

- ◆ 一线程序员撰写，凝聚自己多年开发经验，系统且深入阐释C#开发涉及的**方法**和**实践**
- ◆ 由浅入深剖析C#开发过程中遇到的各个层面的问题，涉及**C#编程**语言基础、常用**.NET Framework**资源、数据库应用等

Practical Guide for C# **Methods & Practices**

程序员
典藏

C#开发实用指南

方法与amp;实践

曹化宇◎著



清华大学出版社

程序员
典藏

C#开发实用指南

方法与amp;实践

曹化字◎著

清华大学出版社
北京

内 容 简 介

本书是一线程序员凝聚自己多年开发经验的结晶之作，书中深入浅出地讲解 C# 编程语言、.NET Framework 资源、常用功能的封装、SQL Server 数据库系统以及 Windows 窗体程序与 ASP.NET 网站项目的开发。

全书内容可以分为五个部分：第一部分（第 1 ~ 9 章）主要介绍 C# 编程语言以及与数据处理的相关内容；第二部分（第 10 ~ 16 章）讨论常见的设计模式及其在 C# 和 .NET Framework 平台中的应用，并且介绍了常用的 .NET Framework 类库资源等；第三部分（第 17 ~ 20 章）讨论 SQL Server 数据库系统的应用，并讲解如何使用 ADO.NET 组件操作数据库和 Excel 文件等；第四部分（第 21 章和第 22 章）讨论了 Windows 窗体项目和 ASP.NET 网站的创建，涉及常用功能的实现以及一些技术和方法的应用特点；第五部分（第 23~25 章）通过三个完整的项目示例，让读者在实战中充分理解不同开发技术与方法的应用技巧。

本书内容安排合理，架构清晰，注重理论与实践相结合，适合作为零基础学习 C# 开发的初学者的教程，也适合作为有一定编程基础的程序员的参考手册。

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# 开发实用指南：方法与实践 / 曹化宇著. —北京：清华大学出版社，2018
（程序员典藏）
ISBN 978-7-302-49283-2

I. ① C… II. ①曹… III. ① C 语言—程序设计—指南 IV. ① TP312.8-62

中国版本图书馆 CIP 数据核字 (2018) 第 002428 号

责任编辑：秦 健

封面设计：杨玉兰

责任校对：徐俊伟

责任印制：李红英

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 刷 者：清华大学印刷厂

装 订 者：三河市铭诚印务有限公司

经 销：全国新华书店

开 本：185mm × 260mm

印 张：36

字 数：899 千字

版 次：2018 年 6 月第 1 版

印 次：2018 年 6 月第 1 次印刷

印 数：1 ~ 2500

定 价：99.00 元

产品编号：073229-01

为什么要写这本书

软件开发是一个充满乐趣和挑战的过程。对于开发人员来讲，学习一种编程语言、一套资源、一系列开发方法与理论，正是进入软件开发世界的必经之路。

易学、好用，相信这是大家对 C# 语言和 .NET Framework 平台的初步感觉。的确，C# 开发入门并不是一件难事，但是，如何掌握大量的 .NET Framework 开发资源，如何合理地应用这些资源，这些就需要深入学习和理解了。

微软的帮助文档是最完整的说明书，但是，只看这些文档并不能帮助我们有效地进行应用软件开发工作，为什么呢？因为这些文档最大的功能就是说明技术的应用，只是告诉开发者如何使用 C# 编程语言和 .NET Framework 类库的基本应用方法。但开发者的任务是如何创建一个应用软件，如何实现应用中的各项功能。例如，在开发工作中，大家可能会遇到以下一些问题：

- ❑ .NET Framework 类库中定义了一系列用于图形图像处理的资源，如 Color 结构处理颜色，Bitmap 类处理位图，Graphics 类处理图形绘制操作等，但是，如何在应用中处理图片的尺寸，如何转换图片格式，如何截取图像的一部分，如何打印出各种尺寸的图片呢？
- ❑ 在 .NET Framework 类库中，System.Data.SqlClient 命名空间中的资源可以用于操作 SQL Server 数据库，但是，如果项目中使用了不少一种数据库呢？如果项目需要更换数据库系统，难道只能完全重写代码吗？
- ❑ 在 Web 项目，如 ASP.NET 网站中，可以使用 C# 语言和 .NET Framework 资源实现很多功能，但是，如何实现验证码功能呢？用户数据如何在网页、代码库和数据库之间进行传递和操作呢？

相信大家在帮助文档中是找不到这些问题的答案的。本书的目的就是给出这一系列问题的答案。本书不但为初学者提供了 C# 语言、.NET Framework 类库，以及开发方法的基础知识，而且为 C# 开发者提供了大量软件功能的实现方法，并封装了很多实用的代码，以及完整项目的综合演示。读者可以在学习、工作中根据需要快速参考这些开发技术与方法。

本书特色

从代码到项目，从部分到整体

本书从基本的 C# 代码开始，逐渐介绍了一些常用的 .NET Framework 类库资源，并讨论了一些基本的软件开发方法。然后，通过 Windows 窗体项目和 ASP.NET 网站项目演示这一系列技术和方法的综合应用。

突出实用性，以功能实现为目标

书中介绍了大量的 .NET Framework 类库资源的基本应用，并以软件功能的实现为目标来组织。本书从基本的应用方法开始，再结合软件功能的实现，真实再现开发技术和方法在项目中的应用特点。

根据功能组织，方便参考索引

本书内容涉及从 C# 语言到 .NET Framework 类库、从代码和结构、从基本应用到功能封装、从技术应用到项目开发。从不同的角度精心组织内容，不但可以在学习中循序渐进，在实际开发中也能够快速参考相关内容。

封装大量实用代码，可直接在项目中应用

本书从实际需求出发，封装了大量的代码，大家可以从了解代码到软件的形成过程，也可以在实际开发中使用这些封装的代码，或者根据需要扩展、修改使用这些代码。当然，大家也可以参考这些内容创建自己的代码库，并合理应用开发方法。要知道，软件开发的过程本来就不是一成不变的。

完整的项目实例，更加接近实战

本书介绍了三个完整的项目，分别是截图工具、Windows 窗体版的账本管理项目，以及 Web 版的个人助理项目。通过这些项目的创建过程，读者可以更加了解从基本的技术应用到完整项目开发的过程，以及不同技术与开发方法在不同类型项目中的综合应用。

读者对象

本书面向 Windows 与 .NET Framework 平台开发者，无论是 C# 的初学者、正在使用 C# 的开发者、Windows 窗体应用开发者，或者是 ASP.NET 网站开发者，本书都能够提供从 C# 语言到 .NET Framework 资源应用的完整内容，以及从代码到软件的完整过程。更重要的是，读者可以从本书开始，迈向可能无限的软件开发世界！

如何阅读本书

本书包括了 C# 编程语言、常用 .NET Framework 类库资源、数据库的应用、Windows 窗体项目和 ASP.NET 网站开发，以及项目的综合演示等内容。

第一部分（第 1 ~ 9 章）主要讨论 C# 编程语言，以及数据处理的相关内容，如值类型与引用类型的应用特点、基本的数据运算、数组与集合、日期与时间等。对于 C# 开发的初学者，可以从第 1 章开始，逐渐学习 C# 编程语言和 .NET Framework 类库的应用。

第二部分（第 10 ~ 16 章）讨论常见的设计模式及其在 C# 和 .NET Framework 平台中的应用，同时介绍了常用的 .NET Framework 类库资源，并结合软件功能实现和开发方法进行

组织与封装。对于初次阅读本书的朋友，应该通读这些内容，然后，在学习或工作中，可以根据需要参考相应的部分，根据实际情况扩展或修改书中封装的代码库，并在自己的项目中灵活应用。

第三部分（第 17 ~ 20 章）讨论 SQL Server 数据库系统的应用，并介绍如何使用 ADO.NET 组件操作数据库和 Excel 文件，然后，通过创建自己的一系列数据组件综合演示设计模式、软件架构和数据处理等知识的综合应用。.NET Framework 平台的应用中，基于数据的应用类型是比较常见的，对于需要在项目中使用数据库的读者，可以在完成这部分内容的学习以后，进一步深入学习数据库系统和软件架构的相关知识。

第四部分（第 21 章和第 22 章）讨论 Windows 窗体项目和 ASP.NET 网站的创建，主要包括一些常用功能的实现，以及一些技术和方法的应用特点。通过这部分的学习，读者可以分别掌握 Windows 窗体项目和 ASP.NET 项目的开发特点。

第五部分（第 23 ~ 25 章）通过 3 个完整的项目示例，让读者在实战中充分理解不同开发技术与方法的应用技巧。建议对软件开发不太熟悉的读者，仔细领会这些项目中开发技术与方法的应用，能够深入理解开发技术与方法的特点，更能够在项目中灵活应用。

进一步学习建议

通过本书的学习，读者应该能够掌握 C# 编程语言和常用的 .NET Framework 类库资源，并能够实现很多软件功能。同时，还可以创建 Windows 窗体、ASP.NET 网站等项目类型，使用功能强大的开发资源灵活构建各种类型的应用软件。当然，如果感兴趣，还可以在本书的基础上深入学习更多、更有趣的开发技术和方法。

如果需要开发大型的软件系统（如 ERP 系统），对于软件架构、人机相互、大型数据库系统的深入学习都是非常有必要的。

如果是开发 Web 项目，本书中 ASP.NET 网站的相关内容主要讨论了服务器端的开发，而对于一个完整的网站来说，HTML、CSS、JavaScript 等技术，以及内容管理、信息架构、响应式页面设计等方面都是值得深入学习的。作为前端设计师，平面设计也是不得不考虑的问题。

对于应用软件来讲，会根据用户的不同而涉及众多的领域，所以，当需要开发某一领域的应用软件时，还应该虚心地向用户了解工作的本质，并与用户交流软件操作的最佳方式，然后，使用合理的技术和方法来实现。只有这样，开发出的软件才可能是良好的效率工具。

本书只介绍了 Visual Studio 系列开发工具的基本功能，即编写和调试代码，以及生成各种项目类型。在实际开发工作中，强大的开发工具是必不可少的，应该进一步深入学习 Visual Studio 的强大功能，以帮助提高工作效率。

下载提示

本书涉及的源程序请读者直接登录清华大学出版社官网（www.tup.com.cn），搜索到本书页面后按照提示进行下载。

勘误和支持

由于作者水平有限，书中难免会出现一些不太合理的地方，而读者的批评和指正，则是我们共同进步的强大动力，您可以将建议反馈给我们，同时，也欢迎大家与作者直接交流，作者的邮箱是 chydev@163.com。

致谢

感谢出版社的编辑老师耐心的交流与指导，使得本书能够顺利地与读者见面。

感谢我的家人，他们为我创造了一个温暖的家、一个安心的工作环境。特别是我的孩子们，他们总是说“爸爸在工作，我不打扰他”，这些正是我快乐生活和努力工作的力量源泉。

谨以此书献给我的家人，以及热爱软件开发的朋友们！

作者

Connect 目 录

第1章 概述..... 1

1.1 编写 C# 代码..... 1

1.1.1 第一个 C# 程序..... 1

1.1.2 语句..... 3

1.1.3 注释..... 3

1.2 命名空间..... 4

1.2.1 资源的组织..... 4

1.2.2 使用 using 语句..... 5

1.3 项目类型..... 5

1.3.1 控制台应用程序..... 6

1.3.2 Windows 窗体应用程序..... 6

1.3.3 ASP.NET 网站..... 7

第2章 数据处理 (一)..... 10

2.1 变量与常量..... 10

2.1.1 变量..... 10

2.1.2 常量..... 11

2.1.3 基本数据类型..... 11

2.1.4 sizeof 运算符..... 11

2.2 值类型与引用类型..... 12

2.3 整数..... 12

2.3.1 算术运算..... 12

2.3.2 增量与减量运算..... 13

2.3.3 位操作..... 13

2.3.4 溢出检查..... 15

2.4 浮点数与 decimal..... 16

2.4.1 类型转换..... 16

2.4.2 算术运算中的类型转换..... 17

2.4.3 处理小数位..... 18

2.5 布尔类型..... 19

2.6 字符串..... 19

2.6.1 转义字符..... 20

2.6.2 逐字符字符串..... 20

2.7 字符..... 21

2.8 枚举..... 21

2.9 结构与类..... 22

2.9.1 字段..... 22

2.9.2 属性..... 23

2.9.3 方法..... 24

第3章 流程控制..... 26

3.1 比较运算..... 26

3.2 条件语句..... 26

3.2.1 if 语句..... 26

3.2.2 ?: 运算符..... 30

3.3 switch 语句..... 30

3.4 循环语句..... 32

3.4.1 for 语句..... 32

3.4.2 foreach 语句..... 34

3.4.3 while 语句..... 34

3.4.4 do-while 语句	35	4.9.2 工厂方法	60
3.5 goto 语句与标签	35	4.9.3 静态构造函数	60
3.6 异常处理	36	4.10 运算符重载	61
3.6.1 try-catch-finally 语句	36	4.11 扩展方法	63
3.6.2 throw 语句	38	4.12 匿名类型与 var 关键字	64
3.6.3 应用中的异常处理	38	4.13 泛型类	65
第 4 章 面向对象编程	40	第 5 章 继承	67
4.1 类与对象	40	5.1 父类与子类	67
4.2 属性与字段	41	5.1.1 构造函数的继承	68
4.2.1 字段	41	5.1.2 唯一没有父类的类 (Object)	70
4.2.2 属性	41	5.2 成员的重写	70
4.2.3 自动属性	43	5.2.1 虚拟成员	70
4.2.4 只读属性	43	5.2.2 重写	71
4.2.5 只写属性	43	5.2.3 隐藏父类成员	71
4.2.6 属性的应用	44	5.3 抽象类与抽象方法	72
4.3 访问级别	44	第 6 章 接口	74
4.4 构造函数与初始化器	45	6.1 创建接口	74
4.4.1 构造函数	45	6.2 实现接口	74
4.4.2 初始化器	46	6.3 接口的继承	75
4.4.3 构造函数链	46	6.4 泛型接口	78
4.4.4 参数默认值	48	6.5 泛型约束	79
4.5 析构函数	48	6.6 using 语句与 IDisposable 接口	79
4.6 方法	49	第 7 章 数组与集合	83
4.6.1 按值或按引用传递参数	50	7.1 数组与 Array 类	83
4.6.2 输出参数	51	7.1.1 多维数组与成员数量	83
4.6.3 参数数组	52	7.1.2 成员访问与查询	84
4.6.4 重载	53	7.1.3 成员排序	86
4.6.5 参数默认值	54	7.1.4 成员反向排列	89
4.6.6 泛型方法	55	7.1.5 数组复制	89
4.7 索引器	56	7.1.6 统计方法	90
4.8 分部类与分部方法	58	7.1.7 其他常用成员	91
4.9 静态类与静态成员	59		
4.9.1 代码封装	60		

7.2 ArrayList 与 List<> 泛型类.....	91	9.1.2 字符串格式化.....	122
7.2.1 成员访问与查询.....	92	9.2 StringBuilder 类.....	123
7.2.2 添加成员.....	93	9.2.1 构造函数.....	124
7.2.3 删除成员.....	94	9.2.2 内容操作.....	124
7.2.4 成员排序.....	94	9.2.3 缓存功能.....	124
7.2.5 成员反向排列.....	95	9.3 空值 (null) 处理.....	125
7.2.6 成员复制.....	95	9.3.1 可空类型.....	126
7.3 Hashtable 与 Dictionary<> 泛型类.....	97	9.3.2 ?? 运算符.....	126
7.3.1 成员访问与查询.....	97	9.3.3 ? 运算符.....	126
7.3.2 修改成员.....	98	9.4 类型判断与转换.....	127
7.4 foreach 语句与枚举器.....	98	9.4.1 Type 类.....	127
7.5 小结.....	102	9.4.2 is 和 as 运算符.....	129
第 8 章 日期与时间.....	103	9.4.3 隐式转换和强制转换.....	129
8.1 DateTime 结构.....	103	9.4.4 装箱与拆箱.....	130
8.1.1 获取日期和时间值.....	104	9.4.5 TryParse() 方法.....	130
8.1.2 日期与时间计算.....	104	9.4.6 Convert 类.....	131
8.2 区域.....	105	9.5 封装类型转换方法.....	131
8.2.1 CultureInfo 类.....	106	9.6 散列.....	132
8.2.2 日历类.....	106	9.6.1 MD5 算法.....	133
8.3 日期与时间格式化.....	106	9.6.2 SHA1 算法.....	134
8.3.1 GetDateTimeFormats() 方法.....	107	9.7 GUID.....	134
8.3.2 ToString() 方法.....	108	9.8 对象的复制.....	135
8.4 中国农历.....	109	9.8.1 浅复制与深复制.....	135
8.5 星期与季度计算.....	113	9.8.2 实现 ICloneable 接口.....	136
8.6 节日判断.....	115	9.8.3 序列化.....	137
8.6.1 固定日期节日.....	115	第 10 章 设计模式.....	140
8.6.2 不固定日期节日.....	116	10.1 策略模式.....	140
8.6.3 给出节日信息.....	117	10.2 单件模式.....	145
第 9 章 数据处理 (二).....	119	10.3 组合模式.....	146
9.1 String 类.....	119	10.4 委托、事件与访问者模式.....	149
9.1.1 常用成员.....	119	10.4.1 委托.....	149
		10.4.2 事件与用户控件.....	151
		10.4.3 访问者模式.....	153

10.5 “三层架构”模式	153	13.2.1 矩形	175
10.5.1 用户界面层	153	13.2.2 椭圆与圆形	176
10.5.2 业务逻辑层	154	13.2.3 线条与多边形	177
10.5.3 数据访问层	154	13.2.4 封闭图形	178
10.6 MVC 模式	154	13.2.5 绘制文本	180
10.7 小结	155	13.2.6 扇形与弧线	181
第 11 章 LINQ 与 Lambda 表达式	156	13.2.7 曲线	183
11.1 LINQ 查询语句	156	13.3 旋转与翻转	185
11.1.1 基本查询	156	13.4 位图截取	186
11.1.2 集合方法	158	13.4.1 截取矩形区域	186
11.1.3 排序	159	13.4.2 截取椭圆或圆形区域	188
11.1.4 分组	159	13.5 封装 CImage 类	190
11.2 Lambda 表达式	160	13.5.1 图像的尺寸问题	190
第 12 章 路径、目录与文件	161	13.5.2 创建 CImage 类	190
12.1 路径	161	13.5.3 基本图形绘制	192
12.1.1 Path 类	161	13.5.4 绘制文本	193
12.1.2 封装常用功能	162	13.5.5 保存与打印	193
12.2 文件	164	第 14 章 获取系统与硬件信息	196
12.2.1 File 类与 FileInfo 类	164	14.1 环境变量	196
12.2.2 文件的读写	165	14.1.1 读取环境变量	196
12.3 目录	166	14.1.2 设置环境变量	197
12.4 ZipFile 类	167	14.2 CPU 信息	198
第 13 章 图形图像	169	14.3 内存信息	200
13.1 常用资源	169	14.3.1 GlobalMemoryStatusEx() 函数	200
13.1.1 Color 结构	169	14.3.2 使用 WMI 获取内存条 信息	202
13.1.2 Bitmap 类	169	14.4 驱动器信息	204
13.1.3 Graphics 类	171	14.4.1 使用 DriveInfo 类	204
13.1.4 格式刷与渐变	171	14.4.2 使用 WMI 获取硬盘信息	205
13.1.5 画笔	173	14.5 操作系统信息	207
13.2 图形绘制	175	14.5.1 获取 Windows 版本	207
		14.5.2 获取计算机与用户名称	208

第 15 章 网络	209	17.4.1 查询条件.....	237
15.1 测试网络连接.....	209	17.4.2 排序 (order by 子句).....	240
15.2 下载与上传文件.....	211	17.4.3 函数.....	241
15.2.1 下载文件.....	211	17.4.4 分组 (group by 子句).....	242
15.2.2 上传文件.....	212	17.4.5 连接 (join 子句).....	243
15.3 发送电子邮件.....	212	17.4.6 自动行号.....	245
第 16 章 正则表达式	216	17.5 视图 (View).....	246
16.1 匹配模式.....	216	17.6 存储过程 (Stored Procedure).....	247
16.1.1 字符匹配.....	216	17.7 事务 (Transaction).....	248
16.1.2 转义字符.....	217	17.8 使用 ADO.NET.....	249
16.1.3 应用规则.....	217	17.8.1 连接数据库.....	249
16.2 Regex 类.....	218	17.8.2 执行 SQL 和调用存储过程.....	252
16.3 封装 CCheckData 类.....	219	17.8.3 使用事务.....	256
16.3.1 验证 E-mail 地址.....	219	17.8.4 脱机组件.....	257
16.3.2 验证手机号.....	220	第 18 章 创建数据基本操作组件	260
16.3.3 验证 18 位身份证号.....	220	18.1 CDataItem 和 CDataCollection 类.....	261
16.3.4 验证用户名格式.....	221	18.1.1 CDataItem 类.....	261
16.3.5 验证是否为汉字.....	222	18.1.2 CDataCollection 类.....	263
16.3.6 验证是否可以转换为数值.....	222	18.2 数据引擎组件.....	267
16.3.7 限制数据范围.....	223	18.2.1 IDbEngine 接口.....	267
第 17 章 SQL Server 数据库	225	18.2.2 CDbEngineBase 基类.....	269
17.1 应用基础.....	225	18.2.3 CSqlEngine 类与 CSql 类.....	270
17.2 准备数据库.....	226	18.3 数据记录操作组件.....	276
17.3 数据表与字段.....	229	18.3.1 IDbRecord 接口.....	276
17.3.1 常用数据类型.....	229	18.3.2 CDbRecordBase 基类.....	278
17.3.2 字段与约束.....	230	18.3.3 CSqlRecord 类.....	281
17.3.3 添加新记录.....	231	18.3.4 CDbRecord 类.....	286
17.3.4 更新记录.....	232	18.3.5 在项目中初始化 CDbRecord 类.....	288
17.3.5 删除记录.....	233	18.4 支持 Access 数据库.....	290
17.3.6 主键.....	234	18.4.1 CAccess 类.....	290
17.3.7 外键.....	235	18.4.2 CAccessEngine 类.....	291
17.4 数据查询.....	235	18.4.3 CAccessRecord 类.....	295

18.4.4 在 CDbRecord 类支持 Access	299	19.6 CSqlQuery 类	321
18.4.5 测试用 Access 数据库	300	19.7 CAccessQuery 类	323
18.5 综合测试	300	19.8 CDbQuery 通用类	325
18.5.1 基本数据操作测试	302	19.9 综合测试	327
18.5.2 用户登录	304	19.9.1 比较运算符查询	329
18.5.3 切换数据库	305	19.9.2 范围查询	331
第 19 章 创建数据查询组件	307	19.9.3 数据列表查询	332
19.1 查询条件	307	19.9.4 空值 (NULL) 查询	332
19.1.1 查询条件类型	307	19.9.5 文本模糊查询	333
19.1.2 条件之间的关系	308	19.9.6 使用 UseNot 属性	334
19.1.3 条件组合	309	19.9.7 组合条件查询	334
19.2 CCondition 类	309	19.10 支持其他数据库	335
19.2.1 CreateCompareCondition() 方法	310	第 20 章 操作 Excel 文件	336
19.2.2 CreateRangeCondition() 方法	311	20.1 使用 OLEDB	336
19.2.3 CreateDateRangeCondition() 方法	312	20.1.1 打开工作表	336
19.2.4 CreateValueListCondition() 方法	312	20.1.2 数据操作	339
19.2.5 CreateFuzzyCondition() 方法	313	20.2 使用 Excel 对象库	340
19.2.6 CreateNullValueCondition() 方法	313	20.2.1 Excel 文档与工作表	341
19.3 CConditionGroup 类	313	20.2.2 单元格	343
19.4 IDbQuery 接口	315	20.2.3 区域 (Range) 与格式	344
19.5 CDbQueryBase 类	315	第 21 章 Windows 窗体应用	346
19.5.1 基本实现	315	21.1 窗体与布局	346
19.5.2 GetCompareOperator() 方法	317	21.1.1 Form 类	346
19.5.3 GetConditionSql() 方法	317	21.1.2 使用 SplitContainer 控件 布局	347
19.5.4 GetConditionGroupSql() 方法	319	21.1.3 控件的 Dock 属性	348
19.5.5 GetSelectSql() 方法	320	21.1.4 MDI 窗体	349
		21.1.5 异形窗体	351
		21.1.6 无标题窗体移动与关闭	352
		21.2 Button 控件	353
		21.3 TextBox 控件	354
		21.4 MaskedTextBox 控件	354

21.5	NumericUpDown 控件	357	22.2.1	使用 IIS Express 测试	399
21.6	CheckBox 控件	357	22.2.2	使用 IIS 测试	401
21.7	RadioButton 与 GroupBox 控件	358	22.2.3	常用目录	403
21.8	列表控件	358	22.2.4	常用文件类型	404
21.8.1	ListBox 和 ComboBox 控件	358	22.2.5	加入封装代码库	405
21.8.2	列表的数据处理	359	22.3	页面与 Web 窗体	407
21.9	CheckedBoxList 控件	361	22.4	常用对象	409
21.10	日期与时间控件	365	22.4.1	Request 对象	409
21.11	菜单	366	22.4.2	Response 对象	411
21.12	通知图标	368	22.4.3	Server 对象	411
21.13	工具栏	369	22.4.4	Session 对象	412
21.14	DataGridView 控件	370	22.5	Web 控件	412
21.14.1	数据访问	370	22.5.1	按钮类控件	412
21.14.2	显示与格式设置	373	22.5.2	文本类控件	413
21.15	TreeView 控件	374	22.5.3	CheckBox 控件	415
21.16	对话框	376	22.5.4	列表类控件	416
21.16.1	信息、警告与错误	376	22.5.5	日期与自定义控件	420
21.16.2	提问对话框	378	22.5.6	Panel 控件	428
21.16.3	输入对话框	379	22.6	文件上传	430
21.16.4	颜色	384	22.7	缓存	433
21.16.5	字体	384	22.8	Ajax 基础	435
21.16.6	打开、保存文件	385	22.9	全站编译	439
21.16.7	选择路径	387	22.10	示例：基于数据库的用户 注册与登录	439
第 22 章 ASP.NET 网站开发	388		22.10.1	实现验证码	439
22.1	网站开发概述	388	22.10.2	注册	444
22.1.1	HTML	388	22.10.3	登录与跳转	447
22.1.2	CSS	389	第 23 章 项目示例 1：截屏程序	452	
22.1.3	JavaScript	392	23.1	实现截屏	452
22.1.4	动态页面技术	394	23.2	实时显示截取内容	455
22.1.5	数据库	397	23.3	响应键盘操作	457
22.2	创建 ASP.NET 网站	397	23.4	保存到剪切板	457
			23.5	添加自动保存选项	457

第 24 章 项目示例 2：迷你账本.....460

24.1 项目概况	460
24.1.1 账目的基本操作	460
24.1.2 多账本管理	460
24.1.3 安全性	460
24.1.4 账目查询	461
24.1.5 账目统计	461
24.2 项目准备	461
24.2.1 创建项目数据库	461
24.2.2 初始化 CAccountBook 项目	463
24.2.3 主窗体	465
24.3 系统与账本操作	467
24.3.1 家长权限	467
24.3.2 账本管理	470
24.3.3 打开账本	474
24.4 添加账目	478
24.4.1 新增支出项	479
24.4.2 新增收入项	481
24.5 账目查询与编辑	481
24.5.1 周期查询	481
24.5.2 编辑账目信息	484
24.5.3 综合查询	485
24.5.4 删除	488
24.6 账目统计	488

第 25 章 项目示例 3：Web 版**个人助手**.....491

25.1 项目概况	491
25.2 项目准备	492
25.2.1 准备数据库	492
25.2.2 项目初始化	493
25.2.3 处理会话数据	495
25.2.4 修改 CVerificationCode 类	498

25.2.5 Web.Config 配置与自定义

控件.....499

25.2.6 ASP.NET 页面模板.....500

25.3 首页.....500

25.4 用户注册.....503

25.4.1 封装代码.....503

25.4.2 注册页面.....505

25.4.3 保存用户信息.....507

25.5 登录.....510

25.6 个人信息.....514

25.7 修改密码.....516

25.8 通讯录功能.....519

25.8.1 准备数据表.....519

25.8.2 CAddrList 类.....520

25.8.3 通讯录主页 (/addrlist/
Index.aspx).....521

25.8.4 查询 (CAddrListQuery 类).....524

25.8.5 编辑联系人 (/addrlist/Edit.
aspx).....525

25.9 账本功能.....529

25.9.1 准备数据库.....529

25.9.2 CAcctBook 和 CAcctRec 类.....530

25.9.3 账本管理.....532

25.9.4 账目查询.....536

25.9.5 账目添加与修改.....548

25.9.6 账目删除.....552

25.9.7 账目统计.....554

附录 A ASCII 码表.....558**附录 B 二进制、十进制与十六进制**

对照表.....559

附录 C 基本数据类型对照表.....560

第 1 章 概述

C# 是 .NET Framework 平台下最重要的编程语言之一，使用 C# 语言和功能强大的类库，可以让开发效率有质的飞越。

本章，我们会了解 C# 代码的编写、组织，以及常见的应用类型。

1.1 编写 C# 代码

很久以来，一些代码狂人都喜欢使用文本编辑工具来编写代码，然后通过命令行编译出可执行程序。C# 代码当然也可以这么干，例如使用记事本编写代码，然后使用 `csc.exe` 命令编译。但是，对于想尽快掌握应用软件开发的朋友来讲，能够使用功能强大的开发工具是非常必要的。

本书，我们使用了微软公司出品的 Visual Studio（简称 VS）家族中的免费版本，如：

- ❑ VS Express for Desktop，在本书中主要用来开发单机程序，如控制台应用程序和 Windows 窗体应用程序。
- ❑ VS Express for Web，在本书中主要用来开发 ASP.NET 网站。
- ❑ 此外，对于一些新的 C# 语言特点或 .NET Framework 资源，我们需要在较新的开发环境中进行测试，如果你只想安装一种开发环境（而且还是免费的），Visual Studio Community 2015 是一个不错的选择。

1.1.1 第一个 C# 程序

这里，启动 VS Express 2013 for Desktop 开发环境，并通过菜单项“文件”→“新建项目”打开新建项目窗口，然后，创建如图 1-1 所示的项目类型。

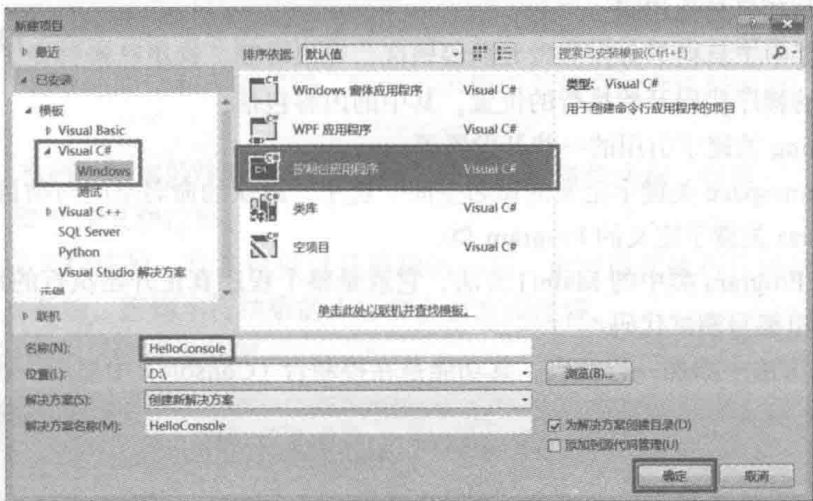


图 1-1 创建控制台应用程序

在这里，选择“Visual C#”下的“Windows”项。然后，在模板中选择“控制台应用程序”，并将项目命名为 HelloConsole，对于项目的存放位置，大家可以自己选择。最后，单击“确定”按钮完成项目的创建。

控件台应用程序，也称为命令程序，是指在命令行环境下使用一系列指令进行操作的软件。在 Windows 7 操作系统中，可以通过 Ctrl+R 快捷键打开“运行”窗口，并执行 cmd 命令打开命令行窗口，然后，可以执行一系列的指令。

回到刚刚创建的 HelloConsole 项目，可以看到，项目中已经自动创建了一些内容，如图 1-2 所示。

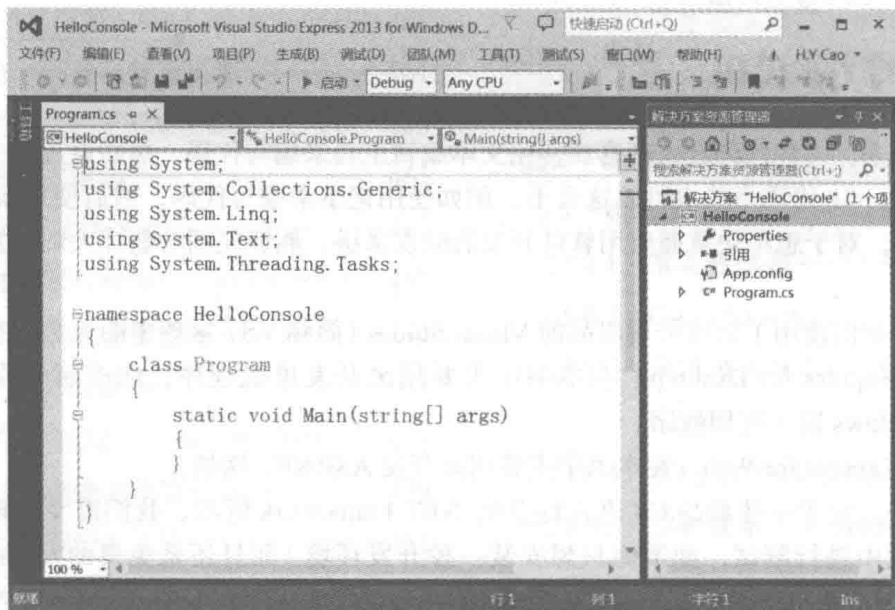


图 1-2 项目默认内容

图 1-2 中的工具栏下方，右侧是“解决方案资源管理器”，显示了项目中包含的内容，如引用资源、代码文件、配置文件等。这些内容都组织在“解决方案”中。实际应用中，一个解决方案可以包含多个项目，但在本书中，一个解决方案一般只会包含一个项目，而且，解决方案名称与项目名称相同。

再看图 1-2 中工具栏下方左侧较大的编辑区，可以看到，这里已经打开了 Program.cs 文件，它是默认的程序代码开始执行的位置，其中的内容包括：

- 使用 using 关键字引用的一些开发资源；
- 使用 namespace 关键字定义的命名空间。这里，默认的命名空间与项目名称相同；
- 使用 class 关键字定义的 Program 类；
- 定义在 Program 类中的 Main() 方法，它就是整个程序真正开始执行的地方。接下来，就在这里编写测试代码。

在 Main() 方法中添加一行代码，其功能是在控制台 (Console) 中显示一些内容。

```
namespace HelloConsole
{
    class Program
    {
        static void Main(string[] args)
```