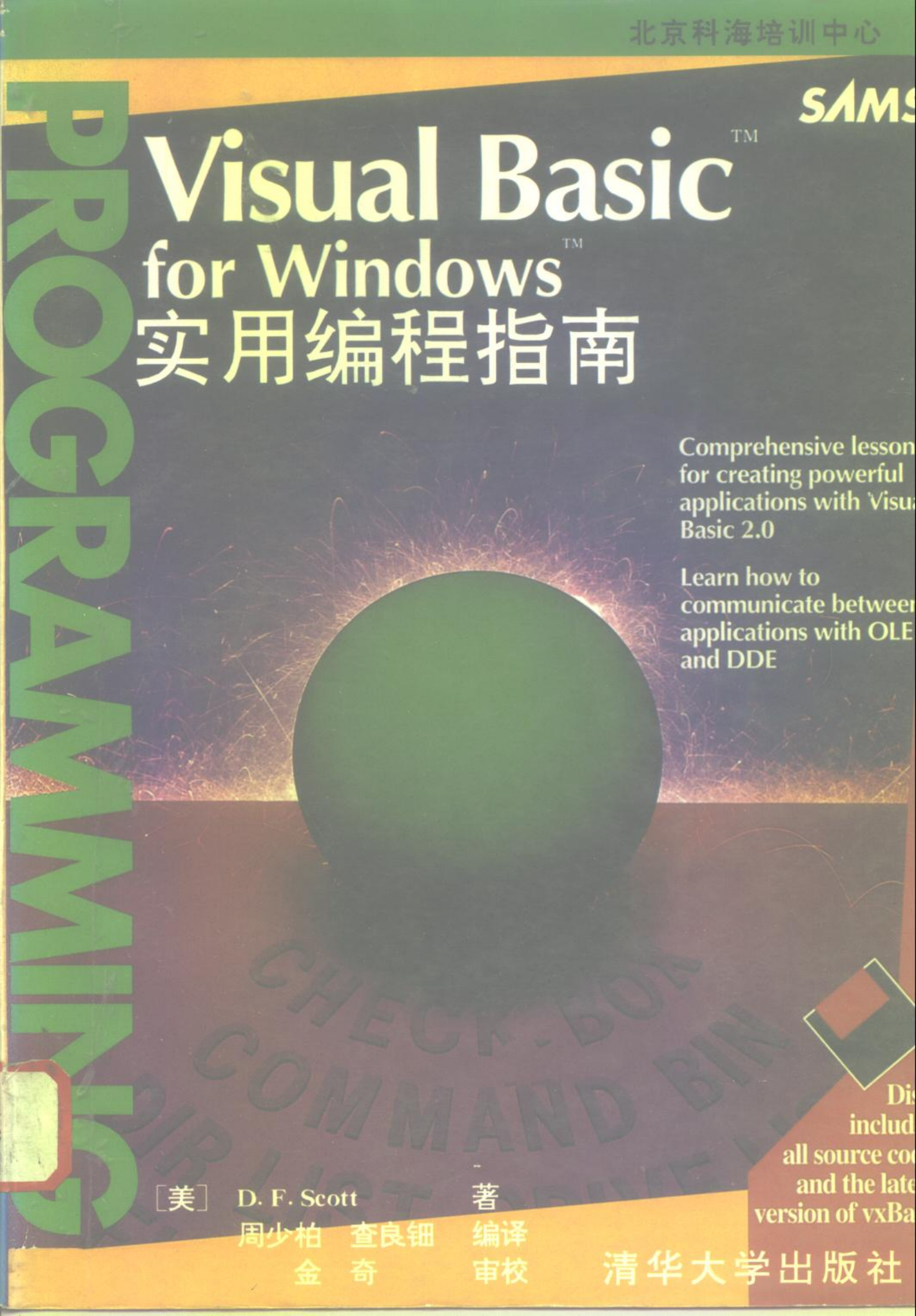
SAMS

PROGRAMMING

Visual Basic™ for Windows™ 实用编程指南

Comprehensive lesson
for creating powerful
applications with Visu
Basic 2.0

Learn how to
communicate betwee
applications with OLE
and DDE



Dis
includ
all source co
and the late
version of vxBa

[美] D. F. Scott 著
周少柏 查良钿 编译
金 奇 审校

清华大学出版社

SAMS 北京科海培训中心

Visual Basic™ for Windows™

实用编程指南

[美] D. F. Scott 著

周少柏 查良钿 编译

金 奇 审校

清华大学出版社

(京)新登字 158 号

Visual Basic™ for Windows™ 实用编程指南

Visual Basic™ for Windows™ Developer's Guide

First Edition

D. F. Scott

图书在版编目(CIP)数据

Visual Basic™ for Windows™ 实用编程指南 / (美) Scortt 著；周少柏，查良钿编译. —北京：清华大学出版社，1994.7

ISBN 7-302-01607-0

Ⅰ.V… Ⅱ:①S… ②周… ③查… Ⅲ.Basic 语言-程序设计-手册 Ⅳ.TP312BA

中国版本图书馆 CIP 数据核字(94)第 08849 号

出版者：清华大学出版社(北京清华大学校内，邮编 100084)

印刷者：北京科普胶印厂

发行者：新华书店总店北京科技发行所

开　本：787×1092　1/16　　印　张：19.25　字　数：468 千字

版　次：1994 年 7 月第 1 版　　1994 年 7 月第 1 次印刷

本社分类号：TP312BA

印　数：0001—8000

定　价：40.00 元

引 言

程序员是理想主义者，他们全心地相信：办公室、商业场所、实验室、甚至在思维的抽象舞台上，都能用数字逻辑语言解释所做的一切。从逻辑上讲，这也许不会都是真的；整个经济和科学社会不太可能全部由微处理器自动化。然而，每一个在程序设计方面出现的新概念，都将使人们向理想更靠近一步，并坚信它最终一定能实现。

在我们当前的生活中显然已越来越离不开微处理器。用代码语言为计算机编程的概念很早就出现了。向市场销售的微处理器依赖于多种语言（主要是 BASIC，还有 Pascal 等），它们都是处理指令的主要机制。80 年代人们在计算中更多的不是使用语言而是使用组装的应用程序（象电子表格或字处理程序等），这些软件也包含某种在后台运行的“宏”语言，但它们大部分是采用揿按钮的方法，因此新的用户可能产生这样的印象，认为似乎掌握计算的技艺只不过要求简单地懂得什么时候揿压按钮。

Microsoft 公司新开发的 Visual Basic 2.0 for Windows Professional Edition 是功能较强的语言版本，它提供如下重要的新特色：

- 新实现的 Visual Basic 语言允许声明引用特定对象的变量。这些对象变量具有图形对象的属性。然而，在任何时候，可根据需要设置它们。于是，可以取任意五个滚动条并对齐它们，或把它们的值置为 0，或在窗体中安排若干正文框的内容而无需用名字引用。
- 新的观察调试系统可在程序运行时监控变量的状态。解释程序可对监控变量进行复杂运算，而不影响应用程序的执行过程。
- 新的网格控件（现在是所有 Visual Basic 版本的特征）。允许数据以表格形式显示，它是对 VB 常用的标准窗体式样的补充。
- 使用新的对象链接与嵌入（Object Linking and Embedding）客户控件，可在 Visual Basic 窗体中按原来格式显示 Windows 应用程序的数据文档，并用 Windows 应用程序代替 VB 应用程序执行文档维护。
- 向 Microsoft SQL 服务器的用户提供 Microsoft 的开放式数据库互联（Open DataBase Connectivity）模型的原型。
- 通过一组 VB 图形对象的内部属性，将这些图形对象与联机帮助文件的各个主题关联，Visual Basic 窗体现在可以直接和 Windows 联机帮助系统相连。对于 Professional Edition，可以用字处理程序编辑这个帮助文件，并用 Microsoft 的 Help Compiler 把它编译在 VB 解释程序能识别的窗体中。

本书讨论应用程序的原理和编程，展示程序设计过程的最终产品以及“原始”产品，还包括为达到最终目标所遇到过的错误和失败。采用科学的编程方法能使程序更精练和更有效。这里提供丰富的实用程序例子，包括涉及“人工智能”的弈棋程序，全书共分十三章。

第一章“使用语言进行计算”首先说明使用应用软件和直接使用高级程序设计语言之间的差异，同时提供许多对编程有用的信息。由于某些已有的软件包未能适应办公环境，所以

不少人开始购置 Visual Basic。既然他们能忍受堆积如山的文档所耗费的时间，他们显然有能力开发办公自动化的程序或程序系统。

第二章“Visual Basic 的结构”讨论语言的文法和措词。BASIC 是“初学者通用符号指令码”的首字缩略词，使用了许多英语的措词。如果仔细观察每一条指令，则可以看到更近似汇编语言的简单操作对象结构。

第三章“应用程序模型”考查程序的结构。应用程序的核心部分把接收到的信息作为数据，并把这些数据传输到用户可以利用的窗体中。为理解如何构造一个应用程序，我们将向读者介绍“信息论”。这种科学能处理知识语言和可用数据表示的工作过程。象库存文件管理那样的应用程序，可根据提供的数据进行合理的分析。

第四章“数据的构成”涉及计算技术中最困难和抽象的问题。像应用程序那样，为了提高效率，方便存取，减少冗余，数据也采用结构设计。这个过程用 Visual Basic 解释程序可能比较难实现，因为它在很大程度上依赖于旧的数据存取（一次一数据的）机制。在真正的多任务系统中，数据格式是具体的，以便多个应用程序可利用同样的数据而不必通过输出/输入过程。

第五章是“自动化的目标”。在商业环境中，程序员的目标是把某些日常的工作模型化，使之在计算机中实现。这并不意味着创建什么新的工作过程，而只是使办公室工作人员有更多机会使用计算机，把纸和笔的工作委托给计算机程序。程序员的目标是实现“自动化”，使得工作过程不再单调乏味。为此，需要先在纸上描述工作过程，如果需要的话用英语而不是 Visual Basic。

第六章“开发与调试”主要探讨程序设计过程。使用“层次”概念开发应用程序，逐一添加工作函数，直到完成整个应用程序。这是一种十分有效的方法，但也容易潜伏错误，因为在某一时刻，对应用程序总目标的了解可能不够全面。也许这就是该方法的主要缺点，但编程过程中错误是难免的。事实上，通过不断的修正错误将使程序更为高效。

第七章“上下文、作用域、关系”解释一些新的定义和声明方法。模块中的变量 X 可能不是出现在另一模块中的同一个 X（当然也可以是）。应用程序中对话框的窗体过程可被多次调用，而每次调用的方法不同，目的在于使对话更灵活生动。应用程序可以包括能相互交换的部件，还可以用已经编写好的部分，为不同的目标创建全新的应用程序。

第八章“与用户通信”，描述应用程序和用户之间的会话。在计算的工作站模型中，用户花费在工作站的时间称为“会话期”。应用程序中采用图形设备来补充术语和数值表达方式，把人与机器之间传递的信息符号化。

第九章“基本设备”，着重讨论与程序的图标和图形交互。现代电子设备采用多行按钮作为它们的控件；为模拟这些设备，Windows 程序中的图形面板也趋向于使用按钮行。

第十章“Windows 环境”介绍 Visual Basic（和其它 Microsoft Windows 应用程序）与 Windows 自身通信所用的指令。Windows 采用它自己的“语言”供内部的应用程序通信。这称为应用程序接口（API）。Windows API 是一系列的程序库，使用它的例程可直接访问 Windows 的图形、数据流和函数。

第十一章“应用程序之间的通信”，研究通信模型的不同用法。Windows 程序可共享资源、工具和总工作量等。

Visual Basic 应用程序可利用 Windows 环境向它提供的各种“流水线”，扩展已有的

Windows 应用程序。

第十二章“算法逻辑学”介绍逻辑方法。在任何程序中，数据代表某些对象，而算法则是代表“动作”或数据处理的过程，它具有良好的结构。算法为计算机描述某些真实世界的数据处理。例如，根据一般的要求从文件中选出记录或把一组记录排序等。

第十三章“智能程序设计”接触到人工智能。这里我们探讨作判定的算法，目的在于确定最好的动作过程。程序能给出所有可预见的选项。在作判断算法中，选项是根据计算的结果取得的，它并不代表人脑的选择。启发式算法尽可能紧密地模拟人类的推理过程，它使用以数学公式表示的启发式或真实世界规则。人们能书写的最复杂的这类程序也许是弈棋等游戏程序。

如果从头到尾读完全书，将能了解 Visual Basic 的整个程序设计过程。对于某些特殊兴趣之处，可反复阅读，并参看有关的插图。那无疑将有益于你所从事的实际工作。

Visual Basic 把人们带进计算技艺之门，使他们成为数学专家、内行，帮助他们更好地理解自己业务的作业进程，从而开辟出富于创新、表达的新途径。

目 录

第二部分 从概念到创造

第三部分 实验性程序设计

第一部分 程序设计过程

第一章 使用语言进行计算

让我们回忆一下上半世纪中有关计算的科学幻想。无论是小说还是电影，你都会看到科幻作家使人类和计算机对话。在科幻计算中，和语言相比，图标不是功能很强的工具。

90年代的程序员可能认为图符和图形的设计是用户通信的基本工具，但不可否认程序设计语言仍然是桥接人类和机器的重要通信工具。它之所以能成为通信的基本构件(思维的原型)是因为通信涉及更多的文字而不是图画。那么，软件市场商弄错了吗？以描述图标作为他们的联编原理的程序员，是不是忽视了向用户提供更为清晰的方法？

本章展示两个处于初始构造阶段的大型 Visual Basic 应用程序，当你仔细地研究它们之后，你会注意到我的重点是讨论高级语言计算与使用组装软件不同之处。你可以看到，语言为解题和自动化任务提供许多独特的方法。

1.1 为什么使用 Visual Basic 语言？

Visual Basic 的目标是生成独特的问题求解应用程序。高级程序设计语言为计算机构造了第一个真正的用户程序。当计算机硬件开发者认为所有计算机用户应成为程序员的时候，组装的软件应用程序就已经出现。但是，虽然有了无数的包装软件，高级语言仍保留着某些重要的领域。

商业界喜欢用高级语言而不是软件程序包构造定制的应用程序，主要原因如下：

- 商业界当前完成工作任务的方法未能用当前选择的预组装软件成功地自动化。
- 商业界中工作人员执行作业的各种方法，由于管理上的安排或个人嗜好，经常变动或调整，而当作业改变时，程序也必须改良。
- 没有经过全面训练的计算机操作工作人员，当他们学会更多有关计算的问题时，可能想使用随时增加功能的程序。假设用户都是些程序员，则希望他们能适应现成的软件和电子表格和其他软件程序包。
- 为了竞争、占领市场和鼓舞士气，公司有权使用有它们自己特色的软件。如果公司选择秘密的商业策略，更不应采用竞争对手所用的同类市场化软件。
- 面向特殊顾客的公司要求高度灵活的软件系统。

1.1.1 为初学者定义 Visual Basic

基于 Microsoft 的 Visual Basic 市场化方法，你可以认为它仅仅是另一种定制的应用程序。该公司不强调"选中和单击"的计算原理(不象 Word for Windows 和 Excel)。对于那些询问 Visual Basic 实际是什么的客户，标准的答复是：Visual Basic 是一种程序设计的指令系

统，它可以用于定制事务应用程序。

在商业或办公环境中，Visual Basic 的真正目的是把正常执行的事务模型化。VB 应用程序能实现自动化以加快和优化每日的任务，使工作少一些重复、也许还更有趣。此外，VB 应用程序不增加用户更多的工作。某些组装软件增加了办公室的工作量，而且工作耗费在维护计算机系统而不是运行事务。Visual Basic 程序员必须避免建造耗时和要求高维护成本的程序包；否则，就没有必要使用它。

人们对 Visual Basic（或任一程序设计语言）最普通的误解是：它是智能系统的主要成分。可能由于含混的市场印象，人们相信 Visual Basic 和其他语言能研究问题、进行分析和提供结论。高级程序设计语言不是智能的；它们不为你阐述任何事情。作为一个成熟的程序员，你应该避免误解，作为程序员，向 Visual Basic 说明是你的责任。作为 Visual Basic 应用程序的作者，我尽量不考虑我是在用 Visual Basic 进行通信。

图 1.1 为高级语言的通信模型。这里的椭圆形对象代表会话的各方，而前头代表会话的事件。电话工程师用通信模型把假想的双方或多方的会话联系起来。在这个特殊模型中，通信模型展示程序员和程序用户之间的双方会话。

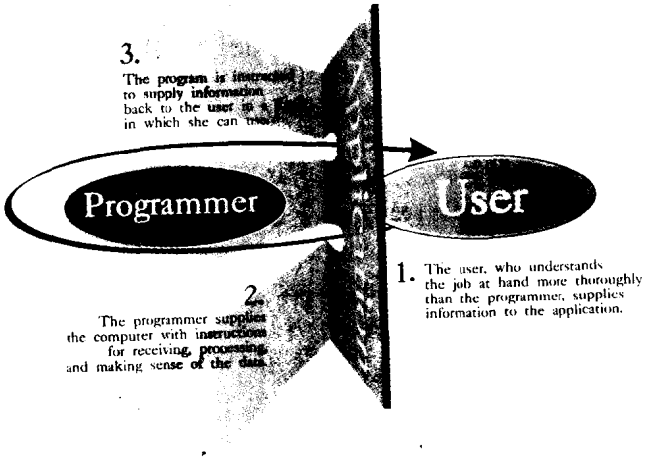


图 1.1 高级语言的通信模型

请注意，事件由用户启动和结束；程序员逻辑上作为他们自身和用户之间数据传输的向导。这不是程序和用户之间或程序员和程序之间的会话。完整的程序设计就是人们之间的会话。这个特殊会话的特点就是程序员预先指示程序如何应答用户所有的咨询。必须预先编制好如何应答用户向应用程序发出的所有可能询问。

图 1.1 把计算机描绘为遮挡程序员和用户两者视线的一堵墙。作为程序的作者，你可能感觉到就象通过磁带录音机和用户说话。在 Visual Basic 的情况下这特别真实（比标准 C++和 QBasic 更真实）。因为系统中有一个创新的事件驱动机制。这个系统“能”使用户更容易输入数据。当设计程序时，设想你已和用户交互而不是和计算机交互。

下面的实例展示了在程序员和用户之间的通信过程中高级程序设计语言的作用。

1.1.2　零售商品库存跟踪系统(实例 1)

若你的客户是一个零售产品商或联营商,他们希望编写一个零售货物的库存、购销和跟踪系统。这种机构比较小,往往是同一个人从供应商或批发商那里购买货物、销售这些货物,及分析这种货物的销售史。因此,采购、开发票、销售点和跟踪模块都不必完全分离。

该程序的用户需要使用的功能是:他们应能检查库存以观察产品是否在库中、放置在什么地方及它的销售价格等。如果产品不在库中,而顾客现在需要它,则用户现在有权初始化采购进程。在任何时候,用户必须能对特殊种类的产品和特定的产品考查销售的情况。

实例 1 的目标如下:

- 主要的库存模型维护一个全部销售货物的数据库。
- 顾客表保存关于带有特权和邮政订货的顾客信息。
- 采购模型维持一个由不同批发商提供的各种货物的最新价格表。这能使购买者确定当前最好的贸易策略。
- 发票和销售点模型使当前库存表可即时更改。
- 销售跟踪模型使用户知道特定货物的销售情况,包括考虑不同的货物分类和商标名。

首要的工作是设计窗体和窗口,而数据是程序的最重要部分,必须首先考虑好。所以,首先设计全程范围的数据变量。组成这个程序主体的数据元素是多变的库存货物。首先让我们研究一下库存货物的特征。除软件工具外,书本、硬件、视频元件和办公家具等都可以作为例子。库存货物的主要特征是它的名称、销售成本和货架价格。

货架中的一格可能有专门的序号。如果制造商对有限的货物提供特殊的优惠。譬如说,对专门标记和编号的货物提供优惠价格。这时,必须在顾客的发票中记录序号。然后,顾客可以复印该发票,并把它送给制造商作为购买的证明。再者,假定顾客返回有缺陷的货物,系统将"不卖"这货物并把它放回库中,原来发票的某处也必须记录这个序号。购销部门需要这个序号以便把该货物退回给制造商(简称为"退回确认"或"R. A. ")。

图 1.2 示出在作决定过程中的原始分段。必须区分货物和产品,所以需要两个编目的数据文件。"货物"是货架上的特定对象或盒子,而"产品"是一组货物的名称。虽然两种编目有相关的元素,但为提高效率,应保持最少的重复。

Visual Basic 处理存储中数据数组的功能还比较弱。解释程序提供三种数据存取方案;其中两个可应用于本程序。顺序存取方案要求把整个数组送入内存,完全在内存中改变或修补数据。当关闭文件时,才把该数组全部存回磁盘。当在网络上编写 Visual Basic 应用程序时,顺序存取是危险的方案。如果两个客户或伙伴沿着网络把同样的数据文件装入存储数组,而每一伙伴改变数组时没有注意到其他伙伴的改变,那么,最后保存的数据文件将重写先前保留的文件,这意味着丢失了一半的修改。无论如何,文件变成 100%的失效。

当需要一次读一个记录时,随机存取方法是比较好的。Visual Basic 允许程序员在引用变量时采用面向对象的语法。Microsoft 发现它很难命名用 Type 子句声明变量这种概念。在 Visual Basic 1,Microsoft 使用术语"用户定义变量",尽管是程序员(不是用户)进行定义。至于 Visual Basic 2 ,Microsoft 试用名字"记录变量",因为在 Visual Basic 的随机存取数据文件

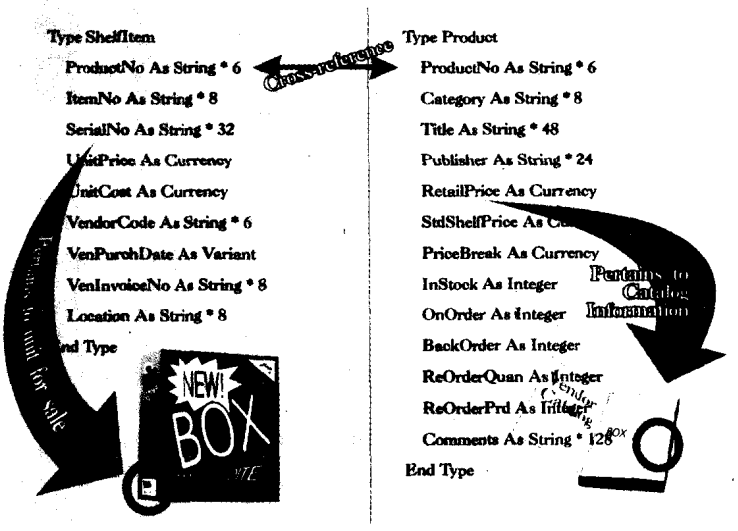


图 1.2 一致性标识的危机

方案中，这涉及到数据记录的构成元素。由于变量不包含记录，Microsoft 在术语的选择上可能有些混淆。因此，本书使用术语“组合变量”以指称在全程或总模块中用 Type 子句声明的变量结构。

> **技术注记** 如果多数据文件中含有相关字段时，选择随机存取方法。

，组合变量方案允许在记录中查找数据字段时，按先主题，后字段的顺序进行。有如下的语法：

```
subject. field
```

从数据库工程师的观点看，前面引用的第一项是它的主题，因为可以使用 subject 作为分类，访问记录中所有的字段。但是，从软件工程师的观点看，引用的第一项是对象；每一字段组成“事件”的一个特征，它按对象编址，因此必须把对象分类，不能混淆。

在任何情况下，都可以利用 Visual Basic 中的 Type 子句描述数据表格和它们的构成字段。在图 1.2 中，可以看出作为货架中的盒子和作为销售产品的货物之间的不同。为 Visual Basic 货物 INVENT1. MAK 编写的前几行代码是对库存系统两个基本数据文件结构的初始估算。我已在程序中加进了一些注释，以说明每一字段在它的组合变量中的作用。

```
Type ShelfItem
  ProductNo As String * 6       ' Key field
  SerialNo As String * 32       ' Number given the item
                                ' by its manufacturer
  UnitPrice As Currency         ' Shelf price give the item set
                                ' by vendor
  UnitCost As Currency          '  Actual Cost of the item set
                                '  by Vendor
```

```
    VendorCode As String * 6        'Code for the vendor that
                                    'sold store this item
    VenPurchDate As Variant         'Date store purchased the
                                    'item from vendor
    VenInvoice As String * 8        'Invoice number of prior
                                    'purchase from vendor
End Type

Type Product
    ProductNo As String * 6         Key field
    Category As Integer             Store-defined code for shelf
                                    'category
    Title As String * 48            'Official title for the
                                    'product
    Publisher As String * 6         'Store-defined code for
                                    'product publisher
    RetailPrice As Currency         'Manufacturer-suggested
                                    'retail price
    OnOrder As Integer              'Amount of product currently
                                    'on order
    BackOrder As Integer            'Amount of product vendors
                                    'placed on back-order
    ReOrderQuan As Integer          'Recommended reorder quantity
                                    'on regular basis
    ReOrderPrd As Integer           'Store code for regular
                                    'reorder
    Comments As String * 128        'Arbitrary comments from
                                    'any user
End Type
```

所有应用程序(扩展的或其他)都在内存或盘上运行数据库,甚至对于只有四种功能的计算器也是如此。存储中数据文件的内容是一个表格。为把内容组织好,表格须包含一系列等长的记录。每个在 Type 声明中出现的成分变量的规则定义了 ShelfItem 和 Product 数据文件的记录长度和结构。记录中的每一变量构成记录的字段。图 1.3 示出怎样安排对象变量 ShelfItem 的数据表格的内容。

现在,请注意两个互相关联的组合变量结构,只有一个变量 ProductNO 在两种类型中同时声明。在此方案中,ProductNO 包含一个 6 数字的标识号。它代表产品的名称。这是个关键的字段,它标识涉及该产品的所有数据表格中的产品。因为它是关键字段。它唯一地标识该产品。

关键字段技术对某些数据库或数据文件方案的结构是很重要的。

技术摘要:关键数据字段

定义:在 Visual Basic 中,关键字段作为一组组合变量的单个独特的成分元素来实现。关键字段可能包含数字或字母的数据。根据定义,它出现在 Visual Basic 货物不只一个的组合变量中。当组合变量代表 Visual Basic 的数据文件时,Type 子句声明的成分元素各自定义数据文件的一个记录。每一数据文件的记录和相同的货物或对象有关。通常,记录中的成分元素用作为标识符——例如,人的社会保险号或汽车的车牌号。可以在多个组合变量中使用同样的标识符,以代表同一个人或货物,采用这种方法,你不必存储货物的所有特征。这些货物遍及 Visual Basic 工程,在存储的大量组合变量中,或在磁盘面积和繁杂的数据文件里。你也

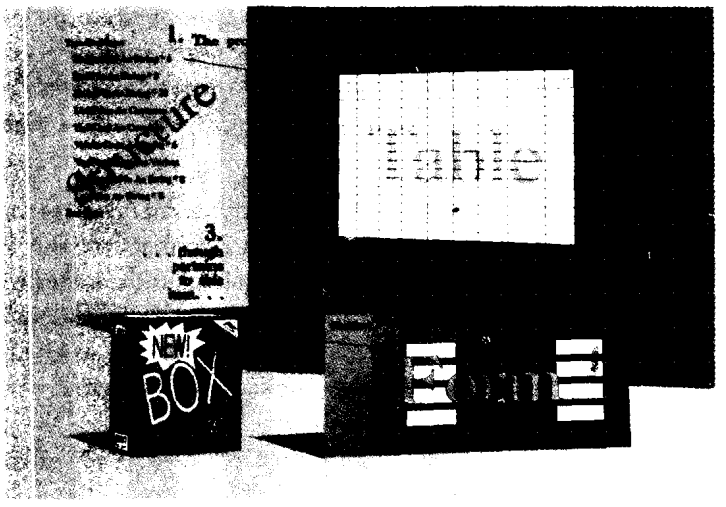


图 1.3 数据表格的双重标识

可以使用关键字段作为排序过程的索引字段。

执行：使用 Type 子句在全程模块中声明一组组合变量。这些组合变量的目标彼此相关，因而它们的 Type 子句都包含一个(只有一个)成分变量。这个变量的名字、定义、和声明是相同的。考虑到形式，关键字段是 Type 子句中声明的第一个组合变量。这些子句使用关键字段作为它们的标识符。

例子：假设你正在为快速润滑汽车服务性企业编写应用程序。该应用程序使用两个数据文件，一个文件包含所有已预先入库的汽车的服务记录，而另一个则包含每辆汽车的发票。你可以使用交通工具的牌照编号，它在大多数汽车的档泥板或在发动机罩下，作为标识符。

两个和同一辆汽车的记录有关的 Type 子句如下：

```
Type ServiceRecord
    VID As String * 36
    ServiceCode(16) As Integer
    ServiceDate(16) As Integer
End Type

Type Invoice
    VID As String * 36
    Number As String * 6
    EntryDate As Variant
    BillAmount As Currency
    Received As Currency
    PymtCode As Integer
End Type

Global CarHistory(1000) As ServiceRecord
Global CustomerHistory(1000) As Invoice
```

你可以使用已经在 ServiceRecord 文件中处理的数据内容生成第一张顾客发票。用下面的指令设置 Invoice 文件的新关键字段：

CustomerHistory(newcar%).VID CarHistory (thisCar%).VID

其中 newcar%是指向 ServiceRecord 文件中最后一个记录的指针，而 thisCar%是指向发票窗体上当前发票记录的指针(指针反映出数据库或数据文件中记录的位置)。

在应用程序的后部：假定这个公司的会计师抽出发票记录以查看与高收费相符的服务。你可以建立指向当前汽车的指针 thisCar%，使用这指令：

Carno$ =CustomerHistory(thisCar%).VID

现在，你可以使用变量 Carno$作为指针进行搜索，以便匹配 CarHistory (searchcar%) VID 的内容。

提醒：为正常地实现数据文件中的关键字段 关键字段中每一表项的内容必须唯一 应该在你的应用程序中实现安全功能，以确保没有两个关键字段的表项相同。为高效地进行工作，你应使程序在每次添加记录时重排序数据文件，并使用关键字段作为排序的索引字段。采用这种方法，如果两个关键字段表项相同，在数据表被排序之后，它们将彼此相邻。

对于货物 INVENT1.MAK，许多内含组合变量的关键字段是 ProductNO，它代表货架上产品的名称或专用标识。

根据采购代理人的看法，一些批发商可能把价格降到低于当前产品目录所印出的价格，目的在于进行商业上的竞争。当作出采购决策，采购代理人接收把价格降低的那些批发商代理人的电话，并对比每一单项的当前目录价格。这些特殊的处理可能难以归类为正常的价格，它们可能为买三样东西，或一样 A 和两样 B 而专门降价。许多目录已经对每一类物品类标价，如果订单的总金额是在某一数目(譬如，1000 美元)之上，则可降低价格。现在零售商品市场中的货物价格是从不固定的；事实上，它是灵活可变的。你必须以某种方式为价格的变化编码。至此，你可以认为产品价格保留在独立的数据文件中。

在这些细节问题上，试考虑一下使用预组装应用程序工作时你该如何做，这样做又有些什么不同。在 80 年代，使用电子表格程序作为数据库管理程序是很普遍的。用户写"库存程序"(一组松散相关的宏)的电子表格列出所有和库存产品有关的数据字段，每项一行，每字段一列。在库存应用模型中，需要分离货物数据文件和产品数据文件，但带有某些两文件共享和重复的标识字段。使用电子表格，那两个文件可以是两个独立的表，也可以是一个大表格中的两段。电子表格用户需要维护这两个表格的关系或为不同的行记录单元的指针，可以想像得到其中的困难。没有什么比文件格式更能说明电子表格作为商业自动化系统的低效。

现在，让我们暂不讨论实例 1，等到第三章"应用程序模型"时再讨论它。当你重新使用这个程序时，你将明白如何建立和分类应用程序中的各种窗体。稍后，你将看到如何设计和使用此程序的更为复杂的数据结构。

1.2 语言的通用性

Florian Coulmas(他在德国 Dusseldorf 大学教授语言)论证说，书写的语言是作为对金钱符号的一种方法而创造的。在 1989 年他的一篇有关正文符号进展的优秀论文"世界的书写系统"中，Culmas 提供论据指出，公元前 8000 年小亚细亚的苏门答腊人使用不同形状的石头代表货币单位的记号。苏门答腊人使用软陶片记录他们的市场交易，把每一种记号刻入陶