

Visual Basic 程序设计教程

主 编 苏 啸 周成义 王 强
副主编 胡艳维 张刚强 张立君
康丽军 张俊才

北京工业大学出版社

内 容 提 要

本书按照“厚基础，强能力”的指导思想，在教学内容的选取、教学的组织和方法上，注重高级语言的语法规则、面向对象程序设计的基本方法的学习和掌握，以及分析和解决实际问题能力的培养，为计算机程序设计的学习与应用打下基础。本书共分 16 章，第 1~4 章为入门知识，其中包括 VB 开发环境的介绍和面向对象程序设计的基本概念；第 5~14 章主要介绍 VB 的语言基础、用户设计界面以及全部内部控件的详细内容；第 15、16 章介绍数据库的访问、ActiveX 控件和 Windows API。

本书适用于高校公共计算机课程教学，通过学习和掌握本课程，可使学生通过计算机等级考试并为提高计算机编程的专业水平打下基础。

图书在版编目 (CIP) 数据

Visual Basic 程序设计教程 / 苏啸, 周成义, 王强主编. —北京: 北京工业大学出版社, 2005.3

ISBN 7-5639-1498-6

I.V… II.①苏…②周…③王… 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 (2005) 第 019394 号

Visual Basic 程序设计教程

主编 苏 啸 周成义 王 强

*

北京工业大学出版社出版发行

邮编: 100022 电话: (010) 67392308

各地新华书店经销

徐水宏远印刷厂印刷

*

2005 年 3 月第 1 版 2005 年 3 月第 1 次印刷

787mm × 1092mm 16 开本 21.75 印张 543 千字

印数: 0001 ~ 5000 册

ISBN 7-5639-1498-6/T·242

定价: 38.00 元

前 言

《Visual Basic 程序设计教程》的出版，得到了很多领导和老师的支持，对此，我们深感荣幸，同时对关心和支持本书并提出宝贵意见和建议的各位教师表示衷心感谢！编写本书的目的有两个：一是适应高校公共计算机课程的教学，让学生能够顺利通过计算机等级考试；二是适应想进一步提高计算机编程水平的需要，即实现计算机专业化教学。

本书主要应用于公共计算机等级考试的教学，也适用于计算机专业的入门教学。公共计算机部分为第1~12章，建议至少64课时。专业部分为第13~16章，建议至少32课时，共计96课时。

Visual Basic 是1991年发布的，从 Visual Basic1.0 到 Visual Basic6.0 中文标准版、专业版、企业版和 VB.NET，微软在不断地改进 Visual Basic。Visual Basic 程序设计发展至今，其包含的概念、内容越来越多，功能越来越强，要想在短时间内学会它也越来越困难。为此，我们按照“厚基础，强能力”的指导思想，在教学内容的选取、教学组织和方法选择上，注重高级语言的语法规则、面向对象程序设计的基本方法，以及分析和解决实际问题能力的培养，为计算机程序设计学习和应用打下基础。本书提供了较多的难度适中的完整范例，有利于学生模仿学习。

为了突出教师授课和学生自学时由浅入深、循序渐进的过程，本书内容安排如下：

本书共分16章，第1~4章是入门知识，其中包括 VB 开发环境的介绍和面向对象程序设计的基本概念。为了使读者理论联系实际，充分体会 VB 可视化编程的好处，以及照顾到后续章节的学习和上机实践，常用的几种控件和输入输出方法的介绍也放在了这一部分加以讲述。第5~14章主要介绍 Visual Basic 的语言基础、用户设计界面以及全部内部控件的详细内容，讲解了多个实例的编程，使读者能够利用编程工具进行程序的调试和错误处理。第15、16章介绍数据库的访问、ActiveX 控件和 Windows API，并且有较完整的实例介绍。

本书由苏啸、周成义、王强担任主编，由胡艳维、张刚强、张立君、康丽军、张俊才担任副主编。参加编写的有（按姓氏笔画为序）：王强、王学周、王雪梅、王淑敏、冯改娥、任玉玲、刘熹、刘秋萍、苏啸、张立君、张刚强、张俊才、宋玉璞、李希勇、周成义、钟友华、胡艳维、康丽军、崔静、崔振远。由于时间紧迫及作者水平有限，书中难免有不足之处，敬请专家和读者批评指正。

编 者

2005年2月

目 录

前 言	(1)	2.2.2 窗体的结构与属性.....	(24)
第 1 章 Visual Basic 概述	(1)	2.3 控件	(34)
1.1 计算机程序设计语言简介	(1)	2.3.1 控件、标准控件.....	(34)
1.1.1 程序设计语言的发展	(1)	2.3.2 控件的命名.....	(35)
1.1.2 程序设计语言的发展趋势	(2)	2.3.3 控件属性值.....	(35)
1.1.3 计算机程序设计方法的 发展概况	(3)	2.3.4 常用控件.....	(36)
1.2 Visual Basic 的基本特点	(3)	2.4 控件的画法和基本操作	(36)
1.3 Visual Basic 的安装、启动和退出	(4)	2.4.1 添加一个新控件.....	(36)
1.3.1 安装 Visual Basic	(4)	2.4.2 移动控件位置和调整控件大小.....	(37)
1.3.2 启动和退出 Visual Basic	(5)	2.4.3 控件的复制和粘贴.....	(37)
1.4 Visual Basic 集成开发环境	(6)	2.4.4 多个控件操作.....	(38)
1.4.1 主窗口	(7)	2.4.5 在工具栏中添加新控件.....	(38)
1.4.2 工程窗口	(8)	2.5 事件驱动	(39)
1.4.3 属性窗口	(9)	2.5.1 事件.....	(39)
1.4.4 窗体设计器.....	(10)	2.5.2 常用事件.....	(40)
1.4.5 代码编辑窗口.....	(10)	小结	(41)
1.4.6 立即窗口.....	(11)	习题	(43)
1.5 开发 Visual Basic 应用程序的 基本步骤	(11)	第 3 章 数据类型及其运算	(44)
1.5.1 Visual Basic 应用程序示例.....	(11)	3.1 数据类型	(44)
1.5.2 基本步骤.....	(13)	3.1.1 基本数据类型.....	(44)
小结	(14)	3.1.2 用户自定义类型.....	(46)
习题	(15)	3.2 常量与变量	(47)
第 2 章 对象及其操作	(16)	3.2.1 常量.....	(47)
2.1 Visual Basic 的对象	(16)	3.2.2 变量.....	(49)
2.1.1 对象的属性、方法和事件.....	(16)	3.3 表达式和运算符	(50)
2.1.2 对象属性设置、对象变量的 声明和引用.....	(18)	3.3.1 算术运算符.....	(50)
2.1.3 对象数组.....	(19)	3.3.2 字符串运算符.....	(51)
2.1.4 对象集合和集合对象.....	(19)	3.3.3 关系运算符.....	(52)
2.1.5 Screen 对象和 Printer 对象	(20)	3.3.4 逻辑运算符.....	(53)
2.2 窗体	(23)	3.3.5 日期运算符.....	(53)
2.2.1 窗体的创建.....	(23)	3.3.6 运算符的优先级.....	(54)
		3.4 Visual Basic 的常用函数	(54)
		3.4.1 算术函数.....	(54)
		3.4.2 字符串函数.....	(54)
		小结	(58)

习题	(59)	5.8 框架控件	(87)
第4章 数据输入、输出	(60)	5.9 焦点与 Tab 顺序	(88)
4.1 数据输出	(60)	5.9.1 设置焦点	(88)
4.1.1 Print 方法	(60)	5.9.2 Tab 顺序	(88)
4.1.2 格式输出 (Format\$)	(61)	5.10 内部控件应用范例	(89)
4.2 InputBox 函数	(62)	小结	(92)
4.3 MsgBox 函数和 MsgBox 语句	(64)	习题	(93)
4.3.1 MsgBox 函数	(64)	第6章 流程控制语句	(95)
4.3.2 MsgBox 语句	(66)	6.1 基本语句	(95)
4.4 字形	(67)	6.1.1 代码编写机制	(95)
4.5 打印输出	(67)	6.1.2 赋值语句	(97)
4.5.1 直接输出	(68)	6.1.3 注释语句	(97)
4.5.2 窗体输出	(68)	6.1.4 程序退出或暂停语句	(97)
小结	(68)	6.1.5 错误处理语句	(98)
习题	(69)	6.2 选择结构	(98)
第5章 Visual Basic 的标准控件	(71)	6.2.1 IF 语句的两种形式	(98)
5.1 文本控件	(71)	6.2.2 块结构条件语句	(99)
5.1.1 标签控件	(71)	6.2.3 If 函数	(102)
5.1.2 文本框控件	(72)	6.3 多路分支结构	(102)
5.2 图形与装饰控件	(74)	6.4 循环控制结构	(105)
5.2.1 图片框属性、事件和方法	(74)	6.4.1 FOR 循环结构	(105)
5.2.2 图像框属性、事件和方法	(75)	6.4.2 当循环控制结构	(107)
5.2.3 图形文件的装入	(76)	6.4.3 Do 循环控制结构	(108)
5.2.4 直线控件	(76)	6.5 多重循环	(109)
5.2.5 形状控件	(77)	6.6 跳转语句	(111)
5.3 命令按钮控件	(78)	6.6.1 无条件跳转语句 (GoTo 语句)	(111)
5.3.1 Caption (标题) 属性	(78)	6.6.2 条件跳转语句 (On - GoTo 语句)	(112)
5.3.2 Default 和 Cancel 属性	(78)	小结	(112)
5.3.3 Style (类型) 与 Picture (图片) 属性	(78)	习题	(113)
5.3.4 Click 事件	(79)	第7章 数组	(114)
5.4 选择控件——复选框和单选按钮	(80)	7.1 数组	(114)
5.4.1 复选框控件 (CheckBox)	(80)	7.1.1 数组的概念	(114)
5.4.2 单选按钮控件 (OptionButton)	(80)	7.1.2 静态数组及声明	(115)
5.5 选择控件——列表框和组合框	(82)	7.1.3 动态数组及声明	(116)
5.5.1 列表框控件	(82)	7.2 数组的基本操作	(117)
5.5.2 组合框控件	(85)	7.2.1 数组的初始化	(117)
5.6 滚动条控件	(86)	7.2.2 数组元素的输入	(117)
5.7 定时器控件	(87)	7.2.3 输出和复制	(118)
5.7.1 Timer 控件属性	(87)	7.2.4 For Each...Next 语句	(119)
5.7.2 Timer 事件	(87)		

7.3 控件数组	(120)	9.2.4 访问键和快捷键	(149)
7.3.1 控件数组的概念	(120)	9.3 弹出式菜单	(151)
7.3.2 控件数组的建立	(120)	9.3.1 弹出式菜单	(151)
7.3.3 自定义数据类型及其数组	(122)	9.3.2 显示弹出式菜单	(151)
7.4 数组应用举例	(124)	9.4 创建菜单控件数组	(152)
7.4.1 分类统计	(124)	9.5 通用对话框	(153)
7.4.2 大量数据的输入和编辑	(125)	9.5.1 通用对话框	(153)
7.4.3 数组排序	(126)	9.5.2 通用对话框的基本属性和 方法	(153)
小结	(129)	9.5.3 通用对话框的显示方法	(154)
习题	(131)	9.5.4 “文件”对话框的使用	(154)
第 8 章 过程概述	(132)	9.5.5 “颜色”对话框的使用	(155)
8.1 Sub 过程	(132)	9.5.6 “字体”对话框的使用	(156)
8.1.1 Sub 过程的建立	(132)	9.5.7 “打印”对话框的使用	(156)
8.1.2 通用过程	(133)	9.5.8 “帮助”对话框的使用	(156)
8.1.3 事件过程	(133)	小结	(156)
8.2 Function 过程	(134)	习题	(157)
8.2.1 Function 过程的定义	(134)	第 10 章 多重窗体应用程序	(158)
8.2.2 使用 Function 过程	(134)	10.1 建立多文档界面	(158)
8.3 调用过程	(135)	10.1.1 建立多文档界面步骤	(159)
8.3.1 调用 Sub 过程	(135)	10.1.2 设置子窗体的大小和位置	(160)
8.3.2 调用 Function 过程	(136)	10.1.3 Dim 语句和 WindowState 属性	(160)
8.3.3 调用其他模块中的过程	(136)	10.1.4 多文档窗体示例	(160)
8.4 子过程和函数的参数传递	(137)	10.2 MDI 的属性、事件和方法	(161)
8.4.1 向过程传递参数	(137)	10.3 多文档界面的几点说明	(162)
8.4.2 按值传递参数	(138)	10.4 综合应用实例	(163)
8.4.3 按地址传递参数	(138)	小结	(166)
8.5 使用可选参数	(138)	习题	(167)
8.5.1 可选参数	(138)	第 11 章 事件过程	(168)
8.5.2 用命名的参数创建简单语句	(140)	11.1 常用的窗体事件	(168)
8.5.3 确定对命名参数的支持	(140)	11.1.1 Load 事件	(168)
小结	(140)	11.1.2 Paint 事件	(169)
习题	(141)	11.1.3 Resize 事件	(170)
第 9 章 菜单设计与通用对话框	(143)	11.1.4 Activate、Deactivate 事件	(171)
9.1 菜单概述	(143)	11.1.5 QueryUnload 事件	(172)
9.1.1 菜单	(143)	11.1.6 Unload 事件	(174)
9.1.2 下拉菜单	(143)	11.2 键盘事件	(175)
9.1.3 弹出式菜单	(144)	11.2.1 KeyPress 事件	(175)
9.2 菜单编辑器	(145)	11.2.2 KeyDown、KeyUp 事件	(176)
9.2.1 菜单编辑器构成	(145)	11.3 鼠标事件	(178)
9.2.2 创建基本菜单控件	(146)	11.3.1 Click 事件	(178)
9.2.3 创建子菜单	(147)		

11.3.2 DblClick 事件	(179)	习题	(227)
11.3.3 MouseMove, MouseDown, MouseUp 事件	(180)	第 14 章 程序调试与错误处理	(228)
11.4 焦点事件	(182)	14.1 错误的概念与类型	(228)
11.4.1 GotFocus 事件	(182)	14.1.1 错误的概念	(228)
11.4.2 LostFocus 事件	(183)	14.1.2 错误的类型	(228)
11.5 拖动事件	(184)	14.2 错误的处理	(229)
11.5.1 DragDrop 事件	(184)	14.2.1 设计错误处理程序	(231)
11.5.2 DragOver 事件	(185)	14.2.2 设置错误捕获	(231)
小结	(187)	14.2.3 编写错误处理例程	(231)
习题	(188)	14.2.4 退出错误处理例程	(232)
第 12 章 文件	(189)	14.2.5 关闭错误处理	(241)
12.1 文件的读写	(189)	14.3 程序调试	(242)
12.1.1 文件类型	(189)	14.3.1 调试基础	(242)
12.1.2 顺序文件的读写	(190)	14.3.2 程序调试方法	(245)
12.1.3 随机文件的读写	(193)	小结	(254)
12.1.4 二进制文件的读写	(197)	习题	(255)
12.2 有关文件系统的语句和函数	(198)	第 15 章 数据库访问	(256)
12.3 FSO 对象模型	(199)	15.1 数据库访问概述	(256)
12.3.1 FSO 对象使用基础	(200)	15.1.1 ADO.NET	(256)
12.3.2 访问驱动器	(201)	15.1.2 ADO	(257)
12.3.3 访问文件夹	(202)	15.1.3 OLE DB	(257)
12.3.4 访问文件	(204)	15.1.4 选择 ADO.NET 或 ADO	(257)
小结	(207)	15.2 ADO 对象模型	(259)
习题	(207)	15.2.1 ADO 对象模型	(259)
第 13 章 绘图	(208)	15.2.2 Connection 对象	(260)
13.1 坐标和颜色	(208)	15.2.3 Recordset 对象 (记录集)	(261)
13.1.1 坐标系统	(208)	15.2.4 Parameters 集合和 Parameter 对象	(261)
13.1.2 颜色	(210)	15.2.5 Fields 集合和 Field 对象	(262)
13.2 绘图方法	(212)	15.2.6 Properties 集合和 Property 对象	(263)
13.2.1 使用 PSet 方法画点	(212)	15.3 使用 ADO 执行查询	(264)
13.2.2 使用 Line 方法画线	(214)	15.3.1 使用 ADO 连接 SQL Server 数据源	(264)
13.2.3 使用 Circle 方法画圆	(218)	15.3.2 执行数据操作	(266)
13.2.4 使用 PaintPicture 方法显示图片	(221)	15.3.3 处理结果	(271)
13.3 直线和形状控件	(223)	15.4 ADO 高级特性	(272)
13.3.1 直线控件	(223)	15.4.1 在 ADO 中使用游标	(272)
13.3.2 形状控件	(224)	15.4.2 在 ADO 中执行事务	(274)
13.4 打印文字和图形	(224)	15.4.3 使用 ADO Stream 对象访问并修改 SQL Server BLOB 数据	(275)
13.4.1 使用窗体的 PrintForm 方法	(225)	15.5 数据环境设计器	(277)
13.4.2 使用 Printer 对象	(225)	15.5.1 数据环境设计器简介	(277)
小结	(226)		

15.5.2 操作数据环境····· (277)	16.2 Windows API 函数····· (317)
15.5.3 创建数据库连接····· (278)	16.2.1 API 函数基本概念····· (317)
15.5.4 数据环境设计器和 Form 窗体中数 据显示控件的绑定····· (283)	16.2.2 API 函数的应用····· (319)
小结····· (292)	小结····· (324)
习题····· (292)	习题····· (324)
第 16 章 ActiveX 控件和 Windows API····· (293)	全国计算机等级考试二级笔试
16.1 ActiveX 控件····· (293)	试卷 (2004 年)····· (325)
16.1.1 ActiveX 控件概念····· (293)	附: 全国计算机等级考试二级笔试
16.1.2 VB 常用的 ActiveX 控件简介····· (294)	试卷 (2004 年)····· (337)

第 1 章 Visual Basic 概述

Visual Basic (可视化 BASIC, 简称 VB) 是微软公司推出的一种新型的现代程序设计语言。VB 虽然以 BASIC 语言为基础, 但是在许多方面都做了重大的改进和突破。VB 的产生使更多的专业、非专业人员都能够参与到开发 Windows 应用程序的行列中, 使计算机程序设计进入了新的阶段。本章介绍计算机程序设计语言及方法的发展概况, VB 的特点、启动和退出、集成开发环境和开发 VB 应用程序的基本步骤。

1.1 计算机程序设计语言简介

人们在使用计算机解决实际问题时, 需要用某种特定的“语言”同计算机交流、沟通, 计算机语言是人类与计算机交流信息的主要途径。这类语言通过语法、语义、描述记号来表述各种运算和处理过程, 能够为计算机所识别、理解、执行, 最终完成某项工作任务。这些语言统称为计算机程序设计语言。

1.1.1 程序设计语言的发展

程序设计语言是与计算机硬件共同诞生、共同发展的, 随着计算机及其应用的不断发展, 它也在发展中不断完善、变革、创新。目前, 程序设计语言的发展可概括为四个阶段。

1. 机器语言

机器语言属于第一代程序设计语言。机器语言是指由 0、1 二进制代码组成的能被计算机直接识别的机器指令的集合。机器语言能直接针对计算机的硬件结构描述各种算法, 因此不需要翻译, 就能够被计算机直接执行, 因此执行速度快。但是, 用机器语言编程非常繁琐, 程序的可读性极差, 程序的修改、调试极不方便。另外, 机器语言是面向机器的, 不同机器的指令系统也不同, 不利于计算机的推广和应用。

2. 汇编语言

汇编语言采用一些助记符号表示机器语言中的指令和数据, 相对于机器语言来说, 比较容易记忆和学习, 所以也称汇编语言为符号语言。它用便于识别的符号, 如英语单词或其缩写作为助记符, 来代替机器指令编写程序, 然后由专门的转换程序, 将这些符号转换为机器语言指令代码。用汇编语言也能够针对计算机的硬件去编写程序, 因此执行速度快。目前, 针对一些实时性要求较高的实际问题, 仍可以采用汇编语言来编写程序。但是它和机器语言一样, 对机器的依赖性较强, 语言的通用性等问题没有得到根本解决。

3. 高级语言

高级语言采用一组通用的英语单词、数学式及规定的符号, 按严格的语法规则和逻辑关

系表述各种运算和处理过程。由于采用这种表达方式编写程序，接近自然语言和数学语言，符合人们的习惯，因此称之为高级语言。高级语言有较强的通用性，用高级语言编写的程序能够在不同的计算机系统上运行。高级语言的种类有许多，适用于不同的应用领域。

世界上出现最早的高级语言是 FORTRAN，是用于编制数值计算程序的主要语言，主要适用于科学计算方面。后来随着 C 语言的广泛应用，FORTRAN 语言逐渐被 C 语言所取代。20 世纪 60 年代以来，出现了很多高级语言，当时比较流行的有 BASIC、COBOL、PASCAL、PL/1 和 ADA 等，其中有些高级语言只适用于大、中型计算机。在 PC 不断发展、广泛应用的过程中，一些高级语言的编译程序才被移植到 PC 上。

PASCAL 语言是一种典型的结构化程序设计语言，具有丰富的数据类型、语句简明、程序结构严谨，控制语句简单、灵活等特点，它已经成为标准的计算机程序设计教学语言。由于 C 语言既适合系统软件设计，也适合应用软件设计。著名的 UNIX 操作系统就是用 C 语言编写的，C 语言是目前国际上公认的优秀的设计语言。

4. 4GL 语言

一般被称为“第四代语言”，是一种交互式、非过程化的、面向用户的编程语言，与高级语言相比它更接近人类自然语言。第四代语言经常被用于访问数据库，目前主要用于数据库开发的查询语言，SQL 就属于 4GL 语言。

1.1.2 程序设计语言的发展趋势

计算机网络的飞速发展和广泛应用，对人们的工作和生活产生了极大的影响，使当今社会发生了极大的变革。最先受到影响的，并对其发展起到很大促进作用的，当然是程序设计语言。Java 的出现就是明显的例子，可以说 Java 语言的出现是网络对程序设计语言发展影响最大的事件。

概括分析，今后程序设计语言的发展将着重在以下几个方面。

1. 智能化逐步提高

新型的计算机语言应该更容易学习、掌握，更便于使用。程序设计语言的发展过程就是一个追求方便、易用的过程。现在的计算机语言已经逐步具有一定的智能化特点，使软件开发逐步走向更高级的阶段。计算机语言的进一步发展将会使更多的用户，不需要专门的、系统的培训，就能够很容易地掌握最新程序设计的方法，应用计算机的智能化程度将会更高。

2. 具有较强的可移植性

程序在不同的计算机或操作系统平台上，不需要重新编译即可执行，真正做到“与系统无关”是人们追求的理想目标。Java 语言在某些方面更实现了这种功能，程序的通用性很强，受到广大用户的欢迎。

3. 适应网络的特性

网络编程语言随着 Internet 的深入和普及，将会得到更快发展和更广泛应用。当前比较流行的 Perl 语言，是开发网站的首选语言之一。简单易学的 VBScript、JavaScript 很适合非专业人员使用。微软最新推出的 Visual Studio.net 中，ASP.net、VC.net、VB.net、C# 等紧密与现有的 Web 开发标准结合，将会被广泛应用在网络应用程序的开发中。在未来网络的需求下，将会产生更好的网络编程语言。

4. 高效与安全性

在 IT 业的发展过程中, 社会对软件的需求增长很快, 要求软件开发具有较高的时效性和安全性, 这对现代程序设计语言提出了更高的要求。

1.1.3 计算机程序设计方法的发展概况

计算机程序设计语言的发展是随着计算机科学技术及其应用的飞速发展在不断丰富和完善的。与此同时, 程序设计的方法也得到不断的改进。人们在不断地探讨、研究, 以使程序设计的方法能够更加系统化、更接近人类自然语言的习惯和思维方式, 使软件开发更加科学化。程序设计方法的发展主要经历了以下几个阶段。

1. 模块化程序设计

将一项较大的开发任务分成若干项独立的小任务, 分头编写各个模块, 然后将这些独立的模块有机地结合起来形成一个整体, 实现系统的功能。

2. 结构化程序设计

结构化程序设计是指仅仅使用三种基本控制结构(顺序、选择和重复)实现程序的设计方法。其基本设计原则是模块化原则、自顶向下原则和逐步求精原则, 在充分的需求分析的基础上按照这一原则去实现系统的功能。

3. 面向对象的程序设计

面向对象的程序设计是一种全新的程序设计和构造软件的方法。该方法更接近人类的思维方式和自然语言的习惯, 能够直接描述客观对象。发展面向对象方法的目的是为了提高软件系统的可重用性、可扩充性和可维护性。与传统的面向过程的程序设计方法比较, 它有两点重大突破: ①操作代码与代码运行过程所需的数据共同封装在一起组成了一种新的数据类型——类; 并且“类”具有继承及扩充机制。②具有人机交互式的可视化程序设计界面。

4. 计算机辅助软件工程 (Case)

计算机辅助软件工程的目标, 就是要实现系统开发过程中各阶段的工作, 能够达到基于计算机的自动化的过程。采用辅助软件工具和方法支持开发, 将系统按功能进行划分, 利用工具对数据建模, 能够自动生成计算机程序, 完成开发设计任务。计算机辅助软件工程的成熟及使用, 还有待一个发展过程。

1.2 Visual Basic 的基本特点

VB 是在 BASIC 语言的基础上发展而来的, 它具有功能强、效率高、简单易学的特点。能够在众多的程序设计语言中成为最受欢迎的 Windows 应用程序的开发语言, 是因为它具有以下基本特点。

1. 可视化的设计平台

VB 提供的可视化的设计平台, 把 Windows 界面设计的复杂性“封装”起来, 使界面设计如同积木游戏一样成为一种享受。用户只需按设计的要求, 非常轻松地用系统提供的工具在屏幕上画出各处对象, 并为其设置相应的属性, VB 自动产生程序界面的设计代码。可视化程序设计为开发 Windows 风格的应用程序, 提供了简化编程难度的有效方法, 大幅度提高了编程的效率。

2. 事件驱动的编程机制

事件驱动是一种适用于图形用户界面的编程方式。当用户在操作界面上点击对象时，该对象就会触发一个事件，此时该事件所对应的程序代码就会被执行，从而完成了指定的操作任务。

3. 能够充分利用系统资源

动态数据交换 (Dynamic Data Exchange, DDE) 是 Windows 操作系统下应用程序间的一种标准通信方式。VB 支持 DDE，并可以实现和其他支持 DDE 的应用程序进行动态数据交换或通信。

VB 支持 Windows 的对象的链接与嵌入技术 (Object Link and Embedding, OLE)，其他应用程序的对象能够链接或嵌入到 VB 应用程序中，例如，Word 文档、Excel 电子表格、图像、声音等，使 VB 能够充分利用其他应用程序的数据。

动态链接库 (Dynamic Link Library, DLL) 是 Windows 最显著的特点之一，VB 支持这项技术。在 VB 程序运行中，需要调用函数库的某个函数时，Windows 就从 DLL 中读出并运行之。例如，可以将用 C、C++、汇编语言等编写的程序添加到 VB 程序中。

4. 具有较强的数据库管理功能

可以直接在 VB 中建立或访问 Access 桌面数据库系统，也能够访问其他外部数据库。VB 提供了开放式数据库链接 (Open Data Base Connectivity, ODBC) 的功能，可以直接访问或通过链接的方式管理并操作后台的大型数据库。现在 VB 还提供了能自动生成 SQL 语句的功能和新的 ActiveX 数据对象 (ADO)。用 VB 设计应用程序的前台界面，通过 ADO 连接后再去访问后台的数据库，是目前开发应用系统的常用方法。

5. 全面支持多媒体技术

VB 提供了多种功能的多媒体控件，能够使用户在较短的时间内，很快掌握编写图、文、声、像等多媒体应用程序的技能。

1.3 Visual Basic 的安装、启动和退出

1.3.1 安装 Visual Basic

1. VB 的版本

1991 年微软公司推出 VB 1.0 之后，其版本在不断更新。1998 年微软公司发布开发工具套件：Microsoft Visual Studio 98，其中含 VB 6.0。VB 6.0 有三个版本，不同的版本是为不同层次的人员和不同的开发需求设计的，用户可以根据自己的情况和需要购买、安装相应的软件。

(1) VB 标准版：它是初学者学习 VB 开发应用程序的学习版本，提供了各种控件和数据库访问的基本功能。

(2) VB 专业版：在标准版的功能基础上，提供了更加完整的工具集和各种附加的功能，为专业人员开发客户/服务器应用程序提供条件。

(3) VB 企业版：包含专业版的全部功能和特征，适合专业人员开发更高性能的分布式应用程序，能够快速访问 Oracle 和 Microsoft SQL Server 等数据库。

2. 基本安装过程

安装 VB 6.0 和安装其他 Windows 应用程序的过程类似。将 VB 6.0 的光盘放入光驱后,一般情况下光盘的安装程序会自动启动,进入安装界面;也可以通过“我的电脑”或“资源管理器”在光盘上找到安装程序: Setup.exe,运行该程序后进入安装界面。下面简要说明安装 VB 6.0 的基本步骤。

(1) 运行安装程序后,出现“Visual Basic 6.0 中文企业版安装向导”的对话框。

(2) 可以在此处阅读 Readme 文件,或者直接单击 [下一步] 按钮,出现“最终用户许可协议”对话框,选择 [接受协议] 单选按钮。单击 [下一步] 按钮,出现“产品和用户 ID”对话框。

(3) 输入正确的产品 ID 号(购买软件可以获得该序列号)、输入姓名和公司名称(输入的文字没有限制),然后单击 [下一步] 按钮,出现“Visual Basic 6.0 中文企业版”对话框。

(4) 选择 [安装 Visual Basic 6.0 中文企业版] 后,单击 [下一步] 按钮,出现“Visual Basic 6.0 中文企业版安装程序”对话框。

(5) 可以选择“典型安装”或者“自定义安装”。如果选择“自定义安装”,会出现对话框,用户在其中选择所需的组件。单击 [更改文件夹] 按钮,可以改变 VB 的默认安装文件夹。

(6) 单击 [继续] 按钮,安装程序将文件复制到硬盘中。复制结束后,需要重新启动计算机。

(7) 计算机再次启动后,出现“安装 MSDN”对话框,如果不需要安装 MSDN,则应该取消 [安装 MSDN] 复选框,单击 [退出] 按钮即可。需要安装 MSDN,可按后面的提示进行操作。

3. 组件的添加和删除

如果需要安装 VB 的新组件,或者删除不需要的组件,要再次运行 VB 的安装程序。在出现的对话框中,可以选择 [添加/删除]、[重新安装] 或 [全部删除] 按钮,再次出现对话框时,根据需要选择相应的选项即可。

1.3.2 启动和退出 Visual Basic

1. 启动

成功安装 VB 后,便可以启动 VB 系统了。运行 VB 系统的方法有多种,可以采用下列方法之一。

(1) 选择 [开始] → [程序] → [Microsoft Visual Basic 6.0 中文版] 程序组,在其中选择 [Microsoft Visual Basic 6.0 中文版] 命令,即可启动 VB 6.0 系统。

(2) 在桌面上双击 [Microsoft Visual Basic 6.0 中文版] 的快捷方式,或者在资源管理器中找到 VB 6.0 安装目录,双击 VB 6.exe 可执行文件,均可运行 VB 系统。

每次启动 VB 系统时,出现如图 1-1 所示的“新建工程”对话框,该对话框共有 3 个选项卡,单击 [新建] 选项卡,可以在其中选择要创建的应用程序的类型,共有 13 种类型的应用程序,选择其中的某一种应用程序类型后,单击 [打开] 按钮,即可建立一个新的 VB 工程。

单击 [现存] 选项卡,可以选择已经存在的工程文件;单击 [最新] 选项卡,可以在最近使用过的工程中选择所需要的工程文件。创造或打开工程文件后,即可进入 VB 的集成开发环境。

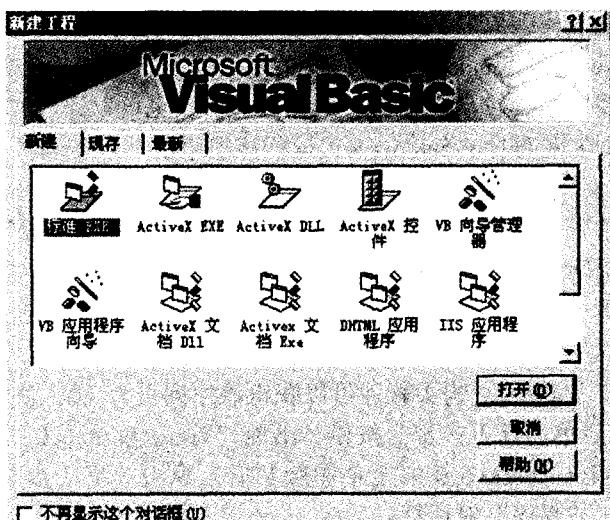


图 1-1 “新建工程”对话框

2. 退出

退出 VB 的方法和退出其他 Windows 应用程序的方法类似，可以选择下列方法之一。

(1) 选择 [文件] → [退出] 命令。

(2) 单击应用程序 [关闭] 按钮，或者按 Alt + F4 键退出当前应用程序。

(3) VB 系统在退出前，会自动判断用户在本次操作中是否修改了工程文件的内容，并询问用户是否保存文件，用户确认后可退出 VB 系统。

1.4 Visual Basic 集成开发环境

Visual Basic 系统为用户开发应用程序提供了一个良好的集成开发环境，如图 1-2 所示。它集成了各种不同的功能，例如，用户界面设计、代码编辑、模块的编译、运行、调试等。该界面由多个窗口构成了 VB 的集成开发环境。开发 Visual Basic 应用程序时，需要将这些窗口配合使用。

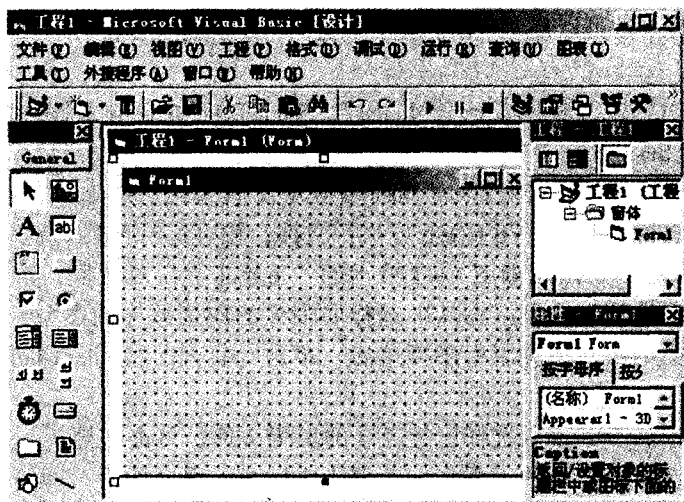


图 1-2 Visual Basic 集成开发环境

1.4.1 主窗口

VB系统的主窗口由标题栏、菜单栏和工具栏组成，为用户提供了开发VB应用程序的各种命令和工具。

1. 标题栏

标题栏与Windows其他应用程序的相似，显示当前的工程文件名和VB的当前工作模式状态，其中的[设计]，表明集成开发环境此时处于设计模式。VB的三种工作模式及其作用如下。

(1) 设计模式。在该模式下，用户可以进行程序界面的设计和代码的编写工作。程序运行后出现错误时，对程序代码的编辑过程，也在设计模式中进行。

(2) 运行模式。程序界面和代码的设计完成后，运行应用程序时处于该模式。VB应用程序运行后，一直处于等待事件发生的状态中，退出应用程序，则回到设计模式。运行阶段不能进行界面和代码的编辑工作。

(3) 中断模式。应用程序运行出现错误时，处于中断模式。该阶段可以编辑代码，重新运行程序，但是程序界面不能够被编辑。

2. 菜单栏

菜单栏中包含VB系统所有的可用命令，这是程序开发过程中用于设计、调试、运行和保存应用程序所需要的命令，共有13个菜单项。

[文件]：主要提供对工程文件的创建、打开、保存和增删等操作命令。将当前应用程序生成可执行文件的操作命令也在其中。

[编辑]：主要提供对应用程序进行编辑的各种操作命令。

[视图]：主要提供设计程序界面、运行和调试程序时各种窗口的切换。

[工程]：主要提供为工程添加窗体、模块、控制、部件等对象的命令。

[格式]：主要提供对窗体控制的对齐、尺寸及间距等格式化的命令。

[调试]：主要提供调试程序的各种命令。

[运行]：主要提供程序启动、中断和停止等命令。

[查询]：主要提供对数据库查询的相关命令。

[图表]：主要提供对图表的新建、设置、添加、显示和修改等命令。

[工具]：主要提供添加过程、过程属性、菜单编辑器、相关选项的设置及应用程序的发布等命令。

[外接程序]：主要提供在VB中进行数据库管理和外接程序管理器的功能。

[窗口]：主要提供窗口的排列和过程文件的切换命令。

[帮助]：启动帮助系统，打开帮助窗口，为用户提供学习和查询VB的相关信息。

3. 工具栏

工具栏中集中了各种用图标表示的按钮，每个按钮对应一个命令，单击按钮，即可执行对应的命令。默认情况下，VB启动后显示标准工具栏，除此，VB还提供了编辑、窗口编辑器和调试等专用的工具栏。可以通过选择[视图]→[工具栏]命令将其他工具栏在这个集成环境中移入或移出。

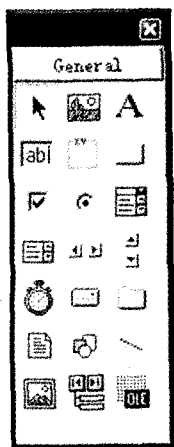


图 1-3 控件工具箱

4. 控件工具箱

控制部件简称控件，是用户设计应用程序界面的工具。VB 的标准工具箱包含建立应用程序所需的各种控件，如图 1-3 所示。另外，VB 还提供了很多 ActiveX 控件，可以将它们添加到工具箱中。如果控件工具箱在集成环境中没有出现，可以选择 [视图] → [工具箱] 命令使其再现。

标准工具箱中包含的控件，按照图 1-3 的顺序依次为：图片框 (PictureBox)、标签 (Label)、文本框 (TextBox)、框架 (Frame)、命令按钮 (CommandButton)、复选框 (CheckBox)、选项按钮 (OptionButton)、下拉式列表框 (ComboBox)、列表框 (ListBox)、水平滚动条 (HScrollBar) 和垂直滚动条 (VScrollBar)、定时器 (Timer)、驱动器列表框 (DriveListBox)、目录列表框 (DirListBox)、文件列表框 (FileListBox)、轮廓 (Shape)、直线 (Line)、图

像框 (Image)、数据控件 (Data) 和 OLE 容器 (OLE) 等。标准控件的功能和使用方法在第 2 章介绍。

工具箱作为用户开发应用程序工具的集合，用户对其使用方法掌握得越多，处理问题的能力越强，可以向工具箱添加自己需要的控件。但是，工具箱并非越大越好，系统允许用户自己定义一张“选项卡”，向其中添加自己常用的控件，如图 1-4 所示。自定义选项卡及添加控件的操作如下：

(1) 在工具箱的空白处，单击鼠标右键，在快捷菜单中选择 [添加选项卡] 命令。

(2) 在打开的对话框中输入选项卡的名称，如 UDF。

(3) 用鼠标将所需的控件拖动到选项卡 UDF 上。

(4) 重复第 (3) 步可以添加多个控件；重复以上过程可以添加多个选项卡。

用户选项卡一经定义，可以随 VB 系统保存，供经常使用，不需要时可以删除。

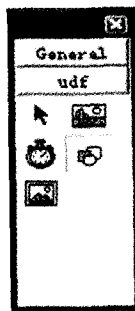


图 1-4 自定义控件工具箱

1.4.2 工程窗口

工程是应用程序各种类型文件的集合，工程文件的扩展名为 .Vbp。它包含的三类主要文件为：窗体文件 (.Frm)、标准模块文件 (.Bas)、类模块文件 (.Cls)。工程文件就是与该工程有关的所有文件和对象的清单，这些文件和对象自动链接到工程。每个工程中的对象和文件也可以供其他工程使用。

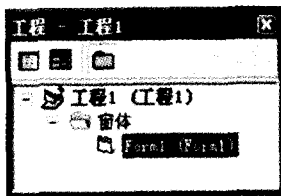


图 1-5 工程窗口

工程窗口如图 1-5 所示，类似 Windows 资源管理器窗口，窗口中列出当前工程中的窗体和模块，以层次化管理方式显示各类文件，而且允许同时打开多个工程。

在工程资源管理器标题栏的下方有三个按钮，其含义和作用如下：

[查看代码]：单击后切换到代码编辑器窗口，查看或编辑

代码。

[查看对象]: 单击后切换到窗体设计器窗口, 查看或设计当前窗体。

[切换文件夹]: 单击后可以在工程中的不同层次之间切换。

1.4.3 属性窗口

在 VB 集成环境中, 属性窗口的默认位置是在工程窗口的下方, 如图 1-6 所示。单击工具栏中的属性窗口按钮或按 F4 键, 可以使隐藏起来的属性窗口再次出现。

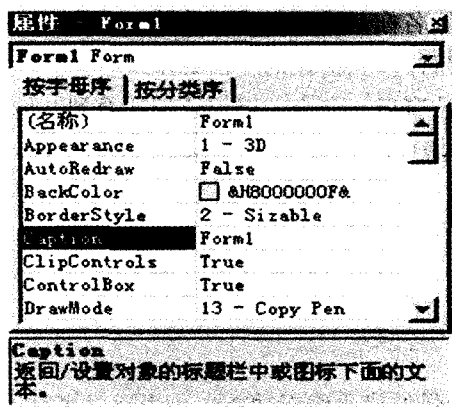


图 1-6 属性窗口

应用程序中的窗体及其控件的属性, 均可以通过属性窗口设置, 例如, 名称、标识、颜色、字体等。属性窗口由以下几部分组成。

(1) 对象下拉列表框: 标识当前对象的名称及其所属的类别, 例如, 图 1-7 中 Form1 是名称, Form 说明是窗体类。单击图 1-6 中对象下拉列表框右边的箭头可列出所选窗体中包含的对象列表。

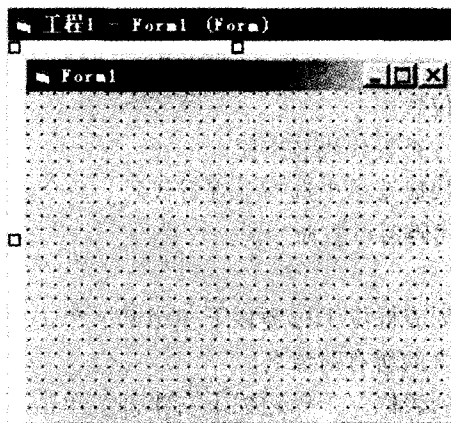


图 1-7 窗体设计器

(2) 选项卡: 可按字母序或分类序两种方式列出所选对象的所有属性。

(3) 属性列表: 该表中列出所选对象的各个属性的默认值, 可以在设计模式或运行模式中重新设置、修改其属性值。不同的对象, 列出的属性也不尽相同。列表左边列出的是各种属性, 右边对应的是属性值。