

指南

吴明哲 审校
林义证 蔡文龙 吴昱欣 编著

Visual C# 2005 完全开发指南

微软认证专家传授高效学习要诀，一次掌握Visual C#四大学习主题

技术要点：

- Visual C# 2005基本语法
- 面向对象程序设计
- 异常处理与文件存取
- 窗口应用程序设计

- 数据访问工具
- ADO .NET 2.0数据库操作
- XML相关技术
- 数据库及报表应用程序开发



光盘内容
书中所有范例源代码

Visual C# 2005 完全开发指南

吴明哲 审校

林义证 蔡文龙 吴昱欣 编著

刘尚滨 改编

科学出版社

北京科海电子出版社

(外埠寄费另计，邮购从速，恕不退货)

内 容 简 介

本书由中国台湾微软大型研讨会讲师联合大学资深程序设计课程教授精心打造。书中详细介绍了 Visual C# 2005 的所有技术要点,并结合当下最流行的应用,通过大量范例,让读者快速掌握 Visual C# 程序设计的核心技术及应用开发过程。

全书共 20 章,第 1 章简单介绍 .NET Framework;第 2~6 章主要介绍 Visual C# 2005 中的数据结构、流程控制、方法以及窗口应用程序的设计基础;第 7~10 章主要介绍对象、类、继承、多态、接口、异常处理和文件访问;第 11~15 章主要介绍窗体设计及常用控件的属性和方法;第 16~18 章主要介绍如何通过数据控件和 ADO.NET 访问数据库;第 19 章介绍如何读写 XML 文件;第 20 章主要介绍 Crystal Reports 报表视图程序的开发方法。配书光盘中收录了书中涉及的所有范例源代码。

本书面向 Visual C# 初、中级用户,既可作为从业人员的自学教材,也可作为高等院校相关专业教师 and 学生的参考用书。

图书在版编目(CIP)数据

Visual C# 2005 完全开发指南/林义证,蔡文龙,吴昱欣编著.

—北京:科学出版社,2008

ISBN 978-7-03-022279-4

I. V… II. ①林…②蔡…③吴… III. C 语言—程序设计—指南 IV. TP312-62

中国版本图书馆 CIP 数据核字(2008)第 082975 号

责任编辑:何立兵 / 责任校对:杨慧芳

责任印刷:科海 / 封面设计:林陶

科学出版社 出版

北京东黄城根北街 16 号

邮政编码:100717

<http://www.sciencep.com>

北京市鑫山源印刷有限责任公司印刷

科学出版社发行 各地新华书店经销

*

2008 年 7 月第一版

开本:16 开

2008 年 7 月第一次印刷

印张:34

印数:000 1-4 000

字数:827 千字

定价:59.00 元(含 1CD 价格)

(如有印装质量问题,我社负责调换)

前 言

自从Internet盛行至今，网络上相关软件的开发技术几乎是每隔不到2年就翻新一次，而在微软公司的软件开发工具中，Visual Studio一直是Windows平台上整合能力最强、功能最齐备的软件开发工具，Visual Studio 2005更针对多项软件开发技术进行了强化。

在软件开发架构上，.NET Framework将成为未来软件开发的通用平台。由于.NET Framework容易移植到各个不同的平台上（包括Windows、UNIX、PDA、手机等），因此Visual C# 2005程序设计师可以直接运用.NET Framework中完整且丰富的类库设计出跨平台的软件系统。另外，通过通用语言运行时（Common Language Runtime, CLR）的接口，在.NET Framework上开发的软件，不论使用哪一种程序语言，都可以互相直接引用，这样在多人共同开发软件系统时，程序设计师就可以使用自己擅长的程序语言来参与软件开发，轻松解决了跨语言集成的问题。

微软公司提出的.NET愿景（Vision）所要建立的信息应用环境，就是让用户可以在任何时刻、任何地点、使用任何设备来存取所需信息，而.NET Framework 与Visual Studio 2005正是用来构建.NET愿景的基础平台与开发工具，通过开发XML Web Service及其相关机制，不但得以整合各类型平台上的开发成果，降低企业e化成本与缩短产品上市时间，更强化了分布式系统的安全性及多功能性，这一切正是构建.NET的“骨与肉”所不可或缺的基石。

本书秉持易学实用的原则，使用浅显易懂的语言与丰富的实际范例，将各种计算机与网络相关技术传递给读者，希望能广受读者的支持与喜爱。本书以这样的原则，期盼能引领读者进入未来软件设计的主流领域。

本书可作为计算机相关专业程序设计课程的教科书。

本书虽然经过多次精心的校对，但难免百密一疏，尚祈读者不吝指正，以期再版时能更趋扎实。读者可通过tsai_wen_lung@pchome.com.tw邮箱询问本书相关的Visual C# 2005问题。

在此声明，本书中所提及的相关产品名称皆是其所属公司的注册商标。

吴明哲 审校
林义证 蔡文龙 吴昱欣 谨识
2008年3月

目 录

第1章 .NET Framework简介	1
1.1 Visual Studio 2005概述	1
1.2 软件开发技术的发展	2
1.2.1 软件分析方法	2
1.2.2 程序设计方法	2
1.2.3 分布式软件技术	3
1.2.4 软件平台	7
1.3 Microsoft .NET	8
1.4 .NET Framework	9
1.4.1 通用语言运行时	10
1.4.2 基础类库	12
1.4.3 ADO .NET与XML	14
1.4.4 Windows Form与Web Form	15
1.4.5 通用语言规范	16
1.4.6 程序语言	17
1.4.7 跨平台的 .NET	17
第2章 C# 2005程序体系结构与数据类型	18
2.1 C# 2005程序体系结构	18
2.1.1 命名空间的声明与完整名称的访问方法	18
2.1.2 C# 程序的基本体系结构	19
2.2 Visual C# 2005数据类型	23
2.2.1 标识符	23
2.2.2 变量	24
2.2.3 C#数据类型	24
2.2.4 变量的声明与初始化	26
2.2.5 字符串数据类型	27
2.3 Visual C# 2005运算符	28
2.3.1 算术运算符	28
2.3.2 关系运算符	29
2.3.3 逻辑表达式	30
2.3.4 位运算符	30
2.3.5 移位运算符	31

2.3.6 复合赋值运算符.....	31
2.3.7 递增及递减运算符.....	32
2.3.8 运算符的优先级.....	33
2.4 控制台应用程序.....	34
2.4.1 新建控制台应用程序项目、程序的编写与执行	34
2.4.2 关闭解决方案	37
2.4.3 打开解决方案	37
2.5 控制台项目格式化输入/输出	38
2.5.1 Write/ WriteLine方法.....	38
2.5.2 Read/ReadLine方法.....	39
2.5.3 如何格式化 (Format) 输出	41
2.5.4 自定义数值格式输出字符串	43
2.5.5 转义控制字符	45
2.6 枚举数据类型	48
2.7 数据类型转换	50
2.7.1 隐含转换 (Implicit Conversion)	50
2.7.2 显式转换 (Explicit Conversion)	51
第3章 流程控制	55
3.1 选择结构.....	55
3.1.1 if...else语句.....	55
3.1.2 if...else if...else语句	57
3.1.3 switch语句	59
3.1.4 ?:三元运算符.....	62
3.2 循环结构.....	64
3.2.1 for语句.....	64
3.2.2 while语句	68
3.2.3 do...while语句.....	69
3.2.4 break语句	70
3.2.5 continue语句.....	70
3.3 习题.....	72
第4章 数组与字符串	73
4.1 数组.....	73
4.1.1 数组的声明	73
4.1.2 数组的初值设定.....	75
4.1.3 foreach语句	77
4.2 不规则数组	80
4.3 ArrayList类	82

4.3.1 ArrayList类常用属性	83
4.3.2 ArrayList类常用方法	83
4.4 字符与字符串	87
4.4.1 字符串的声明和初值设定	87
4.4.2 Visual C# 2005字符类成员	88
4.4.3 Visual C# 2005字符串类成员	91
4.5 习题	94
第5章 方法	95
5.1 方法	95
5.2 如何定义方法	96
5.2.1 定义方法	96
5.2.2 如何调用方法	98
5.3 参数的传递方式	105
5.3.1 传值调用	106
5.3.2 引用调用	108
5.3.3 输出参数	109
5.4 在方法间传递数组	111
5.5 方法重载	113
5.6 块变量、区域变量、静态变量与属性成员	114
5.6.1 块变量 (block level variables)	114
5.6.2 区域变量 (local variables)	115
5.6.3 静态变量 (static variables)	115
5.6.4 对象属性 (非静态成员)	117
5.7 DateTime、Math及Random类	117
5.7.1 DateTime类	117
5.7.2 Math类	118
5.7.3 Random类	120
5.8 习题	121
第6章 窗口应用程序基础操作	122
6.1 Windows应用程序项目	122
6.2 Visual C# 2005面向对象概念简介	123
6.3 第一个Windows应用程序	124
6.4 认识Visual C# 2005集成开发环境	135
6.4.1 Visual C# 2005集成开发环境简介	135
6.4.2 解决方案资源管理器 (Solution Explorer)	139
6.4.3 属性窗口	140
6.4.4 程序代码窗口	141

6.4.5	如何将窗口叠在一起或分开.....	142
6.4.6	如何使用动态帮助.....	143
6.5	控件的操作.....	143
6.5.1	建立控件.....	143
6.5.2	如何选定控件.....	144
6.5.3	如何移动和调整控件.....	145
6.5.4	控件对齐.....	145
6.5.5	如何删除控件.....	145
6.5.6	如何调整控件的前后顺序.....	145
6.5.7	如何调整控件的定位顺序.....	146
6.5.8	如何锁定控件.....	146
第7章	对象与类.....	147
7.1	前言.....	147
7.2	对象与类.....	148
7.2.1	对象.....	148
7.2.2	类.....	150
7.3	面向对象程序设计的特性.....	151
7.3.1	抽象化.....	151
7.3.2	封装.....	157
7.3.3	继承.....	157
7.3.4	多态.....	158
7.3.5	动态绑定.....	159
7.4	对象与类的建立.....	160
7.4.1	如何建立类.....	160
7.4.2	命名空间 (Namespace).....	161
7.4.3	如何建立属性.....	163
7.4.4	如何建立方法.....	168
7.4.5	如何建立索引.....	175
7.4.6	如何建立事件.....	177
7.5	习题.....	179
第8章	继承、多态、接口.....	180
8.1	继承.....	180
8.1.1	类继承.....	180
8.1.2	类成员的访问限制.....	182
8.2	静态成员.....	182
8.2.1	静态成员的使用.....	182
8.2.2	.NET Framework的内存配置.....	184

8.3 多态	185
8.3.1 重载	185
8.3.2 子类如何访问父类的方法或属性	187
8.3.3 动态绑定	189
8.3.4 抽象类	192
8.4 接口与实现	199
8.5 delegate委托类型	204
8.6 习题	206
第9章 异常处理	208
9.1 try...catch...finally语句	208
9.2 异常类	209
9.2.1 常用异常类	209
9.2.2 异常类常用成员	209
9.3 自定义异常处理: 使用throw语句	215
9.4 异常类继承	217
9.5 习题	220
第10章 文件访问	221
10.1 System.IO命名空间常用类简介	221
10.2 DirectoryInfo类	222
10.3 FileInfo类	226
10.4 文件读写	230
10.5 Path类	234
10.6 习题	234
第11章 窗体设计与基础控件开发	235
11.1 窗体和控件	235
11.1.1 窗体和控件常用的成员	235
11.1.2 窗体和屏幕的关系	238
11.1.3 其他属性	239
11.1.4 窗体常用的事件过程	240
11.2 Label标签控件	243
11.3 LinkLabel链接标签控件	246
11.4 ToolTip提示控件	248
11.4.1 ToolTip的使用方式	248
11.4.2 ToolTip常用成员	249
11.5 Button按钮控件	251
11.5.1 Button常用成员	251

11.5.2 程序中如何设置Button属性	252
11.5.3 按钮如何与Enter键和Esc键结合	252
11.6 TextBox文本框控件	253
11.7 RichTextBox富文本框控件	256
11.8 MessageBox.Show方法	262
11.9 习题	266
第12章 常用控件	268
12.1 RadioButton单选按钮控件	268
12.2 CheckBox复选框控件	270
12.3 容器工具	271
12.3.1 GroupBox控件	271
12.3.2 Panel控件	275
12.4 列表工具	276
12.4.1 ListBox控件	276
12.4.2 CheckedListBox控件	279
12.4.3 ComboBox控件	280
12.5 旋转工具	284
12.5.1 NumericUpDown控件	285
12.5.2 DomainUpDown控件	286
12.6 滚动条工具	288
12.6.1 HScrollBar及VScrollBar控件	288
12.6.2 TrackBar控件	291
12.7 日期和时间工具	294
12.7.1 Timer定时器控件	295
12.7.2 MonthCalendar控件	298
12.7.3 DateTimePicker控件	301
12.8 图形处理工具	305
12.8.1 PictureBox控件	305
12.8.2 ImageList控件	309
12.9 习题	312
第13章 工具栏容器	314
13.1 ToolStrip工具栏控件	314
13.1.1 ToolStrip常用成员	314
13.1.2 建立工具栏控件的步骤	315
13.2 MenuStrip菜单控件	324
13.2.1 建立菜单控件	324
13.2.2 菜单控件的常用成员	328

13.3 ContextMenuStrip快捷菜单控件.....	333
13.4 习题.....	337
第14章 键盘、鼠标与共享文件.....	338
14.1 键盘事件简介.....	338
14.1.1 KeyPress事件.....	339
14.1.2 KeyDown和KeyUp事件.....	344
14.2 鼠标事件简介.....	349
14.2.1 Click与DoubleClick事件.....	349
14.2.2 MouseDown、MouseUp与Click事件.....	350
14.3 控件共享事件过程.....	353
14.3.1 使用属性窗口建立共享事件.....	353
14.3.2 建立和取消控件的事件.....	355
14.4 习题.....	364
第15章 通用对话框的应用.....	365
15.1 字体、颜色对话框.....	365
15.1.1 字体对话框.....	365
15.1.2 颜色对话框.....	367
15.2 文件对话框.....	370
15.3 打印工具.....	375
15.3.1 PrintDocument打印文件控件.....	375
15.3.2 PageSetupDialog打印页面设置对话框控件.....	376
15.3.3 PrintPreviewDialog打印预览对话框控件.....	377
15.3.4 PrintDialog打印对话框控件.....	378
15.4 习题.....	382
第16章 数据工具的使用.....	383
16.1 本章概述.....	383
16.2 数据库概念.....	384
16.2.1 数据表.....	384
16.2.2 数据表与数据库.....	385
16.2.3 关系型数据库.....	385
16.3 ADO.NET简介.....	386
16.3.1 ADO.NET.....	386
16.3.2 .NET Data Provider.....	388
16.3.3 DataSet.....	388
16.4 .NET Data Provider.....	389
16.5 使用数据工具建立数据库应用程序.....	390

16.6 数据库的关联查询	398
16.7 数据表记录的修改与浏览	406
16.8 BindingNavigator控件	415
16.9 习题	417
第17章 ADO .NET数据库连接与访问	419
17.1 如何引用ADO .NET命名空间	419
17.2 使用Connection对象打开、关闭数据库连接	421
17.3 使用DataReader对象读取数据表记录	423
17.3.1 Command对象	423
17.3.2 DataReader对象	423
17.3.3 DataReader常用的属性与方法	424
17.3.4 如何使用DataReader读取查询结果	425
17.4 使用DataSet对象读取数据表记录	430
17.4.1 DataSet对象	430
17.4.2 DataAdapter对象	430
17.4.3 如何使用DataAdapter对象	431
17.4.4 如何使用DataTable对象来读取数据	436
17.4.5 如何使用DataTable显示查询结果的所有记录	436
17.5 SQL语法与Command对象	439
17.5.1 Select查询命令	440
17.5.2 使用SQL的注意事项	440
17.5.3 Insert新建命令	445
17.5.4 Delete删除命令	446
17.5.5 Update更新命令	446
17.5.6 如何使用Command对象及SQL语句更新数据库	446
17.5.7 如何使用命名参数及SQL语句	451
17.6 习题	455
第18章 ADO.NET数据库浏览与绑定	456
18.1 DataView对象	456
18.1.1 如何建立DataView对象	456
18.1.2 如何使用DataView对象来排序数据	457
18.1.3 如何使用DataView对象来过滤数据	457
18.2 数据绑定	461
18.3 数据表的关联	465
18.4 使用BindingManagerBase来修改数据表记录	470
18.5 习题	475

第19章 读写XML文件	476
19.1 XML简介	476
19.2 读写XML文件的方法	479
19.2.1 如何读取XML文件中的记录数据	479
19.2.2 如何在XML文件中新建、删除、修改记录数据	480
第20章 Crystal Reports报表视图程序	487
20.1 Crystal Reports 简介	487
20.1.1 Crystal Reports功能概述	488
20.1.2 Crystal Reports可导出的数据格式	488
20.1.3 Crystal Reports支持不同的数据源	488
20.2 重新设置报表的数据源	489
20.3 Windows Form的报表视图程序	493
20.3.1 建立第一个Windows Form报表视图程序	493
20.3.2 设计报表的基本技巧	499
20.3.3 如何设置报表的筛选条件、排序、分组	506
20.4 图表、子报表的设计	511
20.4.1 图表的设计	511
20.4.2 子报表的设计	518
20.5 习题	529

第1章

.NET Framework 简介

1.1 Visual Studio 2005概述

1.2 软件开发技术的发展

1.3 Microsoft .NET

1.4 .NET Framework

1.1 Visual Studio 2005概述

软件系统的规划与配置，以往总需要针对不同操作系统平台、程序语言、软件技术等整合，这样的整合往往费时耗力，更是增加软件复杂度的原因之一。因此为了简化软件规划与设计所需的成本，学术界与商业界不断推出更先进的软件设计技术与产品，以顺应日趋复杂的软件开发需求。

本章将深入介绍整个软件开发环境（包含规划、设计、整合等领域）从过去到现在的进展，以及针对目前的软件开发困境，市场上有哪些解决方案可供选择。当然，本章最后会详细介绍软件开发平台 .NET Framework，看看微软的 .NET 愿景是如何通过 .NET Framework 构建的，以及 Visual Studio 2005 开发环境与 Visual C# 2005 语言所能带来的软件开发优势。

除此之外，我们也将深入剖析分布式软件开发技术，尤其是目前主流的 XML Web Service 技术。在完成整个 Internet 软件开发技术的介绍之后，您将深刻了解如何借助 Visual Studio 2005 软件开发工具和 Visual C# 2005 程序语言，构建出以 .NET Framework 为坚实基础的软件组件与系统，再经由 XML Web Service 将软件系统延伸至 Internet，最后整合 Internet 上不同平台上的不同系统，以及各类客户端设备，勾勒出 .NET 愿景所带来的无限商机。

1.2 软件开发技术的发展

一项新的软件开发技术或平台的推出，往往是因为原有的技术或平台已不足以应付当前软件开发的需求。因此，如果能够了解软件开发技术的演进，就更能清楚地了解.NET Framework这个新一代的软件开发平台与技术是从何而来，有什么值得深入探讨。

以下就分别从软件分析、程序设计、分布式技术与软件平台这4个方面，说明软件开发技术的演进。

1.2.1 软件分析方法

在整个软件系统开发流程的先期，系统分析是相当重要的一部分。在面向对象技术出现之前，结构化程序规划与设计方式一直是软件开发的主流技术。然而为了顺应日益复杂的软件系统，在程序设计技术方面出现了重大变革，那就是面向对象（Object Oriented）技术，于是以往结构化、模块化的软件分析方法已不适用于超大型软件系统，于是OOA（Object Oriented Analysis，面向对象分析）与OOD（Object Oriented Design，面向对象设计）的方法在1989年~1994年增加到50多种，然而不同的分析方法却导致无法适用各类型系统分析的问题。

1997年由Grady Booch、James Rumbaugh、Ivar Jacobson三位软件分析大师与微软、Digital、HP、IBM、Oracle等公司共同整合制订而成的UML（Unified Modeling Language，统一建模语言）软件分析方法，被OMG（Object Management Group，对象管理组）协会认可，正式成为主流的面向对象分析与设计方法。

1.2.2 程序设计方法

现今的程序设计师往往必须埋头于数十万行的程序代码中，殚精竭虑地维持软件系统的可靠性与稳定性，然而传统的单纯的结构化程序设计技术早已无法掌控日益复杂的软件系统，于是在程序设计方法的历史演变中，面向对象（Object Oriented，OO）算是近代软件系统分析、规划与设计概念的一大突破，无疑也是大型软件系统的救世主。

在OO领域中，除了先前提到的OOA与OOD之外，另一项与程序设计密不可分的技术就是OOP（Object Oriented Programming，面向对象程序设计）。所谓OOP就是指使用面向对象的概念来编写程序。有经验的程序设计师甚至可以使用不支持面向对象语法的程序语言编写出面向对象的程序，例如汇编语言；但经验不足的程序设计师则有可能使用诸如C++之类支持面向对象程序语法的程序语言，却编写出一点都不像面向对象的程序。

经常有学生问笔者：如何将长度不到10行的程序代码改用面向对象语言重写？例如：如何将原本使用for循环编写出来的九九乘法表程序，用面向对象的语法重新编写？也有些负责企业MIS或ERP系统的程序设计师问：该如何将系统中的所有程序改成面向对象？似乎所有的人都将面向对象当成万能仙丹，将传统结构化程序设计视为万恶祸首。其实面向对象程序设计技术并不是用来彻底替换原本的程序设计方法，相反，面向对象的目的在于弥

补传统结构化程序设计中大型或复杂系统难以设计与维护的缺点。因此“如何将原本使用for循环编写出来的九九表乘法程序，用面向对象的语法重新乘法写”这样的要求，显然太过吹毛求疵了。

曾经听过这么一个故事，有一个拥有10家分公司的跨国企业，各分公司的MIS系统可以统计某一月份的营收总额，然后将营收报表送交总公司备查。公司的总裁在拿到这10份营收报表之后，想要进一步得到该月份所有分公司的营收总额，于是召开了一次会议来商讨解决方案。在会议中有人提议将目前各分公司的MIS系统改写，建立企业的ERP系统，将各分公司与总公司系统联机，以进行营收总额的统计；也有人建议委托其他软件顾问公司进行系统规划与设计。正当大家为此争执不休时，只见总裁的特别助理拿出计算器，将报表上10个分公司的营收金额相加后，不到30s就算出了该月份全公司的营收总额。

因此，“在适当的地方使用适当的技术来解决问题”才是软件设计中至高的原则，而这“适当的技术”不见得非面向对象不可，也不见得非要用程序设计方法来解决。

1.2.3 分布式软件技术

软件系统如果依功能来分类，可以分成以下几个不同的层次。

(1) 展示 (Presentation)。

在软件系统的展示层中，主要负责用户界面的处理，包括数据的输入与显示、图表的绘制等与数据展示有关的工作。

(2) 商业逻辑 (Business Logic)。

软件系统中都包含了许多逻辑的处理，例如银行利息的计算、转账、账号与密码的检查、数据的加密与解密等，这些工作都属于商业逻辑层次的工作范围。

(3) 数据源 (Data Source)。

所谓的数据源，指的就是软件系统中数据存储的位置，例如Microsoft SQL Server、Oracle、Sybase、Access等数据库，甚至XML数据都可以称作数据源。

从软件系统的架构（以上3个层次的组合方式）来看，软件系统大致上可以分成以下几种类型。

(1) 单机架构 (Stand Alone)。

在单机架构中，展示层、商业逻辑与数据源都位于同一个软件、同一台计算机上，例如用dBase、Access、FoxPro等开发出来的单机应用程序。由于整个系统不需要牵涉到与其他计算机或系统的整合，因此在设计上非常简单，在单机上的效率也不差，适合单人操作使用。然而由于数据位于软件所在的计算机上，因此，如果要进行不同计算机之间的数据交换，往往需要通过文件复制来实现，非常不方便。

假设某银行的系统是采用单机架构开发而成，那么今天在台北市某一分行存款10 000元，这个存款数据只会存放在该行柜台人员所用的计算机上，必须将该笔新的存款数据复制到可移动的磁盘上，然后复制到全省（中国台湾省）其他分行所有柜台人员使用的计算机中，才能在其他分行中看到这笔新的存款数据。如果要进行复杂的转账交易，那么单机架构就明显力不从心了。