

·自学经典·

C#

程序设计

《 自学经典 》

杨光 刘志勇 编著

- ① 内容细致，知识全面，适合初、中级读者学习使用
- ② 案例丰富，所有技术要点均采用示例程序的方式讲解
- ③ 实例代码中注释详细，方便读者理解代码的具体含义
- ④ 由浅入深，循序渐进，强调理论和实践的结合

清华大学出版社



自学经典

C#程序设计自学经典

杨 光 刘志勇 编著

清华大学出版社
北 京

内 容 简 介

本书从 C# 基础开始, 主要介绍了流程控制语句的应用, 数组、集合、结构及枚举的应用, 面向对象编程的基本概念及应用, 索引器、委托、事件和 Lambda 表达式的应用, LINQ 的基础知识, 调试与异常处理, WinForm 应用程序开发基础, WinForms 控件应用, 文件及数据流技术, WPF 编程基础, ADO.NET 操作数据库, 网络编程技术, XML 编程技术, 注册表技术, 线程的基础知识, Windows 应用程序的打包及部署以及 Windows 安全性等内容。

本书结构清晰合理, 案例教学通俗易懂。不仅可以作为大、中专院校以及培训班相关专业的教材, 对于编程爱好者来讲, 同样是一本难得的入门图书。

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#程序设计自学经典 / 杨光, 刘志勇编著. —北京: 清华大学出版社, 2016
(自学经典)

ISBN 978-7-302-42296-9

I. ①C… II. ①杨…②刘… III. ①C 语言—程序设计 IV. ①TP312

中国版本图书馆 CIP 数据核字 (2015) 第 287006 号

责任编辑: 袁金敏 薛 阳

封面设计: 刘新新

责任校对: 徐俊伟

责任印制: 何 芊

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: 27.5

字 数: 673 千字

版 次: 2016 年 3 月第 1 版

印 次: 2016 年 3 月第 1 次印刷

印 数: 1~3000

定 价: 69.00 元

产品编号: 063954-01

前 言

C#是微软公司发布的一种面向对象的、运行于.NET Framework 之上的高级程序设计语言，是微软公司.NET Windows 网络框架的主角。C#是由 C 和 C++衍生出来的面向对象的编程语言，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，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，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。为了广大读者能够更好更快地学习并掌握这门语言，作者精心策划并编写了本书。

本书具有以下特色。

(1) 循序渐进，由浅入深。结构安排合理、内容全面系统。既有最基本的概念，又有实际操作，难度适中，使读者在阅读过程中很顺畅、自然地了解其基本知识。

(2) 内容充实、层次清晰。对于 C#语言的各个知识点，本书对其做了合理的分类与规划，使读者能够更清晰地掌握各个知识点，并且每个知识点都有相应的案例应用。

(3) 语言通俗易懂、简洁明了，全书中没有晦涩的字句。不但适合课堂教学，也适合读者自学使用。

本书共分为 20 章。

第 1 章介绍.NET 平台和 C#语言以及 C#与.NET 的关系。并对.NET 框架的体系结构和 Visual Studio 2012 进行简介。

第 2 章讲解 C# 的基础知识，其中包括常量、变量、数据类型、运算符和表达式以及字符与字符串的处理等。

第 3 章讲解流程控制语句的应用，其中包括选择语句、迭代语句、跳转语句等。

第 4 章讲解数组、集合、结构及枚举的应用等。

第 5 章讲解面向对象编程的基本概念及应用，其中包括类的基本概念及其应用、继承和多态性、接口、抽象类与抽象方法等内容。

第 6 章讲解索引器、委托、事件和 Lambda 表达式的应用。

第 7 章讲解 LINQ 应用。

第 8 章讲解调试和异常处理的相关知识。

第 9 章讲解 WinForm 应用程序开发基础，其中包括 Windows 窗体应用程序的简介以及开发的一般流程、多文档界面等内容。

第 10 章讲解 WinForms 基本控件，其中包括标签控件、按钮控件、文本框控件、列表框控件等。

第 11 章讲解 WinForms 的高级控件的用法。

第 12 章讲解文件及数据流技术的应用。包括文件与文件夹的读写、复制、删除、移动等操作。

第 13 章讲解 WPF 编程基础。

第 14 章讲解 ADO.NET 操作数据库的相关知识。

第 15 章讲解网络编程技术。其中包括 HTTP 网络编程以及套接字网络编程。

第 16 章讲解 XML 编程技术。

第 17 章讲解注册表的相关知识。

第 18 章讲解线程的基础知识，如线程调度、线程同步等。

第 19 章讲解 Windows 应用程序的打包及部署。

第 20 章讲解 Windows 安全性的相关知识。

全书基础知识介绍清晰，理论联系实际，具有很强的操作性。实例介绍知识面广，不但复习了前面所学的内容，而且还增加了一定的创作技巧，从而保证读者能够更好地掌握 C# 程序设计语言。因此，本书不仅可以作为大、中专院校以及培训班相关专业的教材，还可作为程序设计人员和编程爱好者的参考用书。

本书由杨光、刘志勇编著，作者均有多年 C# 程序开发实战经验，在结构安排上更加能够考虑到初学者的需求。另外，参与本书编写的还有张丽、曹培培、胡文华、尚峰、蒋燕燕、杨诚、张悦、李凤云、薛峰、张石磊、孙蕊、王雪丽、张旭、伏银恋、张班班等，由于编者水平所限，加之时间仓促，书中难免有疏漏和不足之处，恳请专家和广大读者指正。

目 录

第 1 章 从零认识 C#	1
1.1 C#简介	1
1.2 .NET 概述	2
1.3 C#与.NET 的关系	2
1.4 .NET Framework 的体系结构	2
1.4.1 公共语言运行库	3
1.4.2 .NET Framework 类库	4
1.5 Visual Studio 2012 简介	4
1.5.1 Visual Studio 2012 新功能	5
1.5.2 安装 Visual Studio 2012	6
1.5.3 卸载 Visual Studio 2012	8
1.5.4 启动 Visual Studio 2012	10
1.6 第一个 C#程序	11
1.6.1 编写第一个 C#程序	11
1.6.2 编译和运行 C#控制台应用程序	12
1.7 应用程序结构	13
1.7.1 控制台应用程序文件夹结构	13
1.7.2 C#程序结构	13
小结	19
第 2 章 C#基础知识	20
2.1 变量与常量	20
2.1.1 C#中的变量	20
2.1.2 C#中的常量	21
2.1.3 变量与常量的初始化	23
2.2 数据类型的分类	24
2.3 运算符和表达式	25
2.3.1 运算符的分类	25
2.3.2 运算符的优先级	30
2.4 字符与字符串的处理	30
2.4.1 char 的使用	30
2.4.2 字符串类 String 的使用	35
2.4.3 可变字符串类 StringBuilder 的使用	52
小结	54
第 3 章 流程控制语句的应用	55
3.1 选择语句的应用	55

3.1.1 简单的 if 条件语句	55
3.1.2 if...else...条件语句	55
3.1.3 if...else if...else 多分支语句	56
3.1.4 嵌套 if 语句	57
3.1.5 switch 多分支语句	58
3.1.6 三元运算符	59
3.2 迭代语句的应用	60
3.2.1 for 循环语句	60
3.2.2 while 循环语句	62
3.2.3 do...while 循环语句	63
3.2.4 foreach 循环语句	64
3.2.5 for、foreach、while 和 do...while 的区别	66
3.2.6 双重循环	66
3.3 跳转语句的应用	67
3.3.1 break 跳转语句	68
3.3.2 continue 跳转语句	69
3.3.3 return 跳转语句	70
3.3.4 goto 语句	72
3.3.5 各跳转语句的区别	73
小结	73
第 4 章 数组与集合、结构与枚举的应用	74
4.1 数组概述	74
4.2 一维数组的声明和使用	74
4.3 二维数组的声明和使用	76
4.4 ArrayList 类	77
4.4.1 ArrayList 类的声明与初始化	77
4.4.2 ArrayList 的使用	77
4.5 Hashtable	86
4.5.1 Hashtable 的属性及其方法	86
4.5.2 Hashtable 元素的添加	87
4.5.3 Hashtable 元素的删除	87
4.5.4 Hashtable 元素的遍历	88
4.5.5 Hashtable 元素的查找	88
4.6 枚举	90
4.6.1 枚举的声明	90
4.6.2 枚举类型与基本类型的转换	91
4.7 结构类型	92
小结	93
第 5 章 面向对象编程的基本概念及应用	94
5.1 类	94
5.1.1 类的概述	94
5.1.2 类的面向对象的概述	94

5.1.3 类的声明及其类成员	95
5.1.4 构造函数和析构函数	98
5.1.5 this 关键字	100
5.1.6 属性	101
5.2 继承	103
5.2.1 继承简述	103
5.2.2 抽象类及类成员	105
5.3 接口	105
5.3.1 接口的介绍及声明	106
5.3.2 实现接口	106
5.4 多态	108
5.5 抽象类与抽象方法的应用	110
5.5.1 抽象类的声明	110
5.5.2 抽象方法的声明	110
5.5.3 如何使用抽象类与抽象方法	111
5.6 密封类与密封方法	113
小结	114
第 6 章 索引器、委托、事件和 Lambda 表达式的应用	115
6.1 索引器	115
6.1.1 索引器的概述及声明	115
6.1.2 索引器的重载	117
6.2 委托	119
6.2.1 委托的基本用法	119
6.2.2 方法与委托相关联	121
6.3 事件	122
6.3.1 事件处理程序	123
6.3.2 事件的应用	124
6.4 Lambda 表达式	125
6.4.1 匿名方法的简介	126
6.4.2 Lambda 表达式简介	126
6.4.3 表达式 Lambda 的应用	126
6.4.4 语句 Lambda 的应用	127
6.4.5 Lambda 表达式中的变量范围	127
小结	128
第 7 章 LINQ 应用	129
7.1 LINQ 基础知识	129
7.1.1 简单的查询	129
7.1.2 函数的支持	131
7.1.3 使用混合的查询和函数语法	132
7.2 LINQ 子句	133
7.2.1 where 子句的应用	134
7.2.2 orderby 子句的应用	135

7.2.3	select 子句的应用	137
7.2.4	多个 from 子句的应用	139
7.2.5	group 子句的应用	140
7.2.6	into 子句的应用	142
7.2.7	let 子句的应用	144
7.2.8	join 子句的应用	146
小结		148
第 8 章	调试和异常处理	149
8.1	程序调试概述	149
8.2	程序错误与程序调试	149
8.2.1	程序错误	149
8.2.2	程序调试	150
8.3	异常类与异常处理	155
8.3.1	异常类	155
8.3.2	异常处理	156
小结		165
第 9 章	WinForm 应用程序开发基础	166
9.1	Windows 应用程序的开发界面	166
9.1.1	创建 Windows 程序	166
9.1.2	解决方案资源管理器	167
9.1.3	窗体设计器和代码编辑器	169
9.1.4	工具箱	169
9.1.5	工具栏	169
9.2	多文档界面	170
9.2.1	多文档界面设置及窗体属性	170
9.2.2	窗体传值技术	172
9.3	开发一个简单的 Windows 应用程序	181
9.3.1	菜单栏	182
9.3.2	工具栏	184
9.3.3	状态栏	186
小结		187
第 10 章	WinForms 基本控件	188
10.1	Control 类	188
10.1.1	Control 类的属性	188
10.1.2	Control 类的事件	189
10.2	标签控件 (Label 控件)	192
10.3	按钮控件 (Button 控件)	193
10.3.1	Button 控件的常用属性	193
10.3.2	Button 控件的应用	194
10.4	文本框控件 (TextBox 控件)	198
10.4.1	TextBox 控件的常用属性	198

10.4.2	TextBox 控件的常用事件	199
10.4.3	TextBox 控件的简单应用	200
10.5	ListBox 控件和 CheckedListBox 控件	201
10.5.1	ListBox 控件的属性	201
10.5.2	ListBox 控件的方法	202
10.5.3	ListBox 控件的事件	203
10.5.4	ListBox 控件的常见用法	203
10.6	消息对话框	206
	小结	208
第 11 章	WinForms 高级控件	209
11.1	单选按钮 (RadioButton)	209
11.1.1	RadioButton 类的常见属性和事件	209
11.1.2	RadioButton 的用法	210
11.2	图片框控件 (PictureBox)	211
11.2.1	PictureBox 类的常见属性和事件	212
11.2.2	PictureBox 控件实例	212
11.3	选项卡控件 (TabControl)	215
11.3.1	TabControl 类的常见属性和事件	215
11.3.2	TabControl 控件实例	215
11.4	进度条控件 (ProgressBar)	217
11.4.1	ProgressBar 类的常见属性	218
11.4.2	ProgressBar 控件实例	218
11.5	ImageList 控件	220
11.5.1	ImageList 类的常见属性	220
11.5.2	ImageList 控件实例	220
11.6	ToolStrip 控件	222
11.6.1	ToolStrip 类的常见属性	222
11.6.2	ToolStrip 相关的伴随类	223
11.6.3	ToolStrip 中的项	223
11.6.4	创建工具栏	224
11.7	ListView 控件	225
11.7.1	ListView 类的常见属性、事件和方法	226
11.7.2	ListView 控件实例	228
11.8	TreeView 控件	231
11.8.1	TreeView 类的属性	231
11.8.2	TreeNode 类的属性	232
11.8.3	TreeView 控件实例	233
11.9	MonthCalendar 控件	235
11.9.1	MonthCalendar 类的属性	235
11.9.2	MonthCalendar 控件实例	236
11.10	DateTimePicker 控件	237
11.10.1	DateTimePicker 类的属性	238
11.10.2	DateTimePicker 控件实例	238

小结	240
第 12 章 文件及数据流技术	241
12.1 System.IO 命名空间	241
12.1.1 System.IO 命名空间中包含的类	241
12.1.2 File 类的常用方法	242
12.1.3 FileInfo 类的方法	244
12.1.4 Directory 类的方法	245
12.1.5 File 类的使用	247
12.1.6 Directory 类的使用	250
12.2 FileStream 文件流类	252
12.2.1 FileMode 枚举对象的成员	252
12.2.2 FileAccess 枚举对象的成员	253
12.2.3 FileStream 类的常用属性	253
12.2.4 FileStream 类的常用方法	253
12.3 StreamReader 类和 StreamWriter 类	254
12.3.1 StreamReader 类	254
12.3.2 StreamWriter 类	255
12.3.3 StreamReader 类与 StreamWriter 类的使用	257
12.4 BinaryReader 类和 BinaryWriter 类	259
12.4.1 BinaryReader 类	259
12.4.2 BinaryWriter 类	260
12.4.3 BinaryReader 类与 BinaryWriter 类的使用	261
小结	262
第 13 章 WPF 编程基础	263
13.1 WPF 概述	263
13.2 WPF 体系结构	264
13.3 WPF 的特点	266
13.4 XAML	267
13.4.1 XAML 简述	267
13.4.2 XAML 的优点	267
13.4.3 XAML 基本语法	268
13.4.4 Application 对象	271
13.5 WPF 布局控件简述	273
13.5.1 Canvas 控件	273
13.5.2 DockPanel 控件	276
13.5.3 Grid 控件	277
13.5.4 StackPanel 控件	278
13.5.5 WrapPanel 控件	279
小结	280

第 14 章 ADO.NET 操作数据库	281
14.1 ADO.NET 简介	281
14.1.1 ADO.NET 的作用	282
14.1.2 ADO.NET 的主要组件	282
14.2 Connection 对象	283
14.2.1 SqlConnection 类的常用属性	283
14.2.2 SqlConnection 类的常用方法	284
14.3 Command 对象	285
14.3.1 SqlCommand 类的创建	285
14.3.2 SqlCommand 类的常用属性	285
14.3.3 SqlCommand 类的常用方法	286
14.3.4 SqlCommand 类的使用	286
14.4 事务处理	289
14.4.1 事务的特性	289
14.4.2 执行事务的步骤	289
14.4.3 事务类 SqlTransaction 类的使用	289
14.5 DataReader 对象	291
14.5.1 SqlDataReader 类的属性	292
14.5.2 SqlDataReader 类的方法	292
14.5.3 SqlDataReader 类的使用	295
14.6 DataSet 对象和 DataAdapter 对象	297
14.6.1 DataSet 对象	297
14.6.2 DataAdapter 对象	301
14.6.3 DataSet 和 SqlDataAdapter 的应用	303
14.7 DataView 对象	304
14.7.1 DataView 类的属性	305
14.7.2 DataView 类的方法	305
14.7.3 DataView 类的使用	306
14.8 DataGridView 控件显示和操作数据	309
14.8.1 DataGridView 类的属性	310
14.8.2 DataGridView 控件的案例教学	311
小结	315
第 15 章 网络编程技术	316
15.1 HTTP 网络编程	316
15.1.1 System.Net 命名空间	316
15.1.2 WebClient 类	318
15.1.3 WebRequest 类和 WebResponse 类	325
15.1.4 WebBrowser 浏览器控件	329
15.2 套接字网络编程	338
15.2.1 TcpClient 类和 TcpListener 类	338
15.2.2 Socket 类	343
15.2.3 UDPClient 类	350
小结	355

第 16 章 XML 编程技术	356
16.1 XML 基础	356
16.2 XML 语法	357
16.2.1 XML 标记、元素和属性	357
16.2.2 XML 的语法规则	358
16.2.3 XML 名称命名规则	360
16.3 操作 XML 文档	361
16.3.1 XML 文档对象模型概述	361
16.3.2 XML 文档的 DOM 实现	361
16.3.3 XML 文档的应用实例	362
16.3.4 装载 XML 文档	362
16.3.5 遍历 XML 文档	363
16.3.6 查询特殊元素和节点	364
16.3.7 修改 XML 文档	365
16.3.8 Save 方法	365
16.4 综合实例	366
小结	370
第 17 章 注册表技术	371
17.1 注册表基础知识	371
17.1.1 简述注册表	371
17.1.2 展示注册表的结构	371
17.2 操作注册表	372
17.2.1 读取注册表中信息	373
17.2.2 创建和修改注册表信息	373
17.2.3 删除注册表中信息	374
17.2.4 情景应用：利用注册表设计注册软件	375
17.3 实战练习：添加“用记事本打开”快捷菜单项	377
小结	378
第 18 章 线程的基础知识	379
18.1 线程简述	379
18.1.1 单线程	379
18.1.2 多线程	380
18.1.3 线程的生命周期	381
18.2 线程调度	382
18.2.1 简述 Thread 类	382
18.2.2 创建线程	383
18.2.3 挂起与恢复线程	384
18.2.4 线程休眠	385
18.2.5 终止与阻止线程	387
18.2.6 情景应用：使用多线程制作端口扫描工具	387
18.3 线程同步	389
18.3.1 简述线程同步机制	389

18.3.2 使用 lock 关键字实现线程同步	391
18.3.3 使用 Monitor 类实现线程同步	391
18.4 综合实例	393
小结	402
第 19 章 Windows 应用程序的打包及部署	403
19.1 安装工具简介	403
19.2 创建部署项目	403
19.3 简单的打包和部署	404
19.4 自定义的打包程序	411
小结	412
第 20 章 Windows 安全性	413
20.1 Windows 应用程序的安全性概述	413
20.1.1 如何创建、卸载域	413
20.1.2 如何实现域间的通信	414
20.2 身份验证和授权	414
20.2.1 标识和 Principal	416
20.2.2 角色	417
20.2.3 声明基于角色的安全性	418
20.3 加密	418
20.3.1 签名	421
20.3.2 交换密钥和安全传输	422
20.4 资源的访问控制	423
20.5 代码访问安全性	424
20.5.1 声明式安全性	424
20.5.2 强制安全性	424
20.5.3 请求权限	425
小结	425

第1章 从零认识 C#

本章将从零开始，为学习 C#打下一个良好的基础，首先会简要介绍.NET 平台和 C# 语言，让读者对.NET 平台和 C#语言有初步的认识，了解 C#与.NET 的关系。此外还会介绍.NET Framework 的体系结构以及如何使用 Visual Studio 2012 编写和执行 C#的控制台程序。

本章主要内容：

- .NET Framework 体系结构
- 使用 Visual Studio 2012
- 编译和执行 C#控制台应用程序

1.1 C#简介

C#是 Microsoft 公司于 2000 年 7 月发布的，它是运行于.NET Framework 之上的一种简单、现代、安全、面向对象的高级程序设计语言。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。

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。最重要的是，C#使得 C++程序员可以高效地开发程序，而绝不损失 C/C++原有的强大的功能。因为这种继承关系，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#。

C#具有以下突出的特点。

- (1) 语法简洁。不允许直接操作内存，去掉了指针操作。
- (2) 彻底的面向对象设计。C#具有面向对象语言所应有的一切特性——封装、继承和多态。
- (3) 与 Web 紧密结合。C#支持绝大多数的 Web 标准，如 HTML、XML、SOAP 等。
- (4) 强大的安全机制。可以消除软件开发中的常见错误，.NET 提供的垃圾回收器能够帮助开发者有效地管理内存资源。
- (5) 兼容性。因为 C#遵循.NET 的公共语言规范(Common Language Specification, CLS)，从而保证能够与其他语言开发的组件兼容。
- (6) 灵活的版本处理技术。因为 C#语言本身内置了版本控制功能，使得开发人员可以

更容易地开发和维护。

(7) 完善的错误、异常处理机制。C#提供了完善的错误和异常处理机制，使程序在交付应用时能够更加健壮。

(8) 强大的类库支持。C#有着数量庞大、功能齐全的.NET 类库的支持，从而可以轻易地完成复杂的加密操作、网络应用操作等。使用 C#可以轻松构建功能强大、开发快捷、运用方便的应用程序。

1.2 .NET 概述

.NET 是微软的新一代技术平台——Microsoft XML Web Services 平台，XML Web Services 允许应用程序通过 Internet 进行通信和共享数据，而不管所采用的是哪种操作系统、设备或编程语言。Microsoft .NET 平台提供创建 XML Web Services，并将这些服务集成在一起，为敏捷商务构建互连互通的应用系统，这些系统是基于标准的，连通的，适应变化的，稳定的和高性能的。从技术的角度，一个.NET 应用是一个运行于.NET Framework 之上的应用程序。

1.3 C#与.NET 的关系

C#是一种相当新的编程语言，C#的重要性体现在以下两个方面。

(1) 它是专门为与 Microsoft 的.NET Framework 一起使用而设计的。(.NET Framework 是一个功能非常丰富的平台，可开发、部署和执行分布式应用程序。)

(2) 它是一种基于现代面向对象设计方法的语言，在设计它时，Microsoft 还吸取了其类似语言的经验，这些语言是近二十年来面向对象规则得到广泛应用后才开发出来的。有一个很重要的问题要弄明白：C#就其本身而言只是一种语言，尽管它是用于生成面向.NET 环境的代码，但它本身不是.NET 的一部分。.NET 支持的一些特性，C#并不支持。而 C#语言支持的另一些特性，.NET 却不支持。但是，因为 C#语言是和.NET 一起使用的，所以如果要使用 C#高效地开发应用程序，理解 Framework 就非常重要了，所以下面将介绍.NET 的核心——.NET Framework 的体系结构。

1.4 .NET Framework 的体系结构

C# 程序在.NET Framework 上运行，它是 Windows 的一个必要组件，如图 1-1 所示为.NET Framework 的体系结构。.NET Framework 包括两大组件：公共语言运行库(Common Language Runtime, CLR)和.NET Framework 类库(Framework Class Library, FCL)。



图 1-1 .NET Framework 的体系结构

1.4.1 公共语言运行库

公共语言运行库提供了异常处理、安全、调试以及任何语言的版本支持等功能。它可以使用各种程序设计语言，并提供跨语言的公共工具集，从而确保了代码之间的互用性。如图 1-2 所示为 .NET 框架的组件。

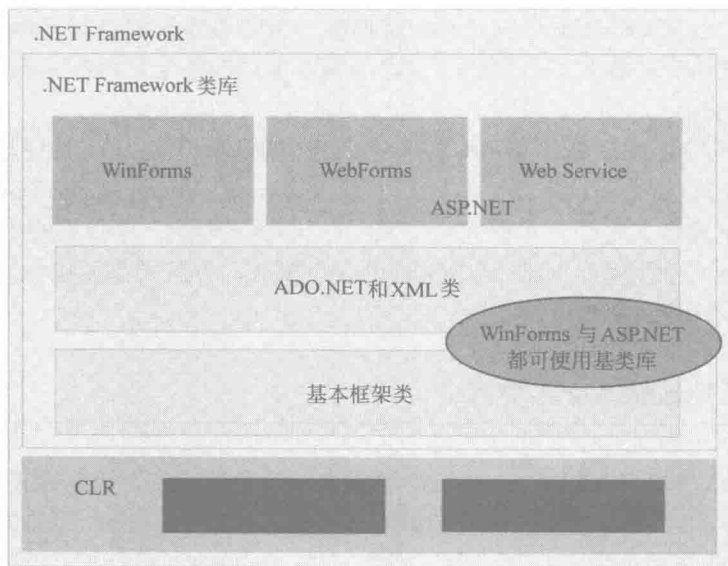


图 1-2 .NET 框架组件

CTS (Common Type System, 通用类型系统) 和 CLS (Common Language Specification, 公共语言规范) 是 CLR 的子集。

CTS 定义了 IL 中的数据类型，如 C# 中的 `int` 类型和 VB.NET 中的 `Integer` 类型都被编译成 `Int32`。

CLS 是 CLR 支持的语言功能的子集，它包括几种面向对象的编程语言的通用功能。