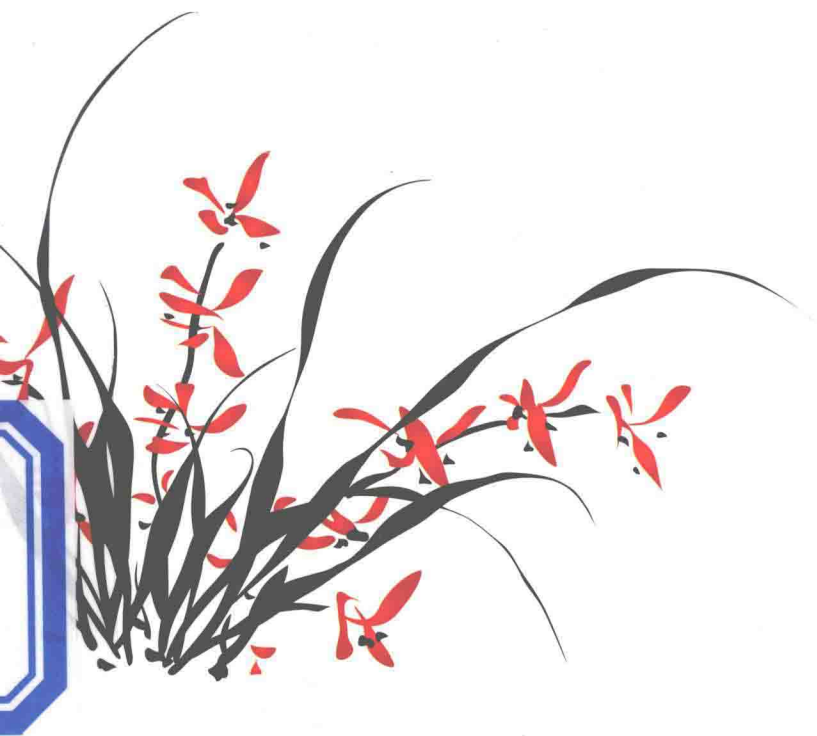


卓越系列

高等学校计算机基础教育课程“十二五”规划教材

Visual Basic 程序设计教程

柴欣 史巧硕 主 编
杨素梅 王建勋 裴祥喜 王旭辉 副主编



加强基础：基础知识讲解详细

注重实践：设计大量典型例题

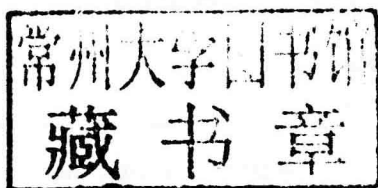
突出应用：源于实际精选案例

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

高等学校计算机基础教育课程“十二五”规划教材·卓越系列

Visual Basic 程序设计教程

主 编 柴 欣 史巧硕
副主编 杨素梅 王建勋 裴祥喜 王旭辉
参 编 (按姓氏音序排列)
陈冀川 曹新国 李建晶
李 娟 宋 洁 赵秀平



中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

内 容 简 介

本书是学习 Visual Basic 语言程序设计的基础教程, 全书系统地讲述了 Visual Basic 语言的基础知识、基本规则及编程方法, 在此基础上, 对 Visual Basic 可视化设计的重要特征如窗体、控件、菜单等的使用进行了深入讲解, 对 Visual Basic 的一些应用进行了详细介绍。

本书加强基础、注重实践, 在内容讲解上采用循序渐进、逐步深入的方法, 突出重点和难点, 使读者易学易懂。

本书适合作为高等院校各专业 Visual Basic 程序设计课程的教材, 也可以作为广大软件开发人员和自学者的参考用书, 还可作为计算机培训班的培训教材, 是初学者的得力帮手。

图书在版编目 (CIP) 数据

Visual Basic 程序设计教程 / 柴欣, 史巧硕主编

—北京: 中国铁道出版社, 2012.5

高等学校计算机基础教育课程“十二五”规划教材·卓越系列

ISBN 978-7-113-14435-7

I. ①V… II. ①柴… ②史… III. ①BASIC 语言—程序设计—高等学校—教材 IV. ①TP312

中国版本图书馆 CIP 数据核字 (2012) 第 050529 号

书 名: Visual Basic 程序设计教程

作 者: 柴 欣 史巧硕 主编

策 划: 吴宏伟

读者热线: 400-668-0820

责任编辑: 孟 欣

编辑助理: 何 佳 王 惠

封面设计: 刘 颖

责任印制: 李 佳

出版发行: 中国铁道出版社 (100054, 北京市西城区右安门西街 8 号)

网 址: <http://www.51eds.com>

印 刷: 北京新魏印刷厂

版 次: 2012 年 5 月第 1 版 2012 年 5 月第 1 次印刷

开 本: 787mm×1092mm 1/16 印张: 19.5 字数: 468 千

印 数: 1~5 000 册

书 号: ISBN 978-7-113-14435-7

定 价: 36.00 元

版权所有 侵权必究

凡购买铁道版图书, 如有印制质量问题, 请与本社教材图书营销部联系调换。电话: (010) 63550836

打击盗版举报电话: (010) 63549504

计算机技术和网络技术的飞速发展,已经深刻改变了人们的工作、学习和生活方式,在当今高度信息化的社会背景下,需要我们了解计算机处理问题的基本思想和方法。计算机程序设计课程作为大学非计算机专业学生的公共必修课,有着非常重要的地位。该课程可以使学生了解计算机编程的思想和方法,培养学生的计算思维。同时,该课程对于激发学生的创新意识、培养自学能力、锻炼实际的编程能力也起着极为重要的作用。

Visual Basic (以下简称 VB) 是目前使用最广泛的面向对象的程序设计语言之一,由此也成为各学校程序设计课程中广泛使用的语言。本书编者长期从事 VB 语言程序设计课程的教学工作,并曾利用 VB 语言开发了多个软件项目,有着丰富的教学经验和开发软件项目的实践经验,对 VB 有着较深入的理解。在编写本书的过程中,编者本着加强基础、注重实践、突出应用的原则,力求使本书达到可读性与适用性的统一。为了便于读者学习,在本书的体系结构和内容上采用了由浅入深、循序渐进的方法。为了提高读者编程技巧,在大部分章节中提供了典型例题。

本书共分为 11 章。第 1~5 章系统地讲述了 VB 语言的基础知识、基本规则及编程方法。其中:第 1 章介绍了 VB 的集成开发环境,并通过一个简单 VB 程序的开发过程,使读者对 VB 有一个感性的认识;第 2~5 章介绍了 VB 的数据类型、程序的基本结构与流程控制语句、数组及过程的基本操作,这些也构成了 VB 程序设计的基础内容。第 6~8 章介绍了 VB 可视化编程方面的知识,重点讲解了窗体、常用控件、ActiveX 控件与系统对象、对话框及菜单的使用。第 9~11 章介绍了 VB 的一些应用,包括 VB 的文件操作与管理、图形与绘图操作及数据库应用等。

为了更好地理论联系实际,达到良好的教学效果,我们还配合本教材编写了《Visual Basic 程序设计实验教程》,在其各章均相应地安排了若干上机实验及练习题,这样可以方便师生有计划、有目的地进行上机操作及课后练习,从而达到事半功倍的教学效果。另外,在本书和实验书中,还针对性地提供了一些接近实际要求的完整程序案例,教师可以以这些程序为范本,组织课程设计的题目。

本书由柴欣、史巧硕任主编,并负责全书的总体策划与统稿、定稿工作;杨素梅、王建勋、裴祥喜、王旭辉任副主编。各章编写分工如下:第 1 章由裴祥喜编写,第 2 章由王旭辉编写,第 3 章由史巧硕编写,第 4 章由李娟编写,第 5 章由赵秀平编写,第 6 章由柴欣编写,第 7 章由李建晶编写,第 8 章由王建勋编写,第 9 章由陈冀川编写,第 10 章由杨素梅编写,第 11 章由曹新国、宋洁编写。在本书的编写过程中,参考了大量文献资料,在此向这些文献资料的作者和出版者深表感谢。

由于时间仓促,编者水平有限,书中难免有疏漏之处,敬请各位专家、读者批评指正。

编 者

2012 年 4 月

第 1 章 Visual Basic 概述	1
1.1 计算机语言与计算机程序基本知识	1
1.1.1 计算机程序设计语言的发展	1
1.1.2 程序设计方法	2
1.2 中文 Visual Basic 6.0 概述	4
1.3 Visual Basic 6.0 的安装和启动	5
1.3.1 Visual Basic 的运行环境	5
1.3.2 Visual Basic 6.0 的安装	5
1.3.3 Visual Basic 6.0 的启动	6
1.4 Visual Basic 6.0 的集成开发环境	6
1.4.1 主窗口	7
1.4.2 窗体窗口	8
1.4.3 属性窗口	8
1.4.4 工程资源管理器窗口	9
1.4.5 代码窗口	9
1.4.6 窗体布局窗口	10
1.4.7 立即窗口	10
1.4.8 工具箱	11
1.4.9 Visual Basic 6.0 的帮助功能	11
1.5 设计一个简单的 Visual Basic 应用程序	14
1.5.1 Visual Basic 应用程序的构成	14
1.5.2 创建 Visual Basic 应用程序的步骤	15
1.5.3 Visual Basic 应用程序举例	15
1.6 事件驱动编程机制	21
1.6.1 对象的概念	21
1.6.2 对象的属性	22
1.6.3 对象的方法	22
1.6.4 对象的事件及事件过程	22
1.6.5 事件驱动编程机制	24
小结	24
习题	24
第 2 章 Visual Basic 语言基础	25
2.1 字符集、标识符和关键字	25

2.1.1	字符集	25
2.1.2	标识符	26
2.1.3	关键字	27
2.2	Visual Basic 的基本数据类型	27
2.2.1	数值型数据	27
2.2.2	字符型数据	29
2.2.3	日期型数据	29
2.2.4	逻辑型数据	30
2.2.5	对象型数据	30
2.2.6	变体型数据	30
2.3	常量和变量	30
2.3.1	常量	30
2.3.2	变量	32
2.4	运算符和表达式	35
2.4.1	运算符	35
2.4.2	表达式	39
2.5	常用内部函数	40
2.5.1	数学函数	40
2.5.2	转换函数	42
2.5.3	字符串函数	42
2.5.4	日期和时间函数	44
2.5.6	Shell()函数	45
2.6	Visual Basic 的编码规则	46
小结		48
习题		48
第3章	Visual Basic 流程控制结构	49
3.1	顺序结构	49
3.1.1	赋值语句	49
3.1.2	数据输出	50
3.1.3	数据输入	57
3.1.4	顺序结构应用程序举例	59
3.2	选择结构	62
3.2.1	If 语句	62
3.2.2	Select Case 语句	70
3.2.3	条件函数	73
3.2.4	选择结构程序设计举例	74
3.3	循环结构	78
3.3.1	For 循环语句	78

3.3.2	Do 循环语句	80
3.3.3	While 循环语句	83
3.3.4	循环的强制退出	84
3.3.5	循环的嵌套	86
3.3.6	几种循环结构的比较	88
3.3.7	循环结构应用程序举例	88
小结		94
习题		95
第 4 章	数组	96
4.1	数组概述	96
4.1.1	数组概念的引入	96
4.1.2	数组的基本概念	97
4.2	定长数组	98
4.2.1	一维数组	98
4.2.2	二维数组	102
4.3	动态数组	108
4.3.1	动态数组的声明	108
4.3.2	与数组操作有关的几个函数	110
4.4	数组应用举例	112
4.4.1	排序	112
4.4.2	查找	116
4.4.3	矩阵计算	117
4.5	控件数组	119
4.5.1	控件数组的概念	120
4.5.2	控件数组的创建	120
4.6	自定义类型及其数组	123
4.6.1	自定义类型的声明	124
4.6.2	自定义类型变量的声明与使用	124
4.6.3	自定义类型数组及其应用	125
小结		127
习题		128
第 5 章	过程	129
5.1	过程概述	129
5.2	Sub 过程	130
5.2.1	事件过程	130
5.2.2	Sub 子过程	130
5.2.3	Sub 子过程的建立	131
5.2.4	Sub 子过程的调用	132

5.3	函数 (Function) 过程	133
5.3.1	函数过程的定义	133
5.3.2	函数过程的建立	133
5.3.3	函数过程的调用	134
5.3.4	Sub 子过程与函数过程的区别	135
5.4	参数的传递	135
5.4.1	值传递	135
5.4.2	地址传递	137
5.4.3	数组传递	138
5.5	过程的嵌套和递归调用	142
5.5.1	过程的嵌套	142
5.5.2	过程的递归调用	143
5.6	变量的作用域	144
5.6.1	全局变量	144
5.6.2	模块级变量	145
5.6.3	过程变量 (局部变量)	146
5.6.4	静态变量	146
5.7	过程的作用域	147
5.7.1	全局过程	147
5.7.2	静态过程	149
5.7.3	模块 (窗体) 过程	150
5.8	过程应用举例	151
小结		152
习题		153
第 6 章	窗体与常用控件	154
6.1	窗体	154
6.1.1	窗体概述	154
6.1.2	窗体的属性、事件和方法	155
6.1.3	窗体的生命周期	157
6.1.4	多重窗体的操作	160
6.2	控件概述	163
6.2.1	控件的分类	163
6.2.2	控件的常用属性	163
6.2.3	控件的默认属性、焦点和 Tab 顺序	167
6.3	命令按钮、标签、文本框控件	168
6.3.1	命令按钮 (CommandButton)	168
6.3.2	标签 (Label)	169
6.3.3	文本框 (TextBox)	170

6.4	单选按钮、复选框和框架	178
6.4.1	单选按钮	178
6.4.2	复选框	179
6.4.3	框架	181
6.5	滚动条控件	184
6.6	列表框与组合框	185
6.6.1	ListBox (列表框) 控件	185
6.6.2	ComboBox (组合框) 控件	191
6.7	时钟 (Timer) 控件	193
6.8	鼠标与键盘事件	195
6.8.1	鼠标事件	196
6.8.2	键盘事件	199
6.9	控件应用举例	203
小结		204
习题		205
第 7 章	ActiveX 控件与系统对象	207
7.1	ActiveX 控件概述	207
7.2	ProgressBar 和 Slider 控件	209
7.2.1	ProgressBar 控件	209
7.2.2	Slider 控件	210
7.3	ImageList 控件与 ImageCombo 控件	211
7.3.1	图像列表 (ImageList)	211
7.3.2	图像组合框 (ImageCombo)	212
7.4	ListView 和 TreeView 控件	213
7.4.1	ListView 控件	213
7.4.2	TreeView 控件	213
7.5	StatusBar (状态栏) 和 ToolBar (工具栏) 控件	215
7.5.1	StatusBar 控件	215
7.5.2	ToolBar 控件	216
7.6	常用系统对象	217
7.6.1	App 对象	217
7.6.2	Clipboard 对象	218
小结		221
习题		222
第 8 章	对话框和菜单	223
8.1	对话框的分类和使用	223
8.1.1	系统预定义对话框	223

8.1.2	通用对话框	226
8.1.3	自定义对话框	230
8.2	菜单的设计和使用	231
8.2.1	窗口菜单设计	231
8.2.2	快捷菜单的设计	236
8.2.3	菜单设计举例	237
小结		239
习题		239
第 9 章	文件基本操作	240
9.1	文件的基本概念	240
9.1.1	数据文件的类型	240
9.1.2	处理文件的基本流程	241
9.2	顺序文件	242
9.2.1	顺序文件的读操作	242
9.2.2	顺序文件的写操作	246
9.3	随机文件	246
9.3.1	定义记录类型	247
9.3.2	随机文件的读操作	247
9.3.3	随机文件的写操作	247
9.4	二进制文件	249
9.4.1	二进制文件的打开和关闭	249
9.4.2	二进制文件的读/写操作	249
9.5	Visual Basic 中的文件控件	251
9.5.1	驱动器列表框	252
9.5.2	目录列表框	252
9.5.3	文件列表框	252
9.5.4	文件系统控件的联动	254
9.6	Visual Basic 文件操作命令和函数	257
9.6.1	文件操作命令	257
9.6.2	文件操作函数	260
小结		261
习题		262
第 10 章	图形和绘图操作	263
10.1	坐标系统	263
10.1.1	坐标系统与对象或控件的关系	263
10.1.2	坐标系统	264

10.2 使用颜色	266
10.2.1 使用控件的颜色属性	266
10.2.2 颜色值的设置	266
10.3 使用绘图控件	267
10.3.1 PictureBox 控件	268
10.3.2 Image 控件	268
10.3.3 Line 和 Shape 控件	269
10.4 使用图形方法绘图	272
10.4.1 图形方法	272
10.4.2 绘图操作	272
小结	279
习题	279
 第 11 章 Visual Basic 的数据库应用	280
11.1 数据库初步	280
11.1.1 数据库的相关知识	280
11.1.2 通过 Visual Basic 访问数据库	282
11.2 Visual Basic 中的数据源及数据控件	283
11.2.1 Visual Basic 的数据源及数据源控件	283
11.2.2 Visual Basic 的数据识别控件	283
11.2.3 利用数据控件创建简单的数据库应用程序	284
11.3 Visual Basic 中的数据库编程模型	286
11.3.1 DAO 模型	286
11.3.2 RDO 模型	288
11.3.3 ADO 模型	288
11.4 用数据库模型编程	289
11.4.1 创建数据库	289
11.4.2 对记录集 (Recordset) 进行操作	291
11.4.3 数据库的事务处理 (Transactions)	294
11.5 SQL 简介	295
11.5.1 SQL 组成	295
11.5.2 SQL 的数据定义功能	296
11.5.3 SQL 的数据处理功能	296
11.5.4 查询生成器	298
小结	299
习题	299
 参考文献	300

第 1 章

Visual Basic 概述

本章首先介绍计算机程序设计语言的基本知识，然后介绍 Visual Basic 6.0 的一些基本概念和集成开发环境，并通过一个简单的例子说明 Visual Basic 应用程序设计的一般过程。

学习目标

- 了解计算机程序设计语言的基本知识。
- 掌握 Visual Basic 6.0 集成开发环境的主要组成部分及使用方法。
- 理解对象的属性、事件和方法的概念，了解 Visual Basic 事件驱动的编程机制。
- 掌握开发 Visual Basic 6.0 应用程序的一般步骤。

1.1 计算机语言与计算机程序基本知识

计算机语言是人与计算机之间交互的语言，是人与计算机之间传递信息的媒介。为了使计算机进行各种工作，需要有一套用于编写计算机程序的数字、字符和语法规则，由这些字符和语法规则组成的各种指令（或各种语句）就是计算机能接受的语言。

计算机程序或者应用程序（通常简称程序）是指一组指示计算机每一步动作的指令，通常用某种程序设计语言编写。

本节将介绍计算机语言与计算机程序的基本知识。

1.1.1 计算机程序设计语言的发展

计算机之所以能自动进行计算，是因为采用了程序存储的原理，计算机的工作体现为执行程序。程序是控制计算机完成特定功能的一组有序指令的集合，编写程序所使用的语言就是计算机语言，也称为程序设计语言，它是人与计算机之间进行信息交流的工具。

从 1946 年世界上第一台计算机诞生起，在短短的 60 多年间，计算机技术迅速发展，程序设计语言的发展从低级到高级，经历了机器语言、汇编语言、高级语言到面向对象语言多个阶段。

1. 机器语言（Machine Language）

计算机能够直接识别和执行的二进制指令（也称机器指令）的集合称为该种计算机的机器语言。早期的计算机程序都是直接使用机器语言编写的，这种语言使用二进制代码，因此编写出的程序难以理解和记忆，目前已很少使用。

2. 汇编语言 (Assembly Language)

用助记符代替机器指令以利于理解和记忆,由此形成了汇编语言。汇编语言实际上是与机器语言相对应的语言,只是在表示方法上采用了便于记忆的助记符号来代替与机器语言相对应的二进制指令代码,因此也称为符号语言。计算机不能直接识别汇编语言,需要经汇编程序转换为机器指令码后才能识别。这种语言的执行效率较高,但由于难以理解,使用较少。

3. 高级语言 (High-Level Language)

机器语言和汇编语言是面向机器的语言,高级语言采用更接近自然语言的命令或语句,使用高级语言编程,一般不必了解计算机的指令系统和硬件结构,只需掌握解题方法和高级语言的语法规则,就可以编写程序。高级语言在设计程序时着眼于问题域中的过程,因此它是一种面向过程的语言。对于高级语言,人们更容易理解和记忆,这也给编程带来很大方便,但它与自然语言还是有较大差别。

4. 面向对象语言 (Object-Oriented Language)

面向对象语言是比面向过程语言更高级的一种语言。面向对象语言的出现改变了编程者的思维方式,使设计程序的出发点由着眼于问题域中的过程转向着眼于问题域中的对象及其相互关系,这种转变更加符合人们对客观事物的认识。因此,面向对象语言更接近于自然语言,是人们对于客观事物更高层次的抽象。

目前,世界上已经设计和实现的计算机语言有上千种,但实际被人们广泛使用的计算机语言不过数十种。

1.1.2 程序设计方法

程序设计即编写程序,是一项创造性的工作。进行程序设计时至少要具备两个方面的知识:一是要掌握一门或一门以上的高级语言;二是要掌握问题求解的方法与步骤,即如何将一个实际问题转化为一系列计算机可以实现的操作步骤,这就是“算法”需要研究的问题。有了正确的算法,就可以利用任何一种程序设计语言编写程序,使计算机进行工作,从而得到正确的结果。此外,为了更有效地完成程序设计工作,还要熟悉程序设计的方法。程序设计的方法也是随着计算机技术的发展而不断进步和完善的。在程序设计的发展过程中,人们对程序的结构进行了深入的研究,并不断地探索。究竟应该用什么样的方法来设计程序,如何保证程序设计的正确性,程序设计的主要方法和技术应如何规范等。经过反复实践,逐渐确定了程序设计的基本技术方法——结构化程序设计方法与面向对象的程序设计方法。

1. 结构化程序设计

结构化程序设计强调从程序的结构和风格上研究程序设计,它将程序划分为三种基本结构,人们可以用这三种基本结构来展开程序,表示一个良好的算法,从而使程序结构清晰、易读易懂且质量好。这三种基本结构为顺序结构、选择结构和循环结构。

(1) 顺序结构

顺序结构是一种最简单、最基本的结构。在顺序结构内,各块按照它们出现的先后顺序依次执行。图 1-1 所示为一个顺序结构形式,从图中可以看出它有一个入口 a 和一个出口 b, A 框和 B 框都是顺序执行的处理框,只有当 A 框执行结束后才能执行 B 框。

(2) 选择结构

选择结构中包含一个判断框,根据给定的条件 P 是否成立而选择执行 A 框或 B 框,当条件成立时,执行 A 框,否则执行 B 框。A 框或 B 框可以是空框,即不执行任何操作,但判断框的两个分支 A 或 B 执行完毕后都必须汇合在一起,从出口 b 退出,然后接着执行其后面的过程。图 1-2 所示的流程图就是选择结构,选择结构的程序中产生了分支,但对于整个的结构而言,它仍然只具有一个入口 a 和一个出口 b。

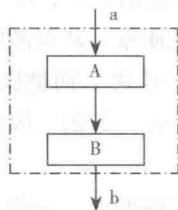


图 1-1 顺序结构流程图

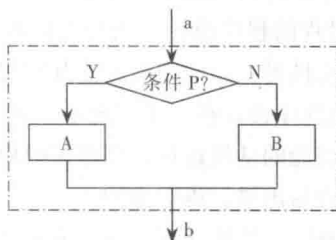


图 1-2 选择结构流程图

(3) 循环结构

循环结构又称重复结构,是指在一定条件下反复执行一个程序块的结构。循环结构也是只有一个入口和一个出口。根据循环条件的不同,循环结构分为当型循环结构和直到型循环结构两种。

① 当型循环的结构如图 1-3 所示,其功能是:当给定的条件 P 成立时,执行 A 框操作,执行完 A 操作后,再判断 P 条件是否成立,如果成立,再次执行 A 操作……如此重复执行 A 操作,直到 P 条件不成立才停止循环。此时不执行 A 操作,而从出口 b 脱离循环结构。

② 直到型循环的结构如图 1-4 所示,其功能是:首先执行一次 A 框操作,然后判断条件 P,当条件 P 不成立时,再次执行 A 框操作,并再次判断给定条件 P 是否成立……如此反复,直到给定的 P 条件成立为止,此时不再执行 A 框,而从出口 b 脱离循环。

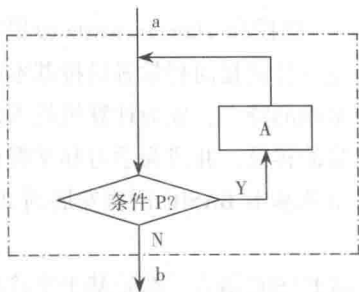


图 1-3 当型循环结构流程图

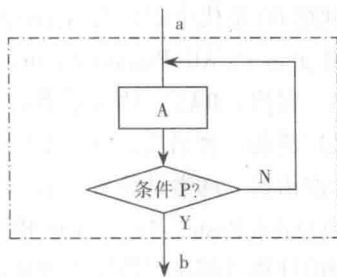


图 1-4 直到型循环结构流程图

由以上三种基本结构构成的程序,称为结构化程序。三种基本结构中的每一种结构都应具有以下特点:

- ① 程序只有一个入口和一个出口。
- ② 程序中没有死语句,即每一条语句都应该有一条从入口到出口的路径通过(至少通过一次)。
- ③ 程序中没有死循环(无限的循环)。

实践证明,任何满足以上三个条件的程序,都可以表示为由以上三种基本结构构成的结构化程序;反之,任何一个结构化程序都可以分解为一个个的基本结构。

结构化程序设计方法使得程序的逻辑结构清晰、层次分明,有效地改善了程序的可靠性和可维护性,提高了程序开发的效率。

2. 面向对象程序设计

结构化程序设计技术虽已使用了几十年,但如下问题仍未得到很好的解决:

① 面向过程的程序设计方法与人们习惯的思维方法仍然存在一定的差距,所以很难自然、准确地反映真实世界。因而用此方法开发出来的软件,有时很难保证质量,甚至需要重新开发。

② 结构化程序设计在方法实现中只突出了实现功能的操作方法(模块),而被操作的数据(变量)处于实现功能的从属地位,即程序模块和数据结构是松散地耦合在一起的。因此当应用程序比较复杂时,容易出错,难以维护。

由于上述缺陷,结构化程序设计方法已不能满足现代化软件开发的要求,一种全新的软件开发技术应运而生,这就是面向对象的程序设计(Object-Oriented Programming, OOP)。

20 世纪 80 年代,在软件开发中各种概念和方法积累的基础上,就如何超越程序的复杂性障碍,如何在计算机系统中自然地表示客观世界等问题,人们提出了面向对象的程序设计方法。面向对象的方法不再将问题分解为过程,而是将问题分解为对象。对象将自己的属性和方法封装成一个整体,供程序设计者使用。对象之间的相互作用则通过消息传递来实现。使用面向对象的程序设计方法,可以使人们对复杂系统的认识过程与系统的程序设计与实现过程尽可能一致。

1.2 中文 Visual Basic 6.0 概述

Visual Basic 从字面上理解是可视的 BASIC 语言,实际上它是一种可视化的应用程序开发工具,用来编制 Windows 环境下的应用程序。

20 世纪 60 年代中期,英国 Dartmouth 大学的 John Kemeny 教授和 Thomas Kurtz 教授开发出 BASIC (Beginner's All-Purpose Symbolic Instruction Code),其主要目的是向初学者讲授基本的程序设计技术。很快,BASIC 语言凭借着其短小精悍、易于学习掌握的特点,成为计算机技术发展史上使用最广泛的一种语言,获得了广大计算机用户和编程人员的喜爱,并成为学习和掌握计算机语言的标准语言。随着计算机技术的不断发展,BASIC 语言也从基本 BASIC 语言发展到 20 世纪 80 年代的 Quick Basic、True Basic 和 Turbo Basic 等。

早期的计算机都是字符操作界面,所有的编程语言,包括 BASIC 语言,都是基于字符界面进行编程开发的。20 世纪 80 年代后期,图形用户界面(GUI)开始成为主流,Windows 迅速风靡全球,人们在为计算机的使用更加方便而高兴的同时,也发现原有的大部分计算机语言已经不能适应这种图形化的软件开发平台,因为要编写 Windows 环境下的应用程序,必须建立相应的窗口、菜单、对话框等各种“控件”,这将使程序的编制变得越来越复杂。

直到 1991 年,Microsoft 公司推出了 Windows 应用程序开发工具 Visual Basic,这种情况才有了根本的改观。Visual 意为“可视化的”,指的是一种开发图形用户界面的方法,在计算机程序设计中引申为可视化的程序设计。程序员不用编写大量代码去描述界面元素的外观、位置等信息,

只要将这些元素用鼠标拖动到具体位置即可(即可视化设计,或所见即所得)。所以,Visual Basic 是基于 BASIC 的可视化的程序设计语言,是一个面向对象的集成开发系统。Visual Basic 一方面继承了其先辈 BASIC 简单、易学易用的特点,另一方面采用了面向对象、设计过程可视化、事件驱动的编程机制、动态数据驱动等先进的软件开发技术,使 BASIC 语言编程技术发展到了一个新的高度,它为广大用户提供了一种“所见即所得”的可视化程序设计方法。

Visual Basic 的诞生使编程技术向前迈进了一大步,它开创了可视化编程的先河。在它的带动下,许多优秀的可视化开发工具如 Delphi、PowerBuilder、Visual FoxPro 等相继问世。这些开发工具各有千秋,但它们都或多或少地从 Visual Basic 中汲取了营养。随着组件对象技术的不断进步,以及 Internet 的不断普及,Visual Basic 也在不断进步。自 1991 年推出 Visual Basic 后,经过 Microsoft 公司的不断努力,Visual Basic 的版本也在不断更新。目前,Visual Basic 的较新版本是 Visual Basic.NET。

本书以目前使用最多的 Visual Basic 6.0 为版本介绍 Visual Basic 的使用,对于 Visual Basic.NET,由于其基本内容与 Visual Basic 6.0 差别很小,本书的大部分内容也适合 Visual Basic.NET。

1.3 Visual Basic 6.0 的安装和启动

Visual Basic 是一整套功能完备的开发工具,使用前需安装到系统中。在开发 Visual Basic 程序时,可以通过几种方法启动 Visual Basic 系统。

1.3.1 Visual Basic 的运行环境

Visual Basic 6.0 是基于 Windows 系统的一个应用程序,本身对软、硬件没有特殊要求。

硬件要求:586 以上的处理器、16 MB 以上内存、100 MB 以上的剩余硬盘空间、CD-ROM 驱动器、鼠标等。

软件要求:Windows 2000、Windows XP 或 Windows 7。

1.3.2 Visual Basic 6.0 的安装

1. 初次安装

Visual Basic 6.0 必须在 Windows 环境下用系统自带的安装程序 Setup.exe 安装。步骤如下:

- ① 启动 Windows。
- ② 插入 Visual Basic 6.0 的安装光盘。
- ③ 运行 Visual Basic 6.0 安装程序 Setup.exe。
- ④ 根据用户要求和计算机配置选择以下三种安装方式之一:
 - 典型安装:将默认内容安装到硬盘上。
 - 自定义安装:按照用户选择的内容安装到硬盘上。
 - 最小安装:系统提取必需的内容安装到硬盘上。

注意:在安装时,如果设置了光盘自动运行,则在插入光盘时,安装程序将被自动加载,然后选择“安装 Visual Basic 6.0”选项,再根据要求选择安装方式。

2. 添加或删除 Visual Basic 6.0 部件

当安装好 Visual Basic 6.0 系统后,有时需要添加或删除某些部件。操作步骤如下:

① 在 CD-ROM 驱动器中插入 Visual Basic 安装光盘。

② 单击“开始”按钮,选择“设置”→“控制面板”命令。

③ 双击“控制面板”中的“添加或删除程序”图标,打开对应的窗口。

④ 选择其中的 Visual Basic 6.0 选项,然后单击“添加/删除”按钮,打开 Visual Basic 6.0 安装程序对话框,其中有三个按钮:

- “添加/删除”按钮:添加新的部件或删除已安装的部件,单击该按钮会弹出“维护”对话框,用户根据需要选中或取消选中部件前的复选框。
- “重新安装”按钮:以前安装的 Visual Basic 6.0 有问题,重新安装。
- “全部删除”按钮:将 Visual Basic 6.0 从系统中全部删除。

1.3.3 Visual Basic 6.0 的启动

完成 Visual Basic 6.0 安装后,可以通过以下两种方式启动 Visual Basic 6.0:

① 在“开始”菜单中选择“程序”命令,然后选择“Microsoft Visual Basic 6.0 中文版”→“Microsoft Visual Basic 6.0 中文版”命令,即可启动 Visual Basic 应用程序。

② 利用 Windows 建立快捷方式的功能,将 Visual Basic 6.0 程序的快捷方式图标放在桌面上。启动 Visual Basic 6.0 时,只要在桌面上双击该图标即可。

用上述任何一种方法启动 Visual Basic 后,都将弹出“新建工程”对话框,如图 1-5 所示。

该对话框有以下三个选项卡:

① 新建:用于建立新的 Visual Basic 应用程序,该选项卡中列出了可创建的 Visual Basic 工程类型,默认类型为“标准 EXE”,用于建立一个标准的 EXE 类型的工程,本书只讨论该工程类型。

② 现存:用于选择和打开现有的 Visual Basic 应用程序工程文件,继续进行编辑、修改。

③ 最新:列出了最近一段时间内编辑过的 Visual Basic 应用程序工程文件。

新建一个工程时,可选择“新建”选项卡,从中选择“标准 EXE”选项并单击“打开”按钮,即可进入 Visual Basic 6.0 应用程序集成开发环境进行新工程的创建。



图 1-5 “新建工程”对话框

1.4 Visual Basic 6.0 的集成开发环境

Visual Basic 6.0 的集成开发环境是开发 Visual Basic 应用程序的平台。熟练掌握 Visual Basic 的集成开发环境的使用方法是设计开发 Visual Basic 应用程序的基础。Visual Basic 的集成开发环境界面如图 1-6 所示,它包括以下几个独立的窗口。