



工业和信息化普通高等教育“十三五”规划教材立项项目

21 世纪高等学校计算机规划教材

21st Century University Planned Textbooks of Computer Science

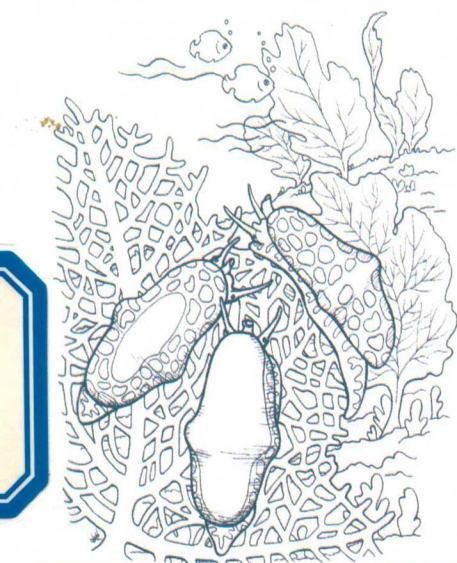
Visual Basic 语言 程序设计教程

Visual Basic Programming Language

吴昊 周美玲 主编

熊李艳 雷莉霞 副主编

- 通俗易懂，易于初学者自主学习
- 内容充实，理论与实践紧密结合
- 面向应用，全方位提升编程技能



高校系列



中国工信出版集团



人民邮电出版社
POSTS & TELECOM PRESS



工业和信息化普通高等教

才立项项目

21 世纪高等学校计算机规划教材

21st Century University Planned Textbooks of Computer Science

Visual Basic 语言 程序设计教程

Visual Basic Programming Language

吴昊 周美玲 主编

熊李艳 雷莉霞 副主编



吴昊 周美玲 主编

熊李艳 雷莉霞 副主编

人民邮电出版社

北京市丰台区右安门内大街 26 号

人民邮电出版社

北京市丰台区右安门内大街 26 号

人民邮电出版社

北京市丰台区右安门内大街 26 号

人民邮电出版社

北京市丰台区右安门内大街 26 号

人民邮电出版社

北京市丰台区右安门内大街 26 号

人民邮电出版社

北京市丰台区右安门内大街 26 号

人民邮电出版社

北京市丰台区右安门内大街 26 号

人民邮电出版社

北京市丰台区右安门内大街 26 号

人民邮电出版社



高校系列

人民邮电出版社

北京

图书在版编目 (CIP) 数据

Visual Basic语言程序设计教程 / 吴昊, 周美玲主
编. — 北京: 人民邮电出版社, 2018.4
21世纪高等学校计算机规划教材
ISBN 978-7-115-47516-9

I. ①V… II. ①吴… ②周… III. ①BASIC语言—程
序设计—高等学校—教材 IV. ①TP312.8

中国版本图书馆CIP数据核字(2018)第047654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,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。本书有配套的实践教材,既帮助学生巩固了所学的知识,又扩展了思路,能进一步加强学生自学能力的培养。

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课程的教材,并可作为计算机等级考试的辅导用书。

◆ 主 编 吴 昊 周美玲

副 主 编 熊李艳 雷莉霞

责任编辑 张 斌

责任印制 沈 蓉 彭志环

◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路 11 号

邮编 100164 电子邮件 315@ptpress.com.cn

网址 <http://www.ptpress.com.cn>

三河市潮河印业有限公司印刷

◆ 开本: 787×1092 1/16

印张: 19.5

2018 年 4 月第 1 版

字数: 498 千字

2018 年 4 月河北第 1 次印刷

定价: 52.00 元

读者服务热线: (010)81055256 印装质量热线: (010)81055316

反盗版热线: (010)81055315

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,我们在多年从事相关教学工作及与其他高校合作的基础上编写了本书和配套的《Visual Basic 语言程序设计实验教程》,目的是希望进一步推动计算机教育改革、全面提升计算机教学质量,改进计算机教学课程体系,推动精品课程教学实践与建设。

本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。与本书配套的实践教程有助于拓展知识、提高创造能力的练习题和实验,既帮助学生巩固了所学知识,又扩展了学生的思路,使学生掌握 Visual Basic 程序设计语言的基本知识和 Visual Basic 程序设计的方法。

参与编写本书的作者长期从事非计算机专业的程序设计教学和教学研究,有较深的理论研究基础和丰富的教学改革实践经验,对计算机课程教学、课程体系进行了深入的探索,对计算机课程建设进行了一定广度和深度的研究。本书由吴昊、周美玲担任主编,编写分工为:周美玲负责第1、5、7章,雷莉霞负责第2、3、11章,熊李艳负责第4、9、12章,吴昊负责第6、8、10章及本书最终的统稿。在编写大纲及书稿编写过程中,我们得到了华东交通大学信息工程学院领导的关心和支持,计算机基础教研室张恒、范萍、丁振凡、刘媛媛、甘岚、杜玲玲、张月园、李明翠等同仁为本书编写的最终完成付出了很多劳动,提供了很多帮助,在此向他们一并表示由衷的感谢。

根据“立体化”教材体系的要求,除配套教材外,本书还提供电子教案、习题答案等相关教学资源,读者可登录人邮教育社区(www.ryjiaoyu.com)下载。

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

2017年12月

目 录

第1章 引言.....1

1.1 程序与程序设计语言.....1

1.2 Visual Basic 简介.....3

1.2.1 Visual Basic 的发展.....3

1.2.2 Visual Basic 的特点.....4

1.2.3 Visual Basic 的启动和退出.....5

1.3 Visual Basic 的集成开发环境.....6

1.4 建立简单的应用程序.....12

1.4.1 设计 Visual Basic 应用程序的步骤.....12

1.4.2 建立简单的应用程序.....14

1.5 面向对象的程序设计语言.....18

1.5.1 对象和类.....18

1.5.2 对象的属性、事件和方法.....19

1.6 窗体.....22

1.6.1 窗体的基本属性.....22

1.6.2 窗体的事件.....26

1.6.3 窗体的方法.....29

1.6.4 焦点和 Tab 键序.....30

第2章 Visual Basic 语言基础.....33

2.1 语言基础.....33

2.1.1 Visual Basic 的字符集.....33

2.1.2 Visual Basic 的词汇集.....34

2.1.3 编码规则与约定.....34

2.2 数据类型.....36

2.3 常量与变量.....39

2.3.1 常量.....39

2.3.2 变量.....41

2.4 运算符与表达式.....43

2.4.1 算术运算符与算术表达式.....43

2.4.2 关系运算符与关系表达式.....45

2.4.3 逻辑运算符与逻辑表达式.....47

2.4.4 字符串运算符与字符串表达式.....48

2.4.5 日期型表达式.....49

2.4.6 运算符的优先级.....50

2.5 常用内部函数.....51

2.5.1 数学函数.....51

2.5.2 字符串函数.....52

2.5.3 转换函数.....54

2.5.4 日期和时间函数.....55

2.5.5 格式输出函数.....56

2.5.6 其他函数.....58

第3章 Visual Basic 程序初步.....59

3.1 结构化程序设计.....59

3.1.1 采用的设计方法.....59

3.1.2 基本结构.....60

3.1.3 特点.....60

3.2 顺序结构.....61

3.2.1 赋值语句.....61

3.2.2 数据的输入和输出.....62

3.2.3 常用语句.....68

3.3 基本控件.....69

3.3.1 标准控件.....70

3.3.2 命令按钮.....73

3.3.3 标签.....75

3.3.4 文本框.....76

第4章 选择结构程序设计.....80

4.1 算法介绍.....80

4.1.1 算法介绍.....80

4.1.2 算法的表示.....84

4.2 条件语句.....87

4.2.1 单分支结构语句.....	87
4.2.2 双分支语句.....	89
4.2.3 If 函数.....	92
4.2.4 多分支结构语句.....	92
4.2.5 Choose 函数.....	98
4.3 选择结构的嵌套.....	99

第 5 章 循环结构程序设计.... 104

5.1 循环语句.....	104
5.1.1 For...Next 循环语句.....	104
5.1.2 Do...Loop 循环语句.....	107
5.1.3 While...Wend 循环语句.....	110
5.2 多重循环.....	112
5.3 其他控制语句.....	114
5.3.1 Exit 语句.....	114
5.3.2 With...End With 语句.....	115
5.3.3 GoTo 语句.....	115
5.4 应用举例.....	116

第 6 章 数组..... 124

6.1 数组的概念.....	124
6.1.1 数组与数组元素.....	125
6.1.2 下标与数组的维数.....	125
6.2 数组的声明与应用.....	126
6.2.1 静态数组的声明.....	126
6.2.2 Array 函数.....	128
6.2.3 数组的应用.....	129
6.3 动态数组.....	131
6.3.1 创建动态数组.....	131
6.3.2 数组刷新语句.....	132
6.4 For Each...Next 循环语句.....	133
6.5 控件数组.....	134
6.5.1 控件数组的概念.....	134
6.5.2 控件数组的创建.....	135
6.5.3 控件数组的使用.....	136
6.6 列表框与组合框.....	136
6.6.1 列表框.....	136
6.6.2 组合框.....	140
6.7 程序举例.....	141

第 7 章 常用控件..... 147

7.1 单选按钮.....	147
7.1.1 常用属性.....	147
7.1.2 常用事件.....	148
7.2 复选框.....	151
7.2.1 常用属性.....	151
7.2.2 常用事件.....	151
7.3 框架.....	153
7.3.1 框架的用途.....	153
7.3.2 常用属性.....	153
7.3.3 常用事件.....	154
7.4 滚动条.....	154
7.4.1 常用属性.....	155
7.4.2 常用事件.....	156
7.5 图片框.....	159
7.5.1 常用属性.....	159
7.5.2 常用方法和事件.....	160
7.6 图像框.....	162
7.6.1 图像框的用途.....	162
7.6.2 常用属性.....	162
7.7 计时器.....	164
7.7.1 常用属性.....	164
7.7.2 常用事件.....	165

第 8 章 过程..... 170

8.1 过程概述.....	170
8.2 Function 过程.....	170
8.2.1 函数过程的定义.....	170
8.2.2 Function 过程的调用.....	173
8.3 Sub 过程.....	174
8.3.1 事件过程和通用过程.....	175
8.3.2 Sub 过程的定义.....	176
8.3.3 Sub 过程的调用.....	176
8.4 参数传递.....	178
8.4.1 按址传递与按值传递.....	179
8.4.2 数组参数的传递.....	183
8.4.3 可选参数与可变参数.....	184
8.4.4 对象参数.....	186
8.5 过程的嵌套与递归调用.....	187

8.5.1 过程的嵌套调用.....	187
8.5.2 过程的递归调用.....	188
8.6 模块	189
8.6.1 窗体模块.....	189
8.6.2 标准模块.....	190
8.6.3 类模块	190
8.7 过程与变量的作用域.....	191
8.7.1 过程的作用域.....	191
8.7.2 变量的作用域.....	191
8.8 变量的生存期.....	195
8.8.1 动态变量.....	195
8.8.2 静态变量.....	195
8.9 多重窗体程序设计	196
8.9.1 多窗体处理.....	196
8.9.2 Sub Main 过程.....	199
8.10 程序举例.....	200

第9章 界面设计 203

9.1 菜单设计	203
9.1.1 菜单编辑器.....	204
9.1.2 下拉式菜单.....	206
9.1.3 菜单的 Click 事件	208
9.1.4 菜单项的控制.....	209
9.1.5 弹出式菜单.....	213
9.2 工具栏和状态栏	215
9.2.1 ActiveX 控件.....	215
9.2.2 工具栏	215
9.2.3 状态栏	218
9.3 对话框	221
9.3.1 预定义对话框.....	221
9.3.2 通用对话框.....	222
9.3.3 自定义对话框.....	227
9.4 多重窗体程序设计与多文档 程序设计.....	228
9.4.1 多窗体程序设计.....	228
9.4.2 多文档程序设计.....	230
9.5 Sub Main 过程	232
9.6 鼠标与键盘事件	233
9.6.1 键盘事件.....	233

9.6.2 鼠标事件	235
------------------	-----

第10章 文件..... 238

10.1 文件的概念	238
10.1.1 数据文件的结构	238
10.1.2 文件类型	238
10.1.3 文件处理的步骤	239
10.2 顺序文件	240
10.2.1 顺序文件的打开与关闭	240
10.2.2 顺序文件的写入操作	241
10.2.3 顺序文件的读出操作	242
10.3 随机文件	245
10.3.1 随机文件的打开与关闭	245
10.3.2 随机文件的写入操作	245
10.3.3 随机文件的读出操作	246
10.4 常用的文件操作语句和函数	248
10.4.1 文件操作语句	248
10.4.2 文件操作函数	249
10.5 文件系统控件.....	249
10.5.1 驱动器列表框	250
10.5.2 目录列表框	250
10.5.3 文件列表框	251
10.5.4 文件系统控件的联动	252
10.6 应用举例	252

第11章 图形操作 256

11.1 图形操作基础.....	256
11.1.1 引例	256
11.1.2 坐标系统	257
11.1.3 绘图属性	259
11.2 图形控件	262
11.3 图形方法和事件	263
11.4 图形的层次.....	267

第12章 数据库应用 269

12.1 数据库基础.....	269
12.1.1 数据库的基本概念	269
12.1.2 关系数据库	272

12.1.3 Visual Basic 数据库应用系统.....	274
12.2 数据管理器的使用	278
12.2.1 创建数据库.....	278
12.2.2 添加数据表.....	280
12.2.3 修改数据表结构.....	282
12.2.4 用户数据的编辑.....	283
12.2.5 数据窗体设计器.....	284
12.3 数据库控件.....	286
12.3.1 数据控件.....	287

12.3.2 数据绑定控件	289
12.3.3 记录集 Recordset 对象	290
12.4 ADO 数据访问对象	293
12.4.1 ADO 控件使用基础.....	293
12.4.2 创建 ADO 控件.....	294
12.4.3 ADO 控件的常用属性、方法与事件	294
12.4.4 ADO 数据绑定控件.....	297

参考文献 304

10.1 数据库应用系统.....	274
10.2 数据管理器的使用	278
10.2.1 创建数据库.....	278
10.2.2 添加数据表.....	280
10.2.3 修改数据表结构.....	282
10.2.4 用户数据的编辑.....	283
10.2.5 数据窗体设计器.....	284
10.3 数据库控件.....	286
10.3.1 数据控件.....	287
10.3.2 数据绑定控件	289
10.3.3 记录集 Recordset 对象	290
10.4 ADO 数据访问对象	293
10.4.1 ADO 控件使用基础.....	293
10.4.2 创建 ADO 控件.....	294
10.4.3 ADO 控件的常用属性、方法与事件	294
10.4.4 ADO 数据绑定控件.....	297
参考文献	304

10.1 数据库应用系统.....	274
10.2 数据管理器的使用	278
10.2.1 创建数据库.....	278
10.2.2 添加数据表.....	280
10.2.3 修改数据表结构.....	282
10.2.4 用户数据的编辑.....	283
10.2.5 数据窗体设计器.....	284
10.3 数据库控件.....	286
10.3.1 数据控件.....	287
10.3.2 数据绑定控件	289
10.3.3 记录集 Recordset 对象	290
10.4 ADO 数据访问对象	293
10.4.1 ADO 控件使用基础.....	293
10.4.2 创建 ADO 控件.....	294
10.4.3 ADO 控件的常用属性、方法与事件	294
10.4.4 ADO 数据绑定控件.....	297
参考文献	304

第1章 引言

Visual Basic 是一种面向对象的可视化的程序设计语言。本章将向大家介绍程序与程序设计语言的概念、Visual Basic 简介与集成开发环境、建立简单的 Visual Basic 应用程序的过程、面向对象程序设计语言的基本概念以及 Visual Basic 中最主要的操作界面——窗体的操作。

1.1 程序与程序设计语言

1. 程序

程序即计算机程序，是指为了得到某种结果而可以由计算机等具有信息处理能力的装置执行的代码化指令序列或者可以被自动转换成代码化指令序列的符号化指令序列或者符号化语句序列。

同一计算机程序的源程序和目标程序视为同一作品。

2. 程序设计

程序设计过程通常包括分析、设计、编码、测试、排错等不同阶段，但通常简单地理解为程序设计是用某种程序设计语言（也称为计算机语言）来编写计算机程序的过程。

用某种程序设计语言编写的代码称为源程序，也称为源代码。源代码是不能直接被计算机执行的，要通过编译程序编译或解释程序解释成目标程序（二进制代码），再通过连接程序连接成可执行程序后才能执行。

3. 程序设计语言

程序设计语言是用于编写计算机程序的语言。语言的基础是一组记号和一组规则。根据规则由记号构成的记号串的总体就是语言。在程序设计语言中，这些记号串就是程序。

程序设计语言包含三个方面，即语法、语义和语用。语法表示程序的结构或形式，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，即表示按照各种方法所表示的各个记号的特定含义，但也不涉及使用者；语用表示程序与使用的关系。

程序设计语言种类很多，但一般来说，各种语言的基本成分不外乎以下 4 种。

① 数据成分，用以描述程序中所涉及的数据。

② 运算成分，用以描述程序中所包含的运算。

③ 控制成分,用以表达程序中的控制结构。

④ 传输成分,用以表达程序中数据的传输。

4. 程序设计语言的分类

自 20 世纪 60 年代以来,世界上公布的程序设计语言已有上千种之多,但是只有很小一部分得到了广泛的应用。从发展历程来看,程序设计语言可以分为 3 类。

(1) 机器语言

机器语言是计算机诞生和发展初期使用的语言,由二进制 0、1 代码指令构成,不同的 CPU 具有不同的指令系统。机器语言是从属于硬件设备的,不同的计算机设备有不同的机器语言。直到如今,机器语言仍然是计算机硬件所能“理解”的唯一语言。机器语言编写的程序是可以直接运行的。

在计算机发展初期,人们就是直接使用机器语言来编写程序的,那是一项相当复杂和烦琐的工作。

机器语言程序难编写、难修改、难维护,需要用户直接对存储空间进行分配,编程效率极低。这种语言已经被渐渐淘汰了。

(2) 汇编语言

汇编语言开始于 20 世纪 50 年代初期。为了克服机器语言的缺点,人们将机器指令中表示操作的代码用英文助记符来表示,如用 ADD 表示加法、MOV 表示数据传递等。

汇编语言指令是机器指令的符号化,与机器指令存在着直接的对应关系,所以汇编语言同样存在着难学难用、容易出错、维护困难等缺点。但是汇编语言也有自己的优点:可直接访问系统接口,汇编程序翻译成机器语言程序的效率高。从软件工程角度来看,只有在高级语言不能满足设计要求,或不具备支持某种特定功能的技术性能(如特殊的输入/输出)时,汇编语言才被使用。

采用汇编语言编写的程序(源程序),必须经过汇编程序(一种语言处理程序)汇编成计算机能够直接识别的机器语言后,才能被计算机执行。

机器语言和汇编语言都属于面向机器的低级语言。

(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:它接近于数学语言或自然语言,同时又不依赖于计算机硬件,编写出的程序能在所有计算机上通用。经过努力,1954 年,第一个完全脱离机器硬件的高级语言——FORTRAN 语言问世了。经过 60 多年的发展,有几百种高级语言出现,具有重要意义的有几十种,影响较大、使用较普遍的有十几种。

高级语言是面向用户的,基本上独立于计算机种类和结构的语言。其最大的优点是:形式上接近于算术语言和自然语言,概念上接近于人们通常使用的概念。高级语言的一个命令可以代替几条、几十条甚至几百条汇编语言的指令。因此,高级语言易学易用,通用性强,应用广泛。

高级语言与计算机的硬件结构及指令系统无关,它有更强的表达能力,可方便地表示数据的运算和程序的控制结构,能更好地描述各种算法,而且容易学习掌握。

用高级语言编写的程序称为高级语言源程序,也不能在计算机中直接执行,必须经过编译或解释程序翻译成机器语言后才能执行。虽然程序翻译占去了一些计算机时间,在一定程度上影响了计算机的使用效率,但是实践证明,高级语言是有效地使用计算机与计算机执行效率之间的一个很好的折中手段。

高级语言并不是特指的某一种具体的语言,而是包括很多种编程语言,如流行的 Java、C、C++、

C#、Visual Basic、Python、Lisp、Prolog、FoxPro 等，这些语言的语法、命令格式都各不相同。

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、面向对象程序设计语言的过程。

高级语言编写程序的编写效率虽然比汇编语言高，但随着计算机硬件技术的日益发展，人们对大型、复杂的软件需求量剧增，而同时因缺乏科学规范、系统规划与测试，程序含有过多错误而无法使用，甚至带来巨大损失。20 世纪 60 年代中后期，“软件危机”的爆发使人们认识到大型程序的编制不同于小程序。解决“软件危机”一方面需要对程序设计方法、程序的正确性和软件的可靠性等问题进行深入研究，另一方面需要对软件的编制、测试、维护和管理方法进行深入研究。结构化程序设计是一种程序设计的原则和方法。它讨论了如何避免使用 GOTO 语句；如何将大规模、复杂的流程图转换成一种标准的形式，使得它们能够用几种标准的控制结构（顺序、分支和循环）通过重复和嵌套来表示。1970 年，第一个结构化程序设计语言 Pascal 语言问世。

到了 20 世纪 70 年代末期，随着计算机应用领域的不断扩大，人们对软件技术的要求越来越高，结构化程序设计语言和结构化程序设计方法又无法满足用户需求的变化，其缺点也日益显露出来，例如代码的可重用性差、可维护性差、稳定性差、难以实现。

结构化程序设计方法与语言是面向过程的，存在较多的缺点，同时程序的执行是流水线式的，在一个模块被执行完成前，不能干别的事，也无法动态地改变程序的执行方向。这和人们日常认识、处理事物的方式不一致。人们认为客观世界是由各种各样的对象（或称实体、事物）组成的，每个对象都有自己的内部状态和运动规律，不同对象间的相互联系和相互作用构成了各种不同的系统，进而构成整个客观世界。为了使计算机更易于模拟现实世界，1967 年挪威计算中心开发出了 Simula 67 语言，它提供了比子程序更高一级的抽象和封装，引入了数据抽象和类的概念，被认为是第一个面向对象程序设计语言。它对后来出现的面向对象语言（如 Visual Basic、C++、Java、C#等）产生了深远的影响。

1.2 Visual Basic 简介

1.2.1 Visual Basic 的发展

Basic 语言诞生于 1964 年，全称为“Beginners All-Purpose Symbolic Instruction Code”，含义为“初学者通用符号指令代码”。它简单易学，一直被大多数初学者作为首选入门的程序设计语言。随着计算机技术的发展，各种 Basic 语言版本应运而生。1976 年前后开发出 DOS 环境下的 GW-Basic，20 世纪 80 年代中期又出现了多种结构化 Basic 语言，如 True Basic、Quick Basic、Turbo Basic、QBasic 等。

1988 年，美国微软（Microsoft）公司推出了 Windows 操作系统，以其友好的图形用户界面（GUI）、简单易学的操作方式和卓越的性能，赢得了广大计算机用户的喜爱，因此，开发在 Windows 环境下的应用程序成为 20 世纪 90 年代软件开发的主流。起初人们在开发 Windows 应用程序时遇到了很大困难，因为要编写 Windows 环境下运行的程序，必须建立相应的窗口、菜单、对话框等各种“控件”，程序的编制很复杂。

1991 年 4 月，Microsoft 公司推出的 Visual Basic 1.0 使这种情况有了根本的改观。时任 Microsoft 公司董事长的比尔·盖茨说，Visual Basic 是“用 Basic 语言开发 Windows 应用程序最强有力的工具”。

Visual 的含义是“可视化的”，指的是开发图形用户界面的方法，它与其他编程软件不同的是不需要编写大量代码去描述界面元素的外观和位置，只要把预先建立好的对象拖放到屏幕上相应的位置即可。Visual Basic 采用的“可视化编程”是面向对象编程技术的简化版，它引入了面向对象和事件驱动的程序设计的新机制，把过程化和结构化编程结合在一起，其解决问题的方式更符合人们的思维习惯，为开发 Windows 应用程序提供了强有力的开发环境和工具。

随着 Windows 操作平台的不断成熟，Visual Basic 的版本也不断升级。自 Visual Basic 1.0 之后，Microsoft 公司又相继推出 Visual Basic 2.0、Visual Basic 3.0、Visual Basic 4.0，这些版本主要应用于 Windows 3.X 环境中 16 位应用程序的开发。1997 年，Microsoft 公司发布了 Visual Basic 5.0，它是一个 32 位应用程序开发工具，可以运行在 Windows 9.X 或 Windows NT 环境中。

随着版本的提高，Visual Basic 的功能也越来越强。自从 5.0 版之后，Visual Basic 推出了中文版，与前几个版本相比，其功能有了很大提升。

1998 年，Microsoft 公司推出了 Visual Basic 6.0。

Visual Basic 6.0 有学习版、专业版和企业版三种版本，以满足不同的开发需要。学习版适用于普通学习者及大多数使用 Visual Basic 开发一般 Windows 应用程序的人员；专业版适用于计算机专业开发人员，包括学习版的全部内容功能以及 Internet 控件开发工具之类的高级特性；企业版除包含专业版全部的内容外，还有自动化构件管理等工具，使得专业编程人员能够开发功能强大的分布式应用程序。

2002 年 2 月，Microsoft 公司推出了 Visual Basic .NET 2002。

2010 年 4 月，Microsoft 公司推出了 Visual Studio 2010。

2014 年 11 月，Microsoft 公司推出了 Visual Studio 2015。

本书使用的是 Visual Basic 6.0 中文企业版。在后面的讲解中将 Visual Basic 简称为 VB。

1.2.2 Visual Basic 的特点

Visual Basic 是在原有的 Basic 语言的基础上发展而来的，它具有 Basic 语言简单易用的特点，同时增加了面向对象和可视化程序设计语言的特点。Visual Basic 的特点主要有以下几点。

1. 面向对象的可视化编程

VB 采用面向对象程序设计方法（OOP），把程序和数据“封装”起来作为一个对象。所谓“对象”，就是指可操作的实体，如窗体、窗体中的标签、文本框、命令按钮等，面向对象编程就是指程序员可根据界面设计要求直接在界面上设计出窗口、菜单、按钮等类型对象并为每个对象设置相应的属性。VB 系统将自动产生界面设计代码，编程人员只需编写实现程序功能的那部分程序代码，从而大大提高程序设计的效率。

2. 事件驱动的编程机制

VB 通过事件来执行对象的操作，通常由用户操作引发某个事件来驱动完成某种功能。例如，命令按钮是一个对象，当用户单击该按钮时，将产生（或触发）一个“单击”（Click）事件，而在发生该事件时，系统将自动执行一段相应的程序（称为“事件过程”），以实现指定的操作或达到运算、处理的目的。

在 VB 中，编程人员只需针对这些事件编写相应的处理代码（即事件过程），这样的代码一般较

短,所以程序既易于编写,又易于维护。

3. 结构化的程序设计

VB 是在结构化的程序设计语言 Basic 语言的基础上发展起来的,加上面向对象的设计方法,因此是更出色的结构化程序设计语言。在事件过程的编写中,要遵循结构化程序设计的原则。

4. 友好的集成开发环境

VB 提供了易学易用的应用程序集成开发环境。在该集成开发环境中,编程人员可以设计用户界面、编写代码和调试程序,甚至可以把应用程序编译生成可执行文件,脱离 VB 环境直接在 Windows 环境下运行。

5. 具有强大的功能

VB 可以对多种数据库系统进行数据访问,支持对象的链接与嵌入(OLE)、动态数据交换(DDE)、动态链接库(DLL)及 Active 等技术,它能够充分利用 Windows 资源,开发出集文字、声音、图像、动画、Web 等对象为一体的应用程序。

1.2.3 Visual Basic 的启动和退出

1. VB 6.0 的启动

VB 6.0 的启动和其他 Windows 应用程序类似,常用的方法如下。

① 通过依次单击“开始”→“所有程序”→“Microsoft Visual Basic 6.0 中文版”目录→“Microsoft Visual Basic 6.0 中文版”启动。

② 双击桌面上的 Visual Basic 6.0 应用程序图标也可以启动 VB 6.0。

启动 VB 6.0 之后,将出现图 1.1 所示的“新建工程”对话框。



图 1.1 “新建工程”对话框

在图 1.1 所示的对话框中,包含以下 3 个选项卡。

“新建”:用来建立新的工程。其列表框中有多种选择,可供用户建立不同类型的程序。

“现存”:选择和打开现有的工程。其操作和 Windows 中选择并打开文件的操作类似。

“最新”:该选项卡的列表框中会按时间顺序列出最近使用过的工程。

图 1.1 所示的对话框显示出可以在 VB 6.0 中使用的工程类型，主要有以下几种。

- ① 标准 EXE：建立标准 Windows 下的可执行文件，它是“新建”选项卡中最基本的类型，也是默认类型。
- ② 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。这种程序也可以包装成可执行 EXE 文件。
- ③ ActiveX DLL：与 ActiveX EXE 程序一样，只是不能包装成 EXE 文件，只能包装成动态链接库。
- ④ ActiveX 控件：用于开发用户自定义的 ActiveX 控件，只能在专业版和企业版中建立。

2. VB 6.0 的退出

如果要退出 VB，可以单击 VB 主窗口右上角的关闭按钮，也可以选择“文件”菜单下面的“退出”命令。退出时，VB 会自动判断用户是否修改了工程的内容。若修改了，会询问用户是否保存文件或直接退出。

1.3 Visual Basic 的集成开发环境

启动 VB，选择默认的新建“标准 EXE”选项后，可以看到图 1.2 所示的界面，这个界面就是 VB 的集成开发环境。

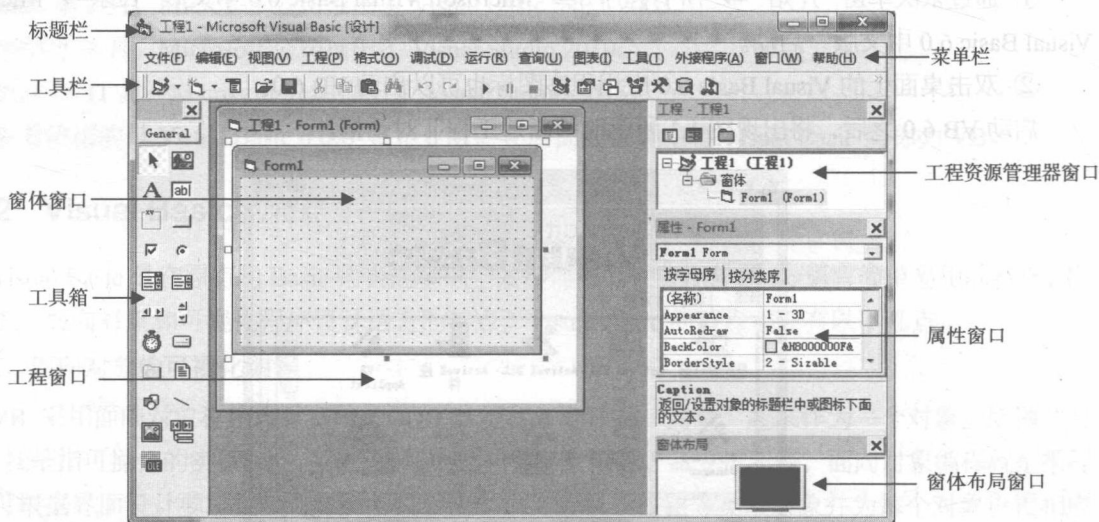


图 1.2 VB 集成开发环境

VB 集成开发环境由标题栏、菜单栏、工具栏、工具箱、窗体窗口、工程窗口、工程资源管理器窗口、属性窗口、窗体布局窗口和隐藏的代码窗口等组成。

1. 标题栏

VB 标题栏和其他 Windows 应用程序一样，由控制菜单框、标题、最小化、最大化、关闭按钮组成。不同的是，在 VB 的标题栏中同时会显示 VB 的工作模式，如“设计”。VB 新建的第一个工程默认为“工程 1”，图 1.2 所示的界面中标题栏显示内容为“工程 1-Microsoft Visual Basic[设计]”。

VB 有三种工作模式，分别如下。

① 设计模式：在这种模式下，用户可进行界面设计和代码的编制，以完成应用程序的开发。

② 运行模式：当用户启动运行程序时，进入运行模式。此模式下不可对代码进行编辑，也不能设计界面。

③ 中断模式：主要用于调试程序，当程序出错运行中断单击“调试”按钮时，进入中断模式。此模式下可以对代码进行编辑，按 F5 键或单击“继续”按钮，程序继续运行；单击“结束”按钮程序停止运行。此模式下，会弹出“立即”窗口，在窗口内可输入简短的命令，并立即执行。

2. 菜单栏

VB 的菜单栏包括 13 个下拉菜单，其中包含了程序设计过程中所需的命令。

① 文件：用于新建、打开、添加、移除、保存、显示最近的工程以及生成可执行文件，保存窗体，打印和退出 VB 系统。

② 编辑：用于对源代码的编辑、查找，显示一些常用的信息等。

③ 视图：用于集成开发环境中各窗口的关闭和打开。

④ 工程：用于对工程中各窗体、模块、控件等的添加以及显示工程的属性。

⑤ 格式：用于窗体、控件的格式设计，如对齐、间距等。

⑥ 调试：用于程序的调试、查错。用户可对代码设置断点，逐步排查错误。

⑦ 运行：用于程序的启动、中断和结束等。

⑧ 查询：用于设计 SQL 查询。

⑨ 图表：用于建立数据库中的表，在设计数据库应用程序时编辑数据库的命令。

⑩ 工具：用于集成开发环境的设置及原有工具的扩展。

⑪ 外接程序：用于为工程增加或删除外接程序。

⑫ 窗口：用于屏幕窗口的层叠、平铺等布局以及列出所有已打开的文档。

⑬ 帮助：安装 MSDN 后，帮助用户系统地学习和掌握 VB 的使用方法及程序设计方法。

3. 工具栏

工具栏是常用命令的快捷方式，使用工具栏可以在编程环境下提供对常用命令的快速访问。单击工具栏上的按钮，系统会执行该按钮所代表的操作。VB 6.0 通常提供编辑、标准、窗体编辑器、调试和自定义工具栏。利用自定义工具栏，用户可定制自己所需的工具栏。通常系统默认的是图 1.3 所示的标准工具栏。

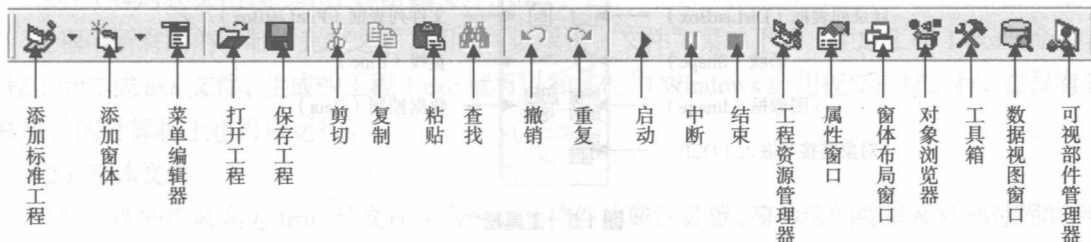


图 1.3 VB 标准工具栏

要显示或隐藏工具栏，可以选择“视图”菜单的“工具栏”命令或在主窗口的工具栏上单击鼠标右键，从弹出的快捷菜单上显示的工具栏名称进行选择。

4. 窗体窗口

窗体窗口也称为“窗体设计器”，位于主窗口的中间部分。窗体窗口如图 1.4 所示。

窗体是 VB 应用程序的主要构成部分，是应用程序的载体。用户可以在窗体中添加控件，通过窗体上的控件交互可控制应用程序的运行，得到想要的结果。窗体窗口具有普通窗口的一切功能，如可被移动、改变大小、缩小成图标及单独关闭等。每个窗体窗口必须有唯一的窗体名，新建窗体默认窗体名为 Form1、Form2……。

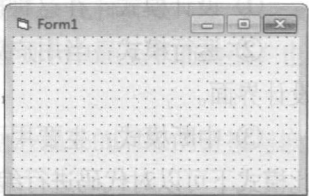


图 1.4 VB 窗体窗口

在设计模式下，窗体是可见的，窗体上的网格点用来帮助用户对添加的控件进行准确定位，它的间距也可以通过“工具”菜单下的“选项”命令进行设置。运行模式下，这些网格点是不显示的。

一个 VB 应用程序至少要有一个窗体窗口。实际应用中，一个 VB 应用程序通常有多个窗体，运行时哪个窗体出现，可通过窗体的属性来设定。

除了一般窗体外，VB 还有一种多文档窗体 MDI (Multiple Document Interface)，它可以包含子窗体，每个子窗体都是独立的，但每个子窗体只能在父窗体内进行移动。

5. 工具箱

工具箱位于主窗口的左侧，用户也可以拖动工具箱的顶部到任意位置。工具箱中的每一个按钮就是一个控件制作工具，对应着一个控件。工具箱中的工具分为两类：一类是标准控件或内部控件，另一类是 ActiveX 控件。ActiveX 控件需要另外加载，默认情况下工具箱上显示的是标准控件。工具箱上有 20 个标准控件，其含义如图 1.5 所示。

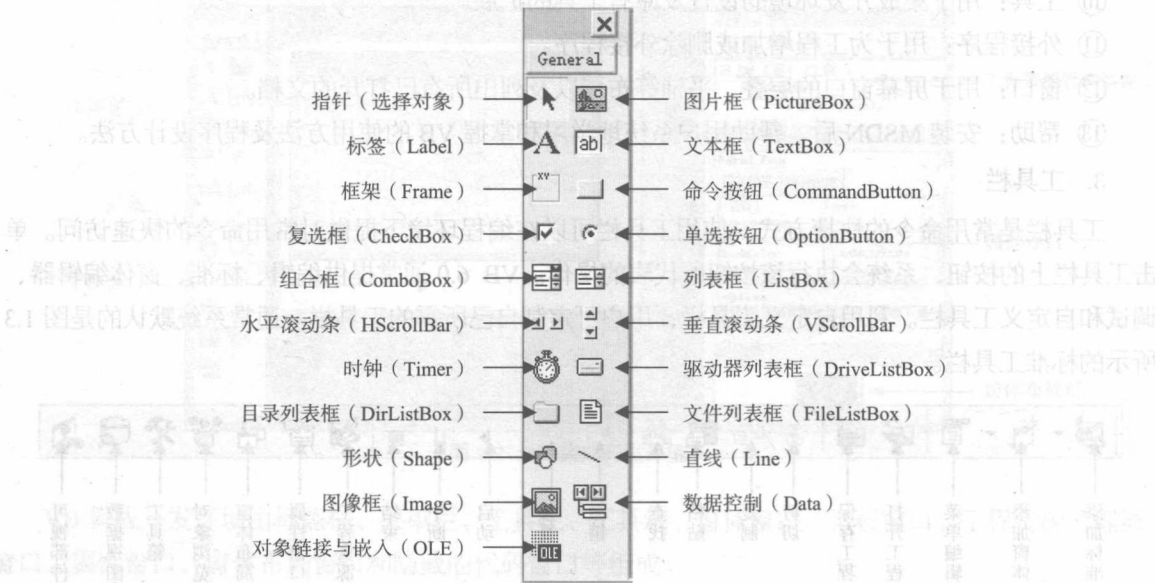


图 1.5 工具箱

指针不是标准控件，只是是否选择对象的标志。若没有控件被选中，指针是凹下去的，如图 1.5 所示，若有控件被选中，指针和其他控件一样是平的。

工具箱上的控件是 VB 应用程序进行界面设计的工具。若把窗体窗口比作盖房子的地基，则工具

箱上的控件就像在地基上建房的砖块、木头。

工具箱上的控件可以根据需要添加和删除,其操作如下。

- ① 在工具箱空白处单击鼠标右键,在弹出的快捷菜单中选择“部件”命令。
- ② 在随后弹出的“部件”对话框中有三个选项卡,分别是“控件”“设计器”和“可插入对象”。选择需要的类别,在其列表框中选中要添加的控件,单击“确定”按钮即可将其添加到工具箱上。删除的操作过程是类似的,只要将添加的控件在列表框中取消选定就可以了。

注意: 只能对添加的控件进行删除,对图 1.5 所示的标准控件不能删除。

6. 工程资源管理器

工程资源管理器窗口用来对工程和工程中包含的对象进行管理,其界面如图 1.6 所示。

一个 VB 应用程序通常对应一个工程,工程中可以添加窗体和模块,图 1.6 所示的界面中工程 1 中包含 2 个窗体 (Form1 和 Form2)、2 个标准模块 (Module1 和 Module2) 和 1 个类模块 (Class1)。

工程和其包含的对象都对应着不同类型的文件,一个 VB 工程可以包含 7 种类型的文件。

(1) 工程文件

工程文件的扩展名为.vbp。当一个应用程序包括两个及以上的工程时,就构成了一个工程组,工程组文件的扩展名为.vbg。用户可以通过“文件”菜单下的“添加工程”命令来添加工程。

工程文件保存的是它所包含的其他对象的名称和位置等信息,保存文件

时通常要先保存其他类型的文件,最后保存工程文件。例如要保存图 1.6 所示的工程 1,应在工程资源管理器中分别选中 Form1、Form2、Module1、

Module2 和 Class1,“文件”菜单下将分别出现“保存 Form1”“保存 Form2”

“保存 Module1”之类的命令,选择对应的命令保存上述 5 个文件,最后保存工程文件。工程文件中将记录工程 1 中包含的上述 5 个文件的名称和位置信息,

下次打开时只要双击打开工程文件,系统会根据工程文件中记录的其他 5 个文件的名称和位置信息将其打开,通常是将其他文件和工程文件保存在

同一文件夹下。当然,顺利打开其他 5 个文件的前提是依据工程文件中记录的名称和位置信息能找到相应文件。若在最后保存工程后,人为地在 Windows 环境中将 5 个文件在硬盘中进行删除或移动,

再通过工程来打开是会出错的。若确实需要删除工程中其中的某个文件如 Module2,正确的操作方法是在工程资源管理器窗口选中 Module2,右键单击,在弹出的快捷菜单中选择“移除 Module2”命令,

最后再次保存工程文件以更新其中包含的对象的名称和位置等信息。

保存工程时还会出现一个工程附属文件.vbw。

工程中所有的内容制作完成之后,用户可以单击“文件”菜单下的“生成工程 1.exe”命令,将工程 1 制作成.exe 文件,生成的工程 1.exe 就可以和其他的 Windows 应用程序一样运行,在没有安装 VB 软件的计算机上也可以运行。

(2) 窗体文件

窗体文件的扩展名为.frm,该文件包含窗体及控件的属性设置、窗体级的变量及外部过程的声明、事件过程及用户自定义过程。每一个窗体都有一个窗体文件。一个应用程序至少应该包含 1 个窗体,最多可达 255 个。窗体也称为窗体模块。

用户可以在工程中添加新建窗体,也可以将已有的窗体添加进来。添加窗体的操作方法是单击“工程”菜单的“添加窗体”命令,将弹出图 1.7 所示的“添加窗体”对话框。对话框有 2 个选项卡,



图 1.6 工程资源管理器