

21世纪高等学校规划教材
华东交通大学教材(专著)基金资助项目

Visual Basic 程序设计教程

熊李艳 周美玲 主编

吴昊 杜玲玲 副主编



21st Century University
Planned Textbooks

21 世纪高等学校规划教材
华东交通大学教材（专著）基金资助项目

Visual Basic 程序设计教程

熊李艳 周美玲 主编
吴昊 杜玲玲 副主编

21st Century University
Planned Textbooks

人民邮电出版社
样书
专用章

人 民 邮 电 出 版 社
北 京

图书在版编目 (C I P) 数据

Visual Basic程序设计教程 / 熊李艳, 周美玲主编.
北京: 人民邮电出版社, 2009. 9
21世纪高等学校规划教材
ISBN 978-7-115-21192-7

I. V… II. ①熊…②周… 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 (2009) 第145856号

内 容 提 要

本书从程序设计概念出发, 循序渐进地讲解了 Visual Basic 编程的基本理论和程序设计方法。内容主要包括 Visual Basic 的程序设计概念、基础知识、算法、程序控制三大结构、常用控件、数组、过程、文件、数据库基础等知识。

为了让初学者更好地掌握 Visual Basic (以下简称 VB) 的使用, 书中通过通俗易懂的语言和大量典型的实例, 帮助学生掌握 VB 程序设计语言的基本知识和 VB 程序设计的方法, 在每章的后面都配有一定数量的习题, 既巩固的所学知识, 又扩展学生的思路, 能进一步提高学生的自学能力。

本书可作为高校非计算机专业学生的计算机程序设计课程的教材, 也可作为成人教育、职业技术教育、工程技术人员及自学者的程序设计的教材, 还可作为计算机等级考试的辅导用书。

21 世纪高等学校规划教材

Visual Basic 程序设计教程

-
- ◆ 主 编 熊李艳 周美玲
副 主 编 吴 昊 杜玲玲
责任编辑 刘 博
 - ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号
邮编 100061 电子函件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
中国铁道出版社印刷厂印刷
 - ◆ 开本: 787×1092 1/16
印张: 17.25
字数: 452 千字
印数: 1—3 500 册
- 2009 年 9 月第 1 版
2009 年 9 月北京第 1 次印刷

ISBN 978-7-115-21192-7

定价: 29.00 元

读者服务热线: (010)67170985 印装质量热线: (010)67129223
反盗版热线: (010)67171154

前言

计算机科学与技术学科的迅速发展,推动着大学计算机教育相关的课程体系、课程内容和教学方法不断更新。为了贯彻落实《教育部关于进一步深化本科教学改革全面提高教学质量的若干意见》精神,深入研讨和推广计算机课程的教改新成果,我们在多年和其他高校合作的基础上自编了《Visual Basic 程序设计教程》和《Visual Basic 程序设计实践教程》教材。教材的编写将进一步推动计算机教学改革,全面提升计算机教学质量,改进计算机教学课程体系,推动精品课程建设。

全书理论教学与实践教学相结合,图文并茂,内容实用,层次分明,讲解清晰,针对高校学生的特点,采用案例教学方式,强调对学生动手能力的培养。通过实际案例,促进学生对基础知识、基本技能的掌握,实践教程配有用于拓展知识、提高创造能力的练习题和实验,既巩固所学知识,又扩展学生的思路,帮助学生掌握 Visual Basic (以下简称 VB) 程序设计语言的基本知识和 VB 程序设计的方法。

参编作者长期从事非计算机专业程序设计教学和教学研究工作,有较深的理论研究基础和教学改革实践基础,对计算机课程教学、课程体系进行了较深入的探索,对计算机课程建设有一定广度和深度的研究。本书由熊李艳、周美玲担任主编,参编人员有熊李艳(第1、3、4、7章)、周美玲(第2、6章)、吴昊(第8、9、11、12章)、杜玲玲(第5、10章),熊李艳负责本书最终的统稿。在制定编写大纲及书稿编写过程中,始终得到了华东交通大学信息工程学院领导的关心和支持,计算机基础部的雷莉霞、范萍、黎海生、刘媛媛、宋岚、张年、丁振凡、蔡体键、段楠楠、李明翠、李蓉、李卓群、莫佳、王鹏鸣、秦永红、游敏给了作者大力帮助,参与了大纲的编写与程序的调试,他们为本书的最终完成付出了很多劳动,在此表示由衷的感谢。

根据“立体化”教材体系的要求,除配套教材外,作者还提供电子教案、习题答案等教材中涉及的相关教学资源。

由于编者水平有限,编写时间仓促,书中难免有欠妥之处,恳请广大读者提出宝贵意见。

编 者

2009年6月

目 录

第 1 章 引言1	2.1.1 Visual Basic 的字符集.....29
1.1 程序与程序设计语言.....1	2.1.2 Visual Basic 的词汇集.....29
1.2 Visual Basic 简介.....3	2.1.3 编码规则与约定.....30
1.2.1 Visual Basic 的发展.....3	2.2 数据类型.....32
1.2.2 Visual Basic 的功能特点.....4	2.3 常量与变量.....36
1.2.3 Visual Basic 的安装和启动.....5	2.3.1 常量.....36
1.3 Visual Basic 的集成开发环境.....7	2.3.2 变量.....39
1.3.1 标题栏.....8	2.4 运算符与表达式.....41
1.3.2 菜单栏.....8	2.4.1 算术运算符与数值表达式.....42
1.3.3 工具栏.....9	2.4.2 字符串运算符与字符串表达式.....43
1.3.4 窗体设计器.....10	2.4.3 关系运算符与关系表达式.....44
1.3.5 工具箱.....10	2.4.4 逻辑运算符与逻辑表达式.....46
1.3.6 工程资源管理器窗口.....11	2.4.5 日期型运算符和日期表达式.....47
1.3.7 属性窗口.....12	2.4.6 混合运算的执行顺序.....48
1.3.8 代码窗口.....13	2.5 常用内部函数.....49
1.4 建立简单的应用程序.....13	2.5.1 数学函数.....49
1.4.1 分析问题、设计算法.....14	2.5.2 字符串函数.....50
1.4.2 设计应用程序用户界面.....14	2.5.3 转换函数.....52
1.4.3 对象属性的设置.....15	2.5.4 日期、时间函数.....53
1.4.4 编写程序代码.....15	2.5.5 格式输出函数.....54
1.4.5 调试运行程序.....16	2.5.6 其他函数.....56
1.4.6 保存程序文件.....17	
1.5 面向对象的程序设计语言.....17	第 3 章 Visual Basic 程序初步57
1.5.1 对象和类.....18	3.1 结构化程序设计.....57
1.5.2 对象的属性、方法和事件.....18	3.2 赋值语句.....58
1.6 窗体.....20	3.3 数据输出和输入.....59
1.6.1 窗体的基本属性.....20	3.3.1 Print 方法.....59
1.6.2 窗体的事件.....24	3.3.2 特殊打印格式.....60
1.6.3 窗体的方法.....25	3.3.3 Cls 方法.....61
1.6.4 焦点与 Tab 键序.....26	3.3.4 输入对话框.....61
	3.3.5 消息对话框.....63
第 2 章 Visual Basic 语言基础29	3.4 常用语句.....65
2.1 语言基础.....29	

3.5 基本控件	66	6.3 动态数组及其声明	118
3.5.1 标准控件	66	6.4 数组的基本操作	119
3.5.2 命令按钮	70	6.4.1 数组元素的赋值	120
3.5.3 标签	72	6.4.2 与数组相关的函数	121
3.5.4 文本框	74	6.4.3 数组的复制	122
第 4 章 选择结构程序设计	79	6.4.4 数组的输出	122
4.1 算法介绍	79	6.4.5 For Each ... Next 循环语句	123
4.1.1 算法介绍	79	6.5 控件数组	124
4.1.2 算法的表示	80	6.5.1 控件数组的概念	124
4.2 条件语句	83	6.5.2 控件数组的建立	125
4.2.1 单分支结构语句	83	6.6 列表框和组合框	126
4.2.2 双分支语句	84	6.6.1 列表框	126
4.2.3 If 函数	87	6.6.2 组合框	132
4.2.4 多支结构语句	87	6.7 数组应用举例	135
4.2.5 Choose 函数	92	第 7 章 常用控件	139
4.3 选择结构的嵌套	93	7.1 单选按钮	139
第 5 章 循环结构程序设计	98	7.1.1 常用属性	139
5.1 循环语句	98	7.1.2 常用事件	140
5.1.1 For...Next 语句	98	7.2 复选框	141
5.1.2 Do...Loop 语句	100	7.2.1 常用属性	141
5.1.3 While...Wend 语句	102	7.2.2 常用事件	141
5.2 多重循环	103	7.3 框架	142
5.3 其他控制语句	105	7.3.1 对象的创建	142
5.3.1 Exit 语句	105	7.3.2 常用属性	143
5.3.2 With ... End With 语句	106	7.3.3 常用事件	143
5.3.3 GoTo 语句	106	7.4 滚动条	145
5.4 应用举例	107	7.4.1 常用属性	145
第 6 章 数组	114	7.4.2 常用事件	146
6.1 数组的概念	114	7.5 图片框	148
6.1.1 引例	114	7.5.1 常用属性	148
6.1.2 数组简介	115	7.5.2 常用方法和事件	149
6.2 静态数组及其声明	116	7.6 图像框	150
		7.7 计时器	151
		7.7.1 常用属性	151
		7.7.2 常用事件	151

第 8 章 过程159	9.2.1 ActiveX 控件.....193
8.1 过程概述.....159	9.2.2 工具栏.....194
8.2 函数过程.....159	9.2.3 状态栏.....197
8.2.1 Function 过程的定义.....159	9.3 对话框.....198
8.2.2 Function 过程的调用.....162	9.3.1 通用对话框.....199
8.3 Sub 过程.....163	9.3.2 自定义对话框.....204
8.3.1 事件过程和通用过程.....163	9.4 多重窗体程序设计与多文档程序 设计.....204
8.3.2 Sub 过程的定义.....164	9.4.1 多窗体程序设计.....204
8.3.3 Sub 过程的调用.....165	9.4.2 多文档程序设计.....207
8.4 参数传递.....167	9.5 Sub Main 过程.....210
8.4.1 按址传递与按值传递.....167	9.6 鼠标与键盘事件.....210
8.4.2 数组参数的传递.....171	9.6.1 键盘事件.....210
8.5 过程的嵌套与递归调用.....173	9.6.2 鼠标事件.....211
8.5.1 过程的嵌套调用.....173	第 10 章 文件213
8.5.2 过程的递归调用.....174	10.1 文件的概念.....213
8.6 模块.....175	10.2 顺序文件.....214
8.6.1 窗体模块.....176	10.3 随机文件.....218
8.6.2 标准模块.....176	10.3.1 用户自定义类型.....219
8.6.3 类模块.....177	10.3.2 随机文件操作.....220
8.7 过程与变量的作用域.....177	10.4 常用的文件操作语句和函数.....224
8.7.1 过程的作用域.....177	10.4.1 文件操作语句.....224
8.7.2 变量的作用域.....178	10.4.2 文件操作函数.....226
8.8 变量的生存期.....181	10.5 文件系统控件.....227
8.8.1 动态变量.....181	10.5.1 驱动器列表框.....227
8.8.2 静态变量.....181	10.5.2 目录列表框.....228
8.9 程序举例.....182	10.5.3 文件列表框.....228
第 9 章 界面设计186	10.5.4 文件系统控件的联动应用.....229
9.1 菜单设计.....186	第 11 章 图形操作231
9.1.1 菜单编辑器.....186	11.1 度量系统.....231
9.1.2 下拉式菜单.....189	11.2 坐标系统.....231
9.1.3 菜单的 Click 事件.....190	11.3 绘图属性.....232
9.1.4 运行时动态改变菜单属性.....191	11.4 图形控件.....234
9.1.5 弹出式菜单.....192	11.5 图形方法与事件.....235
9.2 工具栏和状态栏.....193	

11.6 程序举例	238	12.3 数据库控件	250
第 12 章 数据库应用	240	12.3.1 数据控件	250
12.1 数据库基础	240	12.3.2 数据绑定控件	254
12.1.1 数据库的基本概念	240	12.3.3 记录集对象	255
12.1.2 关系数据库	241	12.4 ADO 数据访问对象	257
12.1.3 Visual Basic 数据库应用 系统	243	12.4.1 ADO 控件使用基础	257
12.2 数据管理器的使用	243	12.4.2 创建 ADO 控件	257
12.2.1 创建数据库	243	12.4.3 ADO 控件的常用属性、方法与 事件	257
12.2.2 打开数据库	245	12.4.4 ADO 数据绑定控件	260
12.2.3 添加数据表	246	附录 1 ASC II 字符集	263
12.2.4 修改数据表结构	247	附录 2 常用内部函数	264
12.2.5 用户数据的编辑	248	附录 3 Visual Basic 常用属性	266
12.2.6 数据窗体设计器	249		

第 1 章

引言

1.1 程序与程序设计语言

程序设计语言（Programming Language）是人们为了描述计算过程而设计的一种具有语法语义的记号，是一组用来定义计算机程序的语法规则。它是一种被标准化的交流技巧，用来向计算机发出指令。一种计算机语言让程序员能够准确地定义计算机所需要使用的数据，并精确地定义在不同情况下应当采取的行动。计算机程序是一组指令的组合，这组指令依据既定的逻辑控制计算机的运行。

对计算机工作人员而言，程序设计语言是除计算机本身之外的所有工具中最重要工具，是其他所有工具的基础。没有程序设计语言的支持，计算机无异于一堆废料。由于程序设计语言的这种重要性，从计算机问世至今的半个多世纪中，人们一直在为研制更新更好的程序设计语言而努力着。程序设计语言的数量在不断增加，各种新的程序设计语言以及现有程序设计语言的新版本在不断面世。目前已问世的各种程序设计语言有成千上万个，但这其中只有极少数得到了人们的广泛认可。

程序设计语言包含 3 个方面，即语法、语义和语用。语法表示程序的结构或形式，亦即表示构成程序的各个记号之间的组合规则，但不涉及这些记号的特定含义，也不涉及使用者；语义表示程序的含义，亦即表示按照各种方法所表示的各个记号的特定含义；语用表示程序与使用的关系。

计算机程序设计语言的发展，总的来说，经历了从机器语言、汇编语言到高级语言的历程。

1. 机器语言

电子计算机所使用的是由“0”和“1”组成的二进制数，二进制是计算机语言的基础。计算机发明之初，人们只能写出一串串由“0”和“1”组成的指令序列交由计算机执行，这种语言，就是机器语言。使用机器语言是十分痛苦的，特别是在程序有错需要修改时，更是如此。而且，由于每台计算机的指令系统往往各不相同，所以，在一台计算机上执行的程序，要想在另一台计算机上执行，必须另编程序，造成了重复工作。但由于使用的是针对特定型号计算机的语言，故而运算效率是所有语言中最高的。机器语言，是第一代计算机语言。

2. 汇编语言

为了减轻使用机器语言编程的痛苦，人们进行了一种有益的改进：用一些简洁的英文字母、符号串来替代一个特定的指令的二进制串，比如，用“ADD”代表加法，“MOV”代表数据传递

等，这样一来，人们很容易读懂并理解程序在干什么，纠错及维护都变得方便了，这种程序设计语言就称为汇编语言，即第二代计算机语言。然而计算机是不认识这些符号的，这就需要一个专门的程序，专门负责将这些符号翻译成二进制数的机器语言，这种翻译程序被称为汇编程序。

汇编语言同样十分依赖于机器硬件，移植性不好，但效率仍然很高，针对计算机特定硬件而编制的汇编语言程序，能准确发挥计算机硬件的功能和特长，程序精炼而质量高，所以至今仍是一种常用而强有力的软件开发工具。

3. 高级语言

从最初与计算机交流的痛苦经历中，人们意识到，应该设计一种这样的语言，这种语言接近于数学语言或人的自然语言，同时又不依赖于计算机硬件，编出的程序能在所有机器上通用。经过努力，1954 年，第一个完全脱离机器硬件的高级语言——FORTRAN 问世了，50 多年来，共有几百种高级语言出现，有重要意义的有几十种，影响较大、使用较普遍的有 FORTRAN、ALGOL、COBOL、BASIC、LISP、NOBOL、P L/1、Pascal、C、PROLOG、Visual C++、Visual Basic、Delphi、Java 等。

高级语言的出现使得计算机程序设计语言不再过度依赖某种特定的机器或环境。这是因为高级语言在不同的平台上会被编译成不同的机器语言，而不是直接被机器执行。最早出现的编程语言之一 FORTRAN 的一个主要目标，就是实现平台独立。

虽然大多数的语言可以既被编译又被解释，但大多数只在一种情况下能够良好运行。在一些编程系统中，程序要经过几个阶段的编译，一般而言，后阶段的编译往往更接近机器语言。这种常用的使用技巧最早在 1960 年代末用于 BCPL，编译程序先编译一个叫做“0 代码”的转换程序，然后再使用虚拟器转换到可以运行于机器上的真实代码。这种成功的技巧之后又用于 Pascal 和 P-code，以及 Smalltalk 和二进制码，虽然在很多时候，中间过渡的代码往往是解译，而不是编译的。

如果所使用的翻译机制是将所要翻译的程序代码作为一个整体翻译，其后运行内部格式，那么这个翻译过程就被称为编译。因此，一个编译器是一个将人可阅读的程序文本（叫做源代码）作为输入的数据，然后输出可执行文件（Object Code）。所输出的可执行文件可以是机器语言，由计算机的中央处理器直接运行，或者是某种模拟器的二进制代码。

如果程序代码是在运行时才即时翻译，那么这种翻译机制就被称作解译。经解译的程序运行速度往往比编译的程序慢，但往往更具灵活性，因为它们能够与执行环境互相作用。

高级语言的发展也经历了从早期语言到结构化程序设计语言，从面向过程到非过程化程序设计语言的过程。相应地，软件的开发也由最初的个体手工作坊式的封闭式生产，发展为产业化、流水线式的工业化生产。

20 世纪 60 年代中后期，软件越来越多，规模越来越大，而软件的生产基本上是各自为战，缺乏科学规范的系统规划与测试、评估标准，其恶果是大批耗费巨资建立起来的软件系统，由于含有错误而无法使用，甚至带来巨大损失，软件给人的感觉是越来越不可靠，以致几乎没有不出错的软件。这一切，极大地震动了计算机界，史称“软件危机”。人们认识到大型程序的编制不同于写小程序，它应该是一项新的技术，应该像处理工程一样处理软件研制的全过程。程序的设计应易于保证正确性，也便于验证正确性。1969 年，提出了结构化程序设计方法，1970 年，第一个结构化程序设计语言 Pascal 语言的出现，标志着结构化程序设计时期的开始。

从 20 世纪 80 年代初开始，在软件设计思想上，又产生了一次革命，其成果就是面向对象的程序设计。在此之前的高级语言，几乎都是面向过程的，程序的执行是流水线似的，在一个模块

被执行完成前，人们不能干别的事，也无法动态地改变程序的执行方向。这和人们日常处理事物的方式是不一致的，对人而言是希望发生一件事就处理一件事，也就是说，不能面向过程，而应面向具体的应用功能，也就是对象（Object）。其方法就是软件的集成化，如同硬件的集成电路一样，生产一些通用的、封装紧密的功能模块，称之为软件集成块，它与具体应用无关，但能相互组合，完成具体的应用功能，同时又能重复使用。对使用者来说，只关心它的接口（输入量、输出量）及能实现的功能，至于如何实现的，那是它内部的事，使用者完全不用关心，C、VB、Delphi 就是典型代表。有许多用于特殊用途的语言，只在特殊情况下使用。例如，PHP 专门用来显示网页；VB 被广泛用于操作系统和编译器的开发（所谓的系统编程）。

高级语言的下一个发展目标是面向应用，也就是说，只需要告诉程序要干什么，程序就能自动生成算法，自动进行处理，这就是非过程化的程序语言。

1.2 Visual Basic 简介

Visual Basic（以下简称 VB）是 Microsoft 公司在 Basic 语言的基础之上推出的软件开发工具，它是一种基于对象的可视化编程语言，是目前 Windows 平台上设计应用程序最为快捷的工具之一。

何谓 Visual Basic？“Visual”指的是开发图形用户界面（GUI）的方法。不需编写大量代码去描述界面元素的外观和位置，而只要把预先建立的对象加到屏幕上，一点即可。如果已使用过诸如 Paint 之类的绘图程序，则实际上已掌握了创建用户界面的必要技巧。

“Basic”指的是 Beginners All-Purpose Symbolic Instruction Code，一种在计算机技术发展历史上应用得最为广泛的语言。Visual Basic 在原有 Basic 语言的基础上进一步发展，至今包含了数百条语句、函数及关键词，其中很多和 Windows GUI 有直接关系。专业人员可以用 Visual Basic 实现其他任何 Windows 编程语言的功能，而初学者只要掌握几个关键词就可以建立实用的应用程序。

Visual Basic 不仅是 Visual Basic 编程语言。Visual Basic 编程系统和 VBA 都使用这一语言。Visual Basic Scripting Edition（VBScript）是广泛使用的脚本语言，它是 Visual Basic 语言的子集。这样，在学习 Visual Basic 中得到的经验可应用到所有这些领域中。从开发个人或小组使用的小工具，到大型企业应用系统，甚至通过 Internet 的遍及全球分布式应用程序，都可在 Visual Basic 提供的工具中各取所需。

1.2.1 Visual Basic 的发展

1991 年，Visual Basic 1.0 正式发布。作为微软起家产品，Visual Basic 1.0 发布伊始就像雷电一般征服了广大开发者，通过完备的开发环境和跨时代的直观性取代了 QBasic 在广大开发者心目中的地位，成为那时新一代的当红软件。

1992 年，随着 Windows 系统被用户接受度的提高和对新平台开发的需求，微软推出了具有部分面对对象特征和支持 VBX（ActiveX）控件的 Visual Basic 2.0。

1993 年初，Visual Basic 3.0 便快速发布了，Visual Basic 3.0 强调几个重要特性，一个是 OLE 2.0 的概念，另一个则是数据库存取，逐步体现出 Visual Basic 作为企业及开发工具的优势。

1995 年，Visual Basic 4.0 发布。Visual Basic 4.0 支持 OCX 控件，同时语言换为 Visual Basic For

Application, 与 Office 95 所采用的宏语言统一起来。这个版本还为 Visual Basic 成为一种 COM 语言奠定了基础。

1997 年, Visual Basic 5.0 发布, Visual Basic 对 COM(这时候叫 ActiveX)提供了最强的支持。同时为了提高效率加入了本地代码编译器。并且对 Visual Basic For Application 语言有重大的完善和丰富。

1998 年, Visual Basic 6.0 发布, 它不仅改进了 Visual Basic 5.0, 更整合了其他开发工具, 成为 Visual Studio 家族性产品。这表明微软公司此时确立了 Visual Basic 的产品定位, 使之成为企业级快速开发的利器。

2002 年, Visual Basic.NET 发布, 对 CLR 有非常完善的支持, 尽量保留着 BASIC 易懂的语法风格和易用性, 以及完成了向面向对象转变的道路。这个版本大大减少了 Visual Basic 语言的关键字(原来共有 120 个关键字), 使其成为了一门现代的、强壮的、面向对象的、简单的可视化开发语言。

2003 年, Visual Basic.NET 2003 发布。这版本改进了 Visual Basic 6.0 升级向导。其次, 在 IDE 方面, Visual Basic.NET 2003 修正了许多错误, 还让事件处理恢复了 Visual Basic 6.0 的开发体验。

2005 年, Visual Basic 2005 发布。Visual Basic 2005 在语言上的有非常大的改动, 如泛型、运算符重载、Partial Type、窗体的默认实例、语法的多种改进、My 关键字等。Visual Basic 2005 能异常便捷地开发 .NET 下的应用。而泛型、运算符重载等功能又可以让 Visual Basic 的开发人员深入 .NET Framework 的全部功能。

2007 年, Visual Basic 2008 发布。Visual Basic 2008 进一步简化了源代码, 并可以实现与使用高级功能的组件进行交互, 并且引入了新的语言功能, 其中包括局部类型推理、对象初始值设定项、匿名类型和扩展方法。这些功能是对一项很重要的新功能——语言集成查询(LINQ)的支持。同时还提供了更完备的向下兼容, 将许多 Visual Basic 6.0 ActiveX 控件升级为 Visual Basic 2008 的有效控件。

本书主要介绍 Visual Basic 6.0 版。Visual Basic 6.0 包括 3 种版本, 分别为学习版(标准版)、专业版和企业版 3 种版本, 以满足不同的开发需求。

3 种版本中, 企业版功能最全, 专业版包括了学习版的功能。用户可根据自己的需要购买不同的版本。对于大多数用户来说, 专业版完全可以满足需要。

本书使用的是 Visual Basic 6.0 中文企业版, 但其内容可用于专业版和学习版, 所有程序都可以在专业版和学习版中运行。

Visual Basic 6.0 是专门为 Microsoft 的 32 位操作系统设计的, 可用来建立 32 位的应用程序。在 Windows 9x、Windows NT 以及 Windows 2000 或 Windows XP 环境下, 用 Visual Basic 6.0 可以自动生成 32 位应用程序。这样的应用程序在 32 位操作系统下运行, 速度更快、更安全, 并且更适合在多任务环境下运行。

1.2.2 Visual Basic 的功能特点

1. 面向对象的程序设计

VB 采用了面向对象的程序设计思想。它的基本思路是把复杂的程序设计问题分解为一个能够完成独立功能的相对简单的对象集合, 所谓“对象”就是一个可操作的实体, 如窗体、窗体中的命令按钮、标签、文本框等。面向对象的编程就好像搭积木一样, 程序员可根据程序和界面设计要求, 直接在屏幕上“画”出窗口、菜单、按钮等不同类型的对象, 并为每个对象设置属性。

2. 事件驱动编程机制

在 Windows 环境下, 程序是以事件驱动方式运行的, 每个对象都能响应多个不同的事件, 每个事件都能驱动一段代码——事件过程, 该代码决定了对象的功能。通常称这种机制为事件驱动。事件可由用户的操作触发, 也可以由系统或应用程序触发。例如, 单击一个命令按钮, 就触发了按钮的 Click (单击) 事件, 该事件中的代码就会被执行。若用户未进行任何操作 (未触发事件), 则程序就处于等待状态。整个应用程序就是由彼此独立的事件过程构成的。

3. 提供应用程序集成开发环境

VB 为编程提供了一个集成开发环境。在这个环境中, 编程者可设计界面、编写代码、调试程序, 直至把应用程序编译成可在 Windows 中运行的可执行文件, 并为它生成安装程序。VB 的集成开发环境为编程者提供了很大的方便。

4. 结构化的程序设计语言

VB 具有丰富的数据类型, 是一种符合结构化程序设计思想的语言, 而且简单易学。

5. 强大的数据库访问功能

VB 利用数据控件可以访问多种数据库, VB 6.0 提供的 ADO 控件, 不但可以用最少的代码实现数据库操作和控制, 也可以取代 Data 控件等控件。

6. 支持对象的链接与嵌入技术

VB 的核心是对对象的链接与嵌入 (OLE) 技术的支持, 它是访问所有对象的一种方法。利用 OLE 技术, 能够开发集声音、图像、动画、字处理、Web 等对象于一体的程序。

7. 网络功能

VB 6.0 提供了 Dynamic HTML 设计工具。利用这种技术可以动态创建和编辑 Web 页面, 使用户可在 VB 中开发多功能的网络应用软件。

8. 多个应用程序向导

VB 提供了多种向导, 如应用程序向导、安装向导、数据对象向导和数据窗体向导, 通过它们可以快速地创建不同类型、不同功能的应用程序。

9. 支持动态交换、动态链接技术

动态数据交换 (DDE) 的编程技术, VB 开发的应用程序能与其他 Windows 应用程序之间建立数据通信。通过动态链接库技术, 在 VB 程序中可方便地调用用 C 语言或汇编语言编写的函数, 也可调用 Windows 的应用程序接口 (API) 函数。

10. 联机帮助功能

在 VB 中, 利用帮助菜单和 F1 功能键, 用户可随时方便地得到所需要的帮助信息。VB 帮助窗口中显示了有关的示例代码, 通过复制、粘贴操作可获取大量的示例代码, 为用户的学习和使用提供方便。

1.2.3 Visual Basic 的安装和启动

1. 安装

安装 Visual Basic 之前, 必须确认计算机满足最低的安装要求, 并阅读安装盘根目录下的 Readme 文件。

为运行 Visual Basic, 必须在计算机上安装相应的硬件和软件系统。这些系统要求包括: Microsoft Windows NT 95 或更高版本, 或 Microsoft Windows NT Workstation 4.0 (推荐 Service Pack 3) 或更高版本; 486DX/66 MHz 或更高的处理器 (推荐 Pentium 或更高的处理器), 或任何运行于 Microsoft

Windows NT Workstation 的 Alpha 处理器；一个 CD-ROM 驱动器；Microsoft Windows 支持的 VGA 或分辨率更高的监视器；16MB RAM for Windows 95、32 MB of RAM for Windows NT Workstation。

将 VB 6.0 中文企业版安装光盘放入光驱，即可自动运行安装程序，如果不能自动运行安装程序，可查找安装文件 SETUP.EXE，双击运行。依照屏幕上的安装指令行事（安装过程中除了要输入产品 ID 外，一般都使用默认的设置，即总是单击“下一步”或“确定”）。

2. 启动

当运行 Visual Basic 安装程序时，允许将程序项置于已存在的程序组中，或在 Windows 中为 Visual Basic 创建一个新的程序组和程序项。这时可准备从 Windows 启动 Visual Basic。

单击任务条上的“开始”，选择“程序”，接着选取“Microsoft Visual Basic 6.0”，然后双击 Visual Basic 图标。

也可以创建一个 Visual Basic 快捷键，并双击该快捷键。

用上面所介绍的任何一种方法启动 Visual Basic 6.0 后，将首先显示版权屏幕，说明此份程序复制的使用权属于谁。稍候，显示“新建工程”对话框，如图 1-1 所示。

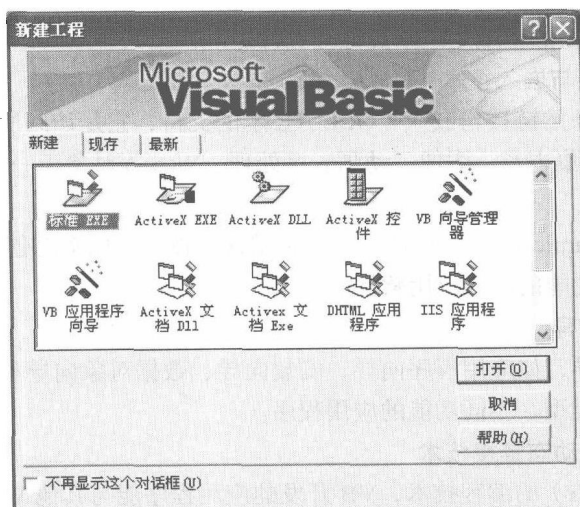


图 1-1 “新建工程”对话框

对话框中显示出可以在 Visual Basic 6.0 中使用的工程类型，主要有以下几种。

标准 EXE：建立标准 Windows 下的可执行文件（.EXE 文件），它是“新建”选项卡中最基本的类型，也是默认类型。

ActiveX EXE：这种程序只能在专业版和企业版中建立，用于建立进程外的对象的链接与嵌入服务器应用程序项目类型。这种程序可包装成可执行文件。

ActiveX DLL：这种程序与 ActiveX EXE 程序是一致的，只是包装不一样，ActiveX DLL 只能包装成动态链接库。

ActiveX 控件：只能在专业版或企业版中建立，用于开发用户自定义的 ActiveX 控件。

VB 应用程序向导：用于在开发环境中建立新的应用程序框架。

数据工程：提供开发数据报表应用程序的框架，选中该图标后，将自动打开数据环境设计器和数据报表设计器。

IIS 应用程序：这种程序是一个生存在 Web 服务器上并响应浏览器请求的应用程序，使用

HTML 来表示它的用户界面，使用编译的 Visual Basic 代码来处理浏览器的请求与响应事件。

外接程序：用于建立 Visual Basic 外接程序，并在开发环境中自动打开连接设计器。

ActiveX 文档 EXE 和 ActiveX 文档 DLL：建立可以在超链接环境中运行的 Visual Basic 应用程序，即 Web 浏览器。

VB 企业版控件：该选项不是用来建立应用程序的，而是用来在工具箱中加入企业版控件图标的。

在对话框中选择要建立的工程类型，初学者应选择“标准 EXE”，然后单击“打开”按钮，会出现图 1-2 所示的集成开发环境界面。

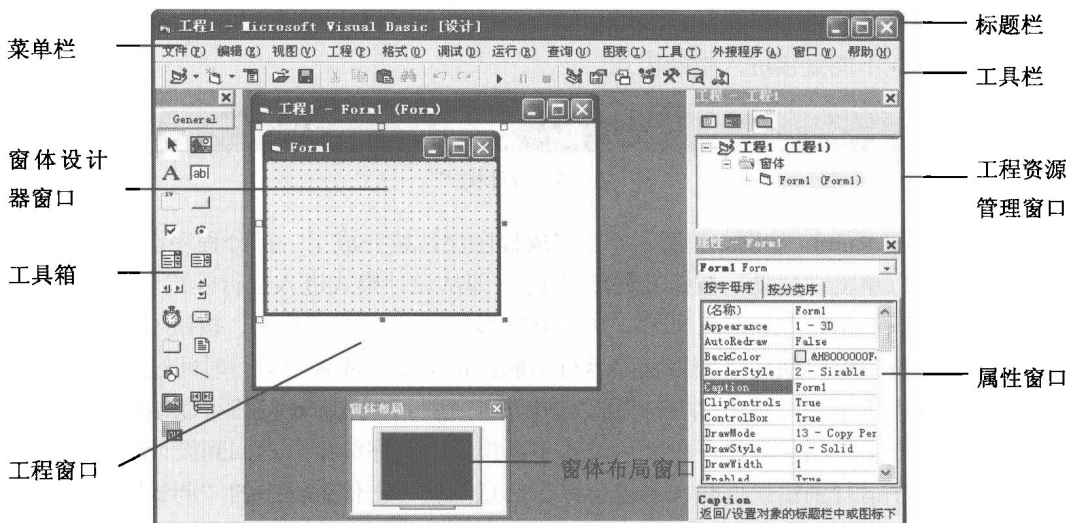


图 1-2 Visual Basic 的集成开发环境

3. 退出

以下 4 种方式，均可以退出 Visual Basic 6.0 应用程序。

- ① 选择“文件”菜单中的“退出”。
- ② 单击窗口右上角关闭按钮。
- ③ 双击窗口左上角的控制图标。
- ④ 按下 Alt+Q 快捷键。

在退出 Visual Basic 时，如果是新建的程序或是已修改过的原有程序没有存盘，系统将显示一个对话框，询问用户是否将其存盘，可以选择“是”（存盘）或“否”（不存盘），这两个选择都可以退出 Visual Basic。

1.3 Visual Basic 的集成开发环境

在启动 VB 6.0 后，可见到如图 1-2 所示的界面，这就是 VB 6.0 的集成开发环境（IDE）。在屏幕上将看到有菜单栏、工具栏、控件工具箱、窗体、工程资源管理器、属性窗口和窗体布局窗口，另外还有隐藏的代码编辑窗口等。

1.3.1 标题栏

标题栏是屏幕顶部的水平条，通常为蓝色，显示当前应用程序的名字。标题栏中含有当前工程的名字以及最小化、最大化、关闭窗口等几个常规按钮。启动 Visual Basic 后，标题栏中显示的信息为：工程 1-Microsoft Visual Basic[设计]。

标题栏中还同时显示当前 Visual Basic 的工作模式，共有 3 种：设计模式（Design）、运行模式（Run）和中断模式（Debug）。

① 设计模式（Design）：进入 Visual Basic 集成环境后，首先进入设计模式，如图 1-3 所示，可进行用户界面的设计，对象属性的设计、代码的编辑、保存文件和编译文件等操作，从而完成程序的开发。在设计模式下只有“启动”按钮可以使用，“中断”按钮和“结束”按钮不能使用，就是说由设计模式不能直接进入中断模式。

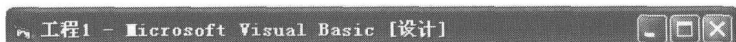


图 1-3 设计模式

② 运行模式（Run）：如图 1-4 所示，在完成程序设计或完成了一部分程序设计时，想要看一下运行结果，可以单击“启动”按钮或快捷键 F5，即从设计模式进入运行模式。在运行模式下，集成环境窗口中只保留菜单栏和工具栏，其他窗口都消失。这时，“启动”按钮不能使用，“中断”按钮和“结束”按钮可以使用。如果程序运行不能正常结束，或者因运行时间过长而要停止程序运行，这时需要人工干预。单击  按钮可以中断程序运行，单击  按钮可以结束程序运行。中断程序和结束程序是不同的，程序中断后进入中断模式，程序结束后返回到设计模式。如果程序存在错误，当程序运行到错误处时，系统会弹出信息框，单击信息框中的“调试”按钮可以转换到中断模式，单击信息框中的“结束”按钮可以回到设计模式。



图 1-4 运行模式

③ 中断模式（Debug）：如图 1-5 所示，在运行模式下，当程序出现错误或单击“中断”按钮时，Visual Basic 都会进入中断模式。在中断模式下，可以修改程序代码，这时鼠标指针指向  按钮，屏幕显示该按钮功能的提示信息是“继续”，单击  按钮后，程序将从中断处继续运行。在中断模式下，“启动”按钮和“结束”按钮都可以使用。单击  按钮将回到运行模式，单击  按钮将回到设计模式。

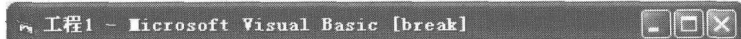


图 1-5 中断模式

1.3.2 菜单栏

菜单栏用于显示所使用的 Visual Basic 命令。除了提供标准“文件”、“编辑”、“视图”、“窗口”和“帮助”菜单之外，还提供了编程专用的功能菜单，如“工程”、“格式”或“调试”。下面给出一些菜单的主要功能。

① 文件：用于创建、打开、添加、移除、保存、另存为、显示最近的工程以及生成可执行文

件，打印和退出系统等子菜单项。

- ② 编辑：用于对工程项目中的文本代码进行操作，包括剪切、复制、粘贴、删除、查找、替换、插入文件等编辑命令。
- ③ 视图：用于对各窗口进行操作，通过选择视图菜单项来显示各窗口。
- ④ 工程：用于在设计时对工程进行管理，设置工程属性，添加、删除窗体、控件等。
- ⑤ 格式：用于窗体、控件的格式设计，如对齐、间距、尺寸等。
- ⑥ 调试：用于程序的调试、查错等。
- ⑦ 运行：用于启动程序或编译执行。
- ⑧ 查询：用于设计数据库 SQL 查询、运行结果查询、语言语法查询等。
- ⑨ 图表：用于建立数据库中的表，如新建、设置、添加、显示、修改图表等。
- ⑩ 工具：用于建立 ActiveX 控件时需要的工具命令，并可以启动菜单编辑器以及配置环境选项。

每个菜单项含有若干个菜单命令，执行不同的操作。用鼠标单击某个菜单项，即可打开该菜单，然后用鼠标单击菜单中的某一条就能执行相应的菜单命令。

1.3.3 工具栏

在编程环境下提供对于常用命令的快速访问。单击工具栏上的按钮，则执行该按钮所代表的操作。按照默认规定，启动 Visual Basic 之后显示“标准”工具栏。附加的编辑、窗体设计和调试的工具栏可以从“视图”菜单上的“工具栏”命令中移进或移出。

工具栏能紧贴在菜单条之下，或以垂直条状紧贴在左边框上，如果将它从菜单下面拖开，则它能“悬”在窗口中。

表 1-1 工具栏图标的名称和作用

图 标	名 称	作 用	快 捷 键
	添加工程	添加一个新工程，相当于“文件”菜单中的“添加工程”命令	
	添加窗体	用于添加新的窗体到工程中，单击其右边的箭头，将弹出一个下拉菜单，可从选择需要的窗体类型	
	菜单编辑器	打开菜单编辑对话框	Ctrl+E
	打开工程	用来打开一个已经存在的 Visual Basic 工程文件	Ctrl+O
	保存工程（组）	保存当前的 Visual Basic 工程（组）文件	
	剪切	把选择的内容剪切到剪贴板	Ctrl+X
	复制	把选择的内容复制到剪贴板	Ctrl+C
	粘贴	把剪贴板的内容复制到当前插入位置	Ctrl+V
	查找	打开“查找”对话框	Ctrl+F
	撤销	撤销当前的修改	Ctrl+Z
	重复	对“撤销”的反操作	
	启动	用来运行一个应用程序	F5
	中断	暂停正在运行的程序（可以单击“启动”按钮或按 Shift+F5 快捷键继续	Ctrl+Break