

ual Basic

Visual Basic

程序设计

张得太 丁源明 主 编 赵双萍 赵锡英 副主编

VISUAL BASIC CHENGXU SHEJI

甘肃科学技术出版社

图书在版编目(CIP)数据

Visual Basic 程序设计 / 张得太, 丁源明 主编. — 兰州: 甘肃科学技术出版社, 2007.1

ISBN 978-7-5424-1126-6

I.V... II.①张...②丁... III.BASIC 语言—程序设计

Visual Basic 程序设计

张得太 丁源明 主 编
赵双萍 赵锡英 副主编

甘肃科学技术出版社

图书在版编目(CIP)数据

Visual Basic 程序设计 / 张得太, 丁源明主编. —兰州:
甘肃科学技术出版社, 2007. 1
ISBN 978-7-5424-1126-6

I. V... II. ①张...②丁... III. BASIC 语言—程序设计
IV. TP312

中国版本图书馆CIP数据核字 (2007) 第007609号

责任编辑:杨丽丽(0931-8773274 gskjyll@126.com)

封面设计:陈妮娜(0931-8773275)

出版发行:甘肃科学技术出版社(兰州市南滨河东路 520 号 0931-8773237)

印 刷:兰州市兰林印刷厂(兰州市城关区东李家湾 318 号)

开 本:787mm×1092mm 1/16

印 张:18.75

字 数:430 千

版 次:2007 年 2 月第 1 版 2007 年 2 月第 1 次印刷

印 数:1~2000

书 号:ISBN 978-7-5424-1126-6

定 价:28.00 元

序 言

自 Windows 操作系统诞生以来,Microsoft 公司推出了 Visual Basic 以下(简称 VB)程序设计语言,作为可视界面应用程序开发的工具,备受广大用户的青睐,并且易学易用,有着广泛的应用基础,成为初学者学习面向对象程序设计的首选语言。

VB 语言即是一种可视化对象的程序设计语言,学习设计和应用视窗界面对象的同时,还需要以算法程序设计为基础。但是,对于没有学习过计算机语言的初学者,一方面需要计算机的基础知识、基本的数据结构和算法基础,另一方面又需要掌握和应用 VB 控件等对象,在学习中不免存在思路的相互转换,产生更多疑惑,对程序语言语法和算法的理解存在偏差,较难明白“做什么?怎么做?”。为了更好地解决这方面的问题,我们在总结多年教学实践经验的基础上,组织编写了这本更适合初学者学习和教学使用的教程,并配有相应的实验习题指导书。

一、本书的特点

1. 创设问题情境。每章的开始以问题的形式提出为什么或做什么,从如何解决问题的角度阐述相关的知识点,细化难点。着重解决“做什么?怎么做?”的问题。

2. 合理组织,分散重点和难点。将 VB 中的对象和语言语法以应用为主线有机地结合起来,从问题出发将难点分布到不同章节,遵循学习者认知的基本规律,达到各个击破、循序渐进的目的。具有知识点、应用主线和应用层面突出的特点。

3. 突出难点和疑点。内容的组织根据知识点的难与易进行细化与简化,达到繁简适当、通俗易懂和言简意赅的目的,突出重难点。

4. 应用为主线。以应用为目的主线贯穿全书,增强目的性,增加趣味性,突出应用目标。

5. 例题、习题难易梯度化。设计不同层次的例题、习题和试验项目,由易到难体现梯度和层次,突出从简到繁、循序渐进的特点。

二、使用本书应注意的问题

1. 避免一气呵成,面面俱到。突出点、线、面的特点,每一章节所突出的知识点说明做什么和怎么做的问题,组成点;与后续章节紧密相连,若干章节构成线;最终目标到达应用层面。

2. 了解每章开篇中提出的问题和学习的目标,注重发现问题、提出问题和解决问题的思路。

3. 加强实践操作,注重程序设计思想,积累解决问题方法。

三、本书的编写组织

参加本书编写的还有近年来担任本门课程授课的教师魏莹、周秀媛、何辉、李晓斌、樊小龙等共同编写完成。在此感谢关心和帮助编辑出版本书的学校领导及各界朋友,并对那些在因特网上提供可用信息而未署名的朋友一并表示感谢。

编 者

2006. 9. 25

图书在版编目(CIP)数据

Visual Basic 程序设计 / 张得大、丁海明主编. — 兰州 :

甘肃科学技术出版社, 2002.1

ISBN 978-7-5424-1264-4

内容简介

本书以 Microsoft Visual Basic 6.0 中文版为背景,介绍了面向对象的基本概念、可视化应用程序设计的基本方法和 Visual Basic 对象应用方法,以程序设计语言的基本语法为基础的算法基本思想和方法,过程及函数,窗体模块、标准模块的设计及应用,以应用为目的的图形处理、文件管理及数据库程序设计与应用等。

本书的优点是分散了学习 Visual Basic 的难点,遵循学习的认知规律和教学规律,创设问题情境,从提出问题进而解决问题的角度引出知识点、剖析难疑点,深入浅出、层次分明、目的明确,适用于教学和初学者学习。

目 录

第 1 章 VB 程序设计基础	1
1.1 程序设计与程序语言	2
1.1.1 计算机基本思想的形成	2
1.1.2 程序设计语言	4
1.1.3 程序与算法	6
1.2 Visual Basic6.0 及面向对象程序设计的基本概念	8
1.2.1 Visual Basic6.0 启动	8
1.2.2 VB 集成开发环境	9
1.2.3 面向对象的基本概念及 VB 中的对象	11
习题 1	18
第 2 章 VB 程序设计的基本方法	20
2.1 VB 程序设计的基本步骤	21
2.2 界面设计的基本方法	23
2.2.1 对象的编辑	23
2.2.2 对象的属性及属性设置	24
2.3 事件及事件过程的编写	29
2.3.1 对象的基本事件	29
2.3.2 事件过程的编写	30
习题 2	36
第 3 章 VB 程序语言基本语法	38
3.1 基本数据类型	39
3.1.1 数据类型的定义	39
3.1.2 常量	41
3.1.3 变量	41
3.2 VB 的运算符和表达式	44
3.2.1 算术运算符和算术表达式	44
3.2.2 关系运算与逻辑运算	46
3.2.3 关系表达式和逻辑表达式	46
3.3 基本语句	47
3.4 常用内部函数	50
3.4.1 函数的语法构成	50
3.4.2 数学函数	50
3.4.3 字符串函数	52
3.4.4 转换函数	53

3.4.5 日期和时间函数	53
3.4.6 随机数函数	54
3.4.7 判断函数(Is 函数)	55
3.5 Print 方法的格式输出	55
习题 3	58
第 4 章 输入、输出程序设计	62
4.1 数据输入	63
4.1.1 InputBox 函数	63
4.1.2 文本框(TextBox 控件)	64
4.2 数据输出	65
4.2.1 标签(Label 控件)	65
4.2.2 MsgBox 函数和语句	66
4.3 对象的焦点设置及应用	67
4.3.1 Tab 键的应用	68
4.3.2 焦点的程序控制	69
习题 4	74
第 5 章 选择结构及应用	77
5.1 选择结构	78
5.1.1 If... Then 语句	78
5.1.2 If... Then... Else 语句	80
5.1.3 If 语句的嵌套	81
5.1.4 If... Then... ElseIf 语句	83
5.1.5 SelectCase 语句	85
5.1.6 If 函数	86
5.2 单选按钮、复选框和框架	87
5.2.1 单选按钮(OptionButton 控件)	87
5.2.2 复选框(CheckBox 控件)	89
5.2.3 框架(Frame 控件)	90
5.2.4 应用举例	91
5.3 键盘事件的应用	95
习题 5	97
第 6 章 循环控制结构及应用	101
6.1 循环控制结构	102
6.1.1 循环控制结构的基本概念	102
6.1.2 循环控制结构语句	103
6.2 列表框、组合框	111
6.2.1 列表框(ListBox 控件)	111
6.2.2 组合框(ComboBox 控件)	116
习题 6	119

第7章 数组与记录	125
7.1 数组	126
7.1.1 数据结构的基本概念	126
7.1.2 数组的声明和引用	127
7.1.3 数组的清除	131
7.1.4 控件数组	132
7.2 数组的应用	134
7.2.1 排序算法	134
7.2.2 程序举例	136
7.3 用户自定义数据类型	141
7.3.1 用户自定义数据类型的语法	141
7.3.2 自定义类型的引用	142
习题7	144
第8章 图形控件与通用对话框	149
8.1 图形控件	150
8.1.1 图片框与图像框	150
8.1.2 直线与形状控件	151
8.2 计时器(Timer)	153
8.3 滚动条(ScrollBar 控件)	156
* 8.4 通用对话框	158
习题8	162
第9章 VB 绘图方法	164
9.1 Visual Basic 的坐标系统	165
9.2 绘制几何图形	166
习题9	173
第10章 过程及标准模块	174
10.1 VB 应用程序的组成结构	175
10.2 过程的定义	176
10.2.1 Sub 过程的定义	176
10.2.2 Sub 过程的调用及参数传递	179
10.2.3 Function 过程	182
10.2.4 Sub 过程与 Function 过程的区别	182
* 10.3 过程的递归调用	183
10.4 其他形式的参数	185
10.4.1 数组参数	185
* 10.4.2 可选参数、可变参数与对象参数	187
10.5 变量和过程的作用域	189
10.5.1 变量的作用域	189
10.5.2 过程的作用域	190

10.6 过程应用举例	190
习题 10	192
第 11 章 多重窗体与菜单编辑器	197
11.1 多重窗体	198
11.1.1 启动对象	198
11.1.2 窗体的加载与卸载	199
11.2 菜单设计	203
11.2.1 菜单编辑器	204
11.2.2 菜单设计步骤	205
11.2.3 菜单的控制	206
11.2.4 建立弹出式菜单	209
习题 11	212
第 12 章 数据文件与文件系统控件	214
12.1 数据文件	215
12.1.1 顺序文件的操作	215
12.1.2 随机文件	218
* 12.1.3 二进制文件	221
* 12.2 文件管理的函数和语句	222
12.3 文件系统控件及应用	223
12.3.1 文件系统控件	223
12.3.2 文件系统控件的应用	227
习题 12	230
* 第 13 章 数据库	233
13.1 数据库基础	234
13.1.1 概述	234
13.1.2 关系数据库及其特点	234
13.1.3 数据库管理器	237
13.2 结构化查询语言 SQL	240
13.3 数据控件(Data Control)	241
13.3.1 数据控件与记录集对象	241
13.3.2 数据控件的常用事件	245
13.3.3 数据控件的应用举例	246
13.4 ADO(Active Data Object)数据控件	248
13.4.1 ADO 对象模型	248
13.4.2 使用 ADO 数据控件	248
13.4.3 使用数据窗体向导	251
习题 13	255
* 第 14 章 多文档界面	259
14.1 多文档界面(MDI)	260

14.1.1MDI 窗体的特点	260
14.1.2MDI 有关的属性、事件和方法	261
14.1.3 建立 MDI 应用程序的步骤	261
14.2 建立工具栏.....	262
14.2.1 利用控件建立工具栏	262
14.2.2 使用“工具栏向导”建立工具栏	266
14.3 状态栏.....	269
14.3.1 为窗体添加状态栏	269
14.3.2Panels 集合	269
14.3.3 在设计时创建状态栏	270
14.3.4 运行时处理状态栏	272
习题 14	276
附录 1 VB 中对象的属性	277
附录 2 VB 中对象的事件	283
附录 3 VB 中对象的方法	286
附录 4 标准 ASCII 码表.....	288
附录 5 标准键盘键的 KeyCode 系统常量	289
参考文献.....	290

第1章 VB 程序设计基础

问题背景:

学习本门课程的目的是学会自己设计程序,那么,什么是计算机语言?计算机语言和自然语言又有什么不同?什么是算法?程序设计与算法有什么联系?什么是面向对象程序设计?VB 程序设计语言中有怎样的对象、属性、方法?如何设计一个简单的 VB 应用程序?

主要内容:

- 程序设计的基本思想
- 程序设计语言简介
- 程序设计与算法
- VB 集成开发环境与面向对象的基本概念
- 简单 VB 应用程序设计

学习指导:

通过学习本章内容,重点掌握程序设计的基本概念,初步建立程序设计的基本思想。认知 VB 面向对象的程序设计的特点,以 VB 的具体可视对象为基础,重点了解对象、属性和方法等基本概念,了解 VB 工程及工程类别,熟悉 VB6.0 集成开发环境、界面操作和工作方式,学习设计简单的 VB 应用程序。

1.1 程序设计与程序语言

在计算机文化基础课程的学习中已初步了解或掌握了计算机的发展过程、基本组成结构和工作原理等基本知识,并且学会了文字处理、表处理等基本操作,已经清楚地知道这些操作是通过程序设计师设计好的程序实现的。程序设计到底是做什么工作?程序设计语言究竟是怎样的语言?让我们再次回顾计算机的发展历史,进一步了解计算机的基本思想与程序设计语言。

1.1.1 计算机基本思想的形成

最早的计算机,可以追溯到 19 世纪。查尔斯·巴贝奇(Charles Babbage, 1791~1871), 22 岁从剑桥获得数学博士学位,当时被认为是一位很具潜力的年轻数学家。在他还是学生的时候,就对计算器的设计产生兴趣。1822 年,查尔斯·巴贝奇成功地设计出用来制作对数和三角函数表的差分机(Difference Engine),其精度可达 6 位小数。由于获得英国皇家学院的资助,他开始应用当时新的数学知识,设计并制造一台功能更广泛的差分机。10 年过去,就在差分机即将完成的时候,他领悟到:一合理想的计算器,必须能够依照指令改变其执行程序,也就是可变程序的概念——这是个前所未有的伟大理想。他立刻弃置了差分机,开始埋头设计分析机 (Analytical Engine) 以实现他的理想。整个设计在 1838 年完成,就连写程序的方法都想好了。可惜当时的技术无法配合, Babbage 终其一生未能完成一台计算器,但留下细致入微的设计图 300 多张、笔记 6000 多页和许多半成品。

分析机的设计构想已经和现在的计算机十分相似,它有“存储库”、“计算室”,在穿孔卡片中存储程序和数据,控制中心类似于今天的 CPU。因此,巴贝奇也被尊称为计算机之父。



图 1.1 Charles Babbage



图 1.2 Ada Byron

巴贝奇在年近半百的时候,结识了当时的名才女爱达·奥古斯塔·洛夫莱斯 (Ada Byron, Augusta Lovelace, 1815~1852), 诗人拜伦的女儿,后来成为伯爵夫人。但是她在满月的时候,父母就离异了。她一生中没机会见到那位诗人父亲。后来根据她的遗嘱,她与父亲并排葬在一起,两人都享年 36 岁。

爱达不但能诗善画,还是当时主要数学家狄·摩根 (Augustus de Morgan 1806~1871) 的学生。因机缘巧合,她参观了巴贝奇的工作室。虽然零件散落满地,屋子中央坐落着一台怪物般的半成品,但在听到解释之后,她说她看到了一个“伟大而美丽的发明”。

爱达参加了巴贝奇一部分的工作,负责分析机的程序设计。指出分析机可以编程,发现了

编程的基本要素(如循环、子程序),建议分析机用二进制存储,并预言分析机可以编制复杂的音乐和图画。

在 19 世纪 40 年代,Babbage 和 Ada 已经认识到:计算程序的核心技术在于重复。试想,如果在今天的程序语言中不许使用 while、for 这些迭代方法,也不许利用 If 或 GoTo 实现分支,那么程序语言将有何用?

Babbage 和 Ada 曾经明确地写下:自动计算器的真正重要之处,在于它可以重复执行一套给定的程序。其重复次数可以在计算前确定,也可以依计算结果而临时决定。显然前者就是 for 循环,后者是 while 循环。所以,最早的程序设计应该归功于爱达。20 世纪 80 年代由美国国防部委托设计的一套程序语言,就以 Ada 命名,以此来纪念这位伟大的女性。

然而 Ada 却把这种可以因软件而改变硬件功能的设计理念,归功于法国人约瑟夫·玛丽·雅卡尔 (Joseph Jacquard, 1752~1834), 法国提花机的发明者(1801 年),第一台能织出复杂图案的自动织布机。

1946 年第一台电子计算机 ENIAC(Electronic Numerical Integrator And Calculator)诞生和 1949 年第一台使用二进制、能存储程序的计算机 EDVAC(Electronic Discrete Variable Automatic Calculator)问世,奠定了现代计算机设计体系结构,EDVAC 的突出优点在于:① 把计算机要执行的指令和处理的数据都用二进制表示;② 把执行指令和处理的数据按顺序编成程序存储到计算机内部让它自动执行。从而解决了程序的“内部存储”和“自动执行”问题。这种由计算器、逻辑控制器、存储器、输入和输出五个部分组成的“存储程序”式计算机确立为冯式结构计算机。这个思想的提出者就是美籍匈牙利科学家冯·诺伊曼(John von Neumann, 1903~1957),因此人们称冯·诺伊曼为现代电子计算机之父。但冯·诺伊曼本人没有说过存储程序的概念是他的发明,却不止一次的提到图灵是现代计算机思想的创始人。



图 1.3 John Von Neumann

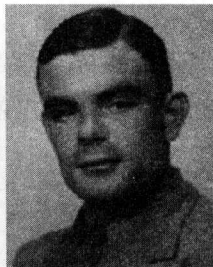


图 1.4 Alan Turing

关于计算理论可以追溯到 1900 年,当时著名的大数学家希尔伯特在世纪之交的数学家大会上给国际数学界提出了著名的 23 个数学问题。其中第十问题是这样的:存在不存在一种有限的、机械的步骤能够判断“丢番图方程——形如 $a_1x_1^{b_1} + a_2x_2^{b_2} + \dots + a_nx_n^{b_n} = c$ 的整系数多项式方程,又称不定方程”是否存在解?这就提出来了有限的、机械的证明步骤的问题,用今天的话说就是算法。但在当时,人们还不知道“算法”是什么。实际上,当时数学领域中已经有很多问题都是跟“算法”密切相关的,因而,科学的“算法”定义呼之欲出。之后到了 20 世纪 30 年代的时候,终于有两个人分别提出了精确定义算法的方法,一个人是图灵,一个人是丘奇。其中图灵提出

来的图灵机模型直观形象,很快得到了大家的普遍接受。

艾伦·麦席森·图灵(Alan Mathison Turing,1912~1954),英国数学家,计算机科学家。图灵最大的贡献就是把算法这样一个基本的、深刻的概念用他的图灵机模型讲清楚了。正是因为图灵奠定的理论基础,人们才有可能发明 20 世纪以来甚至是人类有史以来最伟大的发明——电子计算机。因此人们称图灵为“计算机理论之父”。

“存储程序”的现代计算机思想决定了计算机具有“大脑”,能够按照人们设计好的步骤——符号化的序列,完成确定的任务,这种符号化的序列就是程序设计语言。

1.1.2 程序设计语言

语言是用来表达、交流思想的工具,而程序设计语言则是人们用来向计算机传递信息与下达命令的通信工具。例如,和美国人沟通要用英文,同样地,人若要和计算机沟通的话,就必须使用计算机能懂的语言,这种语言称为程序语言(Programming Language)。而一般我们用来与人沟通的语言则称为自然语言(Natural Language)。

程序语言依据与自然语言的相似度又可以分为三种:机器语言、汇编语言、高级语言。其中高级语言与人类所使用的自然语言最为相近,而机器语言则和人类所使用的语言南辕北辙。除了传统的机器语言、低级语言、高级语言之外,目前的集成开发环境(如 Visual Basic)则朝向更高级语言发展,而在人工智能领域则朝向自然语言发展。

1. 机器语言(Machine Language)

机器语言是计算机硬件唯一看得懂(识别)的语言,换句话说,机器语言的一条语句就是一连串的 0、1 二进制数的一个组合,因此又称为机器码。一般人通常看不懂这些 0、1 所代表的特殊涵义,其实对于计算机而言,这些 0、1 的组合数字,可能代表某种数据,也可能代表某个指令。由于大多数的人无法了解或记忆这一连串 0、1 数字所代表的含义(见表 1.1 所示),因而发展了汇编语言与高级语言。

由于不同的机器(CPU 不同),其机器指令不同,因此机器语言属于与机器相关的程序语言,并且不可移植,一旦更换 CPU,程序就无法使用。

表 1.1 机器语言与汇编语言的对比

处理器	机器语言指令	汇编语言指令	意义
8051	0000010	INC	执行累加 1
	10000100	DIV	执行除法
X86	0000001111001000	ADD CX, AX	执行缓存器加法
	101110010011010000010010	MOV CX,1234H	将 CX 缓存器的值指定 1234H

2. 汇编语言(Assembly Language)

汇编语言是一种接近于机器语言的表示方法,不过是使用人类比较容易记忆的助记码(mnemonics)来对应机器语言的 0、1 组合。在汇编语言中,使用操作码与操作数来表示一连串的 0、1 组合,这些操作码用类似英文的缩写以利人类的记忆与理解,例如:使用 INC 来代表 Increment(累加)。表 1.1 中可以看到机器语言与汇编语言的差异:

(1)汇编语言比机器语言更接近人类所使用的自然语言。

(2)不同的 CPU 所使用的汇编语言也不相同,所以汇编语言也是与机器相关的程序语言,并且不具备可移植性。

(3)由于计算机只“看得懂”机器码,因此汇编语言必须通过汇编程序(Assembler)翻译为机器码。而任何一个完整的汇编语言指令恰好对应一组机器语言的 0、1 串。

同样,不同的 CPU 必须使用不同的汇编语言,并且对于该 CPU 的组织结构有充分认知,因此,汇编语言与机器语言则合称为低级语言(Low-level Language)。这种低级语言仍旧无法被绝大多数的人所接受,因而发展了更接近于人类自然语言的高级语言。

3. 高级语言

高级语言使用更接近人类思维的方式来设计程序,当程序设计完成之后,必须通过专门的翻译程序翻译后才能够被计算机执行。高级语言的表达式通常具有比较强大的功能,因此,一条高级语言的语句可能被翻译成许多的机器码以便完成复杂的工作。如 BASIC、Fortran、Pascal、C 语言等都是高级语言。Visual Basic 既是一种高级语言也是一种可视化集成开发环境。

4. 自然语言与人工智能(Natural Language and AI)

计算机的最主要目的是为了帮助人们解决问题,由于传统的程序语言不易学习和掌握,因此在人工智能的研究中,正积极研究如何使计算机了解人类使用的自然语言(Natural Language),期望当计算机具有处理自然语言的能力之后,任何人都可以是一位自然语言的程序设计师,写程序就变成了一种交代计算机完成工作。例如:您只要对着计算机下达指令“计算机,请把这份信送到最近的邮局”,计算机就会自动执行您的指令。

5. BASIC 及 Visual Basic 简介

BASIC 语言是在 1964 年,由 Dartmouth 学院的 John Kemeny 与 Thomas 共同发展的一套计算机语言,全名是 Beginner's All-purpose Symbolic Instruction Code,简称为 BASIC。顾名思义,BASIC 是一种专为初学者所设计的程序语言,由于 BASIC 语法接近人类所使用的自然语言(英文)与数学算式,因此深受计算机初学者的喜爱。

早期的 BASIC 属于直译语言,因此可以逐行执行,并且立刻看到执行的结果,很容易被初学者所接受。不过由于欠缺结构化的概念,使得程序维护及管理比其他高级语言困难,因此,较少被专业程序设计师拿来设计大中型程序。

在 1988 年 Microsoft 推出的 Quick BASIC 4.5 版(简称 QB)之后,就改进了以上的缺点。QB 程序实现了结构化以及模块化的设计,并且也由直译语言转为编译语言,所以可以编译执行,同时在 1991 年又推出了中文版本,因此,被国内许多教学单位拿来作为教学语言,有不少程序设计师的程序母语也正是 BASIC 语言,这个变化可以说是 BASIC 语言的第一次大转变。

BASIC 语言第二次大转变是在 Windows 操作系统流行之后,由于 Windows 的图形用户接口(GUI;Graphics User Interface)对使用者产生了极大的吸引力,因此,窗口(图形化接口)程序设计的需求也同样被重视。为此,Microsoft 在 1991 年推出了 Visual Basic 1.0 (简称 VB 1.0),1995 年随着 Windows 95 的发布,Visual Basic 4.0 也随之推出。Visual Basic 6.0,简称 VB6.0,是

1998 年 Microsoft 发布的 Visual Studio 98 家族中的成员之一, 是一个 32 位的 Windows 开发工具。

1.1.3 程序与算法

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, 这些实现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) 对数据的操作描述, 即操作步骤, 也就是算法 (Algorithm)。

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。比如, 厨师做菜肴, 需要有菜谱。菜谱一般应包括配料 (用那些原料及各原料的用量) 和操作步骤 (如何使用这些原料按规定的先后步骤进行加工所需要的菜肴)。作为程序设计人员, 必须认真考虑组织数据和设计算法。

2. 算法

算法是解决“做什么”和“怎么做”的问题。程序的操作语句实际上就是算法的具体描述。从事各种工作和活动, 都必须事先想好工作的步骤, 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得的操作步骤——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 $1+2+3+\cdots+100$, 有人可能先进行 1 加 2, 再加 3, 再加 4, $\cdots$, 一直加到 100, 而有的人采取这样的方法, $(1+100)+(2+99)+\cdots+(50+51)=5050$ 。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. 算法的表示方法

算法的表示方法很多, 常用的有自然语言、传统流程图、结构化流程图、伪代码 PAD 图等, 这里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。

适应于计算机的算法, 必须是有限操作序列的集合。

【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的问题:

(1) 累加的和怎样存储 —— 数据描述?

(2) 怎样使得操作步骤的描述简单?

我们采用 0 加 1, 再加 2, 再加 3, $\cdots$, 一直加到 100 的方式, 采用如下方法:

(1) 暂假设有一个可以存放累加和这种数值的存储单元, 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, 如果

1 加到 1000 或 10000 该如何表示呢? 这就类似数学中的数学归纳法, 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: $S+i$ 的值存放到 S 中, 表示为 $S+i \Rightarrow S$, 或用 VB 语言的赋值语句表示就是 $S=S+i$, 其意思是先计算 $S+i$ 的值再赋给 S 。

只要 S 中的初值为 0, i 从 1 到 100 即可解决问题。

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: 使 $S+i$ 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, 执行转到步骤 2, 否则, 继续下一步;

步骤 5: 输出 S 中的值, 计算结束。

用有限的步骤描述算法是算法的特性之一, 称为算法的有穷性; 除此之外, 算法还必须具有有效性, 即能解决实际问题; 算法必须至少有一个输出, 否则就没有意义了。

自然语言描述算法的方法虽然简单, 但并不直观, 不容易看清楚要执行的流程, 传统的流程图表示的算法较好地解决了这个问题, 下面介绍传统流程图的表示方法。

简化的流程图只需要使用 ANSI 规定的其中几个符号, 见图 1.5 所示, 就可以直观形象的将算法表示出来。例 1.1 的流程图算法表示见图 1.6 所示。

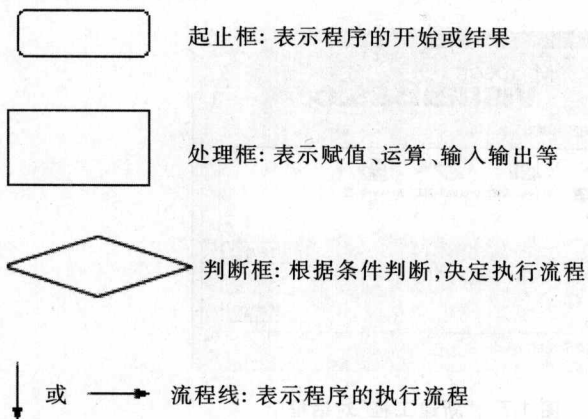


图 1.5 流程图的基本符号

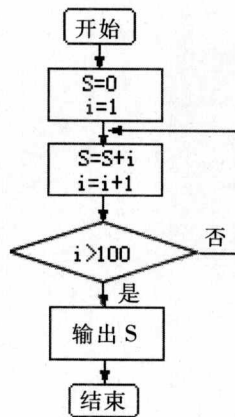


图 1.6 例 1.1 的流程图算法

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, 具体参见第 6 章。“重复是计算机核心技术”, 这个例子的关键就是重复, 是程序设计的核心基础, 也是初学者的难点。

4. 结构化程序设计方法

随着程序设计语言的发展, 程序设计的观念与方法在不断的更新和发展, 20 世纪 80 年代, 结构化程序设计的方法得到很大的发展和广泛的应用。为了解决大而复杂的问题, 结构化程序设计将问题切割成许多小问题(模块), 在程序语言中采用三种基本结构(顺序、选择和循