

全国计算机等级 考试二级教程



教育部考试中心

——Visual Basic 语言程序设计 (2017年版)

高等教育出版社



全国计算机等级考试二级教程

——Visual Basic 语言程序设计

(2017 年版)

Quanguo Jisuanji Dengji Kaoshi Erji Jiaocheng
——Visual Basic Yuyan Chengxu Sheji

教育部考试中心

刘炳文 杨明福 陈定中 编

高等教育出版社·北京

内容提要

本书是在教育部考试中心组织下,依据《全国计算机等级考试二级 Visual Basic 语言程序设计考试大纲(2013年版)》的要求,在上一版的基础上修订而成。主要内容包括 Visual Basic 6.0 中文版的编程环境、常用内部控件的功能和用法、控制结构、数组、过程、菜单、对话框、多窗体程序设计技术和鼠标、键盘事件以及文件处理等。本书对上一版中的部分章节进行了精减,更加突出了 Visual Basic 6.0 中文版的功能及易学易用的编程特色。大量的例题及习题练习则是本书的另一特点。

针对初学者的特点,全书在编排上注意了由简及繁、由浅入深和循序渐进,力求通俗易懂、简捷实用,是应试人员考前必备的教材。

图书在版编目(CIP)数据

全国计算机等级考试二级教程:2017年版. Visual Basic 语言程序设计 / 教育部考试中心编. --北京:高等教育出版社, 2016. 10

ISBN 978-7-04-046559-4

I. ①全… II. ①教… III. ①电子计算机-水平考试-教材②BASIC 语言-程序设计-水平考试-教材 IV. ①TP3

中国版本图书馆 CIP 数据核字(2016)第 235614 号

策划编辑	何新权	责任编辑	何新权	封面设计	王 琰	版式设计	杜微言
责任校对	刁丽丽	责任印制	朱学忠				

出版发行 高等教育出版社
社 址 北京市西城区德外大街 4 号
邮政编码 100120
印 刷 高教社(天津)印务有限公司
开 本 787 mm×1092 mm 1/16
印 张 27.5
字 数 680 千字
购书热线 010-58581118
咨询电话 400-810-0598

网 址	http://www.hep.edu.cn http://www.hep.com.cn
网上订购	http://www.hepmall.com.cn http://www.hepmall.com http://www.hepmall.cn
版 次	2016 年 10 月第 1 版
印 次	2016 年 10 月第 1 次印刷
定 价	50.00 元

本书如有缺页、倒页、脱页等质量问题,请到所购图书销售部门联系调换

版权所有 侵权必究

物 料 号 46559-00

积极发展全国计算机等级考试 为培养计算机应用专门人才、促进信息 产业发展作出贡献 (序)

中国科协副主席 中国系统仿真学会理事长
第五届全国计算机等级考试委员会主任委员
赵沁平

当今，人类正在步入一个以智力资源的占有和配置，知识生产、分配和使用为最重要因素的知识经济时代，也就是小平同志提出的“科学技术是第一生产力”的时代。世界各国的竞争已成为以经济为基础、以科技（特别是高科技）为先导的综合国力的竞争。在高科技中，信息科学技术是知识高度密集、学科高度综合、具有科学与技术融合特征的学科。它直接渗透到经济、文化和社会的各个领域，迅速改变着人们的工作、生活和社会的结构，是当代发展知识经济的支柱之一。

在信息科学技术中，计算机硬件及通信设施是载体，计算机软件是核心。软件是人类知识的固化，是知识经济的基本表征，软件已成为信息时代的新型“物理设施”。人类抽象的经验、知识正逐步由软件予以精确地体现。在信息时代，软件是信息化的核心，国民经济和国防建设、社会发展、人民生活都离不开软件，软件无处不在。软件产业是增长快速的朝阳产业，是具有高附加值、高投入高产出、无污染、低能耗的绿色产业。软件产业的发展将推动知识经济的进程，促进从注重量的增长向注重质的提高方向发展。软件产业是关系到国家经济安全和文化安全，体现国家综合实力，决定 21 世纪国际竞争地位的战略性新兴产业。

为了适应知识经济发展的需要，大力促进信息产业的发展，需要在全民中普及计算机的基本知识，培养一批又一批能熟练运用计算机和软件技术的各行各业的应用型人才。

1994 年，国家教委（现教育部）推出了全国计算机等级考试，这是一种专门评价应试人员对计算机软硬件实际掌握能力的考试。它不限制报考人员的学历和年龄，从而为培养各行业计算机应用人才开辟了一条广阔的道路。

1994 年是推出全国计算机等级考试的第一年，当年参加考试的有 1 万余人，2012 年报考人数已达 549 万人。截至 2013 年年底，全国计算机等级考试共开考 38 次，考生人数累计达 5 422 万人，有 2 067 万人获得了各级计算机等级证书。

事实说明，鼓励社会各阶层人士通过各种途径掌握计算机应用技术，并通过等级考试对他们的能力予以科学、公正、权威性的认证，是一种比较好的、有效的计算机应用人才培养途径，符合我国的具体国情。等级考试同时也为用人单位录用和考核人员提供了一种测评手段。从有关公司对等级考试所作的社会抽样调查结果看，不论是管理人员还是应试人员，对该项考



试的内容和形式都给予了充分肯定。

计算机技术日新月异。全国计算机等级考试大纲顺应技术发展和需求的变化，从2010年开始对新版考试大纲进行调研和修订，在考试体系、考试内容、考试形式等方面都做了较大调整，希望等级考试更能反映当前计算机技术的应用实际，使培养计算机应用人才的工作更健康地向前发展。

全国计算机等级考试取得了良好的效果，这有赖于各有关单位专家在等级考试的大纲编写、试题设计、阅卷评分及效果分析等多项工作中付出的大量心血和辛勤劳动，他们为这项工作的开展作出了重要的贡献。我们在此向他们表示衷心的感谢！

我们相信，在21世纪知识经济和加快发展信息产业的形势下，在教育部考试中心的精心组织领导下，在全国各有关专家的大力配合下，全国计算机等级考试一定会以“激励引导成才，科学评价用才，服务社会选材”为目标，服务考生和社会，为我国培养计算机应用专门人才的事业作出更大的贡献。

前 言

Visual Basic 是 Microsoft 公司成功的编程语言产品之一，在全世界拥有数以百万计的用户。它之所以受到人们的青睐，原因是多方面的，但主要的有两点，一是功能强大，二是容易掌握。Visual Basic 的出现，打破了 Windows 应用程序的开发由专业的 C 程序员一统天下的局面，即使是非专业人员也能胜任，并可在较短的时间内开发出质量高、界面好的应用程序。

随着版本的更新，Visual Basic 已成为真正专业化的大型开发语言和环境，不仅功能越来越强，而且更容易使用。Visual Basic 6.0 所提供的开发环境与 Windows 具有完全一致的界面，使用更方便，其代码效率已达到 Visual C++ 的水平。

为了推动计算机应用人才的成长，国内先后推出一系列有关计算机的考试，且规模在不断扩大。“全国计算机等级考试”是由教育部考试中心组织、深受社会各界欢迎的计算机考试，自 1994 年举办以来，应试人数逐年增加，对计算机技术的普及应用起到了十分重要的作用。本书是根据教育部考试中心最新制定的《全国计算机等级考试二级 Visual Basic 语言程序设计考试大纲（2013 年版）》的要求，在上一版的基础上修订而成。

全书共分十四章，主要内容包括：Visual Basic 程序开发环境、对象和事件驱动的概念、常用内部函数和表达式、数据输入输出、常用标准控件、基本控制结构、数组、过程调用、键盘和鼠标事件过程、菜单程序设计、对话框程序设计、多窗体程序设计、文件处理等。

本书的内容紧扣考试大纲，介绍的是 Visual Basic 6.0 的基础知识，是 Visual Basic 程序设计中最基本的部分，适用于初学者。针对初学者的特点，在体系结构和内容上注意了由简及繁、由浅入深、循序渐进、深入浅出以及理论与实践密切结合。考虑到 Visual Basic 是为编写应用程序而研制的，本书中的例题主要用来加深对概念的理解。只有理解了这些基本概念，才能用 Visual Basic 设计复杂的应用程序。

Visual Basic 6.0 包括 3 种版本，即学习版、专业版和企业版，这些版本是在相同的基础上建立起来的，因此大多数应用程序可以在 3 种版本中通用。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，书中所有程序可以在专业版和学习版中运行。

本书由教育部考试中心组织编写和审定，刘炳文编写了第 1~14 章，杨明福、陈定中编写了第 15 章。谭浩强教授和陈志泊副教授对本书进行了全面的审阅，提出了许多宝贵意见；在本书的编写和出版过程中，教育部考试中心和高等教育出版社给予了大力支持，在此一并表示衷心感谢。

由于笔者水平有限、经验不多，缺点和错误在所难免，诚望专家和广大读者不吝赐教，批评指正。

编 者

目 录

第 1 章 Visual Basic 程序开发环境	1	3.2.3 代码编辑器	43
1.1 可视化与事件驱动型语言	1	3.3 程序的保存、装入和运行	45
1.1.1 可视化界面设计	1	3.3.1 保存程序	45
1.1.2 事件驱动的编程机制	2	3.3.2 程序的装入	47
1.2 Visual Basic 的启动与退出	3	3.3.3 程序的运行	48
1.3 主窗口	5	3.4 Visual Basic 应用程序的结构与 工作方式	50
1.3.1 标题栏和菜单栏	5	3.4.1 Visual Basic 应用程序的构成	50
1.3.2 工具栏	7	3.4.2 事件驱动	51
1.4 其他窗口	9	习题	52
1.4.1 窗体设计器和工程资源管理器	9	第 4 章 Visual Basic 程序设计基础	54
1.4.2 属性窗口和工具箱窗口	12	4.1 数据类型	54
习题	13	4.1.1 基本数据类型	54
第 2 章 对象及其操作	14	4.1.2 用户定义的数据类型	57
2.1 对象	14	4.2 常量和变量	58
2.1.1 Visual Basic 的对象	14	4.2.1 常量	58
2.1.2 对象属性设置	16	4.2.2 变量	61
2.2 窗体	18	4.3 变量的作用域	64
2.2.1 窗体的结构与属性	18	4.3.1 局部变量与全局变量	64
2.2.2 窗体事件	22	4.3.2 默认声明	66
2.3 控件	23	4.4 常用内部函数	67
2.3.1 标准控件(内部控件)	23	4.4.1 转换、数学及日期和时间函数	68
2.3.2 控件的命名和控件值	25	4.4.2 字符串函数	71
2.4 控件的画法和基本操作	28	4.5 运算符与表达式	74
2.4.1 控件的画法	28	4.5.1 算术运算符	75
2.4.2 控件的基本操作	29	4.5.2 关系运算符与逻辑运算符	76
习题	31	4.5.3 字符串表达式与日期表达式	79
第 3 章 简单程序设计	33	4.5.4 表达式的执行顺序	80
3.1 Visual Basic 中的语句	33	习题	81
3.2 编写简单的 Visual Basic 应用 程序	36	第 5 章 数据的输入输出	84
3.2.1 用 Visual Basic 开发应用程序的 一般步骤	36	5.1 数据输出——Print 方法	84
3.2.2 编写 Visual Basic 应用程序	37	5.1.1 Print 方法	84
		5.1.2 与 Print 方法有关的函数	86



5.1.3 格式输出	87	7.1 选择控制结构	154
5.1.4 其他方法和属性	90	7.1.1 单行结构条件语句	154
5.2 数据输入——InputBox 函数	94	7.1.2 块结构条件语句	155
5.3 MsgBox 函数和 MsgBox 语句	97	7.1.3 If 函数	159
5.3.1 MsgBox 函数	98	7.2 多分支控制结构	160
5.3.2 MsgBox 语句	101	7.3 For 循环控制结构	164
5.4 字形	102	7.4 当循环控制结构	169
5.4.1 字体类型和大小	102	7.5 Do 循环控制结构	172
5.4.2 其他属性	104	7.6 多重循环	176
习题	105	习题	179
第 6 章 常用标准控件	109	第 8 章 数组	187
6.1 文本控件	109	8.1 数组的概念	187
6.1.1 标签	109	8.1.1 数组的定义	187
6.1.2 文本框	110	8.1.2 默认数组	191
6.2 图形控件	114	8.2 静态数组与动态数组	192
6.2.1 图片框和图像框	114	8.2.1 动态数组的定义	192
6.2.2 图形文件的装入	116	8.2.2 数组的清除和重定义	194
6.2.3 直线和形状	119	8.3 数组的基本操作	195
6.3 按钮控件	122	8.3.1 数组元素的输入、输出和复制	196
6.3.1 属性和事件	122	8.3.2 For Each...Next 语句	200
6.3.2 应用举例	123	8.4 数组的初始化	201
6.4 选择控件——复选框和单选 按钮	125	8.5 控件数组	203
6.4.1 复选框和单选按钮的属性和 事件	126	8.5.1 基本概念	203
6.4.2 应用举例	127	8.5.2 建立控件数组	204
6.5 选择控件——列表框和 组合框	129	习题	207
6.5.1 列表框	129	第 9 章 过程	215
6.5.2 组合框	133	9.1 Sub 过程	215
6.6 滚动条	138	9.1.1 建立 Sub 过程	215
6.7 计时器	140	9.1.2 调用 Sub 过程	218
6.8 框架	142	9.1.3 通用过程与事件过程	220
6.9 焦点与 Tab 顺序	145	9.2 Function 过程	222
6.9.1 设置焦点	145	9.2.1 建立 Function 过程	222
6.9.2 Tab 顺序	146	9.2.2 调用 Function 过程	224
习题	148	9.3 参数传送	227
第 7 章 Visual Basic 控制结构	154	9.3.1 形参与实参	227
		9.3.2 引用	229
		9.3.3 传值	231
		9.3.4 数组参数的传送	233

9.4 可选参数与可变参数	236	12.1.2 自定义对话框	318
9.4.1 可选参数	236	12.1.3 通用对话框控件	321
9.4.2 可变参数	237	12.2 文件对话框	322
9.5 对象参数	238	12.2.1 文件对话框的结构	322
9.5.1 窗体参数	239	12.2.2 文件对话框的属性	323
9.5.2 控件参数	240	12.2.3 文件对话框举例	326
9.6 局部内存分配	245	12.3 其他对话框	329
习题	247	12.3.1 颜色 (Color) 对话框	329
第 10 章 键盘与鼠标事件	254	12.3.2 字体 (Font) 对话框	330
10.1 KeyPress 事件	254	12.3.3 打印 (Printer) 对话框	333
10.2 KeyDown 和 KeyUp 事件	257	习题	336
10.3 鼠标事件	266	第 13 章 多重窗体程序设计与环境	
10.3.1 鼠标位置	266	应用	340
10.3.2 鼠标键	268	13.1 建立多重窗体应用程序	340
10.3.3 转换参数 (Shift)	271	13.1.1 与多重窗体程序设计有关的语句	
10.4 鼠标光标的形状	274	和方法	340
10.4.1 MousePointer 属性	274	13.1.2 建立界面	341
10.4.2 设置鼠标光标形状	275	13.1.3 编写程序代码	345
10.5 拖放	277	13.2 多重窗体程序的执行与保存	351
10.5.1 与拖放有关的属性、事件和		13.2.1 指定启动窗体	352
方法	277	13.2.2 多窗体程序的存取	353
10.5.2 自动拖放	278	13.3 Visual Basic 工程结构	354
10.5.3 手动拖放	281	13.3.1 标准模块	354
习题	284	13.3.2 窗体模块	355
第 11 章 菜单程序设计	290	13.3.3 Sub Main 过程	356
11.1 Visual Basic 中的菜单	290	13.4 多模块编程	358
11.2 菜单编辑器	291	13.5 系统对象	361
11.3 用菜单编辑器建立菜单	294	13.5.1 App 对象	361
11.4 菜单项的控制	299	13.5.2 Screen 对象	361
11.4.1 有效性控制	299	习题	363
11.4.2 菜单项标记	300	第 14 章 数据文件	366
11.4.3 键盘选择	304	14.1 文件概述	366
11.5 菜单项的增减	305	14.2 文件的打开与关闭	368
11.6 弹出式菜单	308	14.2.1 文件的打开 (建立)	368
习题	312	14.2.2 文件的关闭	370
第 12 章 对话框程序设计	317	14.3 文件操作语句和函数	371
12.1 概述	317	14.3.1 文件指针	371
12.1.1 对话框的分类与特点	317	14.3.2 其他函数	372



14.4 顺序文件	373	14.7 文件基本操作	396
14.4.1 顺序文件的写操作	373	习题	398
14.4.2 顺序文件的读操作	377	附录 1 考试指导	403
14.5 随机文件	382	附录 2 全国计算机等级考试二级	
14.5.1 随机文件的打开与读写操作	383	Visual Basic 语言程序设计	
14.5.2 随机文件中记录的增加与		考试大纲 (2013 年版)	413
删除	389	附录 3 全国计算机等级考试二级	
14.6 文件系统控件	391	Visual Basic 语言程序设计	
14.6.1 驱动器列表框和目录列表框	392	样题及参考答案	418
14.6.2 文件列表框	394	附录 4 习题参考答案	426

Visual Basic 是新一代的可视化程序设计语言，其应用程序设计是在一个集成开发环境（IDE）中进行的。在本章中，将介绍 Visual Basic 6.0 版的集成开发环境。

1.1 可视化与事件驱动型语言

Visual Basic 是一种新型的现代程序设计语言，具有很多与传统程序设计语言不同的特点，其中最主要的特点有两个，即可视化界面设计和事件驱动的编程机制。

1.1.1 可视化界面设计

常用的高级程序设计语言大体上可以分为两类，即面向过程的语言和面向对象的语言。面向过程的程序设计语言把解题的过程看作是数据加工的过程，注重的是算法描述。因此，面向过程的高级语言又称为算法语言。

计算机技术的进一步发展，特别是具有图形用户界面（GUI）的操作系统（如 Windows）的广泛使用，使得面向对象的程序设计思想应运而生。采用了面向对象思想的程序设计语言就是面向对象的程序设计语言。

Visual Basic 是面向对象的程序设计语言，但它与一般的面向对象的程序设计语言不完全相同。在一般的面向对象程序设计语言中，对象由程序代码和数据组成，是抽象的概念；而 Visual Basic 则是应用面向对象的程序设计方法，把程序和数据封装起来作为一个对象，并为每个对象赋予应有的属性，使对象成为实在的东西。在设计对象时，不必编写建立和描述每个对象的程序代码，而是用工具画在界面上，Visual Basic 自动生成对象的程序代码并封装起来。每个对象以图形方式显示在界面上，都是可视的。

用传统的程序设计语言来设计程序时，主要的工作就是设计算法和编写代码，程序的功能和显示结果（包括大量的用户界面）都通过程序语句来实现。在设计过程中看不到界面的实际显示效果，必须编译后运行程序才能观察。如果对界面的效果不满意，还要回到程序中去修改。有时候，这种编程—编译—修改的操作可能要反复多次，大大影响了软件开发效率。



Visual Basic 提供了可视化设计工具,把 Windows 界面设计的复杂性“封装”起来,开发人员不必为界面设计而编写大量程序代码,只需要按设计要求的屏幕布局,用系统提供的工具(控件),在屏幕上画出所需的各种“部件”,即图形对象,并设置这些图形对象的属性,Visual Basic 自动产生界面设计代码,程序设计人员只需要编写实现程序功能的那部分代码。也就是说,程序所需要的用户界面是用 Visual Basic 所提供的可视化设计工具“画”出来的,而不是用程序代码“写”出来的。例如,我们可以根据需要很容易地画出如图 1.1 所示的简易计算器面板,如果用传统的程序设计语言建立这样一个界面,则必须编写大量的程序代码,而且可能要经过反复修改才能达到设计要求。



图 1.1 简易计算器面板

1.1.2 事件驱动的编程机制

Visual Basic 通过事件驱动的方式来实现对象的操作,其程序不是按照预定的“路径”执行,而是在响应不同的事件时驱动不同的事件代码,以此来控制对象的行为。一个对象可能会产生多个事件,每个事件都可以通过一段程序来响应。例如,命令按钮是一个对象,当用户单击该按钮时,将产生一个“单击”(Click)事件,而在产生该事件时将执行一段程序,用来实现指定的操作。

在用 Visual Basic 设计大型应用软件时,不必建立具有明显开始和结束的程序,而是编写若干个微小的子程序,即过程,这些过程分别面向不同的对象,由用户操作引发某个事件来驱动完成某种特定的功能,或者由事件驱动程序调用通用过程来执行指定的操作。这样可以方便编程人员,提高效率。

在传统的面向过程的应用程序中,执行哪一部分代码和按何种顺序执行代码都由程序本身控制。而在面向对象的程序设计中,编程人员要以“对象”为中心来设计模块,而不是以“过程”为中心来考虑应用程序的结构。此外,在事件驱动应用程序中,代码不是按预定的顺序执行,而是在响应不同的事件时执行不同的代码段。

事件是可以由窗体或控件识别的操作。在响应事件时,事件驱动应用程序执行指定的代码。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,甚至由应用程序本身的消息触发。这些事件的顺序决定了代码执行的顺序,因此,在事件驱动应用程序中,每次运行时所执行的代码和所经过的“路径”是不一样的。

Visual Basic 的窗体和每个控件都有一个预定义的事件集,当其中的某个事件发生,并且在相关联的事件过程中存在代码时,Visual Basic 将执行这些代码。

尽管 Visual Basic 中的对象能自动识别预定义的事件集,但必须通过代码判定它们是否响应具体事件以及如何响应具体事件,代码(即事件过程)与每个事件对应。为了让窗体或控件响应某个事件,必须把代码放入这个事件的事件过程之中。

对象所能识别的事件类型有很多种,但多数类型为大多数对象所共有。例如,大多数对象都能识别 Click 事件,即单击事件;如果单击窗体,则执行窗体的单击事件过程中的代码;如



果单击命令按钮,则执行命令按钮的单击事件过程中的代码。此外,某些事件可以在运行期间触发。例如,当在运行期间改变文本框中的文本时,将引发文本框的 Change 事件,如果 Change 事件过程中含有代码,则执行这些代码。

1.2 Visual Basic 的启动与退出

Visual Basic 6.0 是 Visual Studio 6.0 套装软件中的一个成员,它可以和 Visual Studio 6.0 一起安装,也可以单独安装。单独安装的 Visual Basic 6.0 中文版包括 4 张光盘,其中两张为 MSDN。若安装方式不同,则启动方式也略有区别。在这里,假定所使用的 Visual Basic 6.0 是单独安装的。

对于 Windows XP 及以前的操作系统,可以用以下方法启动 Visual Basic 6.0:

第一种方法:使用“开始”菜单中的“程序”命令。操作如下:

- (1) 单击 Windows 环境下的“开始”按钮,弹出一个菜单,把光标移到“程序”命令上,将弹出下一个级联菜单。
- (2) 把光标移到“Microsoft Visual Basic 6.0 中文版”,弹出下一个级联菜单,即 Visual Basic 6.0 程序组。
- (3) 单击“Microsoft Visual Basic 6.0 中文版”,即可进入 Visual Basic 6.0 编程环境。

第二种方法:使用“我的电脑”。操作如下:

- (1) 双击“我的电脑”,弹出一个窗口,然后单击 Visual Basic 6.0 所在的硬盘驱动器盘符,将打开相应的驱动器窗口。
- (2) 单击驱动器窗口中的 vb60 文件夹,打开“vb60”窗口。
- (3) 双击“vb6.exe”图标,即可进入 Visual Basic 6.0 编程环境。

第三种方法:使用“开始”菜单中的“运行”命令。操作如下:

- (1) 单击“开始”按钮,弹出一个菜单,然后单击“运行”命令,将弹出一个对话框。
- (2) 在“打开”栏内输入 Visual Basic 6.0 启动文件的名字(包括路径)。例如“c:\vb98\vb6.exe”。
- (3) 单击“确定”按钮,即可启动 Visual Basic 6.0。

第四种方法:建立启动 Visual Basic 6.0 的快捷方式(具体操作见有关资料)。

在 Windows 7 环境下,可以用以下两种方法启动 Visual Basic:

第一种方法,使用“开始”菜单。操作如下:

- (1) 单击 Windows 环境下的“开始”按钮,弹出一个菜单,单击“所有程序”,将弹出系统安装的程序清单。
 - (2) 单击“Microsoft Visual Basic 6.0 中文版”,显示下一级菜单。
 - (3) 单击该菜单下的“Microsoft Visual Basic 6.0 中文版”,即可进入 Visual Basic 编程环境。
- 第二种方法,建立启动 Visual Basic 6.0 的快捷方式。操作如下:
- (1) 同第一种方法中的(1)、(2)。
 - (2) 用鼠标右键单击“Microsoft Visual Basic 6.0 中文版”,弹出一个菜单,把鼠标光标移



到“发送到”，显示下一级菜单。

(3) 单击“桌面快捷方式”，即可在桌面上建立启动 Visual Basic 6.0 的快捷方式。

建立快捷方式后，只要双击快捷方式图标，即可启动 Visual Basic 6.0。

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，说明此份程序拷贝的使用权属于谁。稍候，显示“新建工程”对话框，如图 1.2 所示。图中所显示的是“新建”选项卡，如果单击“现存”或“最新”选项卡，则可分别显示现有的或最新的 Visual Basic 应用程序文件名列表，可从中选择要打开的文件名。

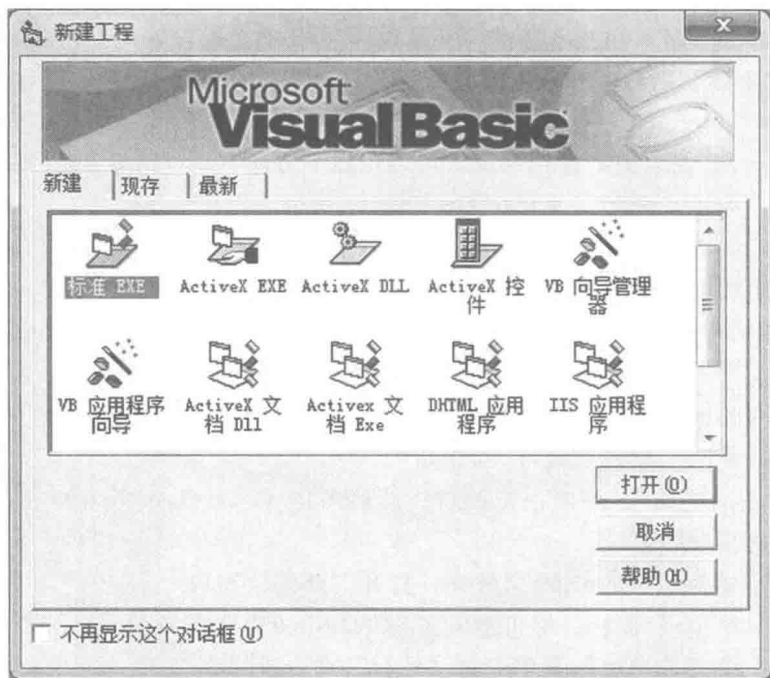


图 1.2 “新建工程”对话框（“新建”选项卡）

“新建”选项卡对话框显示了可以在 Visual Basic 6.0 中使用的工程类型，即可以建立的应用程序，其中“标准 EXE”用来建立一个标准的 EXE 工程，本书将只讨论这种工程类型。

在对话框中选择要建立的工程类型（例如“标准 EXE”），然后单击“打开”按钮，即进入 Visual Basic 6.0 集成环境，如图 1.3 所示。

Visual Basic 6.0 提供了许多种工程类型以满足不同的需要。每次启动 Visual Basic 时，都要显示“新建工程”对话框。在一般情况下，可能主要使用“标准 EXE”工程，因此，没有必要在每次启动 Visual Basic 时显示该对话框，这可以通过选择（见图 1.2 左下角）“不再显示这个对话框”选项来实现（用鼠标单击小方框，使框内有“√”）。选择该选项后，再选择“标准 EXE”，然后单击“打开”按钮，进入 Visual Basic 开发环境。以后再启动 Visual Basic 时，即不再显示“新建工程”对话框，直接进入开发环境。

在默认情况下，Visual Basic 6.0 的集成开发环境为传统的 Windows MDI（多文档界面）方式（图 1.3）；此外，也可以用 SDI（单文档界面）方式启动 Visual Basic 设计。在多数情况

下,使用 SDI 方式可能会更方便。为了把编程环境变为 SDI 方式,可执行“工具”菜单中的“选项”命令,打开“选项”对话框,选择“高级”选项卡,在对话框中选择“SDI 开发环境”选项,然后单击“确定”按钮。这样设置后,退出 Visual Basic,然后重新启动,即可按 SDI 方式进入 Visual Basic 集成开发环境。



图 1.3 Visual Basic 6.0 编程环境 (MDI 方式)

从图 1.3 中可以看出,启动 Visual Basic 后,屏幕上分为若干部分,包括标题栏、菜单栏、工具栏、工具箱、窗体设计器窗口、工程资源管理器、窗体布局窗口和属性窗口。为了能清楚地看到每个部分,这里对原来的各部分进行了缩放和重新排列。读者在启动自己的 Visual Basic 后,所看到的各部分的排列情况可能与图 1.3 所示的有微小差别,一些窗口会重叠。实际上,和 Windows 下的窗口一样,集成开发环境中的每个窗口都可以在屏幕上移动、缩小、放大或关闭。此外,Visual Basic 保存上一次使用时屏幕上各部分最后的排列方式,并作为下一次启动 Visual Basic 后的屏幕布局。

为了退出 Visual Basic,可先打开“文件”菜单,并执行其中的“退出”命令(或按 Alt+Q)。如果当前程序已修改过并且没有存盘,系统将显示一个对话框,询问用户是否将其存盘,此时选择“是”按钮则存盘,选择“否”按钮则不存盘。在上述两种情况下,都将退出 Visual Basic,回到 Windows 环境。

1.3 主窗口

主窗口也称设计窗口。启动 Visual Basic 后,主窗口位于集成环境的顶部,该窗口由标题栏、菜单栏和工具栏组成(参见图 1.3)。

1.3.1 标题栏和菜单栏

1. 标题栏

标题栏是屏幕顶部的水平条,它显示的是应用程序的名字。用户与标题栏之间的交互关系



由 Windows 来处理，而不是由应用程序处理。启动 Visual Basic 后，标题栏中显示的信息为：

工程 1-Microsoft Visual Basic[设计]

方括号中的“设计”表明当前的工作状态是“设计阶段”。随着工作状态的不同，方括号中的信息也随之改变，可能会是“运行”或“Break”，分别代表“运行阶段”或“中断阶段”。这 3 个阶段也分别称为“设计模式”“运行模式”和“中断模式”。

2. 菜单栏

在标题栏的下面是集成环境的主菜单。菜单栏中的菜单命令提供了开发、调试和保存应用程序所需要的工具。Visual Basic 6.0 中文版的菜单栏共有 13 个菜单项，即文件、编辑、视图、工程、格式、调试、运行、查询、图表、工具、外接程序、窗口和帮助。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。例如，单击“文件”，就可以打开文件菜单，如图 1.4 所示。打开菜单后，如果单击“打开工程”，就可以打开已有的工程文件；而如果单击“工程另存为”，就可以保存文件，等等。在以后的叙述中，我们把形如上面的操作记为“执行文件菜单中的‘打开工程’命令”“执行文件菜单中的‘工程另存为’命令”。

菜单中的命令分为 3 种类型，一类是可以直接执行的命令，这类命令的后面没有任何信息（例如“保存工程”）；另一类在命令名后面带有省略号（例如“打开工程”），需要通过打开“对话框”来执行；第三类带有子菜单命令，这类命令的右端有一个箭头（见“格式”菜单）。在用鼠标单击后一类命令后，屏幕上将显示一个对话框，利用该对话框可以执行各种有关的操作。在“文件”菜单中，“新建工程”“保存工程”等是可以直接执行的命令，而“打开工程”“工程另存为”等命令则必须通过对话框来执行。此外，从“文件”菜单可以看出，在有些命令的后面还带有其他信息，例如：

打开工程... Ctrl+O
保存 Form1 Ctrl+S

其中“Ctrl+O”等叫做“热键”（或快捷键）。在菜单中，热键列在相应的菜单命令之后，与菜单命令具有相同的作用。使用热键方式，不必打开菜单就能执行相应的菜单命令。例如，按 Ctrl+O，可以立即执行“打开工程”命令。注意，只有部分菜单命令能通过热键执行。

上面介绍了通过鼠标和热键执行菜单命令的方法。除使用鼠标外，也可以通过键盘执行菜单命令。只有在打开菜单后，才能选择所需要的命令，执行相应的操作。Visual Basic 6.0 提供了多种打开菜单和选择菜单的方法，用户可以根据自己的兴趣或习惯选用其中的一种。

第一种方法，步骤如下：

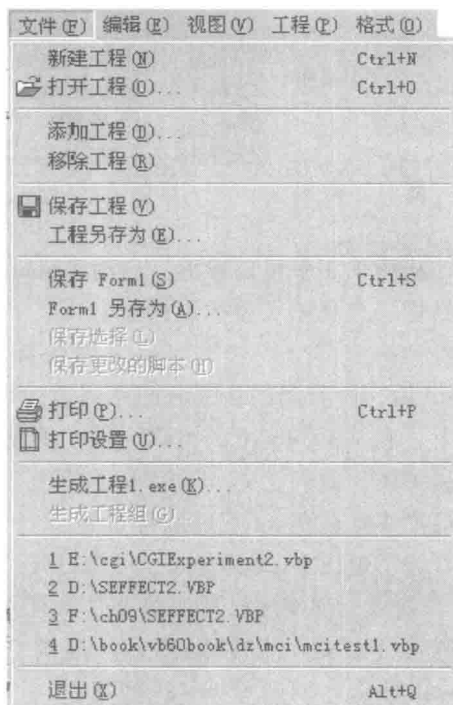


图 1.4 “文件”菜单

(1) 按 F10 或 Alt 键激活菜单栏, 此时第一个菜单项 (“文件”) 被加上一个浅色的框。

(2) 按菜单项后面括号中的字母键打开菜单, 下拉显示该菜单项的命令。菜单被打开后, 各菜单命令后面的括号内都有一个字母。

(3) 按菜单命令后面括号中的字母键, 即可执行相应的命令。

第二种方法, 步骤如下:

(1) 按 F10 或 Alt 键激活菜单栏, 此时第一个菜单项 (“文件”) 被加上一个浅色的框。

(2) 用→或←把条形光标移到需要打开的菜单项上, 按回车键打开该菜单项。

(3) 菜单被打开后, 条形光标覆盖在第一个或上一次执行的菜单命令上。用↑或↓把条形光标移到所需要的命令上, 按回车键即可执行条形光标所在位置的菜单命令。

第三种方法, 步骤如下:

(1) 按下 Alt 键, 不要松开, 接着按需要打开的菜单项后面括号中的字母键, 然后松开 (Alt 键接着松开), 该菜单即被打开。

(2) 按菜单命令后面括号中的字母键, 即可执行指定的菜单命令。

例如, 为了执行 “文件” 菜单中的 “打开工程” 命令, 可以这样操作: 按住 Alt 键, 不要松开, 接着按 F 键, 先后松开 F 键和 Alt 键, 再按 O 键, 即可执行 “文件” 菜单中的 “打开工程” 命令。我们把上述过程记作: Alt+F, O。

除上面 3 种方法外, 有些菜单命令还可以通过热键执行 (见前)。对于没有热键的菜单命令, 只能通过上面 3 种方式执行。

菜单被打开后, 在屏幕上显示相应的菜单命令。如果打开了不当或不需要的菜单, 或者执行菜单命令时打开了不需要的对话框, 可以用 Esc 键关闭。

Visual Basic 应用程序的编辑、编译、连接、运行、调试及文件的打开、保存等都可以通过相应的菜单命令来实现, 其用法与上面介绍的类似。

1.3.2 工具栏

Visual Basic 6.0 提供了 4 种工具栏, 包括编辑、标准、窗体编辑器和调试, 并可根据需要定义用户自己的工具栏。在一般情况下, 集成环境中只显示标准工具栏, 其他工具栏可以通过 “视图” 菜单中的 “工具栏” 命令打开 (或关闭)。每种工具栏都有固定和浮动两种形式。把光标移到固定形式工具栏中没有图标的地方, 按住鼠标左键向下拖动, 或者双击工具栏左端的两条浅色竖线, 即可把工具栏变为浮动的; 而如果双击浮动工具栏的标题条, 则可变为固定工具栏。

固定形式的标准工具栏位于菜单栏的下面, 即主窗口的底部, 它以图标的形式提供了部分常用菜单命令的功能。只要用鼠标单击代表某个命令的图标按钮, 就能直接执行相应的菜单命令。标准工具栏中有 21 个图标, 代表 21 种操作, 如图 1.5 (a) 所示。大多数图标都有与之等价的菜单命令。图 1.5 (b) 是浮动形式的标准工具栏。

表 1.1 列出了标准工具栏中各图标的作用 (表中的编号与图 1.5 (a) 对应)。