

Visual Basic 6.0

程序设计教程

陈学东 常 丹 等 编著



清华大学出版社

Visual Basic 程序设计教程 6.0

陈学东 常 丹 等 编著

清华大学出版社
北京

内 容 简 介

Visual Basic 提供了开发 Windows 应用程序的最迅速、最简捷的方法,在实际程序设计中广泛应用。本书以 Visual Basic 6.0 版本为蓝本,分四部分循序渐进地介绍:一是界面设计,使初学者熟悉、掌握可视化编程和事件驱动编程的思想,并能够创建符合实际需求的 Windows 应用程序界面;二是程序代码设计,对程序设计的基本语法、编程方法、编程规范和常用算法进行较为系统、详细地介绍,引导读者掌握实际应用功能的编程实现;三是数据库访问技术,详细介绍了最新的数据访问技术——ADO 技术的使用;四是面向对象程序设计的初步知识和编程思想。

版权所有,翻印必究。举报电话:010-62782989 13501256678 13801310933

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

本书防伪标签采用清华大学核研院专有核径迹膜防伪技术,用户可通过在图案表面涂抹清水,图案消失,水干后图案复现;或将表面膜揭下,放在白纸上用彩笔涂抹,图案在白纸上再现的方法识别真伪。

图书在版编目(CIP)数据

Visual Basic 6.0 程序设计教程/陈学东,常丹等编著. —北京:清华大学出版社,2005.1
ISBN 7-302-10023-3

I. Visual… II. ①陈… ②常… III. BASIC 语言—程序设计—教材 IV. TP312

中国版本图书馆 CIP 数据核字(2004)第 124668 号

出 版 者:清华大学出版社

<http://www.tup.com.cn>

社 总 机:010-62770175

地 址:北京清华大学学研大厦

邮 编:100084

客户服务:010-62776969

责任编辑:苗建强

版式设计:刘祎森

印 刷 者:北京四季青印刷厂

装 订 者:三河市李旗庄少明装订厂

发 行 者:新华书店总店北京发行所

开 本:185×260 印张:20.25 字数:465 千字

版 次:2005 年 1 月第 1 版 2005 年 1 月第 1 次印刷

书 号:ISBN 7-302-10023-3/TP·6881

印 数:1~5000

定 价:32.00 元

本书如存在文字不清、漏印以及缺页、倒页、脱页等印装质量问题,请与清华大学出版社出版部联系调换。联系电话:(010)62770175-3103 或(010)62795704



Visual Basic(简称 VB)是时下流行的一种功能非常强大的第四代程序设计语言(4GL),它提供了开发 Windows 应用程序的最迅速、最简捷的方法。因其灵活的数据访问、ActiveX 技术、Internet 编程能力等特性,在实际中得到广泛应用。另外,它提供的可视化开发方法,用户在创建应用程序时不需编写大量代码去描述界面元素的外观和位置,只要把预先建立的对象拖拉到屏幕上的合适位置即可,与其他高级语言相比,它的语法规则相对简单,容易理解和掌握,广大学生、应用程序开发人员把它作为学习、掌握和使用 Windows 应用程序开发工具的入门语言。

Visual Basic 6.0 程序设计教程以 VB 6.0 版本作为蓝本,结合多年教学 and 实际系统开发经验,将其分为四部分循序渐进地介绍:

一是界面设计。对初学者不太熟悉的可视化集成开发环境及界面设计的基本对象、属性、方法和事件作了重点介绍,意图使初学者通过大量应用实例的学习和练习,熟悉、掌握 Visual Basic 可视化编程和事件驱动编程的思想,并能够创建符合实际需求的 Windows 应用程序界面,消除初学者对程序设计的畏惧心理,进一步激发其学习兴趣。

二是程序代码设计。对程序设计的基本语法、编程方法、编程规范和常用算法进行较为系统、详细地介绍,引导读者在前面界面设计的基础上分析问题,掌握实际应用中各种功能需求的编程实现。知识点讲解中始终贯穿着编程规范和编程方法,在规范、清晰的程序结构中培养初学者独立发现问题的能力。界面设计与程序代码设计分开介绍,既符合 Windows 应用程序“界面对象创建—对象属性设置—事件驱动代码编写”的编程步骤,又遵循了由易而难、由浅入深的学习规律。同时,初学者在创建了自己的程序界面后,将渴望掌握各种功能的实现方法,使得其对原本枯燥的语法、代码产生浓郁的兴趣。

三是数据库访问技术。针对应用程序、特别是经济、管理类应用程序中数据的访问和处理是核心,本书对 Visual Basic 6.0

管理科学与
信息系统专
业系列教材

中最新的数据访问技术—ADO 技术作了详细的介绍,既提供了直接使用数据窗体向导快速生成数据界面的方法,以满足对编程能力要求不高的读者快速掌握数据访问的需求,也提供了包括输入数据检验在内的、利用数据环境(Data Environment)和 RecorSet 对象对数据访问的深入处理,以满足编程人员对复杂数据访问的更为灵活的处理手段。

四是面向对象编程技术。对 Visual Basic 中面向对象程序设计的相关概念和程序实现作了初步的介绍,以使读者在基本程序设计思想和技巧掌握的基础上,了解面向对象程序设计的基本思想和方法。

分散的、偏重单纯的语法格式或是某些算法模式的 Visual Basic 课程实践,其内容缺乏自主性、系统化,学习结束后初学者通常仍然不知如何着手开发实际应用系统。为促进初学者综合系统开发能力的培养和应用系统开发思路、技巧的掌握,配套出版了《Visual Basic 6.0 实践教程》,采用了“任务驱动式”教学实践模式,围绕实际管理系统分阶段实施,采用每一阶段知识学习后针对知识体系提出任务、读者分析设计方案、上机实践、辅导讲解、问题归纳及提出新任务的模式,并辅助以实例演示系统、多渠道的疑难解答及综合测评方式推进学习实践效果。改变传统的实验辅导为实验引导,改变面向结果、面向独立功能的实验模式为面向过程、面向系统的实验模式,突出自主性、系统化的实践内容。初学者可以结合使用。

本书适合于大专院校学生、初学者的教学用书,也适合于程序设计人员的参考用书。

本书由北京交通大学经济管理学院信息管理系陈学东(第 1、5 章,3.1 节)、常丹(第 2、3、4、6 章)、苟娟琼(第 7、8 章)编著,刘世峰博士、屈波研究员在本书的规划和撰写过程中提出了很多宝贵的建议,并给予了极大的帮助和支持。苑玉节、李志勇、秦建辉等参加了资料的收集和整理工作。此外,本书在写作过程中参考、引用了众多文献,在此对他们致以衷心的感谢。

由于水平有限,Visual Basic 也在不断更新、发展,错误和疏漏之处在所难免,敬请专家和广大读者批评指正。

作 者

2004 年 8 月



第1章 概述 1

1.1 Visual Basic 语言概述 1

1.1.1 Visual Basic 的发展 1

1.1.2 Visual Basic 及 MSDN 的安装 2

1.1.3 Visual Basic 的启动 5

1.2 VB 的程序设计思想 6

1.2.1 对象、属性、方法和事件 6

1.2.2 事件驱动模型 9

1.2.3 交互式开发 9

1.3 可视化集成开发环境 10

1.4 简单应用程序的创建 14

1.4.1 应用程序创建步骤 14

1.4.2 联机帮助系统 18

1.4.3 VB 应用程序构成及保存 22

1.4.4 VB 应用程序的运行 25

本章小结 26

思考与练习 26

第2章 窗体与常用控件 27

2.1 窗体及其应用 27

2.1.1 窗体 27

2.1.2 窗体的属性 29

2.1.3 窗体的方法、事件及其应用实例 37

2.2 常用控件及主要属性 43

2.2.1 常用控件概念 43

2.2.2 常用控件及其属性 44

2.3 控件的方法和事件 58

2.3.1 控件的事件 58

2.3.2 控件的方法 59

2.4 窗体及其基本控件应用实例 60

管理科学与
信息系统专
业系列教材

本章小结	63
思考与练习	63
第 3 章 程序设计语言基础	65
3.1 编程风格	65
3.1.1 容易理解的命名约定的原则	66
3.1.2 变量声明	66
3.1.3 使用适当的数据类型	67
3.1.4 作用域	67
3.1.5 必要的程序注释	69
3.1.6 格式化及缩排的原则	70
3.2 数据类型(含字符串、日期和时间)	72
3.3 变量和常量	79
3.3.1 声明变量	79
3.3.2 常量	81
3.4 数组	82
3.4.1 数组的概念及其声明	82
3.4.2 静态数组	83
3.4.3 动态数组	84
3.4.4 数组的基本操作	85
3.5 运算符和表达式	87
3.5.1 运算符	87
3.5.2 表达式	88
本章小结	89
思考与练习	89
第 4 章 程序控制结构	92
4.1 过程和函数基础	92
4.1.1 Function 过程与调用	93
4.1.2 Sub 过程与调用	95
4.1.3 系统函数	98
4.2 参数传递	109
4.3 选择结构	113
4.4 循环结构	116
4.4.1 循环语句	116
4.4.2 流程控制语句的嵌套	118
4.4.3 退出控制语句	119
4.5 常用算法实例	120

本章小结	130
思考与练习	130
第 5 章 数据库访问技术	134
5.1 数据库访问基础	134
5.1.1 Visual Basic 数据访问机制	134
5.1.2 数据库基本概念	137
5.1.3 结构化查询语言(SQL)	138
5.1.4 数据管理器	143
5.2 使用 ADO 数据控件访问数据	147
5.2.1 ADO Data 控件的常用属性	148
5.2.2 ADO Data 控件的常用方法	151
5.2.3 ADO Data 控件的常用事件	152
5.2.4 使用 ADO Data 控件创建应用程序	155
5.3 使用 DE 环境访问数据	161
5.3.1 创建 Data Environment	161
5.3.2 创建数据绑定窗体	168
5.4 记录游标(Recordset)技术	169
5.4.1 Recordset 对象的引用	169
5.4.2 Recordset 对象的常用属性	169
5.4.3 Recordset 对象的常用方法	175
5.5 数据的检验	179
5.6 用 ADO 技术访问数据示例	182
本章小结	186
思考与练习	186
第 6 章 界面的扩展设计	188
6.1 状态栏、工具栏	188
6.1.1 添加状态栏	189
6.1.2 创建工具栏	193
6.1.3 公共对话框(CommonDialog)	200
6.2 扩展界面控件	205
6.2.1 创建进度指示器(ProgressBar)	205
6.2.2 Slider、UpDown 控件	208
6.2.3 SSTab 控件	212
6.2.4 MsFlexGrid 控件	215
6.3 菜单的设计	218
6.3.1 下拉式菜单的设计	218

6.3.2	弹出菜单	222
6.4	控件数组	223
6.4.1	控件数组的概念	223
6.4.2	控件数组的使用与建立	223
6.5	菜单控件数组	228
6.5.1	创建菜单控件数组	228
6.5.2	使菜单命令有效或无效	228
6.5.3	隐藏菜单	229
6.5.4	运行时添加菜单控件	230
6.6	鼠标和键盘	230
6.6.1	响应鼠标事件	230
6.6.2	响应键盘事件	234
	本章小结	240
	思考与练习	240
第 7 章	深入程序设计	242
7.1	多文档界面(MDI)	242
7.1.1	MDI 内置功能	242
7.1.2	上下级菜单	244
7.1.3	窗体对象	245
7.1.4	由应用程序向导创建 MDI 窗体	246
7.2	文件	248
7.2.1	文件的概念	248
7.2.2	文件系统控件	248
7.2.3	文件的读写	250
7.2.4	文件系统对象及其编程	252
7.3	图形	257
7.3.1	坐标系统	257
7.3.2	绘图属性与事件	258
7.3.3	图形控件	260
7.3.4	绘图方法	262
7.4	程序调试及错误控制	263
7.4.1	Visual Basic 的编译选项	263
7.4.2	Err 对象	265
7.4.3	错误处理程序的类型	265
7.4.4	中央错误处理程序	274
7.4.5	编程原则	278
	本章小结	279

思考与练习	280
第 8 章 面向对象技术及其应用	284
8.1 面向对象技术概述	284
8.1.1 面向对象技术的起源与特征	284
8.1.2 面向对象技术相关概念	286
8.2 VB 面向对象编程	287
8.2.1 VB 中的类与对象	288
8.2.2 VB 面向对象编程特征	293
8.3 ActiveX 部件	299
8.3.1 ActiveX 概述	299
8.3.2 常用 ActiveX 控件举例	303
8.3.3 ActiveX 控件的创建	305
本章小结	312
思考与练习	313

第1章

概 述

Visual Basic(简称 VB)使用可视化的集成开发环境实现应用程序界面和代码的快速创建,并使用事件驱动的编程思想,提高了应用程序的灵活性、方便性。Visual Basic 6.0的联机帮助文档使用 Microsoft Visual Studio 6.0统一的 MSDN(Microsoft Developer Network)帮助方式。本章通过 Visual Basic 运行环境和简单程序示例的接触,引出了 Visual Basic 程序设计思想、集成开发环境和 VB 应用程序的构成、创建、保存、运行和编译。

本章目标

- ✧ 了解 Visual Basic 系统对计算机软、硬件资源的要求;
- ✧ 熟悉 Visual Basic 系统和帮助文件 MSDN 的安装与选项的设置;
- ✧ 熟悉和掌握 Visual Basic 可视化集成开发环境各元素的使用;
- ✧ 熟练掌握 Visual Basic 应用程序创建、编辑、保存和运行、调试的全过程;
- ✧ 通过简单 Visual Basic 应用程序的创建准确掌握可视化编程和事件驱动编程的基本概念(对象、属性、方法和事件)和程序设计思想;
- ✧ 熟练掌握可视化编程和事件驱动编程应用程序创建的步骤和方法。

1.1 Visual Basic 语言概述

1.1.1 Visual Basic 的发展

Visual Basic 是时下流行的一种功能非常强大的第四代程序设计语言(4GL),它提供了开发 Windows 应用程序的最迅速、最简捷的方法。其中,“Visual”指的是图形用户界面(GUI)的可视化开发方法,即用户在创建应用程序时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拖拉到屏幕上的合适位置即可。“Basic”是 Beginners All-Purpose Symbolit Instruction Code(初学者通用指令代码)的缩写。与其他高级语言相比,它的语法规则相对简单,容易理解和掌握。

Basic 语言经历了一个不断发展变化的过程(如图 1-1 所示):

(1) 早期的 Basic 语言功能较为简单,只提供了解释程序,只能满足简单程序设

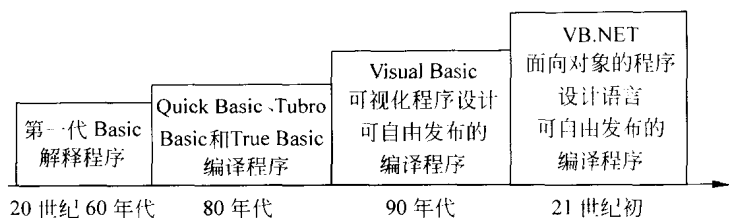


图 1-1 Basic 语言发展演变

计的要求,为开发者提供的开发工具也只是简单的行编辑。

(2) 20 世纪 80 年代,Borland 公司率先在其 PC 机软件产品中提供集成环境,免去了记忆大量命令的麻烦,为软件开发者提供了以往没有的方便高效的软件开发环境。这一时期的 Basic 语言在保留原有的解释功能的基础上,提供了与其他高级语言一样的编译功能,其中较有代表性的有 Quick Basic、Tubro Basic 和 True Basic 等。

(3) 随着计算机技术的飞速发展,计算机操作系统也不断完善。Microsoft 公司推出的 Windows 系统提供了一个图形化的用户界面(GUI)、多任务环境、动态数据交换(DDE)、对象链接和嵌入(OLE)等功能的操作环境。Windows 操作系统为用户提供了前所未有的先进功能,同时也为软件的开发增加了难度。各种基于 Windows 的可视化程序设计语言纷纷出台,像 Visual Basic、Visual C++、Delphi、Visual Foxpro 等。

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。另外,目前许多应用软件中提供的 VBA(Visual Basic for Application)和广泛使用的脚本语言 VBScript(Visual Basic Scripting Edition)都是 Visual Basic 的子集。不论是 Microsoft Windows 应用程序的资深专业开发人员还是初学者,Visual Basic 都为他们提供了整套工具,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 的遍及全球的分布式应用程序,都可在 Visual Basic 提供的工具中各取所需。

(4) 21 世纪初,为了使 Visual Basic 的开发人员从面向对象的设计方法中受益并使企业网站应用程序的开发更简单,Microsoft 公司推出了具有面向对象编程语言所有特点的 Visual Basic .NET(VB.NET),带来了许多新的功能,比如继承的实现,重载和参数化构造函数,支持完全的面向对象的编程功能的实现。另外,随着现代化的语言结构的调整,如结构化的异常处理,代码更易于维护,这使得开发人员将可以使用简洁明了的自由线程来编写高扩展性的代码。

本书选取 Visual Basic 6.0 中文版作为蓝本介绍。

1.1.2 Visual Basic 及 MSDN 的安装

1. 安装前的软硬件需求

针对不同的应用需求,Visual Basic 6.0 提供了学习版、专业版和企业版三个不同的版本,功能逐版增强,读者可以根据自己开发的需要选择安装。运行 Visual Basic 企业版

的系统软硬件要求如下：

- 486DX/66MHz 或更高的处理器(推荐 Pentium 或更高的处理器),或任何运行于 Windows NT Workstation 的 Alpha 处理器;
- Windows 95 或更高版本,或 Windows NT Workstation 4.0(推荐 Service Pack 3)或更高版本;
- 一个 CD-ROM/DVD-ROM 驱动器和鼠标器;
- 16MB RAM for Windows 95、32MB RAM for Windows NT Workstation;
- 最小 90MB 硬盘空间,如果安装 MSDN 另需 70MB 硬盘空间。

2. 安装 Visual Basic 6.0 系统文件

(1) Visual Basic 6.0 可以从单独的 VB 6.0 安装光盘或 Microsoft Visual Studio 6.0 套装盘安装,一般插入安装盘后系统会自动运行 AutoPlay.exe,也可通过运行 Setup.exe 启动安装向导程序。

(2) 系统启动安装向导,分别提示用户阅读“README”文件、同意“最终用户许可协议”、输入合法产品的 ID 号(可在安装光盘附带的用户手册上找到)、用户姓名及公司、服务器安装程序(注:如无特殊要求,请选择“安装 Visual Basic 6.0 中文企业版(I)”)。

(3) 依照屏幕上的安装指令选择或输入后,系统进行 Visual Basic 6.0 安装初始化设置并提示用户“选择安装类型”,如图 1-2 所示。

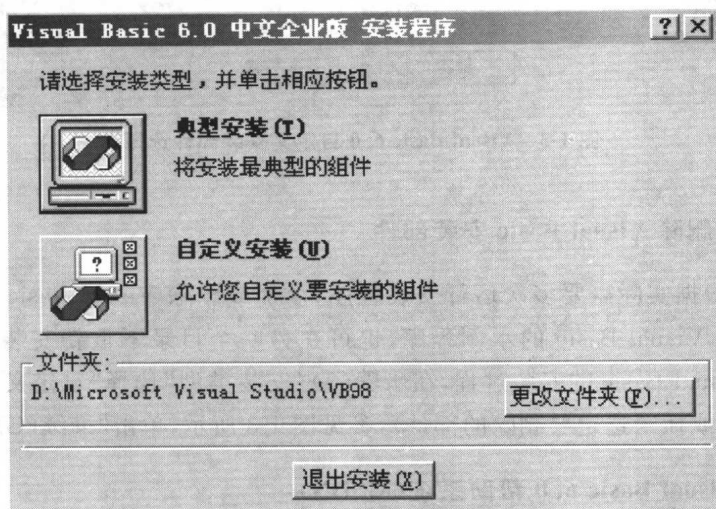


图 1-2 Visual Basic 6.0 安装类型选择

VB 6.0 有两种安装方式:典型安装和自定义安装。初学者可以选择“典型安装”方式,按系统默认方式选择安装部件,并进入步骤(5);选择“自定义安装”,将进入步骤(4),要求用户指定要安装的部件。

系统默认的安装路径是在“Program Files”下创建“Microsoft Visual Studio\VB98”,用户可单击“更改文件夹(F)...”按钮重新指定安装位置。

(4) “自定义安装”方式下允许用户调整需要安装及不需要安装的项目,如图 1-3 所

示。其中选中某一项后,单击“更改选项(G)...”按钮可以对该项目下的明细项目作进一步的选择,系统自动根据用户所选项目检测并显示目标盘上可用的磁盘空间及安装文件所需的磁盘空间。

(5) 设置好安装部件(若选择“典型安装”,则直接进入该步),系统开始向硬盘上复制系统文件并显示安装进度、Visual Basic 功能介绍等,文件复制完成后,弹出“安装成功”提示画面,单击“确定(O)”按钮完成安装。

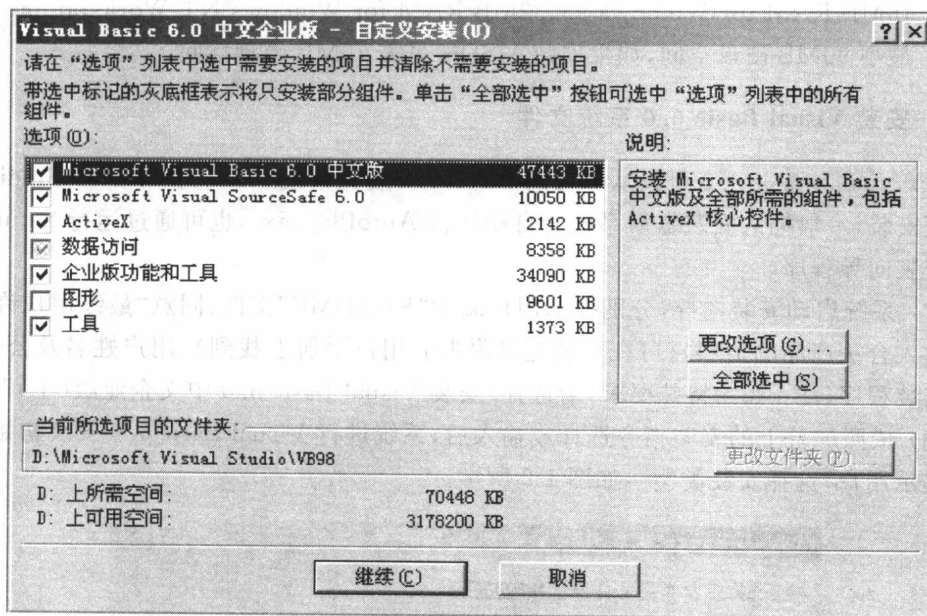


图 1-3 Visual Basic 6.0 自定义安装部件选择

3. 添加或删除 Visual Basic 安装部件

用户可以根据实际需要多次运行安装程序,添加或删除 Visual Basic 安装部件。例如,可添加安装 Visual Basic 的示例程序,也可在另一个目录下重新安装 Visual Basic。具体步骤同 Visual Basic 的安装过程,在步骤(3)“安装类型”选择“自定义安装”,将要添加的部件选中(或撤销选定要删除的部件),参见图 1-3 所示,单击“继续(C)”按钮即可。

4. 安装 Visual Basic 6.0 帮助系统(MSDN)

从 Microsoft Visual Studio 6.0 开始,所有的联机帮助文档都采用全新的 MSDN (Microsoft Developer Network) 帮助方式。MSDN Library 中涉及 MSDN 查阅器的系统文件、目录索引文件、Microsoft Visual Studio 开发产品要使用的帮助文件和大量示例代码,是学习和使用 Microsoft Visual Studio 开发工具的非常重要的参考资源。

安装完 Visual Basic 6.0,系统会提示用户选择“是否安装 MSDN Library”,用户可以插入 MSDN 光盘(MSDN 单独存放于两张光盘中)继续安装,或在以后需要时直接使用 MSDN 光盘安装。

MSDN 安装过程中需要在“典型”、“完全”和“自定义”三个安装选项中做出选择。“典型”安装方式只将最小的文件集复制到硬盘上,包括 MSDN 查阅器的系统文件、目录索引文件,以及 Visual Studio 开发产品要使用的帮助文件,用户使用帮助文件时需要从光盘上读取 MSDN Library 帮助信息。

“完全”安装方式会将 MSDN 帮助系统的所有文件安装在本地硬盘,由于该方式占用硬盘资源大,而且通常使用者不会用到 Visual Studio 6.0 的所有编程工具,因此不推荐采用这种安装方式。

“自定义”安装方式中可以指定在本地硬盘安装的 MSDN Library 文件,如用户可以选择只将 Visual Basic 的帮助文档装入本地硬盘。如果在使用帮助文件时所选择的内容尚未安装在本地硬盘,系统会提示插入 MSDN Library CD。

1.1.3 Visual Basic 的启动

成功安装 VB 后,就可通过单击 Windows 任务栏中的“开始”按钮,从“程序”菜单下找到“Microsoft Visual Basic 6.0 中文版”启动 Visual Basic,显示如图 1-4 所示界面。另外,经常使用 VB 的用户可以在 Windows 桌面或快速启动工具栏中设置 VB 的快捷启动图标,以简化启动时的查找。

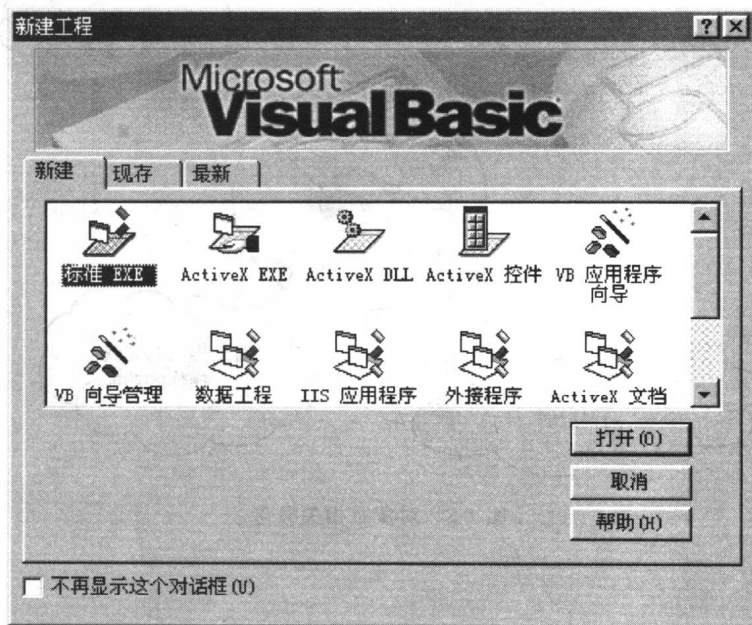


图 1-4 Visual Basic 启动界面

VB 启动界面窗口中以选项卡提供了三种应用程序打开方式:

- 新建选项卡。创建一个新的应用程序。该方式下提供了几种 Visual Basic 常用应用程序创建的模板,普通的应用程序可选择“标准 EXE”。
- 现存选项卡。从磁盘文件中选择并打开一个已经创建好的应用程序。

- 最新选项卡。给出了系统最近打开过的应用程序列表,可以从中选择并打开某一个应用程序。

选择某一项新建应用程序模板(如“标准 EXE”)或某一已有应用程序,单击“打开(O)”按钮即进入 Visual Basic 应用程序可视化的集成开发环境。

1.2 VB 的程序设计思想

在使用 Visual Basic 开发应用程序之前,先要理解 Visual Basic 的程序设计思想和一些关键概念。

1.2.1 对象、属性、方法和事件

Visual Basic 是基于 Windows 应用程序开发的程序设计语言,属于第四代高级程序设计语言(4GL),其程序设计的核心是由对象及响应各种事件的程序代码构成。与之相关的概念包括对象、对象的属性、对象的方法和对象的事件,可以用图 1-5 所示模型形象化地展现这几个概念及其相互间关系。

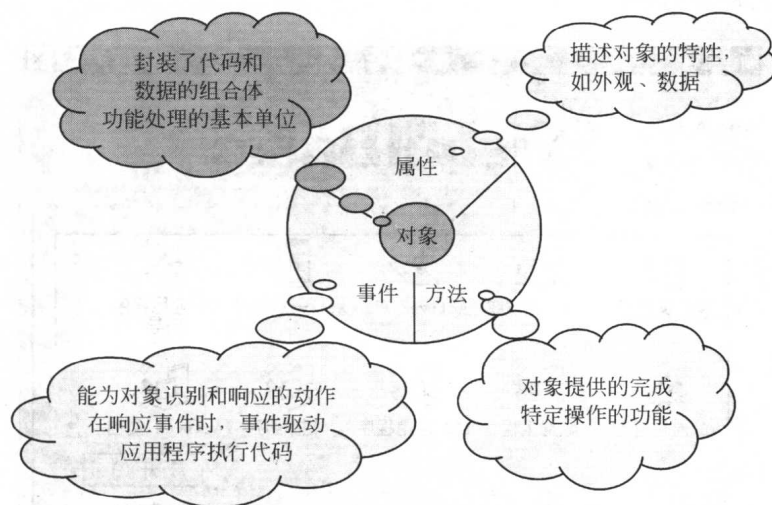


图 1-5 对象及相关概念

1. 对象

对象是 Visual Basic 程序设计的核心, Visual Basic 中界面的设计、数据的存取、功能的实现等都是通过对象实现的,可以说 Visual Basic 程序设计过程中到处都有对象存在。对象是封装了数据和代码的组合物,是 Visual Basic 中功能处理的基本单位。正是由于对象同时包含了功能代码和数据,这比传统的编写代码方法更容易维护。

对象是现实生活中具体事物的抽象,如大家工作使用的计算机是一个对象,同时构

成计算机的各组成部件,如显示器、键盘、CPU等本身也是一个个的对象。每个对象都有其各自的特性、它所能完成的功能和对外界刺激做出的反应。如显示器有颜色、显示屏尺寸等特征,可以在显示屏的某一位置显示某种文字或图形,用户调整明暗度后,以不同的亮度显示信息。再如图 1-5 所示的图片,大圆、小圆、云状说明、线条等都是不同的对象。这些在程序设计语言中以对象及其属性、方法、事件来模拟表示。

同种对象的抽象与集合构成了类,它包含了对对象的属性描述和行为特征的定义,是用于创建对象实例的模板。即类是对象的定义,对象是类的实例。如计算机是一个抽象的概念,描述了所有计算机的构成、性能、它能实现的行为特征等,而具体的某一台计算机则是计算机的一个实例,有着明确的构成、性能及外观特征值等。

每个对象都必须有用于惟一识别该对象的名称,由程序设计人员以一个容易理解的名称命名。如图 1-5 中“对象”所在的小圆命名为 ConceptObject,“对象”的云状说明命名为 DescripObject。

2. 对象的属性

属性是描述对象特性的数据,如颜色、大小、位置、数据等。通过改变对象的属性就可改变对象的特性。如图 1-5 中,对象 ConceptObject 有半径、圆心坐标、颜色、显示文字等属性。

对象的属性有专有的名称,如用 BackColor 描述 ConceptObject 的背景颜色,用 Font 描述其显示文字的字体等。如果要改变对象的外观特性或数据时,通过设置属性的值改变。例如,通过改变 ConceptObject 对象的 Font 属性,就可以改变 ConceptObject 上显示内容的字体。同样的,也可以通过读取对象属性的值获取该对象的相关数据。对属性的引用使用“对象名.属性名”的语法格式。

示例 1.1 将 ConceptObject 显示文字的字体设置为“宋体”。

```
ConceptObject.Font = "宋体"
```

示例 1.2 要将 DescripObject 的背景颜色设置为与 ConceptObject 的背景颜色一致。

需要读取 ConceptObject 的背景颜色值,并将其赋给 DescripObject 的 BackColor 属性,使用如下代码:

```
DescripObject.BackColor = ConceptObject.BackColor
```

大部分属性可以在设计时(即在 Visual Basic 集成开发环境的“属性”窗口中)设置而无需编写任何代码;而有的属性在设计时是不可用的,只有通过代码在运行时设置,如文本框中被用户选中的字符。一部分属性在运行时可以设置并可读取值,这样的属性叫做读写属性,如示例 1.2 中的 BackColor 属性;而一部分属性在运行时只能读取,不能设置,这样的属性叫做只读属性,如对象的名称属性。

3. 对象的方法

通常对象都能完成某些功能,这就是对象的方法,它是对象提供的完成特定操作的