

附赠光盘



SAMS

计算机技术

译林

精选系列

Visual Basic

6 面向对象程序设计

〔美〕 Deborah Kurata

著

潇湘工作室 译

人民邮电出版社



计算机技术译林精选系列
Visual Basic 6 面向对象程序设计

- ◆ 著 (美) Deborah Kurata
 - 译 潇湘工作室
 - 责任编辑 俞 彬
 - ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号
北京顺义振华印刷厂印刷
新华书店总店北京发行所经销
 - ◆ 开本: 787 × 1092 1/16
印张: 30.25
字数: 744 千字 1999 年 7 月第 1 版
印数: 1 - 5 000 册 1999 年 7 月北京第 1 次印刷
著作权合同登记 图字: 01 - 99 - 0433 号
ISBN 7-115-07904-8/TP·1169
-

定价: 60.00 元

本书以一个业务联系管理系统的软件设计与构造作为主线，介绍利用面向对象的编程方法开发一个软件系统的全过程。从开始设计，到中间的开发、测试、文档编制，最终到如何实现这个系统。本书分为三部分：第一部分介绍面向对象编程的基本概念以及支持这些概念的 Visual Basic 特性；第二部分介绍面向对象设计的方法学。它以实例说明软件设计的每个步骤；第三部分介绍使用 Visual Basic 6 如何具体实现一个业务联系管理系统，这里不仅提出编程中考虑的因素，还有丰富的代码实例，可以让读者马上就能切身体会到面向对象编程的好处。

本书不仅适用使用 Visual Basic 6 的程序员，还适于对面向对象程序设计感兴趣的各个水平的人员及大专院校计算机专业师生。

版权声明

Deborah Kurata:Doing Objects in Visual Basic 6

Authorized translation from the English language edition published by SAMS Publishing.

Copyright © 1999 by SAMS Publishing.

All rights reserved. For sale in Mainland China only.

本书中文简体字版由人民邮电出版社和美国 SAMS 出版公司合作出版，未经出版者书面许可，本书的任何部分不得以任何方式复制或抄袭。

版权所有，翻印必究。

作为有多年工作经验的专业软件人员，在工作和学习的过程中，我们编写过许多程序，也为了进行软件维护和改造看过很多程序。说到对此的体会，简直是一言难尽。

程序代码有两种极端情况：一种是具有软件工程思想的人（多数是科班出身）编写的，条理清晰，注释适当，看这种代码简直是一种享受；另一种（多数是“半路出家”的人编写的）是代码从头到尾没有一句注释，而且用 goto 语句转来转去，把你转得晕头转向，就像是看汇编语言一样（当然比那要好些）。对于这样的代码，在那些百思不解的地方，我们曾经问过编写它的人这是什么意思，结果往往是他在看了半天后告诉你：

“我也不知道”。

说实话，每次看这种代码，我们都会感觉自己老了几岁。

当然，程序员都是些聪明而勤奋的人，他们运用自己的智慧让机器做许多事情。但是，编程就是编写代码吗？

本书以一个业务联系管理系统的软件设计与构造作为主线，介绍利用面向对象的方法开发一个软件系统的全过程。从开始设计，到中间的开发、测试、文档编制，最终到如何实现这个系统。本书大致分为三部分：

- 第一部分介绍面向对象编程的基本概念以及支持这些概念的 Visual Basic 特性。
- 第二部分介绍面向对象设计的方法学。它以实例说明软件设计的每个步骤。
- 第三部分介绍使用 Visual Basic 6 如何具体实现一个业务联系管理系统，这里不仅有编程中的考虑因素，还有丰富的代码实例，可以让你马上就能切身体会到面向对象编程的好处。

在翻译本书的过程中，一个译者（他曾经开发过许多软件系统，但却是砍班出身）感叹，“这本书太好了！许多话让我茅塞顿开”。

通过本书的学习，你便可以使用 Visual Basic 6 以面向对象的方式创建一个出色的软件系统，你创建的系统从内到外，都将出类拔萃，高人一筹。

何乐而不为？

本书的翻译工作由潇湘工作室承担，潇湘工作室由清华大学的莘莘学子以及从事计算机开发与教学的科技工作者组成，曾经翻译和编写过大量的计算机科技书。参加本书翻译工作的有贺军、贺民、王晓娟、龚亚平、刘传凯、肖展业、张宝银、陈代川等，另外，肖志钢、吴少波、李伟、陈武、黄素芳、朱明、王雷等人也做了大量的工作。为了保证翻译的质量和前后术语使用的一致，由贺军对全书统稿。

潇湘工作室

1999 年 3 月

从把工作代码汇集到一起，到构造完整的应用程序，其间还有着很大的距离；然而却很少有人讨论如何走过这段路程。当我第一次想写一本有关 Visual Basic(VB)的书时，我就想要研究这个问题。我想要给 Visual Basic 程序员提供实用的方法去设计他们应用程序，我希望他们能够事先做好计划，而不要到事后去追悔莫及。我想要提供给用户开发结构良好的应用程序的专门技术，在这些应用程序里，你可以找到你上个月写的代码，丝毫不会影响其他代码就可以修改它。这就是本书的宗旨所在。

看一下在公司或在家里使用的应用程序，你将发现其中一些比另外一些要好得多。最好的是那些看起来漂亮，使用简单，而且容易领会其含义的程序。如果你有机会看看幕后的内容，并了解这些应用程序的内部，那么你也将发现，一些应用程序看起来比其他的要好得多。最好的是那些容易修改，容易使用，且容易让人领会的程序。

我们如何才能使所有的程序在内部和外部看上去都不错？怎样才能为 Visual Basic 应用程序设计出较好的总体结构和构造？当我发现 Visual Basic 的面向对象特性后，我就知道自己已经找到了答案。

一、使用对象的原因

现在，Visual Basic 所处的语言生命期的阶段类似于几年前当 C++ 出现时 C 所处的情况。C 程序员看 C++ 中的面向对象特性而感到奇怪，“什么是对象？什么时候应该使用它们？为什么使用对象？”。

简单地说，对象就是应用程序里所涉及的东西，诸如时间跟踪系统里的雇员和时间表，购物系统里的顾客和发票，或者棒球应用程序里的队员、球队和比赛。

至于何时应该使用对象的问题，我提议，即使你对它持怀疑态度，也应该永远使用它们。任何应用程序的每个部分都利用对象来做。业务对象，比如雇员、时间表、顾客等等，都可以用对

象来实现。实现细节，比如提供编辑菜单选项、把数据保存到文件等等，也都可以以对象来实现。你要怎样使用对象是个人的选择。然而，我期待着当你用 Visual Basic 的面向对象特性时，你将被它所吸引，而你也将会永远使用它。

用 Visual Basic 使用对象有许多益处，例如：

- 对象帮助你依据现实世界的东西来思索应用程序，这有助于设计过程。
- 对象在一个编程单元内定义其全部的数据和对数据的处理。这使得查找代码以及管理软件开发的复杂性变得更加简单。它们也可以简化多个程序员的开发过程。
- 对象预包装功能可以很容易地在整个企业中重用。例如，你可以开发一个对象来计算公司定价模型，并在任何需要这个函数的应用程序中重用它，包括在 Excel 中！
- 对象使你可以沉浸在使用新技术的兴奋之中。使用对象，可以建立 ActiveX EXE 和 ActiveX DLL，以便在预编译的组件里封装功能。你可以建立自己的 ActiveX 控件；可以用 HTML 应用程序或者 IIS 应用程序来建立 Web 应用程序；可以使用新的数据访问特性，诸如 Data Environment（数据环境）设计器和支持数据的类。

本书阐明且详细介绍这些的目的是帮助你了解什么时候、怎样及为什么使用对象。

二、本书的读者

本书的前两部分介绍面向对象概念，并对设计面向对象的应用程序进行深入的论述。这些信息适于各个水平的 Visual Basic 开发人员。它同样适于管理人员、设计人员、技术人员、测试人员以及对面向对象设计的概况有兴趣的其他任何人。

本书的第三部分阐述使用 Visual Basic 实现面向对象应用程序的过程。提供了详尽的讨论和代码示例。本书这一部分假定你熟悉 Visual Basic，即了解如何创建窗体和编写代码。如果你在以前从来没有用过 Visual Basic，你将需要首先读一些介绍文章，然后再读本书的第三部分。

三、本书的内容

本书组织成如下三部分。

1. 第一部分：基础知识

第一部分的各章提供面向对象设计的基础知识，讲述了面向对象基本概念以及支持这些概念的 Visual Basic 特性。该基础包括了对软件开发过程的总结。它给出了准备面向对象设计，包括建立应用程序的目标和需要的有关知识。

2. 第二部分：设计框架

本书的第二部分介绍面向对象设计（OOD）的 GUIDS Methodology（方法学）。它通过实例说明，使你可以掌握设计的每个步骤。这部分内容如下：

- 以目标为中心的设计。本部分集中于应用程序目标来定义应用程序所涉及的对象，提供一些技术来帮助你定义对象、它们的属性、方法及在应用程序中使用它们的场合。
- 用户界面设计。本部分讲述设计应用程序的用户界面的过程，给出了一些提示来帮助开发良好的用户界面。
- 以实现为中心的设计。这里介绍如何实现应用程序，叙述开发一个精致的对象模

型的过程。在讨论体系结构如何影响设计的同时，提出了三层体系结构。其中包括了对数据访问技术、数据联编选项和应用程序层之间的数据传递技术的详尽讨论。

- 数据设计。这里介绍应用程序中的数据，讲述了对数据分类以及应用程序如何定义存储数据的技术，还提供了设计关系数据库的一般介绍。

- 构造的策略。本部分提出了开始构造前要考虑的策略，讨论了诸如购买或者建立编码标准以及测试计划等策略。

当学完这部分时，你就了解了 OOD 的基础知识。

3. 第三部分：构造应用程序

第三部分详细讲述了如何运用 Visual Basic 的面向对象特性。

- 第十章讲述构造类模型及从类创建对象的方法，内容包括如何在类里定义常量和事件以及如何处理错误。此外，还回答了 12 个有关类的常问问题。

- 第十一章描述了一些高级的特性，如使用选择和使用类持续性把对象保存成一个文件。它详述了在一个窗体内如何给类下定义。然后提供了对程序调试的建议，并回答常见的有关类的 10 个技术问题。

- 第十二章叙述了为类如何定义和实现多个接口。通过在几个类里实现一个接口，可以平衡多态性来简化代码。尽管 Visual Basic 不提供实现继承，但是可以用接口继承和代表来完成实现继承。

- 第十三章提供对 ActiveX 的讨论。它详述了如何构造 ActiveX EXE, ActiveX DLLs, 和 ActiveX 控件。提供了对 HTML 应用程序, IIS 应用程序(web-类)和 ActiveX 文档的讨论。这章也给出了在创建 ActiveX 组件时要做什么和不做什么。

- 第十四章叙述通用的数据访问并详细地论述了如何运用 Visual Basic 里所提供的数据库工具。它演示了如何使用 Data Environment 设计器建构数据库应用程序、ADO Data 控件、ActiveX 数据对象 (ADO) 以及知道数据的类。而后详述了构造三级面向对象的数据库应用程序的过程。

- 第十五章描述如何处理应用程序的细节。这包括使用资源文件、对 Registry 进行设置、把事件登记到 NT 事件日志簿或者任何日志文件、使用 Windows API、创建弹出菜单、创建 What 's This 帮助，及增加应用程序的图标。它也提供由前面章节里讨论的从组件构造大系统的概述。

每章提供大量的代码示例来演示相应的技术。这些代码例子全部收录在本书配套光盘中，附录 A 里讨论了这些示例。

有关 Visual Basic 6 的信息和对本书里材料的任何更正都可以在 InStep Technologies Web 网站 <http://wwwiv.insteptech.com> 地址找到。

四、本书的使用方法

你可以在这本书中快速找到 Visual Basic 6 里可用的新特性。或者，你能通过这本书来了解面向对象概念以及怎样把它们运用于 Visual Basic。还可以把这本书作为指导老师，看一遍全部的例子和示例应用程序。本书可以作为 Visual Basic 里的面向对象特性的参考书。你可以按最适合自己需要的方式使用这本书。

我已经尽一切力量以清楚而简明的方式来叙述本书中所提出的概念，并用 Visual Basic (Enterprise Edition)在 Windows NT 下测试了代码例子。如果你对改进本书的内容有建议或者发现不正确的或者不清楚的地方，希望你给我写信。那时,我会将你的说明合并到本书的将来版本。你可以给我发电子邮件，地址为 deborahk@insteptech.com。

第一部分 基础知识

第一章 VB 中的面向对象基础	3
1.1 什么是 OO	3
1.2 面向对象概念介绍	4
1.2.1 什么是对象	4
1.2.2 什么是类	5
1.2.3 什么是方案	7
1.2.4 对象的相关性	7
1.3 面向对象系统的基本元素	9
1.3.1 抽象: 集中要点	9
1.3.2 封装: 隐藏私有部分	10
1.3.3 继承: 获得重用性	10
1.3.4 多态性: 相同的行为不同的实现	11
1.4 Visual Basic 的 OO 特性	12
1.4.1 在类模块中定义类	12
1.4.2 用属性过程提供公共访问	12
1.4.3 定义多态方法	12
1.4.4 生成事件	12
1.4.5 封装类常量	12
1.4.6 用 Object Browser 查看类	13
1.4.7 创建对象	13
1.4.8 使用窗体作为对象	13
1.4.9 使用对象组	13
1.4.10 定义多接口	13
1.4.11 用 ActiveX 组件公布对象	14
1.4.12 开发数据库应用程序	14
1.4.13 开发基于 Web 的应用程序	14
1.4.14 创建三层应用程序	14

1.5	Visual Basic 6 新特性	15
1.5.1	数据工具: 访问数据更容易	15
1.5.2	Web 开发: 用 VB 创建 Web 应用程序	16
1.5.3	新的面向对象特性: 使用 OO 的更多理由	16
1.5.4	IDE 新特性: 优秀的工具	17
1.5.5	其他新特性: 增强的分类	18
1.6	小结	18
1.7	补充读物	19
1.8	思考题	20
第二章	软件开发过程	21
2.1	软件开发过程概述	21
2.1.1	从想法开始	23
2.1.2	建立需求	23
2.1.3	计划和安排工程	23
2.1.4	确定总体结构	24
2.1.5	开始建造	24
2.1.6	开始检查	25
2.1.7	维护应用程序	25
2.2	面向对象方法的益处	25
2.2.1	语言相同	25
2.2.2	设计现实世界的模型	26
2.2.3	估计时间	26
2.2.4	调整设计时间	27
2.2.5	使用户正确认识原型	27
2.2.6	管理编程的复杂性	27
2.2.7	简化多程序员开发	28
2.2.8	适应变化	28
2.2.9	管理特别订单	28
2.2.10	准备临时交付	29
2.2.11	度量进度	29
2.2.12	管理维护	29
2.2.13	重用组件	29
2.3	业务联系管理系统实例研究	30
2.4	小结	30
2.5	补充读物	31
2.6	思考题	32
第三章	设计的必备条件	33

3.1	从想法开始.....	33
3.1.1	想法从何而来	33
3.1.2	明确陈述想法并评估想法	35
3.2	确定需求.....	36
3.2.1	定义以目标为中心的需求	37
3.2.2	创建工程组	38
3.2.3	核心组的作用	39
3.2.4	准备工作	39
3.2.4	陈述目标	40
3.2.5	设置范围	41
3.2.6	确定需要	42
3.2.7	需要转换为需求	44
3.2.8	区分需求的主次	45
3.3	计划和安排工程.....	46
3.3.1	以目标为中心的计划	46
3.3.2	工程时间安排	47
3.2.3	重点集中目标	49
3.2.4	工程计划和时间安排的 10 个不正确说法	50
3.4	小结.....	54
3.5	补充读物.....	55
3.6	思考题.....	56
 第二部分 设计框架		
 第四章 面向对象总体结构 的 GUIDS 方法		
59		
4.1	GUIDS 方法学	59
4.1.1	以目标为中心的设计	61
4.1.2	用户界面设计	61
4.1.3	以实现为中心的设计	61
4.1.4	数据设计	62
4.1.5	构造的策略	62
4.1.6	建模合并到过程中	62
4.2	小结.....	63
4.3	思考题.....	63
 第五章 面向目标的设计.....		
65		
5.1	什么是以目标为中心的设计	65

5.2 准备以目标为中心的设计	67
5.2.1 确定对象	67
5.2.2 定义行为	67
5.2.3 列出属性	68
5.3 描述对象.....	68
5.3.1 带着目标开始	69
5.3.2 为对象编制文档	69
5.3.3 检验需求	70
5.4 建立关系模型.....	73
5.4.1 容器：“有一个”关系	74
5.4.2 子类：“是一个”关系	75
5.4.3 合作：“使用一个”关系	75
5.5 使用方案验证对象.....	76
5.6 编制以目标为中心的设计文档	77
5.7 小结.....	77
5.8 补充读物.....	78
5.9 思考题.....	79
第六章 用户界面设计.....	81
6.1 什么是“好”的用户界面设计.....	81
6.2 创建意图明确的设计.....	83
6.2.1 确定用户目标	83
6.2.2 定义特性	84
6.2.3 提供思维模型	85
6.2.4 布置特性	85
6.2.5 选择界面风格	87
6.2.6 布置屏幕	90
6.2.7 添加功能	90
6.3 掌握用户界面的基本原理	91
6.3.1 保持简单	91
6.3.2 提供答案，而不是疑问	91
6.3.3 富有灵活性	92
6.3.4 应当宽容	92
6.3.5 善于记忆	93
6.3.6 信任用户	94
6.3.7 适应环境	94
6.3.8 直接开始	94
6.4 验证用户界面设计.....	95
6.4.1 验证特性	95

6.4.2 复审业务规则	95
6.4.3 创建原型	96
6.4.4 验证可用性	97
6.5 用户界面设计文档编制	97
6.6 小结	97
6.7 补充读物	98
6.8 思考题	99
第七章 以实现为中心的设计	101
7.1 什么是以实现为中心的设计	101
7.2 以实现为中心的设计准备	103
7.2.1 单一的结构	103
7.2.2 文件服务器结构	104
7.2.3 双层客户机/服务器结构	104
7.2.4 三层客户机/服务器结构	105
7.3 定义系统结构	106
7.3.1 物理结构	107
7.3.2 逻辑结构	107
7.4 选择数据访问结构	108
7.4.1 数据访问选择	108
7.4.2 选择数据传送机制	111
7.4.3 选择数据访问机制	112
7.5 设计组件	114
7.5.1 用户界面层组件	115
7.5.2 服务器层组件	116
7.5.3 数据层	123
7.5.4 附加组件	123
7.6 设计类	126
7.6.1 定义属性	126
7.6.2 定义方法	128
7.6.3 定义接口	128
7.6.4 定义事件	129
7.7 考虑实现问题	130
7.7.1 性能需求	130
7.7.2 资源限制	130
7.7.3 安全需求	130
7.7.4 使用状态转换图设计动态状态	131
7.7.5 评价现有设计模式	134
7.8 验证以实现为中心的设计	134

7.8.1 再访设计方案	135
7.8.2 保证完成合作	135
7.8.3 创建原型	135
7.9 以实现为中心的设计文档编制	135
7.10 小结	135
7.11 补充读物	136
7.12 思考题	138
第八章 数据设计	139
8.1 什么是数据设计	139
8.2 数据的完全分类	140
8.3 在数据文件中有效存储数据	141
8.3.1 什么时候使用数据文件	141
8.3.2 与数据文件接口	142
8.4 灵活使用数据库	142
8.4.1 何时使用数据库	143
8.4.2 使用数据库接口	143
8.5 在注册表注册配置数据	144
8.5.1 何时使用注册表	144
8.5.2 与注册表接口	144
8.6 永久的和临时的内部数据	145
8.6.1 与内部数据的接口	145
8.7 设计关系数据库	145
8.7.1 定义表	146
8.7.2 定义主关键字	146
8.7.3 规范化过程	147
8.7.4 调整性能	148
8.8 公共数据访问结构	148
8.8.1 本地数据配置	148
8.8.2 网络数据配置	149
8.8.3 数据服务器配置	149
8.8.4 数据瞬态配置	150
8.8.5 其他考虑因素	151
8.9 数据设计文档编制	151
8.10 小结	152
8.11 补充读物	152
8.12 思考题	153
第九章 建造的策略	155

9.1 什么是策略.....	155
9.2 购买与创建.....	157
9.2.1 有什么外部控件	157
9.2.2 需要什么样的控件	158
9.2.3 灵活性如何	158
9.2.4 什么是风险	158
9.2.5 使用开发工具提高效率	159
9.3 编码标准.....	159
9.3.1 为什么需要编码标准	159
9.3.2 有几种标准	160
9.3.3 注释清晰	160
9.3.4 代码排列结构	162
9.3.5 将变量的作用域减到最小	165
9.3.6 设置句法标准	166
9.3.7 错误处理	168
9.4 命名约定.....	170
9.4.1 好的命名和坏的命名	170
9.4.2 对象命名	171
9.4.3 菜单命名	173
9.4.4 模块命名	173
9.4.5 例程命名	173
9.4.6 变量命名	174
9.4.7 常量命名	175
9.5 配置管理/源代码控件	175
9.5.1 检验时间	175
9.5.2 管理改变请求	176
9.6 检查和测试过程.....	177
9.6.1 编写无错代码	177
9.6.2 在离开时测试	177
9.6.3 集成测试	178
9.6.4 测试整个系统	179
9.7 文档资料和帮助系统.....	179
9.7.1 定义文档资料	179
9.7.2 设计帮助系统	179
9.8 实现计划和时间表.....	179
9.9 小结.....	180
9.10 补充读物.....	180
9.11 思考题.....	181