职业技术学院）、张新彩（中原工学院信息商务学院）、张娓娓（西安交通大学城市学院）、安强强（榆林学院）、燕菊维（解放军信息工程大学）等老师提出的宝贵意见。

相关的教学课件可以从出版社网站（<http://press.bjtu.edu.cn>）和作者博客 <http://blog.csdn.net/crcr> 下载，也可以发邮件至 nwuchenrui@126.com 索取。

因时间仓促，书中不足之处和错误在所难免，恳请广大读者不吝赐教。

编者
2012年1月

目 录

第 1 章 C#语言概述	1
1.1 .NET 框架概述	1
1.1.1 公共语言运行时	2
1.1.2 基础类库	3
1.1.3 .NET 框架编程模型	3
1.1.4 .NET 程序的编译与运行	3
1.1.5 C#与 .NET 的关系	5
1.2 Visual Studio 2008. NET 集成开发环境	6
1.2.1 Visual Studio 2008. NET 的新特性	7
1.2.2 Visual Studio 2008. NET 简介	8
1.2.3 Visual Studio 2008. NET 中的其他窗口	10
1.2.4 MSDN 简介	11
1.3 Visual C#. NET 三种应用程序结构	13
1.3.1 用 Visual C#. NET 编写控制台应用程序	13
1.3.2 用 Visual C#. NET 编写 Windows 应用程序	15
1.3.3 用 Visual C#. NET 编写 ASP. NET Web 应用程序	17
1.4 C#应用程序的一般结构	18
1.4.1 命名空间 NameSpace	18
1.4.2 程序结构	20
1.5 断点设置与程序调试	22
1.5.1 如何开始调试	22
1.5.2 设置断点	23
1.5.3 定位错误	23
1.5.4 调试过程	23
1.5.5 显示调试信息	23
1.6 代码编写命名建议	24
习题	24
第 2 章 C#编程基础	25
2.1 数据类型	25
2.2 不同数据类型之间的转换	29
2.2.1 显式转换与隐式转换	29
2.2.2 装箱和拆箱	31
2.3 常量和变量	32

2.3.1 常量	32
2.3.2 变量	32
2.3.3 变量的作用范围（作用域）	33
2.4 运算符与表达式	34
2.4.1 运算符	34
2.4.2 运算符优先级	38
2.4.3 表达式	39
2.5 控制台应用程序与格式化输出	39
2.5.1 控制台输出	39
2.5.2 控制台输入	41
2.5.3 字符串的格式化输出	41
2.6 C#流程控制语句	41
2.6.1 选择语句	42
2.6.2 循环语句	46
2.6.3 跳转语句	50
2.6.4 异常处理语句	52
习题	56
第3章 常用数据类型	57
3.1 数组	57
3.1.1 数组的声明与初始化	57
3.1.2 创建数组实例	59
3.1.3 一维数组	60
3.1.4 多维数组	65
3.1.5 交错数组	68
3.1.6 数组的方法和属性	69
3.1.7 数组和字符串之间的转换	71
3.2 字符串	72
3.2.1 String 类	72
3.2.2 StringBuilder 类	79
3.3 集合	81
3.3.1 ArrayList 数组列表	81
3.3.2 Stack 堆栈	85
3.3.3 Queue 队列	87
3.3.4 Hashtable 哈希表和 SortedList 排序列表	88
3.3.5 BitArray 位数组	90
3.4 日期与时间处理	90
3.5 数学运算	92
3.6 随机数	93
3.6.1 产生随机数的类 System. Random	93

3.6.2 随机数方法 Next 的应用	93
习题	95
第4章 面向对象的编程基础	96
4.1 类	96
4.1.1 C#类的声明和对象的创建	96
4.1.2 类的成员	98
4.1.3 类的构造函数和析构函数	99
4.1.4 静态成员和实例成员	100
4.1.5 分部类	102
4.2 结构类型	102
4.2.1 结构类型的声明	103
4.2.2 结构变量	103
4.3 方法	105
4.3.1 方法的定义与调用	105
4.3.2 静态方法和非静态方法	110
4.3.3 方法的重载	111
4.4 属性与索引器	112
4.4.1 属性	112
4.4.2 索引器	114
习题	118
第5章 面向对象的高级编程	119
5.1 类的继承	119
5.1.1 继承	120
5.1.2 抽象类和密封类	127
5.2 多态	130
5.2.1 虚方法	130
5.2.2 实现多态性	131
5.3 接口	133
5.3.1 定义接口	133
5.3.2 实现接口	134
5.3.3 显式接口成员实现	135
5.4 委托与事件	136
5.4.1 委托	136
5.4.2 事件	140
5.5 反射	143
5.5.1 System.Reflection 命名空间	144
5.5.2 如何使用反射获取类型	144
5.5.3 获取程序集元数据	146
5.6 序列化与反序列化	147

5.6.1 二进制序列化与反序列化	147
5.6.2 XML 序列化与反序列化	150
5.7 .NET 泛型编程	152
5.7.1 为什么要使用泛型	152
5.7.2 定义泛型方法	153
5.7.3 定义泛型类	155
5.7.4 使用泛型集合类	157
习题	160
第6章 窗体、控件和组件	162
6.1 窗体与对话框	162
6.1.1 窗体应用程序的创建、启动和停止	162
6.1.2 窗体的常用属性和方法	163
6.1.3 窗体的常用事件	166
6.1.4 单文档窗体和多文档窗体	167
6.1.5 对话框	168
6.2 控件共有的基本操作	171
6.2.1 控件常用属性和事件	171
6.2.2 控件的常用鼠标与键盘事件	172
6.3 容器类控件和常用组件	173
6.3.1 分组控件 (Panel、GroupBox)	173
6.3.2 工具提示组件 (ToolTip)	174
6.3.3 错误提示组件 (ErrorProvider)	175
6.3.4 定时组件 (Timer)	176
6.4 文本操作类控件	177
6.4.1 标签控件 (Label、LinkLabel)	177
6.4.2 文本控件 (TextBox、RichTextBox、MaskedTextBox)	178
6.5 选择操作类控件	183
6.5.1 列表控件 (ListBox、ComboBox)	183
6.5.2 复选控件 (CheckBox、CheckedListBox)	187
6.5.3 单选控件 (RadioButton)	188
6.5.4 日期时间选择控件 (DateTimePicker)	188
6.6 图像操作类控件	190
6.6.1 图像与动画控件 (PictureBox)	190
6.6.2 图像列表组件 (ImageList)	192
6.7 菜单、工具栏与状态栏	193
6.7.1 菜单控件 (MenuStrip)	193
6.7.2 快捷菜单控件 (ContextMenuStrip)	195
6.7.3 工具栏控件 (ToolStrip)	196
6.7.4 状态栏控件 (StatusStrip)	196

6.8 视图操作类控件	196
6.8.1 列表视图控件 (ListView)	196
6.8.2 树形视图控件 (TreeView)	200
6.9 其他常用控件	204
6.9.1 选项卡控件 (TabControl)	204
6.9.2 面板复合控件 (SplitContainer)	205
6.9.3 任务栏图标组件 (NotifyIcon)	206
习题	207
第7章 目录与文件管理	210
7.1 系统环境相关类	210
7.1.1 Environment 类	210
7.1.2 System.IO 命名空间	214
7.1.3 DriveInfo 类	214
7.2 目录管理	215
7.2.1 Directory 类与 DirectoryInfo 类	215
7.2.2 Path 类	216
7.2.3 目录的创建、删除和移动	217
7.2.4 FolderBrowserDialog 对话框	218
7.3 文件管理	219
7.3.1 File 类与 FileInfo 类	219
7.3.2 文件的复制、删除与移动	220
7.3.3 文件属性	222
7.3.4 OpenFileDialog 与 SaveFileDialog 对话框	222
7.4 文件的读写	225
7.4.1 文件编码	225
7.4.2 文本文件的读写	225
7.4.3 二进制文件的读写	234
习题	237
第8章 图形图像和多媒体编程	239
8.1 图形图像绘制基础知识	239
8.1.1 GDI + 概述	239
8.1.2 Graphics 类	240
8.1.3 颜色	241
8.1.4 坐标	243
8.1.5 Paint 事件	243
8.2 绘制基本图形	243
8.2.1 创建画笔	243
8.2.2 绘制直线	245
8.2.3 绘制矩形	246

8.2.4 绘制多边形	246
8.2.5 绘制曲线	247
8.2.6 绘制椭圆和弧线	248
8.3 创建画刷填充图形	249
8.4 图像处理	252
8.4.1 显示图像	252
8.4.2 保存图像	253
8.4.3 图像的平移、旋转和缩放	255
8.4.4 彩色图像变换灰度图像	256
8.5 文字处理	258
8.5.1 创建字体	258
8.5.2 绘制文本	259
8.6 应用案例	260
8.6.1 可擦写图形轮廓的实现	260
8.6.2 单机版五子棋游戏	262
习题	267
第9章 ADO. NET 和数据操作	269
9.1 ADO. NET 简介	269
9.1.1 ADO. NET 设计目标	269
9.1.2 ADO. NET 数据访问模型	270
9.2 ADO. NET 数据库访问对象	272
9.2.1 SqlConnection 对象	272
9.2.2 SqlCommand 对象	274
9.2.3 DataTable 对象	282
9.2.4 DataSet 对象	284
9.2.5 SqlDataAdapter 对象	288
9.2.6 SqlDataReader 对象	291
9.2.7 SqlParameter 对象	293
9.3 存储过程	296
9.3.1 概述	296
9.3.2 创建存储过程	298
9.3.3 调用存储过程	300
9.3.4 应用举例	301
9.4 语言集成查询 (LINQ)	303
9.4.1 LINQ 简介	304
9.4.2 查询表达式	306
9.4.3 使用 LINQ	309
9.4.4 应用举例	312
习题	315

第 10 章 水晶报表	317
10.1 水晶报表基础知识	317
10.1.1 简介	317
10.1.2 嵌入式与非嵌入式水晶报表	318
10.1.3 报表节	319
10.1.4 拉模式和推模式	320
10.2 水晶报表的设计与显示	321
10.2.1 拉模式水晶报表	321
10.2.2 推模式水晶报表	324
习题	329
第 11 章 Web 应用程序设计基础	330
11.1 页面设计基础	330
11.1.1 HTML	330
11.1.2 CSS	335
11.1.3 JavaScript	341
11.2 Web 应用程序中的常用对象	347
11.2.1 Request	347
11.2.2 Reponse	348
11.2.3 Application	349
11.2.4 Session	350
11.3 网页切换与网页间的数据传递	352
11.3.1 网页的切换	352
11.3.2 网页间的数据传递	354
习题	358
第 12 章 ASP.NET Web 服务器控件	360
12.1 标准控件	360
12.1.1 控件属性	360
12.1.2 控件事件	361
12.1.3 标签控件 (Label)	362
12.1.4 文本框控件 (TextBox)	363
12.1.5 按钮控件 (Button, LinkButton 和 ImageButton)	365
12.1.6 单选控件 (RadioButton 和 RadioButtonList)	367
12.1.7 复选框控件 (CheckBox 和 CheckBoxList)	369
12.1.8 列表控件 (DropDownList 和 ListBox)	371
12.1.9 文件上传控件 (FileUpload)	374
12.1.10 日历控件 (DropDownList 和 ListBox)	376
12.1.11 视图控件 (MultiView 和 View)	378
12.1.12 向导控件 (Wizard)	379
12.2 验证控件	381

12.2.1 必填控件 (RequiredFieldValidator)	382
12.2.2 验证范围控件 (RangeValidator)	384
12.2.3 比较验证控件 (CompareValidator)	385
12.2.4 正则表达式验证控件 (RegularExpressionValidator)	387
12.2.5 验证总结摘要控件 (ValidationSummary)	388
12.3 导航控件	391
12.3.1 站点地图 (Web. sitemap)	391
12.3.2 站点地图路径控件 (SiteMapPath)	392
12.3.3 菜单控件 (Menu)	393
12.3.4 树视图控件 (TreeView)	393
12.4 数据操作控件	394
12.4.1 SQL 数据源控件 (SqlDataSource)	394
12.4.2 网格视图控件 (GridView)	399
12.4.3 数据列表控件 (DataList)	410
习题	414
第13章 Web 应用程序开发实例	416
13.1 系统分析与总体规划	416
13.1.1 需求分析	416
13.1.2 总体规划	417
13.1.3 功能模块	417
13.2 系统框架设计	418
13.2.1 三层架构	418
13.2.2 项目架构搭建	421
13.2.3 数据库设计	425
13.2.4 数据访问层实现	430
13.3 前台功能模块设计	435
13.3.1 会员用户登录页面	435
13.3.2 注册页面	437
13.3.3 购物流程	439
13.3.4 发布商品	444
13.3.5 商品搜索	447
13.4 后台功能模块设计	448
13.4.1 管理员用户登录页面	448
13.4.2 管理商品	450
习题	454
参考文献	456

第 1 章

C#语言概述

教学提示:

本章主要介绍 C#语言的基本理论知识, 并对 C#、C++、Java 三种语言进行对比, 还介绍了 .NET 框架的概念以及开发 .NET 应用程序的运行环境 Visual Studio 2008, 最后介绍了 C# 三种应用程序结构及 C#强大的程序调试功能。

教学目标:

了解 C#语言与其他语言的异同, 为什么要学习 C#语言? 初步了解 C#语言程序结构, 并通过上机操作, 初步掌握程序调试方法。

1.1 .NET 框架概述

.NET 是微软开发的一种面向网络、支持各种用户终端的开发平台。Visual Studio 2008.NET 环境是 Microsoft 为实现其 .NET 技术的开发工具, 在构建基于 Windows 平台的各种程序时, 它为程序员提供了非常出色的开发环境。利用 Visual Studio 2008.NET, 用户可以非常轻松地创建具有自动伸缩能力的可靠的应用程序和组件。

Visual Studio 2008.NET 集成开发环境包含 Visual Basic.NET、Visual C++ .NET、Visual C# .NET 三种编程语言, 并允许它们之间共享工具, 这有助于创建混合语言解决方案。

Visual C#.NET 是微软公司针对 .NET 平台推出来的一门新语言, 作为 .NET 平台的第一语言, 也是微软公司推出的下一代主流程程序开发语言。Visual C#.NET (简称 C#语言) 几乎集中了所有关于软件开发和软件工程研究的最新成果, 如面向对象、类型安全、组件技术、自动内存管理、跨平台异常处理、版本控制、代码安全管理等。它在设计、开发程序界面的时候和以前的某些程序开发语言有所不同, 它既有 Visual Basic 快速开发的特点, 又不乏 C++语言强大的功能, 所以 C#很可能成为将来最主要的软件开发语言。

Visual C#.NET 提出了很多新的功能、概念和观点, 掌握 Visual C#.NET 不仅要掌握语法, 还需要理解并运用这些新的功能、概念和观点, 当然也需要掌握 .NET 框架, 理解 CLR (Common Language Runtime, 公共语言运行时)。

.NET 框架 (Framework) 是一组用于帮助开发应用程序的类库集, Visual Studio.NET 开发平台需要此类库集的支持。 .NET 框架体系结构如图 1-1 所示。

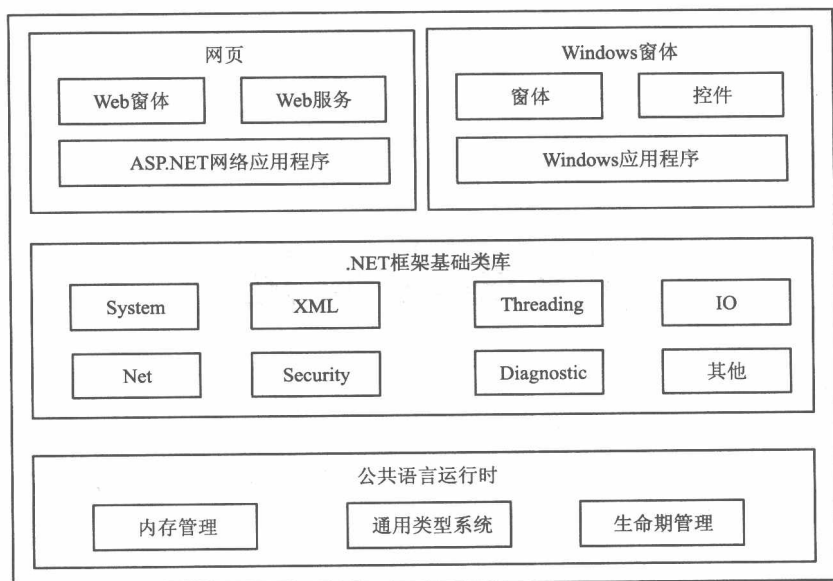


图 1-1 .NET 框架的整体结构

1.1.1 公共语言运行时

1. 公共语言运行时 (CLR)

CLR 是 .NET Framework 的基础，提供诸如内存管理、线程管理和远程处理等核心服务，强制实施严格的类型安全，以确保代码的安全性和可靠性。事实上，代码管理的概念是运行时的基本原则。以运行时为目标的代码称为托管代码，而不以运行时为目标的代码称为非托管代码。

2. 即时 (Just-In-Time) 编译

.NET 编译器，可以直接将源程序编译为 .EXE 或者 .DLL 文件，但是编译出来的程序代码并不是 CPU 能直接执行的机器代码，而是一种中间语言 (IL: Intermediate Language) 代码，在 IL 代码被调用执行时，CLR 的 Class Loader 会将需要的 IL 代码装入内存，然后再通过即时 (Just-In-Time) 编译方式将其临时编译成 CPU 可直接执行的机器代码 (二进制文件)。我们看到的程序运行，其实是经过 JIT 编译后的二进制文件在执行。这时可能有很多朋友会问，这种二次编译是否多此一举，具体实现中是否很麻烦，是否会影响程序的运行速度？二次编译的确是影响了程序运行的速度，但它却为实现跨平台带来了可能。其实这种编译过程犹如 Java 中的 JVM (Java 虚拟机)。正是 JVM 才使得 Java 能够开发出跨平台的应用程序。二次编译是 CLR 在 .Net 框架下自动实现 IL 文件到二进制文件的转变，它可自动完成，并不需要人员的参与，所以它并不会给程序执行带来麻烦。

3. 垃圾回收机制

在应用程序开发过程中，内存管理曾经是一件令人痛苦的事情。内存管理并不科学，应用程序将不断消耗系统资源并最终导致操作系统崩溃。.NET 通过垃圾回收器来管理程序进程中分配的内存，这样，开发人员就不用进行额外的劳动来保证将所有分配的内存都正确地释

放给系统，也避免了内存泄漏。

1.1.2 基础类库

类库是一个由 .NET Framework SDK (Software Development Kit, 软件开发工具包) 中包含的类、接口和值类型组成的库。该库提供对系统功能的访问，是建立 .NET Framework 应用程序、组件和控件的基础。

类库 (Class Library) 提供了 Internet 和企业级开发所需要的各种功能，完全支持 Web 标准及其应用，而且使用简单，扩充方便。类库被分配到不同的命名空间下，System 命名空间是 .NET Framework 中基本类型的根命名空间。

1.1.3 .NET 框架编程模型

.NET 框架编程模型和传统编程模型有所不同，如图 1-2 所示。

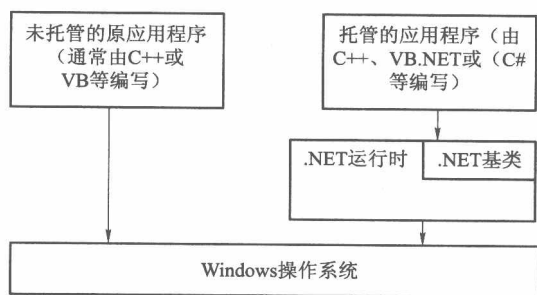


图 1-2 .NET 框架编程模型

传统的编程模型是上层应用直接依附在操作系统之上。例如，我们在 Windows 环境中编程，程序运行于 Windows 之上。但在 .NET 平台中，.NET 框架位于操作系统与上层应用之间，上层应用创建于 .NET 框架之上，而不像过去一样直接运行在操作系统之上。正是在操作系统和应用程序之间有了 .NET 框架，才使得应用程序的平台无关性成为可能。

.NET 框架是 Microsoft 为开发应用程序创建的一个富有革命性的新环境。这句话最有趣的地方是它的含糊不清，但这是有原因的。注意这句话没有说“在 Windows 操作系统上开发应用程序”。尽管 .NET Framework 发布的第一个版本是运行在 Windows 操作系统上，但是以后将推出可运行在其他操作系统上的版本，这些操作系统包括 FreeBSD、Linux 和 Macintosh，甚至个人数字助手类设备。这就是说 .NET 具有平台独立性，是可移植的，这的确是一个很大的突破。

要进一步理解 .NET 框架编程模型，还需要了解两个新的概念：MSIL 和 JIT。

1.1.4 .NET 程序的编译与运行

1. MSIL 和 JIT

在编译使用 .NET 框架创建的代码时，不是立即创建操作系统特定的本机代码，而是

把代码编译为微软中间语言（Microsoft Intermediate Language, MSIL）代码，这些 MSIL 代码不专用于任何一种操作系统，也不专用于任何一种语言，有些类似于 JAVA 的字节码。

因为代码在编译阶段没有直接编译成本机代码，所以在执行应用程序时，必须完成更多的工作，这就是 Just-In-Time (JIT) 编译器的任务。

JIT 把 MSIL 编译为专用于某种操作系统和目标机器结构的本机代码，只有这样，操作系统才能执行应用程序。这里编译器的名称 Just-In-Time，反映了 MSIL 仅在需要时才编译的特性。

过去，常常需要把代码编译为几个应用程序，每个应用程序用于特定的操作系统和 CPU 结构，这通常是一种优化形式（例如，为了让代码在 AMD 芯片上运行将更快），但更多时候是必需的（例如分别运行在 Windows 和 Linux 操作系统上）。现在就不必要了，顾名思义，JIT 编译器使用 MSIL 代码，而 MSIL 代码是独立于机器、操作系统和 CPU 的。目前有几种 JIT 编译器，每种编译器都用于不同的结构，我们总能找到一个合适的编译器创建所需的本机代码。这样，用户需要做的工作就比较少了，实际上，用户不必考虑与系统相关的细节，只需要把注意力放在代码的功能上就足够了。

2. 程序集

在编译应用程序时，创建的 MSIL 代码存储在一个程序集中，程序集包括可执行的应用程序文件（这些文件可以直接在 Windows 上运行，不需要其他程序，其扩展名为.exe）和其他应用程序使用的库（其扩展名是.dll）。

除了包含 MSIL 外，程序集还包含元数据（即程序集中包含的数据信息）和可选的资源（MSIL 使用的其他数据，例如声音和图片文件）。元数据允许程序集是完全自我描述的，不需要其他信息就可以使用程序集。也就是说，我们不再需要把应用程序所需要的数据添加到系统注册表中，因此，部署应用程序就非常简单了，只需把文件复制到远程计算机中的目录下即可。

当然，不必把运行应用程序所需要的所有信息都安装到一个地方。我们可以编写一些程序集，执行多个应用程序所要求的任务。此时，通常把这些可重用的程序集放在所有应用程序都可以访问的地方。在 .NET 框架中，这个地方是“全局程序集高速缓冲存储器”（Global Assembly Cache），有相应的工具可以帮助把程序集放在高速缓冲存储器中。

3. 托管代码

在把代码编译为 MSIL，再用 JIT 编译器把它编译为本机代码后，CLR 的任务还没有完全完成。用 .NET 框架编写的代码在执行时是托管的，即 CLR 管理着应用程序，其方式是管理内存、处理安全性，以及允许进行跨语言调试等。相反，不在 CLR 控制之下运行的应用程序是非托管的，某些语言如 C++ 可以用于编写这类应用程序，例如，访问操作系统的低级功能。使用 C# 主要编写在托管环境下运行的代码，它们使用 CLR 的托管功能，让 .NET 自己与操作系统进行交互，当然也可以编写在非托管环境下运行的代码，但需要特别标注。

图 1-3 所示为传统的代码编译与运行过程，图 1-4 所示为 .NET 中代码的编译和运行过程。