



中国高等职业技术教育研究会推荐
高职高专系列规划教材

Visual Basic 程序设计

■主编 周耿烈 赵双萍 张明新



西安电子科技大学出版社
<http://www.xduph.com>

中国高等职业技术教育研究会推荐

面向 **21** 世纪高职高专系列规划教材

Visual Basic 程序设计

主 编 周耿烈 赵双萍 张明新

西安电子科技大学出版社

2009

内 容 简 介

本书共分 13 章, 主要内容包括 Visual Basic 6.0 程序设计概述, Visual Basic 6.0 基本概念和基本操作, Visual Basic 语法基础, 窗体和基本控件, 基本控制结构, 数组与记录, 过程, 常用内部控件, 程序调试、错误处理与发布, 多功能用户界面设计, 图形处理, 文件处理和数据库应用基础。

本书重点突出, 在结构体系上循序渐进, 深入浅出, 简明精练, 详略得当, 图文并茂; 各章节前后呼应, 针对性强, 可读性好。每章后都附有本章小结和习题, 可以帮助读者巩固所学知识, 提高学习效率和兴趣。

本书可作为高职高专学生的教材, 亦可作为计算机等级考试培训教材或参考书。

★本书配有电子教案, 有需要者可从出版社网站下载, 免费提供。

图书在版编目(CIP)数据

Visual Basic 程序设计 / 周耿烈, 赵双萍, 张明新主编. —西安: 西安电子科技大学出版社, 2009. 2

面向 21 世纪高职高专系列规划教材

ISBN 978 - 7 - 5606 - 2187 - 6

I. V… II. ①周… ②赵… ③张… III. Basic 语言—程序设计—高等学校: 技术学校—教材
IV. TP 312

中国版本图书馆 CIP 数据核字(2009)第 003068 号

策 划 杨丕勇

责任编辑 张晓燕

出版发行 西安电子科技大学出版社(西安市太白南路 2 号)

电 话 (029)88242885 88201467 邮 编 710071

网 址 www.xduph.com 电子邮箱 xdupfxb001@163.com

经 销 新华书店

印刷单位 西安文化彩印厂

版 次 2009 年 2 月第 1 版 2009 年 2 月第 1 次印刷

开 本 787 毫米×1092 毫米 1/16 印 张 22.375

字 数 532 千字

印 数 1~4000 册

定 价 32.00 元

ISBN 978 - 7 - 5606 - 2187 - 6/TP · 1116

XDUP 2479001-1

*** 如有印装问题可调换 ***

本社图书封面为激光防伪覆膜, 谨防盗版。

前 言

计算机技术的飞速发展,促进了计算机基础教育的发展。根据我国当前教学改革和建设的需要,教育部提出了计算机文化基础、计算机技术基础和计算机应用基础三个层次的教学体系。计算机程序设计语言是高职院校各专业的一门基础课程,属于计算机技术基础教育,是当代大学生必须掌握的知识。

随着图形用户界面的普遍使用,面向过程的程序设计语言已不能适应社会的需要,当代大学生应具备使用当今流行的系统平台和开发工具开发应用程序的初步能力。微软公司推出的 Visual Basic 是一种面向对象和采用事件驱动机制的结构化、可视化高级程序设计语言。它是在 Basic 语言的基础上发展起来的,是当前广泛使用的计算机高级程序设计语言。

本书充分考虑到高等职业教育的培养目标、教学现状以及长远的发展方向,结合作者多年从事 Visual Basic 的教学经验编写而成。本书内容丰富,深入浅出,循序渐进,力求具有可读性、实用性和先进性。书中每章开始列出本章要点,在章末有本章小结与之呼应,有利于学生掌握完整的知识体系。书中配有大量例题,使读者能迅速掌握有关概念及一些编程技巧,这些例题都经过仔细的调试。每章后还配有大量的习题,以巩固所学的知识。

本书共 13 章,简单介绍如下:

第 1 章介绍了程序设计语言的发展,程序与算法的概念,Visual Basic 的发展概况、特点及 Visual Basic 6.0 的集成开发环境,创建 Visual Basic 应用程序的步骤以及帮助功能的使用。

第 2 章介绍了 Visual Basic 面向对象程序设计的有关概念、工程的概念及开发环境的设置。

第 3 章介绍了 Visual Basic 语法基础,包括常量、变量的基本概念,运算符、表达式和内部函数的使用。

第 4 章介绍了窗体、命令按钮、标签、文本框控件等基本控件的常用属性、事件和方法及其使用,另外本章对鼠标和键盘事件、焦点事件、Tab 键顺序等也做了介绍。

第 5 章介绍了 Visual Basic 6.0 语句控制结构,包括顺序结构、选择结构、循环结构的常用格式和控制语句流程的方法。

第 6 章介绍了数组的基本概念、一维数组和二维数组的定义与引用方法、控件数组的使用等。

第 7 章介绍了 Sub 过程和 Function 函数、参数的传递方法、过程的递归调用和变量及过程的作用域。

第 8 章介绍了单选按钮、复选框、框架、列表框、组合框、图片框、图像框、计时器以及滚动条等常用内部控件的使用。

第9章介绍了 Visual Basic 程序调试、错误处理和应用程序发布的基本操作。

第10章介绍了多功能用户界面设计,包括对话框、菜单、工具栏、状态栏、多重窗体和 MDI 窗体等的设计和使用。通过学习本章,读者将学会如何设计更加多姿多彩的用户界面。

第11章介绍了图形的基本概念、坐标系统、绘图属性、图形控件、绘图方法。

第12章介绍了文件的基本概念,文件的分类及各种文件的打开、关闭、读写方法,常用文件操作语句、函数及文件系统控件的使用方法。

第13章介绍了数据库的基础知识,在 Visual Basic 中创建、访问数据库和数据的添加、删除、查询方法,重点介绍了 Data 控件及 ADO 控件的使用方法。

本书以应用为中心,以初学者为对象,以提高程序设计能力为宗旨,为读者使用 Visual Basic 开发 Windows 平台下的应用程序提供捷径。

本书配有《Visual Basic 程序设计实验与习题指导》(周耿烈、赵双萍、张明新主编,西安电子科技大学出版社出版,2009年),供自学、自测用。

本书由周耿烈、赵双萍、张明新主编,周秀媛、王娟、张睿敏参加编写。第1~4章及附录由周耿烈编写;第5、9章由王娟编写;第6、7、13章由赵双萍编写;第8、10章由周秀媛编写;第11、12章由张睿敏编写,全书由周耿烈统稿。

由于时间仓促,作者水平有限,书中难免存在疏漏和不足,敬请读者不吝指正。

编 者

2008年12月

目 录

第 1 章 Visual Basic 6.0 程序设计概述 .. 1	2.2 属性、方法和事件概述 21
1.1 程序设计概述 1	2.3 事件驱动 23
1.1.1 程序设计语言 1	2.4 工程的概念及操作 25
1.1.2 程序与算法 2	2.4.1 工程的组成 25
1.2 Visual Basic 的发展和功能特点 5	2.4.2 创建和打开工程 26
1.2.1 Visual Basic 的发展 5	2.4.3 保存工程 26
1.2.2 Visual Basic 的功能特点 6	2.5 工程组的概念及操作 27
1.3 Visual Basic 6.0 的集成开发环境 7	2.5.1 添加工程 27
1.3.1 Visual Basic 6.0 的集成开发 环境概述 7	2.5.2 移除工程 27
1.3.2 Visual Basic 6.0 的主窗口 8	2.6 环境设置 28
1.3.3 工具箱 9	本章小结 30
1.3.4 工程资源管理器 9	习题 2 31
1.3.5 属性窗口 10	第 3 章 Visual Basic 语法基础 32
1.3.6 窗体设计窗口 10	3.1 数据类型 32
1.3.7 窗体布局窗口 10	3.1.1 标准数据类型 32
1.3.8 对象浏览器窗口 11	3.1.2 自定义数据类型 35
1.3.9 代码编辑窗口 11	3.2 常量和变量 35
1.4 建立简单的 Visual Basic 应用程序 12	3.2.1 常量 35
1.4.1 建立一个简单的 Visual Basic 应用程序 12	3.2.2 变量 37
1.4.2 建立 Visual Basic 应用程序 的一般步骤 14	3.3 运算符和表达式 40
1.5 Visual Basic 6.0 帮助系统简介 15	3.3.1 算术运算符和算术表达式 40
1.5.1 使用 MSDN Library 在线帮助 15	3.3.2 字符串运算符和字符串表达式 42
1.5.2 上下文帮助 16	3.3.3 关系运算符和关系表达式 42
1.5.3 运行“帮助”中的代码示例 17	3.3.4 逻辑运算符和逻辑表达式 43
本章小结 17	3.3.5 运算符的优先级 44
习题 1 17	3.4 常用的内部函数 44
第 2 章 Visual Basic 6.0 基本概念 和基本操作 20	3.4.1 数学函数 44
2.1 类与对象的概念 20	3.4.2 字符串函数 46
	3.4.3 数据类型转换函数 47
	3.4.4 日期和时间函数 48
	3.4.5 测试函数 49
	3.4.6 其他函数 49

3.5 Visaul Basic 语言字符集及编码规则	52	5.5.2 暂停语句(Stop 语句)	128
3.5.1 Visaul Basic 的字符集	52	5.5.3 结束语句(End 语句)	128
3.5.2 编码规则	52	5.5.4 With 语句	129
3.5.3 命令格式中的符号约定	53	5.6 常用算法举例	129
本章小结	53	5.6.1 累加、累乘	129
习题 3	55	5.6.2 求素数	131
第 4 章 窗体和基本控件	60	5.6.3 最大、最小值问题	131
4.1 窗体	60	5.6.4 穷举法	132
4.1.1 窗体的建立与保存	60	5.6.5 递推法	132
4.1.2 窗体的基本属性	61	本章小结	133
4.1.3 窗体的主要事件	66	习题 5	134
4.1.4 窗体的主要方法	69	第 6 章 数组	140
4.1.5 控件的添加与操作	72	6.1 数组的基本概念	140
4.2 鼠标事件	75	6.2 静态数组	140
4.3 键盘事件	80	6.2.1 一维数组	140
4.4 焦点与 Tab 顺序	83	6.2.2 二维数组	142
4.5 基本控件	85	6.3 动态数组	145
4.5.1 控件分类及命名	85	6.4 控件数组	146
4.5.2 命令按钮	86	6.5 数组的特殊操作	148
4.5.3 标签控件	89	6.5.1 使用 Erase 语句对数组重新 初始化	148
4.5.4 文本框控件	91	6.5.2 使用 Array 函数为数组赋值	149
本章小结	96	6.5.3 For Each...Next 语句引用数组	150
习题 4	97	6.5.4 数组的复制	151
第 5 章 基本控制结构	103	6.6 数组的应用	151
5.1 结构化程序设计概述	103	6.6.1 排序算法	151
5.2 顺序结构	104	6.6.2 程序举例	154
5.2.1 赋值语句	104	6.7 用户自定义数据类型	156
5.2.2 数据的输入	105	本章小结	158
5.2.3 数据的输出	107	习题 6	159
5.3 分支结构	111	第 7 章 过程	165
5.3.1 If 语句	111	7.1 Sub 过程	165
5.3.2 Select Case 语句	116	7.1.1 Sub 过程的定义	166
5.4 循环结构	119	7.1.2 Sub 过程的调用	168
5.4.1 For...Next 循环结构	119	7.2 Function 过程	169
5.4.2 While...Wend 循环结构	121	7.2.1 Function 过程的定义	169
5.4.3 Do...Loop 循环结构	122	7.2.2 Sub 过程与 Function 过程的区别	171
5.4.4 循环结构的嵌套	125	7.3 参数传递	171
5.5 辅助控制语句	128	7.3.1 形参与实参	172
5.5.1 注释语句	128		

7.3.2 传址与传值	172	9.3 错误捕获与处理	231
7.4 过程的嵌套和递归调用	175	9.3.1 错误捕获	231
7.4.1 过程的嵌套调用	175	9.3.2 错误处理	232
7.4.2 过程的递归调用	175	9.4 应用程序的发布	233
7.5 其他形式的参数	177	本章小结	241
7.5.1 数组参数	177	习题 9	242
7.5.2 可选参数与可变参数	179	第 10 章 多功能用户界面设计	243
7.5.3 对象参数	180	10.1 概述	243
7.6 变量和作用域	180	10.2 通用对话框	244
7.6.1 Visual Basic 工程的组成	180	10.2.1 文件对话框	245
7.6.2 Sub Main 过程	181	10.2.2 “颜色”对话框	247
7.6.3 变量的作用域	181	10.2.3 “字体”对话框	248
7.6.4 过程的作用域	183	10.2.4 “打印”对话框	250
7.7 应用举例	184	10.2.5 “帮助”对话框	251
本章小结	187	10.3 自定义对话框	252
习题 7	188	10.4 菜单设计	252
第 8 章 常用内部控件	194	10.4.1 菜单编辑器	253
8.1 选择类控件与框架	194	10.4.2 菜单的设计	255
8.1.1 单选按钮	194	10.4.3 菜单的控制	257
8.1.2 复选框	197	10.4.4 建立弹出式菜单	260
8.1.3 框架	199	10.5 工具栏和状态栏设计	261
8.2 列表类控件	201	10.5.1 工具栏	262
8.2.1 列表框	201	10.5.2 状态栏	265
8.2.2 组合框	207	10.6 多重窗体与多文档窗体	266
8.3 图像显示控件	210	10.6.1 多重窗体	266
8.3.1 图片框	210	10.6.2 多文档窗体	268
8.3.2 图像框	212	本章小结	271
8.4 其他控件	213	习题 10	272
8.4.1 定时器	213	第 11 章 图形处理	276
8.4.2 滚动条	217	11.1 图形操作基础	276
本章小结	220	11.2 绘图属性	278
习题 8	221	11.2.1 当前坐标	278
第 9 章 程序调试、错误处理与发布	225	11.2.2 线宽与线型	279
9.1 程序中的错误类型	225	11.2.3 颜色属性	279
9.2 程序的调试	226	11.3 图形控件	281
9.2.1 程序的三种工作模式	226	11.3.1 Shape 控件	281
9.2.2 调试工具栏	227	11.3.2 Line 控件	283
9.2.3 跟踪调试	228	11.4 绘图方法	283
9.2.4 调试窗口	229	11.4.1 PSet 方法	283

11.4.2 Line 方法	284	13.1.1 概述	318
11.4.3 Circle 方法	286	13.1.2 关系数据库及其特点	319
11.4.4 Cls 方法	289	13.2 数据库管理器 VisData	321
11.4.5 PaintPicture 方法	290	13.3 数据控件	324
本章小结	291	13.3.1 Data 数据控件	324
习题 11	292	13.3.2 数据控件的常用方法	328
第 12 章 文件处理	294	13.3.3 数据控件的常用事件	329
12.1 文件的基本概念	294	13.3.4 数据控件的应用	329
12.2 文件的打开与关闭	295	13.4 结构化查询语言 SQL	331
12.3 文件的读写操作	297	13.5 ADO 数据控件	333
12.3.1 顺序文件的读写操作	297	13.5.1 ADO 对象模型	333
12.3.2 随机文件的读写操作	300	13.5.2 使用 ADO 数据控件	333
12.3.3 二进制文件的读写操作	302	13.5.3 使用绑定控件	336
12.4 常用文件操作语句和函数	303	13.6 数据窗体模板	336
12.5 文件系统控件	306	本章小结	340
12.5.1 驱动器列表控件	306	习题 13	341
12.5.2 目录列表控件	307	附录	344
12.5.3 文件列表控件	308	附录 1 对象的命名前缀及默认属性	344
本章小结	313	附录 2 变量的命名前缀	345
习题 12	314	附录 3 键代码	346
第 13 章 数据库应用基础	318	附录 4 可捕获的错误	348
13.1 数据库概述	318	参考文献	350

第1章 Visual Basic 6.0 程序设计概述

本章要点:

- (1) 程序设计语言的发展, 程序与算法的基本概念;
- (2) Visual Basic 6.0 的功能及特点;
- (3) Visual Basic 6.0 集成开发环境的主要组成部分及简单使用;
- (4) Visual Basic 6.0 应用程序的开发步骤;
- (5) Visual Basic 6.0 的联机帮助功能。

1.1 程序设计概述

1.1.1 程序设计语言

一个完整的计算机系统由硬件系统和软件系统组成。硬件系统是指构成计算机的设备实体, 软件系统则是指为了运行、管理和维护计算机而编写并运行在硬件上的各种程序。

计算机所做的任何工作, 必须通过执行相应的程序来完成, 而任何程序都是用计算机语言编写的。

计算机语言的种类非常多, 总的来说可以分成机器语言、汇编语言、高级语言三大类。其中前两类依赖于计算机硬件, 因机器而异, 有时称为低级语言, 而高级语言与计算机硬件基本无关, 是目前使用非常广泛的程序设计语言。

1. 机器语言

机器语言是一种用二进制数“0”和“1”表示计算机各种操作的程序设计语言。它是计算机唯一能够识别的语言。一条机器指令能够控制计算机执行一种操作。由于不同计算机的指令系统不同, 所以使用的机器语言也不相同。用机器语言编写的程序, 能够直接被计算机理解并执行, 运行速度快, 占用内存少, 但缺点是难以记忆和识别, 可读性差, 容易出错, 对程序设计人员的要求极高, 可移植性极差。

2. 汇编语言

汇编语言是一种用助记符来表示各种基本操作的程序设计语言。相对于机器语言, 汇编语言比较直观, 易于记忆、检查和编程。汇编语言的实质和机器语言是相同的, 都是直接对计算机硬件操作, 只不过指令采用了英文缩写的标识符, 更容易识别和记忆。它同样需要编程者将每一步具体操作命令的形式写出来。因此, 汇编源程序一般比较冗长、复

杂、容易出错，通用性和可移植性差。计算机不能直接执行汇编语言程序，必须经过汇编程序将其翻译成机器语言程序后才能在计算机上执行，这需要程序员对计算机的结构非常了解。汇编语言的优点是显而易见的，使用汇编语言能完成一般高级语言所不能实现的操作，生成的可执行文件比较小，而且执行速度快。

3. 高级语言

高级语言是一种用接近自然语言和数学语言的语法、符号描述基本操作的程序设计语言。它是目前绝大多数编程者的选择。高级语言的出现是计算机技术发展的里程碑，它大大提高了编程效率，使人们能够开发出功能越来越强的程序。与汇编语言相比，它不但能将许多相关的机器指令合成为单条指令，而且去掉了与具体操作有关但与完成工作无关的细节。高级语言不依赖于计算机硬件，能在所有的计算机上使用。

高级语言的发展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(即面向对象程序语言)的过程。自第一个高级语言 FORTRAN 问世以来的五十多年来，共有几百种高级语言出现，有重要意义的有几十种。其中，影响较大，使用较普遍的有 FORTRAN、ALGOL、COBOL、Basic、LISP、Pascal、C、PROLOG、MATLAB、C++、Visual C、Visual Basic、Delphi、Java 等。

(1) 面向过程的语言。常用的面向过程的语言有 C、FORTRAN、Basic、Pascal 等。

使用面向过程的语言编写应用程序时，先将一个应用程序按照自上向下、由粗到细、逐步求精、模块化设计的原则进行分析，然后将其转化为顺序、选择和循环三种基本结构，再细分为具有各种特定功能的模块(也叫过程)。

(2) 面向对象的语言。面向对象程序设计(OOP)技术汲取了结构化程序中好的思想，并将这些思想与一些新的、强大的理念相结合，从而为程序设计工作提供了一种全新的方法。通常在面向对象的程序设计中，把整个现实世界或其中一部分看做是由不同种类的对象组成的有机整体。同一类型的对象既有共同点，也有各自不同的特征。各种类型的对象之间通过发送消息进行联系，消息能够激发对象作出反应，从而构成一个运动的整体。当前使用较多的面向对象的语言有 C++、Visual Basic、Delphi、Java 等。

1.1.2 程序与算法

程序(Program)是用特定的计算机语言编写的，用以指挥计算机完成预定功能的一段文字。算法(Algorithm)是程序处理问题的步骤和方法。

1. 程序

狭义地讲，程序是计算机指令的集合；广义地讲，程序是完成一定工作或任务的操作步骤的总称。计算机语言程序就是用相应语言来描述“实现某种功能”的操作步骤的语句序列，这些描述操作步骤的语句被翻译成计算机能够识别的计算机指令，在程序运行时能够被计算机直接识别而执行。

一个程序应包括以下两方面的内容：

- (1) 对数据的描述，即在程序中要指定数据的类型和数据的组织形式；
- (2) 对数据操作的描述，即操作步骤，也就是算法。

数据是操作的对象，操作的目的是对数据进行加工处理，以得到所期望的结果。作为

程序设计人员, 必须认真考虑和设计数据结构和算法。

2. 算法

算法解决“做什么”和“怎么做”的问题。程序的操作语句实际上就是对算法的具体描述。从事各种工作和活动, 都必须事先想好操作步骤, 只不过解决简单的问题一般不需要写出操作步骤。要让计算机解决问题, 不论简单与复杂, 都必须写出计算机能懂的操作步骤——程序, 计算机才能按程序设计者的设计步骤解决问题。

对同一个问题, 可以有不同的解题方法和步骤, 即算法可以不同。例如, 求 $1 + 2 + 3 + \dots + 100$, 有人可能先进行 1 加 2, 再加 3, 再加 4, 一直加到 100, 而有的人会采取这样的方法: $(1 + 100) + (2 + 99) + \dots + (50 + 51) = 5050$ 。当然还可以用其他的方法。为了有效地解决问题, 不仅需要考虑算法的正确性, 还要考虑算法的质量。

3. 算法的特性

一个方法要成为可以在程序设计中使用的算法, 需要具备以下特性:

- (1) 有穷性。一个算法要在有限的步骤内解决问题, 计算机程序不能无限地运行下去。
- (2) 确定性。算法中操作步骤的顺序和每一步的内容都应当是确定的, 不应当是含糊不清的。同一算法也不能有不同的解释存在, 即不能具有二义性。
- (3) 有零个或多个输入。输入就是从外部取得必要的信息。一个算法可以有多个输入, 例如输入一个年份, 判断是否是闰年; 一个算法也可以没有输入, 例如计算 $5 + 4$ 。
- (4) 有一个或多个输出。算法的目的就是求解, 即得到最终的结果。输出是与输入有着某些特定关系的量。一个算法得到的最终结果就是输出, 没有输出的算法是没有意义的。
- (5) 可执行性。一个算法应当是可以由计算机执行的, 而算法中描述的操作都可以通过计算机的运行来实现。

4. 算法设计要求

对算法的设计要求一般应包括以下几个方面:

- (1) 正确性。一个算法应当能够解决具体问题。算法的正确性(Correctness)包括:
 - ① 不含逻辑错误;
 - ② 对于几组输入数据能够得出满足要求的结果;
 - ③ 对于精心选择的典型、苛刻的输入数据都能得到要求的结果;
 - ④ 对于一切合法的输入都能输出满足要求的结果。
- (2) 可读性。算法应该以能够被人理解的形式表示, 即具备可读性(Readability)。太复杂的、不能被程序员所理解的算法不应在程序设计中出现。
- (3) 健壮性。健壮性(Robustness)指算法具有抵御“恶劣”输入信息的能力。当输入数据非法时, 算法也能适当地作出反应或进行处理, 而不会产生莫名其妙的输出结果。例如, 当输入三个边的长度值计算三角形的面积时, 一个有效的算法应该在三个输入数据不能构成三角形时报告输入错误, 并能够返回一个表示错误或错误性质的值, 同时中止程序的执行。
- (4) 高效率与低存储量。高效率与低存储量是优秀程序员追求的目标。效率指的是算法执行时间, 一个问题如果有多个算法可以解决, 则执行时间短的算法效率高。存储量指算

法执行过程中所需要的最大存储空间。高效率与低存储量的需求均与问题的规模有关。占用存储量最小、运算时间最少的算法就是最好的算法。但是在实际中,运行时间和存储空间往往是互相矛盾的,要根据具体情况选择。

5. 算法的表示方法

算法的实质是一种逻辑关系。对于这样一种关系,可以用多种方式表达。常用的表达方式有自然语言、传统流程图、N-S 结构化流程图、伪代码 PAD 图等。其中,传统流程图和 N-S 流程图是最常用的算法表示方法。下面举例介绍最简单的自然语言和传统流程图,目的是方便读者理解程序或设计程序。

【例 1-1】 写出计算 $1+2+3+\cdots+100$ 的算法。

分析:在写算法时需要考虑以下问题:

- (1) 累加的和怎样存储,即数据描述问题。
- (2) 使对操作步骤的描述最简单。

若采用 0 加 1,再加 2,再加 3,一直加到 100 的计算方式,则算法如下:

- (1) 假设有一个可以存放累加和的存储单元,其名称为 S;

(2) 按照这种方法求和,需要进行 100 次加法,显然,不可能写成 100 步的操作,否则,要是 1 加到 1000 或 10000 该如何表示呢?可以用数学归纳法,比如,不论数列(有规律的)是有限的还是无限的,可以通过归纳推理,用通项公式来简化表示。算法的描述本质上也是如此,方法是假设 i 表示第 i 次累加,第 i 次的加法可以描述成:将 $S+i$ 的值存放到 S 中,表示为 $S+i \Rightarrow S$,或用 Visual Basic 语言的赋值语句表示为 $S=S+i$,其含意是先计算 $S+i$ 的值,再赋给 S 。

用自然语言的表示方法描述算法如下:

- 步骤 1 将 S 置 0, i 置 1;
- 步骤 2 将 $S+i$ 赋给 S , 即 $S=S+i$;
- 步骤 3 使 i 的值增 1, 即 $i=i+1$;
- 步骤 4 如果 i 的值不大于 100, 则转到步骤 2, 否则, 继续下一步;
- 步骤 5 计算结束, 输出 S 中的值。

自然语言描述算法的方法虽然简单,但并不直观,采用流程图表示算法更加简单、直观。下面介绍传统流程图的表示方法。

传统流程图只需要使用美国国家标准化学会(American National Standard Institute, ANSI)规定的几个符号,就可以直观形象地将算法表示出来。一些常用的流程图符号在 Word 中都可以通过“绘图”命令来绘制,如图 1-1 所示。其中,  表示开始;  表示结束;  表示选择;  表示执行语句;  表示输入/输出数据;  表示预先定义的过程。用箭头表示流程的流向。

流程图由上述 6 种基本符号再加上适当的语句组成。例 1-1 的流程图算法表示如图 1-2 所示。如果用计算机语言将此算法描述出来,就成为这种语言的源程序。

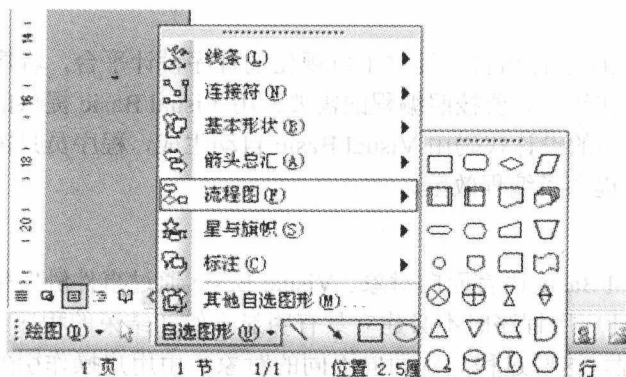


图 1-1 流程图符号

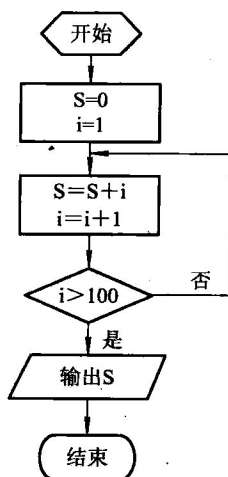


图 1-2 例 1-1 的流程图算法

1.2 Visual Basic 的发展和功能特点

1.2.1 Visual Basic 的发展

Visual Basic 是 Microsoft 公司于 1991 年推出的, 为开发 Windows 应用程序而设计的强有力的编程工具, 是一种具有良好的图形用户界面(Graphic User Interface, GUI)的可视化程序设计语言。Visual Basic 最初由 Basic 语言发展而来, 但从 Basic 到 Visual Basic 的变化是质的飞跃。这种变化不仅是语言功能的大大增强, 更主要的是程序设计方式的改变和程序界面类型的改变, 以及编程机制的改变。Basic 语言是基于过程的程序设计语言, 而 Visual Basic 是基于对象的事件驱动机制的程序设计语言。Basic 语言的编程界面是字符界面, 设计的程序是基于 DOS 平台的字符界面程序; Visual Basic 的程序开发界面是可视化的图形界面, 开发的应用程序也是 Windows 图形界面程序。在可视化开发环境中, 编程是一种更轻松、愉快和高效的智力活动。

Visual Basic 有多个版本, 从 1.0、2.0、3.0、4.0、5.0 到 6.0 版本, 功能不断增强。Visual Basic 6.0 于 1999 年推出, 深受用户欢迎, 目前仍被广泛使用。Visual Basic 6.0 版本之后就是 Visual Basic.Net。从 Visual Basic 6.0 到 Visual Basic.Net 又一次大的变化, 在概念上、框架上和编程方式上都有了变化, 但这种变化没有从 Basic 到 Visual Basic 的变化那么大。掌握 Visual Basic 6.0 之后再学习 Visual Basic.Net 就不是很难的事了, 而且绝大部分基于 Visual Basic 6.0 开发的程序很容易升级成 Visual Basic.Net 程序。本书以 Visual Basic 6.0 为蓝本来介绍。

Visual Basic 6.0 有三个不同的版本, 即学习版、专业版和企业版, 三个版本包含的内容多少不一样(学习版内容最少, 专业版居中, 企业版内容最全面)。

1.2.2 Visual Basic 的功能特点

1. 具有面向对象的可视化工具

Visual Basic 是基于面向对象的程序设计语言, 提供了可视化的程序设计平台, 程序设计人员不必为界面的设计编写大量的代码, 只要按照编程的需要, 用 Visual Basic 提供的工具箱在界面上添加各种对象即可, 界面的设计代码由 Visual Basic 自动生成, 程序员只需编写实现功能的那部分代码, 从而大大提高了编程效率。

2. 事件驱动的编程机制

传统的高级语言面向过程, Visual Basic 语言面向对象。Visual Basic 通过事件触发来执行相应的事件过程(事件驱动), 在设计应用程序时不必建立具有明显开始和结束的程序, 而只需编写若干微小的子程序, 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, 或由事件驱动程序调用通用过程来执行指定的操作。过程也可以由应用程序本身的消息触发, 还可以由操作系统或其他应用程序的消息触发。因此, Visual Basic 应用程序代码较短, 使得程序既易于编写又易于维护。

3. 结构化的程序设计语言

Visual Basic 是在 Basic 语言的基础上发展起来的, 它吸收了其他结构化设计语言的优点, 具有丰富的数据类型, 众多的内部函数, 模块化、结构化的程序机制, 结构清晰, 简单易学。

4. 友好的 Visual Basic 集成开发环境

在 Visual Basic 集成开发环境中, 用户可设计界面、编写代码和调试程序, 把应用程序编译成可执行文件, 直至把应用程序制作成安装盘, 以便能够在没有 Visual Basic 系统的 Windows 环境中运行。

5. 开放的数据库功能与网络支持

Visual Basic 提供了一些接口来实现网络以及数据库的链接。Visual Basic 拥有很强的数据库管理能力, 利用数据控件可以访问多种数据库系统, 可以通过直接访问或建立连接的方式访问并操作后台数据库。对后台数据库的访问主要是通过 ADO 控件或 ODBC 功能实现的, 而对后台数据库的操作是通过 Visual Basic 提供的简单命令集实现的, 也可以使用 SQL 语言以及一些高级的 ActiveX 控件或 API 函数。

Visual Basic 6.0 提供了 DHTML(Dynamic HTML)设计工具。这种技术可以使 Web 页面设计者动态地创建和编辑页面, 使用户在 Visual Basic 中开发多功能的网络应用软件。

6. 充分利用 Windows 资源

Visual Basic 程序开发与 Windows 系统紧密相连, 充分利用了 Windows 操作系统的环境和功能。用户可以方便地调用 Windows 的函数和应用程序, 利用 Windows 操作系统的环境及网络功能开发出各种类型的应用程序。

Visual Basic 提供的动态数据交换(DDE)编程技术, 可以在应用程序中实现与其他 Windows 应用程序建立动态数据交换和在不同的应用程序之间进行通信的功能。

Visual Basic 提供的对象链接与嵌入(OLE)技术把每个应用程序都看做一个对象,将不同的对象链接起来,嵌入到某个应用程序中,可以得到具有声音、影像、图像、动画和文字等各种信息的集合式文件。

1.3 Visual Basic 6.0 的集成开发环境

1.3.1 Visual Basic 6.0 的集成开发环境概述

Visual Basic 6.0集成开发环境(IDE)提供了整套工具,方便用户开发应用程序。它在一个公共环境里集成了许多不同的功能,例如设计、编辑、编译和调试。

当启动 Visual Basic 6.0时,可以见到如图1-3所示的对话框,其中会提示选择要建立的工程类型。使用 Visual Basic 6.0可以生成13种类型的应用程序。

在图1-3所示的“新建工程”对话框中有3个选项卡。

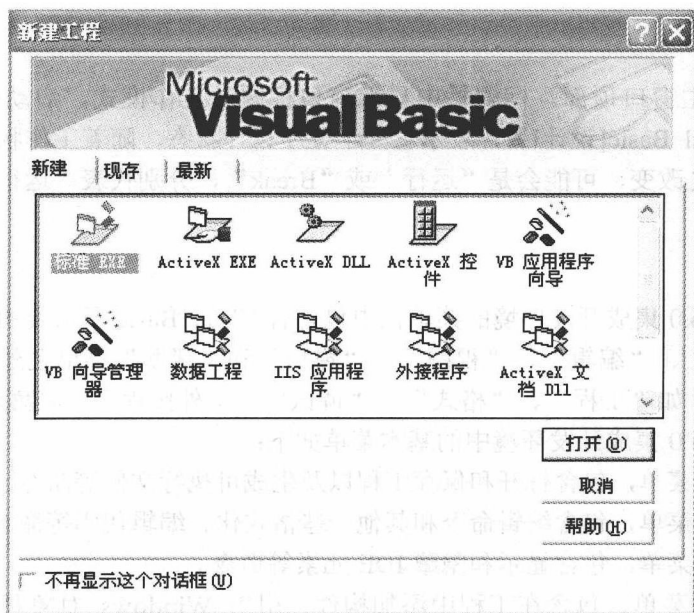


图 1-3 Visual Basic 6.0 中可以建立的工程类型

- (1) “新建”选项卡中列出了13种可生成的工程类型,这些是用户可以创建的文件类型。
- (2) “现存”选项卡中列出了可以选择和打开的现有工程。
- (3) “最新”选项卡中列出了最近使用过的工程,用户可以选择和打开一个需要的工程。

选择“新建”选项卡中的“标准 EXE”工程图标并单击“打开”按钮,可以打开如图1-4所示的 Visual Basic 6.0集成开发环境窗口。

需要说明的是,一般启动时可能见不到图1-4中的“立即”窗口、“本地”窗口、“监视”窗口和代码编辑窗口。在 Visual Basic 6.0集成环境中的窗口都可以通过“视图”菜单中的相应命令来打开和关闭。



图 1-4 Visual Basic 6.0 集成开发环境

1.3.2 Visual Basic 6.0 的主窗口

1. 标题栏

标题栏位于主窗口顶部。标题栏中显示窗口标题及工作模式，启动时它显示“工程1-Microsoft Visual Basic[设计]”，表示它处于程序设计状态，随着工作状态的不同，方括号中的内容也随之改变，可能会是“运行”或“Break”，分别代表“运行状态”或“中断状态”。

2. 菜单栏

Visual Basic 6.0 集成开发环境的菜单栏中包含有 Visual Basic 所需要的命令。它除了提供标准的“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”、“调试”、“外接程序”等菜单。

Visual Basic 6.0 集成开发环境中的基本菜单如下：

- (1) “文件”菜单，包含打开和保存工程以及生成可执行文件等命令。
- (2) “编辑”菜单，包含编辑命令和其他一些格式化、编辑代码等命令。
- (3) “视图”菜单，包含显示和隐藏 IDE 元素等命令。
- (4) “工程”菜单，包含在工程中添加构件、引用 Windows 对象和工具箱新工具等命令。
- (5) “格式”菜单，包含对齐窗体控件等命令。
- (6) “调试”菜单，包含一些通用的调试命令。
- (7) “运行”菜单，包含启动、设置断点和终止当前应用程序运行等命令。
- (8) “查询”菜单，包含操作数据库表时的查询命令以及其他数据访问命令。
- (9) “图表”菜单，包含操作 Visual Basic 工程时的图表处理命令。
- (10) “工具”菜单，包含建立 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。
- (11) “外接程序”菜单，包含可以随意增删的外接程序。默认时这个菜单中只有“可视化数据管理器”选项，通过“外接程序管理器”命令可以增删外接程序。