



高等学校“十一五”规划教材

Visual Basic 程序设计

主 编 王国权 顾泽元

煤炭工业出版社

高等学校“十五”规划教材

Visual Basic 程序设计

主 编 王国权 顾泽元

副主编 张绍兵 董 军

主 审 石 岩

煤炭工业出版社

·北 京·

内 容 提 要

本书以 Visual Basic 6.0 软件系统为环境,系统地介绍了 Visual Basic 程序设计的基本概念、理论和方法。全书共分为 12 章,内容包括 Visual Basic 程序设计概述、Visual Basic 程序设计基础、Visual Basic 程序设计结构、常用控件、数组及应用、过程、应用程序界面设计、文件操作、绘图应用程序设计、数据库管理应用程序设计、多媒体应用程序设计、网络应用程序设计等。本书既注重理论知识,又突出实践和应用,深入浅出,有利于学生对基础知识的理解和编程能力的提高。

本书是高等学校非计算机专业本、专科学生学习面向对象程序设计的通用教材,也可供参加计算机等级考试的读者学习与参考。

图书在版编目(CIP)数据

Visual Basic 程序设计/王国权,顾泽元主编. —北京:
煤炭工业出版社,2006.11
高等学校“十一五”规划教材
ISBN 7-5020-2964-8

I.V… II.①王…②顾… III.BASIC 语言—程序
设计—高等学校—教材 IV.TP312

中国版本图书馆 CIP 数据核字 (2006) 第 116453 号

煤炭工业出版社 出版
(北京市朝阳区芍药居 35 号 100029)
网址:www.cciph.com.cn
北京京科印刷有限公司 印刷
新华书店北京发行所 发行

*

开本 787mm×1092mm¹/₁₆ 印张 18³/₄
字数 453 千字 印数 1—6,000
2006 年 12 月第 1 版 2006 年 12 月第 1 次印刷
社内编号 5763 定价 32.00 元

版权所有 违者必究

本书如有缺页、倒页、脱页等质量问题,本社负责调换

前 言

根据教育部高等学校计算机教学指导委员会 2003 年发布的《关于进一步加强高校计算机基础教学的几点意见》指导精神,按照“1+X”课程体系改革的要求,我们编写了《Visual Basic 程序设计》这本教材。其内容反映了面向对象的程序设计方法,目的是满足高校非计算机专业公共计算机基础课教学的需要。本书教学内容丰富,结构合理,实践性强,深入浅出,既注重基本概念、基本理论、基本方法,又突出实践性和实用性。通过使用本教材,可以使学生掌握 Visual Basic 程序设计的基本概念、基本理论、基本方法和基本应用技能,培养学生分析问题和解决问题的能力、创新精神和实践能力,为应用计算机解决本专业领域相关问题奠定基础。

本书兼顾实际教学内容的需要和全国计算机等级考试二级 Visual Basic 考试大纲(2004 版),并结合了高校课程建设与教学改革的新成果。在讲解基本概念、基本理论、基本方法的基础上,通过应用实例或例题进行程序设计实践,操作步骤清晰,有利于学生编程能力和自学能力的提高。本书章节安排合理,符合教学过程和学生学习的实际要求。每章后都配有大量习题,并在附录中给出了全国计算机等级考试二级 Visual Basic 考试大纲、考试样题、程序调试、错误处理及编程规范,有利于学生的基本训练和复习,使学生掌握程序调试和编程规范等基本技能,培养学生良好的编程规范。

本书按 70 学时编写,其中实验不少于 30 学时,各学校可根据教学大纲的要求适当取舍。

本书由王国权、顾泽元任主编,张绍兵、董军任副主编,石岩主审。全书由王国权统稿。其中,第 1 章、第 2 章、第 4 章由王国权编写,第 8 章、第 9 章、第 10 章由顾泽元编写,第 3 章、第 5 章、第 6 章由张绍兵编写,第 7 章、第 11 章、第 12 章由董军编写,附录由石岩编写。在本书编写过程中,得到了黑龙江科技学院各级领导的大力支持和帮助,在此表示感谢!

由于编者水平有限,书中难免有疏漏之处,敬请各位专家和读者批评指正。

编 者
2006 年 9 月

目 录

第1章 Visual Basic 程序设计概述	(1)
1.1 程序设计的基本概念	(1)
1.2 面向对象的程序设计方法	(3)
1.3 Visual Basic 6.0 简介	(5)
1.3.1 Visual Basic 6.0 的功能特点及版本	(5)
1.3.2 Visual Basic 6.0 的安装、启动及退出	(6)
1.3.3 Visual Basic 6.0 的集成开发环境	(10)
1.4 Visual Basic 6.0 应用程序设计步骤	(12)
1.4.1 Visual Basic 6.0 的窗体	(12)
1.4.2 Visual Basic 6.0 的程序管理结构	(13)
1.4.3 Visual Basic 6.0 的程序管理操作	(14)
1.4.4 Visual Basic 6.0 程序设计步骤	(16)
1.4.5 简单应用程序设计实例	(17)
习题一	(21)
第2章 Visual Basic 程序设计基础	(24)
2.1 Visual Basic 程序语言的基本概念	(24)
2.1.1 标识符	(24)
2.1.2 书写规范	(24)
2.2 数据类型	(25)
2.2.1 基本数据类型	(26)
2.2.2 常量	(27)
2.2.3 变量	(28)
2.2.4 自定义数据类型	(32)
2.2.5 枚举数据类型	(33)
2.3 运算符与表达式	(34)
2.3.1 算数运算符与算数表达式	(34)
2.3.2 关系运算符与关系表达式	(34)
2.3.3 逻辑运算符与逻辑表达式	(35)
2.3.4 字符串运算符与字符串表达式	(36)
2.4 运算符的优先级	(36)
2.5 常用内部函数	(37)
2.5.1 数学函数	(37)
2.5.2 字符串函数	(38)

2.5.3 数据类型转换函数·····	(39)
2.5.4 日期和时间函数·····	(39)
2.5.5 随机函数·····	(40)
2.5.6 输入输出函数·····	(40)
2.6 应用实例·····	(43)
习题二·····	(44)
第3章 Visual Basic 程序设计结构·····	(47)
3.1 顺序结构·····	(47)
3.1.1 顺序结构的概念与流程·····	(47)
3.1.2 顺序结构的基本语句·····	(47)
3.2 选择结构·····	(49)
3.2.1 单行结构条件语句·····	(49)
3.2.2 If 函数·····	(50)
3.2.3 块结构条件语句·····	(50)
3.2.4 条件语句的嵌套·····	(51)
3.2.5 多分支选择结构语句·····	(53)
3.3 循环结构·····	(55)
3.3.1 For···Next 语句·····	(55)
3.3.2 Do···Loop 语句·····	(56)
3.3.3 While···Wend 语句·····	(59)
3.3.4 Goto 语句·····	(60)
3.4 循环的嵌套·····	(60)
3.4.1 循环嵌套的概念·····	(60)
3.4.2 循环嵌套的应用举例·····	(61)
习题三·····	(62)
第4章 常用控件·····	(67)
4.1 控件的分类·····	(67)
4.2 控件的通用特性·····	(68)
4.3 常用标准控件·····	(69)
4.3.1 标签(Label)控件·····	(70)
4.3.2 命令按钮(CommandButton)控件·····	(71)
4.3.3 文本框(TextBox)控件·····	(72)
4.3.4 框架(Frame)控件·····	(73)
4.3.5 选项按钮(OptionButton)控件·····	(73)
4.3.6 复选框(CheckBox)控件·····	(75)
4.3.7 列表框(ListBox)控件·····	(78)
4.3.8 组合框(ComboBox)控件·····	(79)
4.3.9 滚动条(ScrollBar)控件·····	(80)
4.3.10 时钟(Timer)控件·····	(82)

4.3.11 图像(Image)控件	(83)
4.3.12 图片框(PictureBox)控件	(83)
4.4 应用实例	(84)
习题四	(88)
第5章 数组及应用	(92)
5.1 数组的声明	(92)
5.1.1 一维数组的声明	(92)
5.1.2 一维数组的引用	(93)
5.1.3 二维数组的声明	(93)
5.1.4 二维数组的引用	(93)
5.1.5 动态数组的建立	(94)
5.2 数组的操作	(94)
5.2.1 数组操作语句和函数	(94)
5.2.2 For Each...Next 语句	(97)
5.3 数组的应用举例	(98)
5.3.1 一维数组的应用举例	(98)
5.3.2 二维数组的应用举例	(98)
5.3.3 动态数组的应用举例	(101)
5.4 控件数组	(101)
5.4.1 创建控件数组	(102)
5.4.2 控件数组的应用举例	(104)
习题五	(105)
第6章 过程	(110)
6.1 过程的定义与分类	(110)
6.1.1 过程的定义	(110)
6.1.2 过程的分类	(110)
6.2 Sub 过程	(111)
6.2.1 事件过程	(111)
6.2.2 自定义过程	(112)
6.3 函数过程	(115)
6.3.1 函数过程的定义	(115)
6.3.2 函数过程的调用	(115)
6.4 过程参数的传递	(117)
6.4.1 形式参数与实际参数	(117)
6.4.2 参数传递的方式	(117)
6.4.3 数组参数的传递	(122)
6.4.4 对象参数的传递	(124)
6.5 可选参数和可变参数	(125)
6.5.1 可选参数	(125)

6.5.2 可变参数	(127)
6.6 过程的嵌套和递归调用	(128)
6.6.1 过程的嵌套调用	(128)
6.6.2 过程的递归调用	(129)
6.7 鼠标与键盘事件过程	(130)
6.7.1 鼠标事件过程	(130)
6.7.2 键盘事件过程	(135)
习题六	(140)
第7章 应用程序界面设计	(145)
7.1 菜单的设计	(145)
7.1.1 下拉式菜单	(145)
7.1.2 弹出式菜单	(148)
7.2 对话框的设计	(149)
7.2.1 CommonDialog 控件	(149)
7.2.2 通用对话框的应用	(154)
7.3 工具栏的设计	(156)
7.3.1 ImageList 控件	(157)
7.3.2 ToolBar 控件	(157)
7.4 多重窗体的设计	(161)
7.4.1 创建 MDI 窗体	(162)
7.4.2 加入 MDI 子窗体	(162)
7.4.3 MDI 窗体与子窗体的交互操作	(162)
习题七	(163)
第8章 文件操作	(166)
8.1 文件的基础知识	(166)
8.2 文件系统操作	(168)
8.2.1 文件的打开与关闭	(168)
8.2.2 文件管理函数与语句	(170)
8.3 顺序文件	(175)
8.3.1 顺序文件的打开与关闭	(175)
8.3.2 顺序文件的读写操作	(176)
8.3.3 顺序文件应用实例	(178)
8.4 随机文件	(181)
8.4.1 随机文件的打开与关闭	(181)
8.4.2 随机文件的读写操作	(182)
8.4.3 随机文件记录的维护	(183)
8.4.4 随机文件应用实例	(183)
8.5 二进制文件	(188)
8.5.1 二进制文件的操作	(188)

8.5.2 二进制文件应用实例	(189)
8.6 文件系统控件	(190)
8.6.1 DriveListBox 控件	(190)
8.6.2 DirListBox 控件	(191)
8.6.3 FileListBox 控件	(193)
8.6.4 文件系统控件的组合	(195)
8.6.5 应用实例	(196)
习题八	(198)
第9章 绘图应用程序设计	(202)
9.1 图形操作基础	(202)
9.1.1 图形坐标系统	(202)
9.1.2 绘图颜色	(204)
9.1.3 绘图属性	(205)
9.2 绘图方法	(206)
9.2.1 PSet 方法	(206)
9.2.2 Line 方法	(207)
9.2.3 Circle 方法	(207)
9.2.4 Point 方法	(208)
9.2.5 应用实例	(208)
9.3 Line 控件与 Shape 控件	(210)
9.3.1 Line 控件	(210)
9.3.2 Shape 控件	(210)
9.4 Image 控件和 PictureBox 控件	(211)
9.4.1 Image 控件	(211)
9.4.2 PictureBox 控件	(212)
9.4.3 应用实例	(212)
习题九	(214)
第10章 数据库管理应用程序设计	(216)
10.1 数据库基础知识	(216)
10.1.1 数据库概念	(216)
10.1.2 关系数据库及其结构	(217)
10.1.3 SQL 语言	(218)
10.2 可视化数据管理器	(219)
10.2.1 启动可视化数据管理器	(219)
10.2.2 利用可视化数据管理器创建及管理数据库	(219)
10.2.3 数据窗体设计器	(224)
10.3 数据库控件	(225)
10.3.1 Adodc 控件	(225)
10.3.2 数据绑定控件	(228)

10.4 Adodc 控件的高级成员	(231)
10.4.1 CommandType 属性	(231)
10.4.2 RecordSet 属性	(231)
10.4.3 Refresh 方法	(232)
10.5 应用实例	(233)
习题十	(238)
第 11 章 多媒体应用程序设计	(240)
11.1 MMControl 控件	(240)
11.1.1 MMControl 控件的属性	(240)
11.1.2 MMControl 控件编程步骤	(242)
11.1.3 实例	(242)
11.2 MediaPlayer 控件	(244)
11.2.1 MediaPlayer 控件的属性	(244)
11.2.2 MediaPlayer 控件的方法	(245)
11.2.3 应用实例	(246)
11.3 API 多媒体函数控制方法	(247)
11.3.1 API 函数声明	(247)
11.3.2 API 多媒体函数	(248)
11.3.3 应用实例	(248)
习题十一	(249)
第 12 章 网络应用程序设计	(251)
12.1 网络基础知识	(251)
12.2 Internet Transfer 控件	(252)
12.2.1 Internet Transfer 控件的属性	(252)
12.2.2 Internet Transfer 控件的方法	(254)
12.2.3 Internet Transfer 控件的事件	(256)
12.2.4 Internet Transfer 控件的应用实例	(257)
12.3 WebBrowser 控件	(259)
12.3.1 WebBrowser 控件的属性	(259)
12.3.2 WebBrowser 控件的方法	(259)
12.3.3 WebBrowser 控件的事件	(261)
12.3.4 WebBrowser 控件的应用实例	(261)
习题十二	(264)
附 录	(265)
附录一 Visual Basic 程序调试与错误处理	(265)
附录二 Visual Basic 编程规范	(270)
附录三 全国计算机等级考试二级 Visual Basic 考试大纲	(273)
附录四 全国计算机等级考试二级 Visual Basic 样题	(275)
参考文献	(287)

第 1 章 Visual Basic 程序设计概述

本章主要介绍程序设计涉及的基本概念、面向对象的程序设计方法的基础知识,以及 Visual Basic 6.0 的集成开发环境、窗体的组成、程序管理结构等,并以一个简单的 Visual Basic 6.0 应用程序设计实例说明 Visual Basic 程序设计的基本步骤。

1.1 程序设计的基本概念

在进行程序设计之前,应了解程序设计的基本概念,包括程序、算法、程序设计语言、程序设计等基本知识。

1. 程序

程序是操作计算机完成特定任务指令的集合,程序由程序设计语言来实现。Visual Basic 语言就是一种程序设计语言,是用来编写程序的。为了满足人们各种特定的需要,计算机工作者开发了各种工具软件,例如文字处理、表格处理、图形处理、多媒体管理以及各种系统管理工具等,都是一些专用的程序集,用户在界面上与计算机进行交流,就是在调用程序集中的子程序。然而,人们要求计算机做的工作千差万别,现代社会对计算机的要求也是无止境的。计算机厂家不断地制造功能越来越强大的计算机系统,软件开发者也不断地设计出满足不同需要的应用程序。不管计算机结构如何,它要干什么工作,人们总是用某种形式的程序与计算机进行交流的。

2. 算法

算法是对解决问题步骤的描述,或者说用来描述程序的实现步骤。算法本身也可以采用不同的方式进行描述,常用描述算法的方法有自然语言描述、伪代码描述、程序流程图描述等,其中程序流程图分为美国国家标准学会(ANSI)提出的标准流程图和两位美国学者提出的 N-S 结构流程图。一般情况下,本书采用标准流程图来描述程序的算法。

3. 程序设计语言

程序设计语言是由字、词和语法规则构成的指令系统,也称为算法语言。它是人与计算机交流的工具。用户使用程序设计语言,可以告诉计算机做什么,计算机就会根据指令的要求逐条地执行,并且把执行结果告诉用户。

计算机应用的领域很广泛,为了适应不同的需要,往往程序设计语言又各具特点,如有适合于编写系统软件的语言、进行科学计算的语言、数据库管理的语言、图形设计的语言以及人工智能的语言等,还有一些语言具备多种功能。从应用角度说,难以对程序设计语言进行严格分类。随着计算机科学的发展及应用领域的迅速扩展,各种语言的版本都在不断地变化,功能在不断更新和增强,每个时期都有一批语言在流行,又有一批语言在消亡。因此,应该掌握程序设计语言中本质性的、规律性的东西。

根据程序设计语言提供的指令能否被计算机直接执行的情况,一般可以把程序设计语言分为如下三种:

(1) 机器语言。机器语言也称二进制语言,它的代码全部由二进制符号“0”和“1”按不同的方式排列组合而组成。用二进制语言编写的程序能够被计算机直接识别执行,是最早期的计算机程序设计语言,与计算机硬件有关。机器语言的特点是执行快,但是二进制代码记忆难度大,易于出错。

(2) 汇编语言。汇编语言也称符号语言,是用一些英文缩写等助记符来表示二进制代码的程序设计语言。用汇编语言编写的程序,只需要经过简单的翻译就可以被计算机执行。汇编语言相对于机器语言而言,容易记忆,增强了程序的可读性,但是不能被计算机执行,必须通过一个汇编程序被翻译成机器语言后才能执行。

机器语言和汇编语言都是面向计算机的程序设计语言,每种计算机的 CPU(中央处理器)都有一套自己的指令系统,称为机器语言。机器语言编写的程序执行速度快,资源要求低,通常用于编写直接与机器交互的程序,如控制程序等。

(3) 高级语言。由人们易于接受的、接近人类语言的描述方式构成的指令系统,称为高级语言。高级语言是以汇编语言为基础逐渐形成的,更加接近人类的自然语言的一种程序设计语言。一种高级程序设计语言往往只有一百几十条词汇、若干条规则,代码简短、便于记忆、易学易用。高级语言提供常用的数据描述和对数据操作的规则描述,这些规则是脱机的,程序员只需要专注于问题的求解,不必关心机器内部结构和实现。

通常所说的“程序设计语言”是指高级语言,用高级语言编写的程序称为“源程序”。计算机不能直接识别源程序代码,必须翻译成二进制程序代码才能在机器上运行。翻译方式有两种:一种称为解释方式,另一种称为编译方式。解释方式是由解释程序对源程序逐个语句一边翻译,一边执行,这种方式执行速度慢,便于观察调试程序;编译方式是由编译程序把源程序全部翻译成二进制程序,编译后的程序称为“目标程序”,一旦编译成功,目标程序就可以反复高速执行。每种高级语言都配有解释或编译系统,解释方式交互性强,而编译方式速度更快。Visual Basic 提供解释和编译两种执行方式。源程序和目标程序都可作为文件保存下来。

Visual Basic 语言是 20 世纪 90 年代 Microsoft 公司以结构化 Basic 语言为基础,以事件驱动为运行机制而开发的一种通用的可视化程序设计语言。除了提供常规的编程环境外,还提供了可视化设计工具,便于程序员建立图形对象,巧妙地把 Windows 编程的复杂性“封装”起来。它不仅具有传统的程序设计语言的功能,而且随着版本的改进,功能越来越强大,不但可以作为多媒体软件制作工具、实现数据库管理,而且还具有网络功能等。

4. 程序设计

程序设计是指为了利用计算机解决某种特定的问题,使用某种程序设计语言,编写计算机执行的指令序列。程序设计是一项在遵守程序设计语言语法前提下的创造性工作,一般需要完成数据描述和数据处理等工作。

(1) 数据描述。数据描述是指把被处理的信息描述成计算机可以接受的数据形式,如整数、实数、字符、数组等。使用计算机进行信息处理时,必须把信息转换成可被机器识别的“数据”,如数字、文字、图形、声音等。不管什么数据,计算机都以二进制形式进行存储和加工处理。数据是信息的载体,信息依靠数据来表达。有些数据,可以通过程序设计语言直接用“数据类型”描述,如数值和字符等;有些数据,一般的程序设计语言没有提供直接定义,但可以作

为外部文件使用,如 Visual Basic 可以在界面或程序代码中使用图形、声音等文件。

(2) 数据处理。数据处理是指为了获取所需的数据和有用的资料而对数据进行输入、输出、整理、计算、存储、维护等一系列处理工作。通常可用一个子程序来实现数据的某种操作,数据可以通过外部设备,如键盘、磁盘输入,也可以在程序内部使用初始化、赋值等方式获得;程序的执行结果可以输出到屏幕、打印机、文件,或者传送给其他程序。

在程序设计中,编写源程序一般常出现三类错误:一是编译错误,即编译程序时发现的语法错误,如表达式不符合语法规则等;二是运行错误,即执行目标程序时发现错误,如除数为 0 等;三是逻辑错误,即编译和运行时均不能发生的错误,如由表达式书写错误带来程序运行结果错误等。因此,程序一般要经过反复的调试、验证才能完善投入使用,编写的程序应力求具备正确性、易读性、运行高效性。

1.2 面向对象的程序设计方法

目前程序设计方法主要分为两类:一类是结构化的程序设计方法,另一类是面向对象的程序设计方法。

结构化的程序设计方法是 1966 年被提出的,其思想是把程序设计分为顺序结构、选择结构、循环结构。每种结构的程序流程不同,当需要按照语句的先后次序执行时,采用顺序结构;当需要根据条件选择地执行程序时,采用选择结构;当需要根据条件成立与否来决定是否循环执行程序时,采用循环结构。不论程序简单,还是复杂,都可以通过上述 3 种基本结构来实现,提高了程序的清晰度和可维护性。程序设计结构的内容详见第 3 章。结构化的程序设计方法是面向数据和程序的方法,这种方法把数据和程序作为相互独立的实体,在编写程序时,使数据和程序保持一致是程序员的一个沉重的负担。

面向对象的程序设计方法是在结构化的程序设计方法基础上发展起来的,与结构化的程序设计方法有本质的不同,它是把数据和程序组合起来作为一个对象,每个对象除了传递消息之外,相互之间没有其他联系。因此,使程序员摆脱了具体的数据格式和程序的束缚,可以集中精力研究和设计要处理的对象。

在 Visual Basic 中,体现了面向对象程序设计和结构化程序设计两种思想,即总体是面向对象的程序设计思想,而在每个对象内部编程时则采用结构化的编程思想。

1. 类

抽象是人认识事物时的一种行之有效的方法,通过抽象可以反映出诸多事物共同拥有的本质特征。程序设计语言也可以进行不同程度的抽象,结构化程序设计语言主要依靠基本结构对程序进行抽象,编程人员需要考虑整个程序的流程,依据流程的不同采用不同结构实现程序的功能。在面向对象的程序设计中,抽象的方法是通过“类”的概念来体现的,“类”是具有共同属性和操作的多个对象的相似特征的统一描述,反映了一个群体事物的共同数据特征和功能特征。“类”可以分多个层次抽象,较高抽象层次的类称为“父类”,较低抽象层次的类称为“子类”。例如,“车”可以抽象成一个类,这类事物具有一些共同特性。如果把“车”作为父类,则可以把“轿车”、“赛车”、“摩托车”等作为子类,每个子类中的事物也具有共同特性,不同子类中的事物都具有一部分不同的特性。

2. 对象

在 Visual Basic 中,对象是指程序和数据的组合。简单地说,就是把对象当作一个单位来处理。“类”是抽象的概念,类中的每个具体事物称为该类的“对象”。对象是类的实例,所以每个对象都有自己的特性值。对象是动作的主体,一个复杂的对象可以由若干个简单的对象组成。在 Visual Basic 中,对象主要包括“窗体”和“控件”等。窗体是程序运行时的一个窗口;控件是窗体上的界面元素,如命令按钮和文本框等。每个对象的静态特征有对象的“属性”,动态特征有对象的“事件”和“方法”。在设计对象时,只要将对象添加在界面上,不必编写建立和描述对象的程序代码。

3. 对象的属性

对象的属性是对象的数据,用来描述对象的静态特征。属性有属性值,改变对象的属性值就可以改变对象的特性,如窗体对象的 Width 和 Height 属性用来决定其宽度和高度。不同的属性取值决定了这个对象不同于其他对象。每个对象包含的属性不同,但有些属性是很多对象共同有的,如 Width 和 Height 属性。

在 Visual Basic 程序中,设置对象的属性,要按照如下的引用格式:

对象名.属性名=属性值

例如,一个窗体对象的名字是 Form1,则通过程序代码设置其高度的语句是:

Form1.Height=1000

说明:每个对象都有一个惟一的标识,也就是对象的“名称”属性。要引用哪个对象,就要引用这个对象的名称,如果省略对象名称则表示对当前窗体属性进行设置。

4. 对象的方法

对象的方法是指对象可以进行的动作或行为,用来描述对象的动态特征。对象的方法是系统已经实现的一些函数,用户只需要直接使用即可。引用对象的方法的语法格式如下:

对象名.方法名[参数列表]

例如,窗体对象有 Print 方法,可以在窗体上显示数据,在窗体 Form1 上打印输出“计算机程序设计”的语句是:

Form1.Print "计算机程序设计"

说明:如果省略对象名则表示引用当前窗体的方法。

5. 对象的事件

在 Visual Basic 中,每个对象都有一些系统预先定义好的、对象可以执行的动作,这些能被对象识别的动作称为事件,事件用来描述对象的动态特征。事件能够触发对象进入活动状态,当对象被事件触发时就可以执行对应的事件程序代码。

Visual Basic 为事件驱动的编程机制,应用程序是由事件驱动的,即只有当事件发生时,响应事件的程序代码才会运行。Visual Basic 程序运行的过程就是通过事件触发某个对象,随着该对象的活动又触发新的事件,新事件又触发另一个对象,对象就是以这种方式联系在一起的,如果没有事件发生,则整个程序就处于停滞状态。Visual Basic 编程的核心就是为每个事件编写响应事件的过程代码,不同对象响应不同事件的过程就构成了应用程序。

事件也有自己的名称,如命令按钮具有“单击事件”,事件名称的关键字是 Click。当用户单击命令按钮时,就会触发 Click 事件。其实每个事件都是一个函数,当事件被触发时,Visual Basic 就会转去执行该函数中的代码,实现相应的功能。如果在程序运行时单击命令按钮来完

成相应的功能,就要在该命令按钮的 Click 事件中编写代码。在 Visual Basic 中,事件的语法格式如下:

```
Private Sub 对象名_事件名()
```

```
    事件过程代码
```

```
End Sub
```

例如,命令按钮对象的名称是 Command1,其 Click 事件的代码如下:

```
Private Sub Command1_Click()
```

```
    事件过程代码
```

```
End Sub
```

注意:窗体的事件格式为 Form_事件名()。

1.3 Visual Basic 6.0 简介

Visual Basic(Basic, Beginner's All-purpose Symbolic Instruction Code)是由 Microsoft 公司开发的可视化、面向对象的程序设计语言。由于 Visual Basic 具有简单易学、可视化、面向对象、事件驱动的特性而受到用户欢迎,也成为专业人员开发 Windows 应用程序得心应手的开发工具。

1.3.1 Visual Basic 6.0 的功能特点及版本

1. Visual Basic 6.0 的功能特点

Visual Basic 具有强大的 Windows 应用程序开发功能,具有以下主要特点:

(1) 面向对象的可视化编程工具。是指应用面向对象的程序设计方法,把程序和数据封装成一个对象,并且大多数对象都是可视化的。用户可以利用这些对象方便地设计应用程序的图形界面,而不需要编写大量的程序代码去描述界面元素的外观和位置,并且可以通过对象属性的设置及方法和事件的使用,灵活地设计程序。

(2) 采用事件驱动编程机制。开发人员通过编写对象事件过程的程序代码来响应用户对对象的操作,在响应不同的事件时执行不同的代码。事件可以由用户操作触发,也可以由来自操作系统或其他应用程序的消息触发,甚至可以由应用程序本身的消息触发。事件的顺序决定了程序代码的执行顺序,应用程序每次运行时执行程序代码的顺序是不同的。每个事件都能驱动一段程序的运行,程序员的主要任务就是编写响应每个事件的程序代码。

(3) 集成的开发环境。Visual Basic 提供了集成开发应用程序的环境,用户可以在这个环境中,完成应用程序的界面设计、编写程序代码、调试程序、发布程序等所有程序设计步骤。

(4) 强大的数据库管理功能。Visual Basic 利用数据库控件可以直接或间接地访问数据库管理系统,如 Microsoft Access、Microsoft SQL Server、Oracle 等,Visual Basic 6.0 中增加了强大 ADO(ActiveX Database Object)控件,其使用更加方便,访问速度更快。

(5) ActiveX 技术。Visual Basic 支持对象链接与嵌入(OLE)技术,借助这种技术可以更方便地开发 Visual Basic 应用程序和扩展应用程序的功能。

(6) 联机帮助功能。与其他 Windows 软件一样,在 Visual Basic 中,可以利用菜单或在程序设计时按 F1 键,方便地获得 MSDN(Microsoft Develop Network Library)的帮助信息。

2. Visual Basic 6.0 的版本

Visual Basic 是随着 Windows 3.0 操作系统的推出而产生的。1991 年推出 Visual Basic

1.0 版,1992 年 11 月发布了 Visual Basic 2.0 版,1993 年 5 月推出了 Visual Basic 3.0 版,1995 年 9 月推出了 Visual Basic 4.0 版,1997 年 3 月发布了 Visual Basic 5.0 版,1998 年发布了 Visual Basic 6.0 版。目前 Visual Basic 6.0 版包含 3 个版本,可供不同需要的用户使用。

(1) 学习版(Learning Edition):是 Visual Basic 6.0 的基础版本,功能最简单,可以使学习编程的人员轻松入门和开发 Windows 应用程序。

(2) 专业版(Professional Edition):在学习版的基础上,增加了 ActiveX 控件、Internet 控件等工具,为专业人员提供了功能强大的开发环境。

(3) 企业版(Enterprise Edition):在专业版的基础上,增加了开发分布式应用程序等功能,并包括整套的 BackOffice 工具。

本书是以 Visual Basic 6.0 中文企业版为开发环境进行介绍的。

1.3.2 Visual Basic 6.0 的安装、启动及退出

1. Visual Basic 6.0 安装的硬件环境要求

(1) 处理器:80486 或者更高的处理器。

(2) 内存:对于 Windows NT 或更高版本的操作系统,至少需要 32 MB 内存。

(3) 硬盘:学习版的典型安装需要 48 MB 空间,完全安装需要 80 MB 空间;专业版的典型安装需要 48 MB 空间,完全安装需要 80 MB 空间;企业版的典型安装需要 128 MB 空间,完全安装需要 147 MB 空间;另外安装 MSDN 帮助文档需要 67 MB 空间。

(4) 显示器:VGA 或者更高分辨率的显示器。

(5) 光驱:8 倍速以上的 CD-ROM。

2. Visual Basic 6.0 安装的软件环境要求

(1) 操作系统:Microsoft Windows 95/98, Microsoft Windows NT4.0(Pack3)、Microsoft Windows XP 或者更新版本的操作系统。

(2) 浏览器:Microsoft Internet Explore 4.01 或者更高的版本。

2. Visual Basic 6.0 的安装过程

硬件环境和软件环境都符合安装条件后,一般安装过程需要经过以下步骤:

(1) 将 Visual Basic 6.0 中文企业版光盘放入光驱,启动后显示“Visual Basic 6.0 中文企业版安装向导”对话框,如图 1-1 所示。

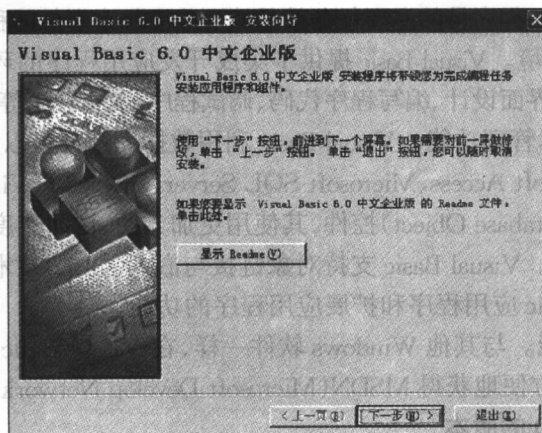


图 1-1 “Visual Basic 6.0 中文企业版安装向导”对话框

(2) 在图 1-1“Visual Basic 6.0 中文企业版安装向导”对话框中,单击“下一步”按钮,打开“最终用户许可协议”对话框,如图 1-2 所示。

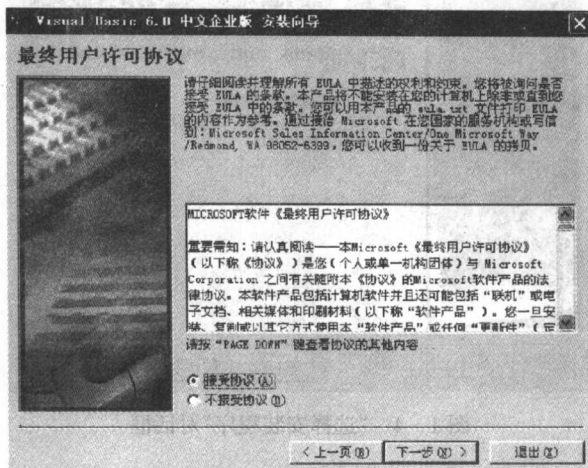


图 1-2 “最终用户许可协议”对话框

(3) 在图 1-2 的“最终用户许可协议”对话框中,选择“接受协议”后,单击“下一步”按钮,打开“产品号和用户 ID”对话框,如图 1-3 所示。

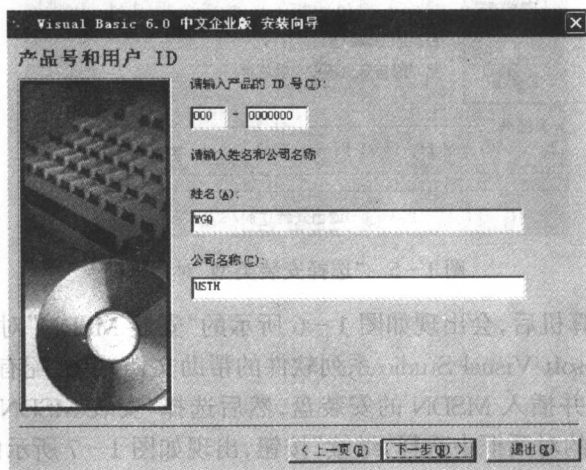


图 1-3 “产品号和用户 ID”对话框

(4) 在图 1-3 的“产品号和用户 ID”对话框中,输入产品 ID 号和用户信息后,单击“下一步”按钮,打开如图 1-4 所示的“选择安装程序”对话框。

(5) 在图 1-4 的“选择安装程序”对话框中,选择“安装 Visual Basic 6.0 中文企业版”,然后单击“下一步”按钮,出现如图 1-5 所示的对话框。如果选择“典型安装”系统会自动安装一些最常用的组件;如果选择“自定义安装”,用户可以根据实际需要有选择地安装组件。然后安装程序将文件复制到硬盘中,完成 Visual Basic 6.0 中文企业版的安装,并要求重新启动计算机,更新系统配置以使安装程序生效。