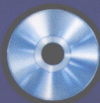


光盘内容  
书中所有范例的完整源代码



# 指南

吴明哲 审校  
曹祖圣 蔡文龙 林义证 编著  
刘尚滨 改编

# Visual Basic 2005 完全开发指南

## 技术要点:

- Visual Basic 2005基本语法
- 面向对象程序设计
- 窗口应用程序开发
- ADO.NET数据访问工具
- XML相关技术
- ASP .NET Web应用程序开发
- 移动Web应用程序开发
- 数据库及报表应用程序开发

科海“完全开发指南”系列编程图书  
中国台湾微软特聘讲师及顾问精心打造

开发  
指南

# 指南

# Visual Basic 2005 完全开发指南

吴明哲 审校  
曹祖圣 蔡文龙 林义证 编著  
刘尚滨 改编

科学出版社

北京科海电子出版社  
[www.khp.com.cn](http://www.khp.com.cn)

## 内 容 提 要

本书由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 VB 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 VB 程序设计的核心技术及应用开发过程。

全书共 23 章,第 1 章简单介绍 .NET Framework;第 2~6 章主要介绍 VB 2005 中的数据结构、流程控制语句、过程以及窗口应用程序的设计基础;第 7~10 章介绍对象、类、继承、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 章讲解 ADO.NET 的使用,包括 DataSet、DataTable 等,以及如何通过 SQL 语言和数据对象访问数据库;第 19~23 章主要介绍 ASP.NET Web 应用程序、Mobile Web 应用程序以及 Crystal Reports 报表视图程序的开发方法。

附书光盘收录了书中涉及的所有范例源码。

本书面向 Visual Basic 初、中级用户,既可作为从业人士的自学教材,也可作为高等院校相关专业教师和学生的参考用书。

### 图书在版编目 (CIP) 数据

Visual Basic 2005 完全开发指南/曹祖圣,蔡文龙,林义证编著.  
—北京:科学出版社,2007  
ISBN 978-7-03-019925-6

I. V… II. ①曹… ②蔡… ③林… III. BASIC 语言—程序设计  
IV. TP312

中国版本图书馆 CIP 数据核字 (2007) 第 139407 号

责任编辑:何立兵 / 责任校对:李玉茹  
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>

北京科普瑞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

\*

2007 年 10 月第一版 开本:16 开  
2007 年 10 月第一次印刷 印张:37.75  
印数:0001~4000 字数:919 千字

定价:63.00 元 (1CD)

(如有印装质量问题,我社负责调换)

# 前 言

自从Internet盛行至今,在网络上相关软件的开发技术几乎是每隔不到两年就翻新一次,而在微软的软件开发工具中, 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,新版的 Visual Studio 2005更针对多项软件开发技术进行强化,尤其是 Visual Basic 2005,不但延续以前版本易学易用的特性,让初学者能够轻松走进程序设计的殿堂,更完整地支持了以前版本所欠缺的面向对象机制,这样,使用Visual Basic 2005开发大型的软件系统,能够降低软件设计的复杂度和维护的成本。

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 Basic 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库(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,都可以直接互相引用,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编程语言来参与软件开发,直接解决了跨语言集成的问题。

微软提出的.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与相关的机制,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与多功能性,这一切正是构建出.NET的骨与肉所不可或缺的基石。

本书秉持着易学易懂的原则,使用浅显易懂的语言与丰富的实际范例,将各类计算机与网络的相关技术传递给读者,因此广受读者的支持与喜爱。期盼本书能引领读者进入未来软件设计的主流领域。

读者可通过tsai\_wen\_lung@pchome.com.tw信箱询问本书相关的VB 2005问题。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。

在此声明,本书中所提及之相关产品名称皆各所属公司之注册商标。

曹祖圣 林义证 蔡文龙 谨识

# 目 录

<b>第1章 .NET Framework介绍</b> .....	1	2.2.3 逻辑表达式 .....	26
1.1 Visual Studio 2005概述 .....	1	2.2.4 复合赋值运算符 .....	27
1.2 软件开发技术的演进 .....	2	2.2.5 运算符的优先级 .....	28
1.2.1 软件分析的方法 .....	2	2.3 控制台应用程序 .....	29
1.2.2 程序设计方法 .....	2	2.3.1 新建控制台应用程序项目、程序的 编写与运行 .....	29
1.2.3 分布式软件技术 .....	3	2.3.2 关闭项目 .....	33
1.2.4 软件平台 .....	7	2.3.3 打开项目 .....	33
1.3 Microsoft .NET .....	8	2.4 控制台项目格式化输入/输出 .....	34
1.4 .NET Framework .....	9	2.4.1 Write/WriteLine方法 .....	34
1.4.1 公共语言运行时 .....	10	2.4.2 Read/ReadLine方法 .....	34
1.4.2 基础类库 .....	12	2.4.3 如何格式化输入/输出 .....	36
1.4.3 ADO .NET与XML .....	14	<b>第3章 流程控制</b> .....	<b>39</b>
1.4.4 Windows Form与Web Form .....	15	3.1 选择结构 .....	39
1.4.5 公共语言规范 .....	16	3.1.1 If...Then...Else 语句 .....	39
1.4.6 程序语言 .....	17	3.1.2 If...Then...Elseif 语句 .....	41
1.4.7 跨平台的 .NET .....	17	3.1.3 Select Case 语句 .....	42
<b>第2章 VB 2005数据类型与运算符</b> .....	<b>18</b>	3.1.4 If函数 .....	45
2.1 变量与数据类型 .....	18	3.1.5 Switch函数 .....	46
2.1.1 变量的命名规则 .....	18	3.1.6 Choose函数 .....	47
2.1.2 变量的种类与数据类型 .....	19	3.2 循环结构 .....	48
2.1.3 变量的声明和初始值设定 .....	20	3.2.1 For...Next 语句 .....	48
2.1.4 常数 .....	23	3.2.2 Do While...Loop语句 .....	51
2.2 VB 2005运算符 .....	24	3.2.3 Do...Loop While语句 .....	53
2.2.1 算术运算符 .....	24	3.2.4 Do Until...Loop语句 .....	54
2.2.2 关系运算符 .....	25		

3.2.5 Do...Loop Until语句 .....	54	5.3.3 如何定义Function过程 .....	91
3.2.6 While ...End While语句 .....	55	5.3.4 如何调用Function过程 .....	91
3.2.7 Continue .....	55	5.4 传值调用及引用调用 .....	93
3.3 习题 .....	56	5.4.1 传值调用 .....	93
<b>第4章 数组与字符串 .....</b>	<b>58</b>	5.4.2 引用调用 .....	95
4.1 数组 .....	58	5.5 变量的生命期 .....	97
4.1.1 数组的声明 .....	58	5.6 过程自变量顺序可变性 .....	100
4.1.2 数组的初始值设定 .....	60	5.7 过程自变量可选择性 .....	101
4.1.3 For Each...Next .....	62	5.8 在过程间传递数组 .....	102
4.1.4 ReDim语句 .....	64	5.9 数组元素可变性 .....	104
4.1.5 IsArray函数 .....	66	5.10 过程重载 .....	105
4.2 不规则数组 .....	66	5.11 习题 .....	106
4.3 ArrayList类 .....	68	<b>第6章 窗口应用程序基础操作 .....</b>	<b>107</b>
4.3.1 ArrayList类常用属性 .....	68	6.1 Windows应用程序项目 .....	107
4.3.2 ArrayList类常用方法 .....	69	6.2 VB 2005面向对象观念介绍 .....	108
4.4 字符串的声明与初始值设定 .....	72	6.3 第一个Windows应用程序 .....	109
4.5 VB传统字符串函数 .....	73	6.4 认识VB 2005的集成开发环境 .....	120
4.6 VB 2005字符串类的成员 .....	75	6.4.1 VB 2005集成开发环境介绍 .....	120
4.7 习题 .....	79	6.4.2 解决方案资源管理器 (Solution Explorer) .....	124
<b>第5章 过程 .....</b>	<b>80</b>	6.4.3 属性窗口 .....	125
5.1 过程 .....	80	6.4.4 程序代码窗口 .....	126
5.2 常用内置函数与数学类 .....	82	6.4.5 如何将窗口叠在一起或分开 .....	127
5.2.1 VB 2005类型转换函数 .....	82	6.4.6 如何使用动态说明 .....	128
5.2.2 VB 2005常用数值函数 .....	84	6.5 控件的操作 .....	128
5.2.3 VB 2005常用日期函数 .....	85	6.5.1 建立控件 .....	128
5.2.4 VB 2005数学类 .....	86	6.5.2 如何选定控件 .....	129
5.3 过程的定义 .....	88	6.5.3 如何移动和调整控件 .....	129
5.3.1 如何定义Sub过程 .....	88	6.5.4 控件对齐 .....	129
5.3.2 如何调用Sub过程 .....	89		

6.5.5 如何删除控件.....	130	8.2.2 .NET Framework的内存配置.....	165
6.5.6 如何调整控件的前后顺序.....	130	8.3 多态.....	166
6.5.7 如何调整控件的定位顺序.....	130	8.3.1 重载.....	166
6.5.8 如何锁定控件.....	131	8.3.2 子类如何访问父类的方法或 属性.....	167
<b>第7章 对象与类.....</b>	<b>132</b>	8.3.3 动态绑定.....	169
7.1 前言.....	132	8.4 接口与实现.....	177
7.2 对象与类.....	133	8.5 Delegate委托类型.....	182
7.2.1 对象(Object).....	133	8.6 习题.....	184
7.2.2 类(Class).....	135	<b>第9章 异常处理.....</b>	<b>185</b>
7.3 面向对象程序设计的特性.....	136	9.1 Try...Catch...Finally语句.....	185
7.3.1 抽象化(Abstraction).....	136	9.2 异常类.....	186
7.3.2 封装(Encapsulation).....	141	9.2.1 常用异常类.....	186
7.3.3 继承(Inheritance).....	141	9.2.2 异常类常用成员.....	186
7.3.4 多态(Polymorphism).....	142	9.3 自定义异常处理:使用Throw语句.....	191
7.3.5 动态绑定(Dynamic binding).....	143	9.4 异常类继承.....	192
7.4 对象与类的建立.....	144	9.5 习题.....	195
7.4.1 如何建立类.....	144	<b>第10章 文件访问.....</b>	<b>196</b>
7.4.2 命名空间(Namespace).....	145	10.1 System.IO命名空间常用类介绍.....	196
7.4.3 如何建立属性.....	146	10.2 DirectoryInfo类.....	197
7.4.4 如何建立方法.....	151	10.3 FileInfo类.....	202
7.4.5 如何建立事件.....	156	10.4 文件读写.....	206
7.4.6 全局对象的事件定义.....	158	10.5 Path类.....	209
7.5 习题.....	160	10.6 习题.....	209
<b>第8章 继承、多态、接口.....</b>	<b>161</b>	<b>第11章 窗体设计与基础控件开发.....</b>	<b>210</b>
8.1 继承.....	161	11.1 窗体和控件.....	210
8.1.1 类继承.....	161	11.1.1 窗体和控件常用的成员.....	210
8.1.2 类成员的访问限制.....	163	11.1.2 窗体和屏幕的关系.....	213
8.2 共享成员.....	163		
8.2.1 共享成员的使用.....	163		

11.1.3 其他属性.....	214	12.5.1 NumericUpDown控件.....	258
11.1.4 窗体常用的事件过程.....	215	12.5.2 DomainUpDown控件.....	260
11.2 Label标签控件.....	218	12.6 滚动条工具.....	261
11.3 LinkLabel链接标签控件.....	221	12.6.1 VScrollBar&HScrollBar控件.....	261
11.4 ToolTip提示控件.....	222	12.6.2 TrackBar控件.....	264
11.4.1 ToolTip的使用方式.....	223	12.7 日期和时间工具.....	266
11.4.2 ToolTip常用成员.....	223	12.7.1 Timer定时器控件.....	266
11.5 Button按钮控件.....	225	12.7.2 MonthCalendar控件.....	269
11.5.1 Button常用成员.....	225	12.7.3 DateTimePicker控件.....	273
11.5.2 Button程序中属性的设置.....	226	12.8 图形处理工具.....	277
11.5.3 按钮如何与Enter键和ESC键结合.....	226	12.8.1 PictureBox控件.....	277
11.6 TextBox文本框控件.....	227	12.8.2 ImageList控件.....	282
11.7 RichTextBox富文本框控件.....	229	12.9 习题.....	285
11.8 InputBox输入函数.....	230	<b>第13章 工具栏容器.....</b>	<b>287</b>
11.9 MsgBox函数与MessageBox.Show方法.....	236	13.1 ToolStrip工具栏控件.....	287
11.9.1 MsgBox语法.....	236	13.1.1 ToolStrip常用成员.....	287
11.9.2 MessageBox.Show方法.....	237	13.1.2 建立工具栏控件的步骤.....	289
11.10 习题.....	241	13.2 MenuStrip菜单控件.....	297
<b>第12章 常用控件.....</b>	<b>242</b>	13.2.1 建立菜单控件.....	297
12.1 RadioButton单选按钮控件.....	242	13.2.2 菜单控件的常用成员.....	301
12.2 CheckBox复选框控件.....	244	13.3 ContextMenuStrip快捷菜单控件.....	306
12.3 容器工具.....	245	13.4 习题.....	309
12.3.1 GroupBox控件.....	245	<b>第14章 键盘、鼠标与共享事件.....</b>	<b>311</b>
12.3.2 Panel控件.....	249	14.1 键盘事件介绍.....	311
12.4 列表框工具.....	250	14.1.1 KeyPress事件.....	311
12.4.1 ListBox控件.....	250	14.1.2 KeyDown和KeyUp事件.....	316
12.4.2 CheckedListBox控件.....	253	14.2 鼠标事件介绍.....	320
12.4.3 ComboBox控件.....	254	14.2.1 Click与DoubleClick事件.....	321
12.5 旋转工具.....	258		



14.2.2	MouseDown、MouseUp与Click 事件.....	322	16.3	ADO .NET简介.....	356
14.3	控件共享事件过程.....	325	16.3.1	ADO .NET.....	356
14.3.1	使用Handles来建立共享事件 过程.....	325	16.3.2	.NET Data Provider.....	358
14.3.2	使用AddHandler及AddressOf新增 事件过程.....	326	16.3.3	DataSet.....	358
14.3.3	使用RemoveHandler及AddressOf 删除事件过程.....	327	16.4	.NET Data Provider.....	359
14.4	习题.....	334	16.5	使用数据工具建立数据库应用程序....	360
<b>第15章</b>	<b>通用对话框的应用.....</b>	<b>335</b>	16.6	数据库的关联查询.....	368
15.1	字体、颜色对话框.....	335	16.7	数据表记录的修改与浏览.....	377
15.1.1	字体对话框.....	335	16.8	BindingNavigator控件.....	386
15.1.2	颜色对话框.....	337	16.9	习题.....	388
15.2	文件对话框.....	340	<b>第17章</b>	<b>ADO .NET数据库联机与访问... 389</b>	
15.3	打印工具.....	344	17.1	如何引用ADO .NET命名空间.....	389
15.3.1	PrintDocument打印文件控件.....	345	17.2	使用Connection对象打开、关闭数据库 的连接.....	391
15.3.2	PageSetupDialog打印页面设置 对话框控件.....	346	17.3	使用DataReader对象读取数据表 记录.....	392
15.3.3	PrintPreviewDialog预览打印 对话框控件.....	348	17.3.1	Command对象.....	392
15.3.4	PrintDialog打印对话框控件.....	348	17.3.2	DataReader对象.....	393
15.4	习题.....	351	17.3.3	DataReader常用的属性与方法....	394
<b>第16章</b>	<b>数据工具的使用.....</b>	<b>353</b>	17.3.4	如何使用DataReader读取查询 结果.....	395
16.1	本章概述.....	353	17.4	使用DataSet对象读取数据表记录.....	399
16.2	数据库概念.....	354	17.4.1	DataSet对象.....	399
16.2.1	数据表.....	354	17.4.2	DataAdapter对象.....	399
16.2.2	数据表与数据库.....	355	17.4.3	如何使用DataAdapter对象.....	400
16.2.3	关系型数据库.....	355	17.4.4	如何使用DataTable对象取得 数据.....	404
			17.4.5	如何使用DataTable显示查询 结果的所有记录.....	405

17.5 SQL语言与Command对象 .....	408	19.3 习题 .....	452
17.5.1 Select查询命令 .....	408	<b>第20章 ASP .NET Web应用程序 .....</b>	<b>454</b>
17.5.2 使用SQL的注意事项 .....	409	20.1 ASP .NET Web应用程序 .....	454
17.5.3 Insert新建命令 .....	413	20.2 第一个ASP .NET Web 应用程序 .....	455
17.5.4 Delete删除命令 .....	413	20.3 ASP .NET数据库应用程序 .....	459
17.5.5 Update更新命令 .....	414	20.4 XML Web Service.....	469
17.5.6 如何使用Command对象及SQL 更新数据库 .....	414	20.4.1 XML Web Service简介 .....	469
17.5.7 如何使用具名参数及SQL语法 .....	418	20.4.2 如何建立Web Service.....	470
17.6 习题 .....	422	20.4.3 如何调用本机的Web Service.....	473
<b>第18章 ADO .NET数据库浏览与绑定... 423</b>		20.4.4 如何调用网络上的Web Service... 477	
18.1 DataView对象 .....	423	20.4.5 使用Web Service的限制.....	480
18.1.1 如何建立DataView对象.....	423	20.5 习题 .....	491
18.1.2 如何使用DataView对象来排序 数据 .....	424	<b>第21章 用户控件..... 493</b>	
18.1.3 如何使用DataView对象来过滤 数据 .....	424	21.1 建立窗口应用程序的用户控件 .....	493
18.2 数据绑定 .....	427	21.1.1 如何创建含有属性的用户控件 ... 493	
18.3 数据表的关联 .....	432	21.1.2 如何创建含有方法的用户控件 ... 499	
18.4 使用BindingManagerBase来修改数据 记录 .....	436	21.2 建立ASP .NET Web应用程序的用户 控件 .....	508
18.5 习题 .....	442	<b>第22章 Mobile Web应用程序 .....</b>	<b>516</b>
<b>第19章 读写XML文件 .....</b>	<b>443</b>	22.1 安装手机仿真器 .....	516
19.1 XML简介 .....	443	22.1.1 Mobile Web 应用程序简介 .....	516
19.2 读写XML文件 .....	446	22.1.2 下载手机仿真器 .....	517
19.2.1 如何读取XML文件中的记录 数据 .....	446	22.1.3 安装手机仿真器 .....	521
19.2.2 如何在XML文件中新增、删除、 修改记录数据 .....	447	22.1.4 测试手机仿真器 .....	525
		22.2 Mobile Web Form开发 .....	527
		22.3 多窗体界面切换 .....	530
		22.4 数据库连接 .....	534
		22.5 习题 .....	537

**第23章 Crystal Reports报表视图程序 .. 539**

23.1 Crystal Reports简介 .....	539
23.1.1 Crystal Reports功能概述 .....	539
23.1.2 Crystal Reports可导出的数据 格式 .....	540
23.1.3 Crystal Reports支持的数据源 .....	540
23.2 重新设置报表的数据源 .....	541
23.3 Windows Form的报表视图程序 .....	545
23.3.1 建立第一个Windows Form报表 视图程序 .....	545
23.3.2 设计报表的基本技巧 .....	551

23.3.3 如何设置报表的条件、排序、 分组 .....	559
23.4 图表、子报表的设计 .....	564
23.4.1 图表的设计 .....	564
23.4.2 子报表的设计 .....	571
23.5 Web Form的报表视图程序 .....	581
23.6 报表XML Web Service .....	587
23.6.1 如何建立报表XML Web Service .....	587
23.6.2 如何使用报表XML Web Service .....	589
23.7 习题 .....	591

# 第 1 章

## .NET Framework 介绍

---

- 1.1 Visual Studio 2005概述
- 1.2 软件开发技术的演进
- 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、软件技术进行集成，然而这样的集成往往费时耗力，更成为增加软件复杂度的祸首之一，因此为了简化软件规划与设计所需的成本，学术界与商业界便不断推出更为先进的软件设计技术与产品，以满足日趋复杂的软件开发需求。

在这一章中，我们将深入了解整个软件开发环境，包含规划、设计、集成等领域，从过去到现在的发展，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，当然最后我们会详细介绍未来的软件开发平台“.NET Framework”，看看微软的.NET愿景是如何通过 .NET Framework 来加以构造的，以及新一代Visual Studio 2005开发环境与Visual Basic 2005语言所能带来的软件开发优势。

除此之外，我们也将针对分布式软件开发技术，尤其是目前主流的XML Web Services技术，加以深入的剖析，在完成整个Internet软件开发技术的介绍之后，你将深刻了解到如何通过Visual Studio 2005软件开发工具，以及Visual Basic 2005程序语言，构造出以.NET Framework为坚实基础的软件组件与系统，再通过XML Web Services将软件系统延伸至Internet，最后集成Internet上不同平台上的不同系统，以及各类型的客户端设备，勾勒出 .NET 愿景所带来的无限商机。

## 1.2 软件开发技术的演进

一项新的软件开发技术或平台的推出，往往起因于原有的技术或平台已不足以应付目前软件开发的需求，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，有什么值得我们深入探讨的。

以下我们就分别以“软件分析”、“程序设计”、“分布式技术”与“软件平台”四个部分，说明过去和现在的软件开发技术演进。

### 1.2.1 软件分析的方法

在整个软件系统开发流程的早期，系统分析占着相当重要的一部分，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（OO，Object Oriented）技术，于是以往结构化、模块化的软件分析方法已经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年到1994年之间增加到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）协会所认可，正式成为主流的面向对象分析与设计的方法。

### 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以上的程序代码中，殚精竭虑地维持软件系统的可靠度与稳定性，然而传统单纯的结构化程序设计技术在今日却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，面向对象算是近代软件系统分析、规划与设计概念的一大突破，也无疑是大型软件系统的救世主。

在OO的领域中，除了先前所提到的OOA与OOD之外，另一项与程序设计密不可分的技術就是OOP（Object Oriented Programming，面向对象程序设计），所谓OOP指的就是“使用面向对象的概念来编写程序”，是的，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非常面向对象的程序，例如汇编语言（不要怀疑，笔者就曾经写过），但经验不足的程序设计师也有可能使用诸如C++之类支持面向对象语法的程序语言，却编写出一点都不像面向对象的程序。

笔者经常被学生问：如何将手边长度不到十行的程序代码改用面向对象重写？譬如：如何将原本使用For循环编写出来的九九表程序，改由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：该如何将系统中所有的程序改成面向对象？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淘汰原本的程序设计方法,相反的,面向对象的目的在于克服传统结构化程序设计中对于大型或复杂系统难以设计与维护的缺点,于是“如何将原本使用For循环所编写出来的九九表程序,改由面向对象的语法重新编写”这样的要求,显然就太过吹毛求疵了。

曾经听过这么一个故事,有一个拥有十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十份营收报表之后,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式,在会议中有人建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,将报表上十个分公司的营收金额相加后,不到30s就算出该月份全公司的营收总额。

是的,“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法则,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,可以分成以下几种不同的层次。

#### 1. 展示 (Presentation)

在软件系统的 Presentation层次中,主要负责进行用户界面的处理,包含数据的输入与显示、图表的绘制等与数据表现有关的工作。

#### 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、转帐、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,甚至是 XML 数据都可以称作是数据源。

以软件系统的体系结构(以上三个层次的组合方式)来看,大致上可以分成以下几个类型。

#### 1. 单机体系结构 (Stand Alone)

在单机体系结构中,展示、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集成,因此在设计上非常单纯,在单机上的效率也不差,适合单人操作使用,然而由于数据是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达成,非常不方便。

假设银行的系统是采用单机的体系结构开发,那么今天在台北某一分行存款10 000元,

这个存款数据只会存放在该办理人员所用的计算机上，必须将该笔新的存款数据复制到磁盘上，然后复制到全省其他分行的所有职员使用的计算机中，这样才能在其他分行中看到这笔新的存款数据，如果还要进行复杂的转帐交易，那么单机体系结构就明显力不从心了。

## 2. 主从式体系结构 (Client/Server)

为了解决单机体系结构数据源分散、难以同步的问题，我们可以将数据源独立出来，如图1-1所示。

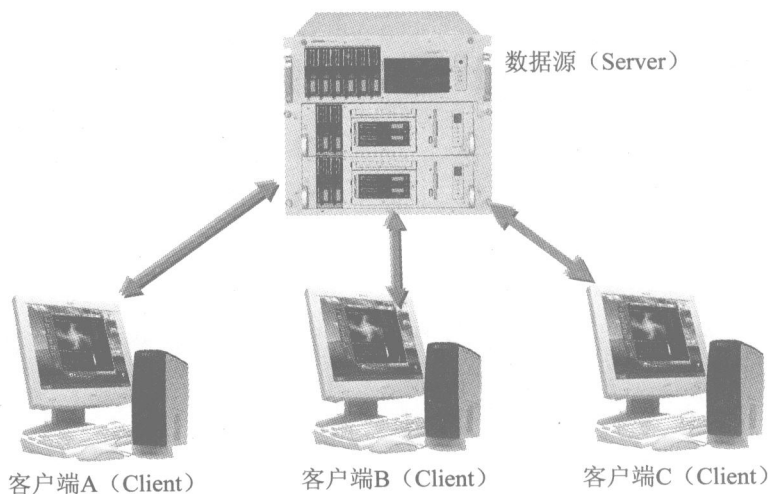


图 1-1

于是不论是使用哪一个客户端的系统，其数据源都是来自同一个位置，因此客户端A所存储的数据，客户端B立即就可以存取得到，省去许多数据同步的问题，而这个数据源通常是数据库管理系统，例如Microsoft SQL Server。

除了后端的数据源独立出来之外，客户端当然得扮演展示层的角色，然而商业逻辑在这个体系结构中该放在哪里呢？

### Heavy Client (肥客户)

将商业逻辑直接用程序语言编写好与展示层一同放在客户端上，称之为Heavy Client，由于商业逻辑需要大量的运算，因此放在客户端上会造成客户端执行效率下降，再加上一旦商业逻辑有更动，就必须重新将新版的程序安装在所有的客户端上，在软件安装部署与管理上非常耗时耗力。

### Thin Client (瘦客户)

将商业逻辑与数据源放置在一起，客户端单纯做展示层工作，称之为Thin Client，在这个体系结构中由于商业逻辑单独放在数据源所在计算机上，因此在修改商业逻辑时就比Heavy Client要方便许多，缺点则是会降低数据源端的执行效率，放置在数据源端的商业逻辑通常使用存储过程 (Stored Procedure) 的方式来处理，一般中小企业都采取这样的软件

体系结构。存储过程是SQL Server上可供调用执行的Server端程序。

### 3. 三层式体系结构 (3-Tiers)

不论将商业逻辑放置在数据源端，还是放置在客户端，两者都有其优缺点，因此如果将商业逻辑独立出来，形成三层式体系结构（如图1-2所示），就可以兼具两者的优点，避免缺点。

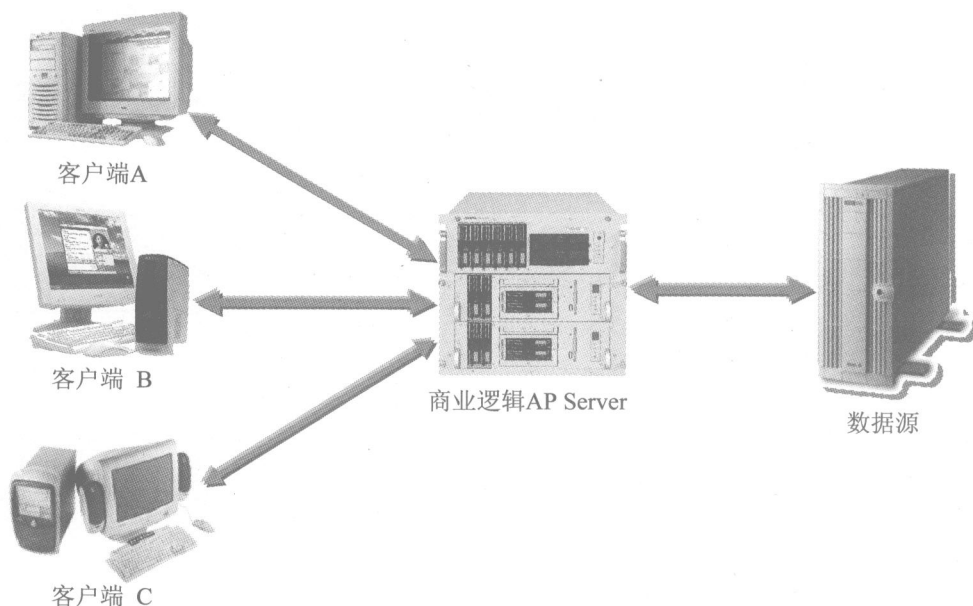


图 1-2

由于将商业逻辑独立出来，因此不论是在商业逻辑的更新与维护上，还是对于客户端与数据源端的执行效率上都有莫大的帮助，其中负责处理商业逻辑的那台Server，也就是业界俗称的AP Server。

### 4. 多层式体系结构 (N-Tiers)

三层式体系结构中客户端程序一旦修改，例如更改操作界面，就必须将新版客户端程序一一安装到所有的客户端计算机上，在客户端维护上仍然非常不方便，另外商业逻辑层由于需要大量运算资源，因此往往会造成整体系统的瓶颈，因此在多层次体系结构中（如图1-3所示），我们可以将商业逻辑层再加以细分，分层负责各类型运算处理，例如转帐交易中会使用到存款与取款的处理程序，我们就可以将转帐与存取款处理程序分别放在两部计算机上，将运算需求分散，以此提升整体商业逻辑的执行效率。

另外，再配合使用ASP、PHP等网站服务器（例如 IIS）端程序绘制客户端操作界面与处理客户端输入数据，这样客户端只要有浏览器就可以使用系统，将来如果需要修改用户界面，也只要修改ASP程序，客户端只要单击浏览器上的刷新按钮即可得到最新的用户界面了。



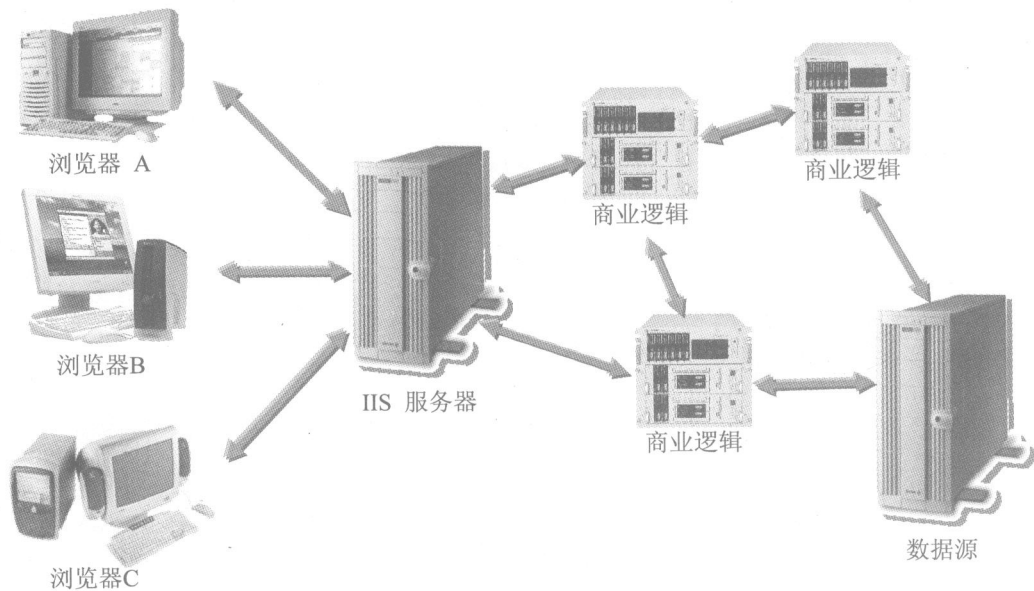


图 1-3

## 5. Windows DNA 2000

微软在实现N-Tiers体系结构上，提供了Windows DNA2000体系结构（前身是Windows DNA），在这个体系结构中，使用COM+（Component Object Model，组件对象模型）作为组件设计规范，配合Windows 2000上的组件服务，构造出多层次体系结构中所需要的商业逻辑组件，后端配合Microsoft SQL Server，前端可以使用Windows应用程序，或者使用ASP开发的Web应用程序。

在这个体系结构中，足以集成所有以微软为主的解决方案，但是对于非微软的系统，例如IBM、Java等，就产生不易集成的缺点，原因在于COM+ 通过 DCOM(Distributed COM)通信协议进行数据传输，再加上必须支持COM+ 规范接口的存取才能调用COM+ 组件，其他系统未必支持。

## 6. .NET

.NET在分布式系统支持上，强调使用XML（eXtensible Markup Language，可扩展标记语言）做为传输的标准数据格式，并且使用Internet上标准的HTTP与SOAP（Simple Object Access Protocol，简单对象访问协议）通信协议来传输 XML 数据（如图1-4所示），因此几乎所有可以连上Internet的系统都支持，十分适合用在异构系统的集成上。

在组件规范上，使用标准的XML Web Service规范将组件Web化，XML Web Service可以看做一种没有用户界面的网站，用来提供客户端所需的商业逻辑服务，客户端只需要支持XML、HTTP、SOAP，就可以和XML Web Service沟通，不限任何操作系统平台、程序语言、硬件设备，如图1-5所示。