



面向“十二五”高等学校精品规划教材◎电气信息类
高等教育课程改革项目研究成果

Visual Basic 程序设计项目教程

主编 薛红梅 张永强

Visual Basic 程序设计项目教程

主 编	薛红梅	张永强	
副主编	申艳光	刘志敏	王彬丽
	马丽艳	王瑞林	
编 委	李震平	杜 巍	张艳丽
	杨 丽	宁振刚	范永健

 北京理工大学出版社

BEIJING INSTITUTE OF TECHNOLOGY PRESS

内 容 简 介

本书通过项目“学生管理系统”引出VB知识点,重点介绍了VB的基本知识及其应用。每章采用“项目目标—项目分析—项目实现—知识进阶—项目交流”教学五部曲的项目化教学模式,用项目引领教学内容,集成基于项目学习和探究式学习的一体化主动学习方法,强调了理论与实践相结合,以实际应用为目标,突出了对学生基本技能、实际操作能力及工程师职业能力的培养,符合学生思维的构建方式。

本书共分8章,包括软件工程的基本知识、Visual Basic 集成开发环境与基本概念、程序设计基础、用户界面设计、文件、菜单与工具栏、图像与MDI窗体、ActiveX 控件与多媒体、数据库应用等内容。

本书可作为工科院校计算机程序设计课程的教材,并可为不同层次的Visual Basic 程序设计人员学习提供参考。

版权专有 侵权必究

图书在版编目(CIP)数据

Visual Basic 程序设计项目教程 / 薛红梅, 张永强主编. —北京: 北京理工大学出版社, 2010. 12

ISBN 978 - 7 - 5640 - 3962 - 2

I. ①V… II. ①薛… ②张… 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(2010)第218952号

出版发行 / 北京理工大学出版社

社 址 / 北京市海淀区中关村南大街5号

邮 编 / 100081

电 话 / (010)68914775(办公室) 68944990(批销中心) 68911084(读者服务部)

网 址 / [http:// www. bitpress. com. cn](http://www.bitpress.com.cn)

经 销 / 全国各地新华书店

印 刷 / 北京国马印刷厂

开 本 / 787 毫米 × 1092 毫米 1/16

印 张 / 16.75

字 数 / 390 千字

版 次 / 2010 年 12 月第 1 版 2010 年 12 月第 1 次印刷

印 数 / 1 ~ 4000 册

定 价 / 32.00 元

责任校对 / 陈玉梅

责任印制 / 边心超

图书出现印装质量问题,本社负责调换

前言

“什么是工程教育的正确方法”一直以来是一个备受关注的话题。工程教育的目的是将学生培养成为“整装待发”的工程师，也就是在其从事职业前具备较好的工程能力和扎实的技术基础知识。

新型的工程教育需要回答这些问题：未来工程课程的核心内涵是什么？哪些能力的培养必须安排在课程计划中？哪些可以在毕业后的工作过程中积累？从抽象知识到实际应用之间的逻辑顺序是什么？

CDIO (Conceive — Design — Implement — Operate, 构思—设计—实施—操作/运营) 改革, 以一体化和实用方式回应了工程教育的历史和未来的挑战, 使学生知道如何在现代团队环境下构思、设计、实施及运行复杂且具有高附加值的工程产品、过程和系统。自 2000 年起, 世界范围内以 MIT 为首的几十所大学操作实施了 CDIO 模式, 迄今已取得显著成效, 深受学生欢迎, 得到产业界高度评价。

建设符合我国实际需求的适应大工程理念和 CDIO 工程新的教育模式的教材体系是本教材改革的核心内容之一, 它将有助于课程体系和教学内容更加合理和科学, 有助于学生以主动的、实践的方式学习和获取工程能力, 包括个人的科学和技术知识、终身学习能力、交流和团队工作能力, 以及在社会及企业环境下建造产品和系统的能力。

本教材特色如下:

1. 融入 CDIO 理念, 采用新的教学五部曲

本教材采用以项目实例“学生管理系统”为导向的教学模式, 集成基于项目学习和探究式学习的一体化主动学习方法。采用“项目目标—项目分析—项目实现—知识进阶—项目交流”教学五部曲的项目化教学模式, 用项目引领教学内容, 强调了理论与实践相结合, 突出了对学生基本技能、实际操作能力及工程师职业能力的培养, 符合学生思维的构建方式。通过项目设计, 激发学生学习兴趣, 培养获取知识(自主学习)、共享知识(团队合作)、运用知识(解决问题)、总结知识(技术创新)和传播知识(沟通交流)的能力与素质, 训练其职业道德修养和社会责任意识, 提高学生认知能力, 从而为学生提供真实世界的学习经验。

2. 围绕现代工程师应具备的素质要求, 多方位、多角度培养学生的工程能力

教材中利用“想想议议”“知识进阶”“项目交流”“角色模拟”“调研与分析”“能力拓展与训练”“问题卡片”等栏目多方位、多角度培养学生的工程能力, 包括终身学习能力、团队工作和交流能力、在社会及企业环境下建造产品的系统能力等。

“想想议议”是一些启发性较强、难度不太大的问题, 旨在培养学生善于观察、勤于思考、勤于讨论的良好学习习惯和品质。有一些“想想议议”问题与社会、生活和科技发展紧密联系, 旨在培养学生解决实际问题的能力。

“知识进阶”和“项目交流”中包括一些思维密度较大、思维要求较高的问题和要求, 旨在培养学生的系统思维能力、发散思维能力、创新思维能力、沟通能力、适应变化的自信和勇气以及团队协作创新的工作理念。

“思辩题”旨在培养学生的批判性和创造性思维。

“角色模拟”主要是通过模拟工程师与真实世界之间的互动,通过项目分析、设计与实现,旨在培养学生工程实践应用能力,培养学生在团队中有效合作、有效沟通、有效管理的能力,提高学生应用工程知识的能力和解决真实世界问题的能力。

“能力拓展与训练”有利于激发学生的自主探究性,在拓展创作中实现自我价值,并培养主动学习、经验学习和终身学习的能力。

“问题卡片”旨在培养学生主动学习、自主学习的能力和积极的态度。

3. 贴近学生生活,倡导“快乐学习”理念

本教材精选贴近学生生活具有趣味性和实用性的项目实例“学生管理系统”,按照教学规律和学生的认知特点将知识点融于项目实例中。

4. 强调人文素质培养

每章后附有“你我共勉”,旨在培养学生良好的学习态度和职业道德。

总之,本书在适度的基础知识与理论体系覆盖下,突出了工程教育的教学方法论,主要特色在于采用创新教学方法和学习环境为学生提供真实世界的学习经验,力求达到 CDIO 改革的总体目标,使学生更深入地掌握技术基础知识、领导新产品、过程和系统的建造与运行,理解研究和技术发展对社会的重要性和战略影响。

本教材共分为 8 章,内容包括程序设计基础、用户界面设计、文件、菜单与工具栏、图形操作、ActiveX 控件与多媒体、数据库。

本教材由薛红梅、张永强任主编,申艳光、刘志敏、王彬丽、马丽艳、王瑞林任副主编,统稿工作由薛红梅、刘志敏和王彬丽完成。各章编写分工为:第 1 章由张永强编写,第 2 章由薛红梅、王瑞林、范永健编写,第 3 章由刘志敏编写,第 4 章由申艳光编写,第 5 章由李震平、宁振刚编写,第 6 章由张艳丽、杨丽编写,第 7 章由马丽艳编写,第 8 章由王彬丽编写,附录由杜巍编写。在编写过程中得到了河北工程大学领导和教师们的大力支持,在此表示深深的敬意和感谢。

限于作者的水平及时间的仓促,加之对 CDIO 理念的研究尚处初探阶段,书中难免存在不足之处,恳请读者批评和指正,以使其更臻完善!

本书有配套的《Visual Basic 程序设计实验实训》,同时提供电子课件和项目素材,可以发邮件至 xind-jj@hebeu.edu.cn 索取或登录 www.bitpress.com.cn 下载。

编 者

目 录

第1章 绪论	1
1.1 软件及软件工程	1
1.1.1 什么是软件	1
1.1.2 软件工程与软件危机	1
1.1.3 软件工程过程与软件生命周期	2
1.1.4 软件项目开发流程	3
1.1.5 软件开发文档	3
1.2 算法和程序设计	4
1.2.1 算法	4
1.2.2 程序设计	8
1.2.3 程序设计方法	8
1.3 学生管理项目介绍	9
1.3.1 系统分析	9
1.3.2 需求分析	10
1.4 软件开发工具的选择	11
1.4.1 影响开发工具选择的因素	11
1.4.2 Visual Basic 6.0 简介	12
1.5 认识集成开发环境	14
项目交流	19
基本知识练习	20
能力拓展与训练	20
第2章 程序设计基础	22
2.1 项目一 系统登录界面和主界面	22
2.1.1 项目目标	22
2.1.2 项目分析	22
2.1.3 项目实施	23
2.1.4 相关知识	31
2.2 项目二 学生成绩输入	42
2.2.1 项目目标	42
2.2.2 项目分析	43
2.2.3 项目实施	43
2.2.4 相关知识	45
2.3 项目三 学生成绩评定	57
2.3.1 项目目标	57

2.3.2 项目分析	58
2.3.3 项目实施	58
2.3.4 相关知识	59
2.4 项目四 学生成绩统计	65
2.4.1 项目目标	65
2.4.2 项目分析	65
2.4.3 项目实施	66
2.4.4 相关知识	66
2.5 项目五 学生成绩统计分析	72
2.5.1 项目目标	72
2.5.2 项目分析	72
2.5.3 项目实施	72
2.5.4 相关知识	74
2.6 项目六 小助手——简易计算器	80
2.6.1 项目目标	80
2.6.2 项目分析	81
2.6.3 项目实施	81
2.6.4 相关知识	84
2.7 项目七 小助手——组队方法	93
2.7.1 项目目标	93
2.7.2 项目分析	93
2.7.3 项目实施	93
2.7.4 相关知识	94
2.8 知识进阶	100
2.8.1 编码约定	100
2.8.2 结构化编码	102
项目交流	103
基本知识练习	104
能力拓展与训练	105
第3章 用户界面设计	107
3.1 项目 档案管理之信息录入	107
3.1.1 项目目标	107
3.1.2 项目分析	107
3.1.3 项目实施	108
3.1.4 相关知识	111
3.2 知识进阶	117
3.2.1 界面所涉及的元素	117
3.2.2 界面属性	117
3.2.3 统一管理 VB 控件的界面属性	118

项目交流	119
基本知识练习	119
能力拓展与训练	120
第4章 文件	123
4.1 项目 档案管理之信息存储	123
4.1.1 项目目标	123
4.1.2 项目分析	123
4.1.3 项目实施	124
4.1.4 相关知识	126
4.2 知识进阶	139
4.2.1 常用的文件操作语句、函数和属性	139
4.2.2 使用 FSO 对象模型操作文件	141
项目交流	144
基本知识练习	144
能力拓展与训练	145
第5章 菜单与工具栏	147
5.1 项目一 菜单设计	147
5.1.1 项目目标	147
5.1.2 项目分析	147
5.1.3 项目实施	148
5.1.4 相关知识	151
5.2 项目二 工具栏设计	155
5.2.1 项目目标	155
5.2.2 项目分析	156
5.2.3 项目实施	156
5.2.4 相关知识	158
5.3 知识进阶	159
项目交流	161
基本知识练习	161
能力拓展与训练	161
第6章 图形操作与多文档窗体	163
6.1 项目 休闲娱乐——绘图板	163
6.1.1 项目目标	163
6.1.2 项目分析	163
6.1.3 项目实施	164
6.1.4 相关知识	168
6.2 项目 休闲娱乐——电子相册	179
6.2.1 项目目标	179
6.2.2 项目分析	180

6.2.3 项目实施	180
6.2.4 相关知识	180
6.3 知识进阶	184
项目交流	188
基本知识练习	188
能力拓展与训练	189
第7章 AcitveX 控件与多媒体	191
7.1 项目 休闲娱乐——多媒体播放器	191
7.1.1 项目目标	191
7.1.2 项目分析	191
7.1.3 项目实施	192
7.1.4 相关知识	196
7.2 知识进阶	202
项目交流	203
基本知识练习	204
能力拓展与训练	204
第8章 数据库的应用	206
8.1 项目 学生管理数据库设计	206
8.1.1 项目目标	206
8.1.2 项目分析	207
8.1.3 项目实施	208
8.1.4 相关知识	213
8.2 知识进阶	234
项目交流	236
基本知识练习	237
能力拓展与训练	237
附录一 常用算法综合举例	239
附录二 错误调试	248
附录三 软件职业道德规范	256
附录四 VB 常用术语释义	257
附录五 ASCII 码表	258
参考文献	259

第1章 绪 论

本章要点：

- 什么是软件及软件工程？
- 什么是软件危机？
- 软件项目开发流程有哪些步骤？
- 什么是算法和程序设计？

1.1 软件及软件工程

1.1.1 什么是软件

根据国际化标准组织的定义，软件是与计算机系统操作有关的程序、过程、规则，以及任何有关的文档资料和数据。程序是计算机可以执行的程序以及与程序有关的数据，文档是用来描述、使用和维护程序及数据所需要的图文资料。

软件系统按功能可将软件划分为系统软件和应用软件。

1. 系统软件

系统软件是计算机系统的底层管理软件，它与计算机硬件紧密配合，管理与硬件相关的数据输入、处理和输出，使计算机系统的各个部分协调、高效地工作，如操作系统、数据库管理系统等。

2. 应用软件

应用软件是为解决某种专门问题而设计的软件。它包括应用软件包，以及为解决科研及生产中的实际问题而由用户设计的应用软件，如文字处理软件、CAD 软件、城市交通监管系统、生产设备的自动控制系统软件等。

想想议议：

到现在为止，我们学过和使用过哪类软件？

1.1.2 软件工程与软件危机

软件工程概念的出现源自软件危机。

在 20 世纪 60~70 年代，出现了软件危机。所谓软件危机，是指在软件开发和维护过程中所遇到的一系列严重问题。

具体地说，在软件开发维护过程中，软件危机主要表现在以下几方面。

(1) 主观盲目地制订软件开发计划，对工作量估计不足，进度计划无法遵循，使得开发工作的完成时间一再拖延，经费预算经常超支。

(2) 不重视软件测试工作，提交给用户的软件质量差，在运行中暴露出大量的问题。在

应用领域工作的不可靠软件，轻者影响系统的正常工作，重者发生事故，甚至造成生命财产的重大损失。

(3) 开发过程没有统一的规范化方法和管理流程，设计和实现过程的文档资料不完整，开发人员各行其是，互相之间接口的统一问题常常被忽视，软件结构不清晰。这些都使得软件难以维护。

为了消除软件危机，通过认真研究解决软件危机的方法，认识到软件工程是使计算机软件走向工程科学的途径，逐步形成了软件工程的观念，开辟了工程学新兴领域——软件工程学。

所谓软件工程是应用于计算机软件的定义、开发和维护的一整套方法、文档、实践标准和工序。软件工程包括 3 个要素，即方法、工具和过程。方法是完成软件工程项目的技术；工具支持软件的开发、管理、文档生产；过程支持软件开发的各个环节的控制、管理。

软件工程的中心思想是把软件产品看做一个工程产品来处理。把需求计划、可行性研究、工程审核、质量监督等工程化的概念引入软件生产当中，以使其满足工程项目的 3 个基本要素：进度、经费和质量。

1.1.3 软件工程过程与软件生命周期

1. 软件工程过程

ISO 9000 定义：软件工程过程是把输入转化为输出的一组彼此相关的资源和活动。

定义支持了软件工程过程的两方面内涵：其一，软件工程过程是指为获得软件产品，在软件工具支持下由软件工程师完成的一系列软件工程活动，包括软件规格说明、软件开发、软件确认、软件演进 4 种基本活动；其二，从软件开发的观点看，软件工程过程是使用适当的资源（包括人员、硬软件工具、时间等），为开发软件进行的一组开发活动，在过程结束时将输入（用户要求）转换为输出（软件产品）。

2. 软件生命周期

通常，将软件产品从提出、实现、使用、维护到停止使用的过程称为软件生命周期。也就是说，软件生命周期是指一个软件从提出开发要求到该软件停止使用的整个时期。一般包括可行性研究与需求分析、设计、实现、测试、交付使用以及维护等活动，这些活动可以有重复，执行时也可以迭代。

(1) 可行性研究与计划制订。确定待开发软件系统的开发目标和总的要求，给出它的功能、性能、可靠性以及接口等方面的可能方案，制订完成开发任务的实施计划。

(2) 需求分析。对开发软件提出的需求进行分析并给出详细定义。编写软件规格说明书及初步的用户手册，提出评审。

(3) 软件设计。系统设计人员和程序设计人员应该在反复理解软件需求的基础上，给出软件的结构、模块的划分、功能的分配以及处理流程。

(4) 软件实现。把软件设计转换成计算机可以接受的程序代码。

(5) 软件测试。在设计测试用例的基础上，检验软件的各个组成部分，编写测试分析报告。

(6) 运行和维护。将已交付的软件投入运行，并在运行使用中不断地维护，根据新提出

的需求进行必要的扩充和删改。

1.1.4 软件项目开发流程

软件项目其实是一个逐步演绎的过程，包括启动、计划、实施和交付4个阶段。在每一个阶段中，设计的业务有需求分析、设计、编码、测试、交付。在每一个业务处理中，项目的担当者还需要对项目进行进度管理、质量管理、成本管理、团队管理和风险管理。

1. 项目启动

系统分析员作为客户与项目团队之间的桥梁，应该和客户进行很好的沟通，了解业务，为接下来的系统设计做好业务基础。一般采取的方法是到客户那里进行实地问卷，考察交流。当系统分析员向客户描绘系统应该实现的功能与客户达成共识后，才进入系统的设计阶段。进入设计阶段，系统分析员不能够只为了实现业务而随意地设置系统构件，这个时候不但要考虑系统的功能，还要考虑系统的性能和系统的扩展性。当系统的业务要求和性能要求满足客户的需求后，进入下一个阶段，如果不符合，则继续进行这一阶段。

2. 项目计划

在这个阶段，应该做的是完成项目进度表，人员的组建，系统环境的设置，还有项目的风险分析，开发采用的语言，代码的编码规约。这些基本上可以通过系统设计图纸所描述的系统架构来设置。

3. 项目实施

在这个阶段，编码与测试是主要的任务。程序员编写系统设计图纸中构件的具体实现代码。编写出来的代码应符合编码规约中的要求。为了防止错误，程序员之间可以互相检查编写出来的代码。好的编码方式是采用测试驱动开发的方法。编写完代码后，程序员还应该自己进行测试，测试通过后才能够提交。为了跟踪项目进度情况，应该在每天结束工作以前开会，在会议上登记当日工作的完成进度，登记遇到的问题，并且在会议上进行解决。

4. 项目交付

大的项目交付一般采用的是分期交付。当完成某一个模块后就进行交付。在这个阶段交付的项目应该按照需求分析上面罗列的清单进行交付，交付的项目一般包括用户使用说明书、软件代码和编译后可运行的系统。

想想议议：

与用户沟通获取需求的方法有哪些？

1.1.5 软件开发文档

软件开发文档是软件开发人员、管理人员、维护人员、用户和计算机之间的桥梁。文档编写是软件开发过程中的一项重要工作，在软件开发工作中占有突出的地位和相当大的工作量，没有文档的软件，不能称为软件产品。

软件开发文档可以分为系统文档和用户文档两大类。系统文档描述的是系统设计、实现和测试等各方面的内容；用户文档则是对系统的功能及使用方法的描述，但并不涉及这些功能的实现过程。

1. 系统文档

系统文档是指从问题定义、需求说明到验收测试计划这样一系列与系统实现有关的文档，它对于理解程序和维护程序来说是极为重要的。

系统文档通常包括可行性研究报告、项目开发计划、软件需求说明书、总体设计说明书、详细设计说明书、开发进度表、测试计划、测试报告和项目开发总结报告等。

2. 用户文档

用户文档详细地描述了软件的功能、性能和用户界面，以及如何使用软件等具体细节，它是帮助用户了解系统的“窗口”。

用户文档至少应包括功能描述、安装文档、使用手册、操作指南。

想想议议：

作为软件开发人员应具备什么样的素质和能力？

1.2 算法和程序设计

1.2.1 算法

1. 算法

所谓算法是指对解题方案准确而完整的描述。

对于一个问题，如果可以通过一个计算机程序，在有限的存储空间内运行有限长的时间而得到正确的结果，则称这个问题是算法可解的。但算法不等于程序，也不等于计算方法。一个算法的优劣可以用空间复杂度与时间复杂度来衡量。

想想议议：

求自然数 $1+2+\cdots+100$ 的和，解决这个问题有多少种算法，并比较这些算法的优劣？

2. 算法的基本特征

1) 可行性

针对实际问题设计的算法，人们总是希望能够得到满意的结果。但一个算法又总是在某个特定的计算工具上执行的，因此，算法在执行过程中往往要受到计算工具的限制，使执行结果产生偏差。

2) 确定性

算法的确定性，是指算法中的每一个步骤都必须是有明确定义的，不允许有模棱两可的解释，也不允许有多义性。

3) 有穷性

算法的有穷性，是指算法必须能在有限的时间内做完，即算法必须能在执行有限个步骤之后终止。算法的有穷性还应包括合理的执行时间的含义。因为如果一个算法需要执行千万年，显然失去了实用性。

4) 拥有足够的情报

一个算法是否有效，还取决于为算法所提供的信息是否足够。通常，算法中的各种运算

总是要施加到各个运算对象上，而这些运算对象又可能具有某种初始状态，这是算法执行的起点或是依据。因此，一个算法执行的结果总是与输入的初始数据有关，不同的输入将会有不同的结果输出。当输入不够或输入错误时，算法本身也就无法执行或导致执行有错。一般来说，当算法拥有足够的情报时，此算法才是有效的，而当提供的情报不够时，算法可能无效。

3. 算法的基本要素

一个算法通常由两种基本要素组成：一是对数据对象的运算和操作；二是算法的控制结构。

1) 算法中对数据的运算和操作

每个算法实际上是按照解题要求选择合适的操作所组成的一组指令序列。因此，计算机算法就是计算机能处理的操作所组成的指令序列。

通常，计算机可以执行的基本操作是以指令的形式描述的。一个计算机系统能执行的所有指令的集合，称为该计算机系统的指令系统。计算机程序就是按解题要求从计算机指令系统中选择合适的指令所组成的指令序列。在一般的计算机系统中，基本的运算和操作有以下四类：

- 算术运算主要包括加、减、乘、除等运算；
- 逻辑运算主要包括“与”“或”“非”等运算；
- 关系运算主要包括“大于”“小于”“等于”“不等于”等运算；
- 数据传输主要包括赋值、输入、输出等操作。

2) 算法的控制结构

一个算法的功能不仅取决于所选用的操作，而且还与各操作之间的执行顺序有关。算法中各操作之间的执行顺序称为算法的控制结构。

算法的控制结构给出了算法的基本框架，它不仅决定了算法中各操作的执行顺序，而且也直接反映了算法的设计是否符合结构化原则。描述算法的工具通常有传统流程图、N-S 结构化流程图、算法描述语言等。一个算法一般可以用顺序、选择、循环 3 种基本控制结构组合而成。

4. 算法设计基本方法

计算机解题的过程实际上是在实施某种算法，这种算法称为计算机算法。计算机算法不同于人工处理的方法。以下是工程上常用的几种算法设计方法，在实际应用时，各种方法之间往往存在着一定的联系。

1) 列举法

列举法的基本思想是，根据提出的问题列举所有可能的情况，并用问题中给定的条件检验哪些是需要的，哪些是不需要的。

2) 归纳法

归纳法的基本思想是，通过列举少量的特殊情况，经过分析，最后找出一般的关系。

3) 递推法

所谓递推，是指从已知的初始条件出发，逐次推出所要求的各中间结果和最后结果。其中初始条件或是问题本身已经给定，或是通过对问题的分析与化简而确定。

4) 递归法

人们在解决一些复杂问题时,为了降低问题的复杂程度(如问题的规模等),一般总是将问题逐层分解,最后归结为一些最简单的问题。这种将问题逐层分解的过程,实际上并没有对问题进行求解,而只是当解决了最后那些最简单的问题后,再沿着原来分解的逆过程逐步进行综合,这就是递归的基本思想。

5) 减半递推法

实际问题的复杂程度往往与问题的规模有着密切的联系。因此,利用分治法解决这类实际问题是有意义的。所谓分治法,就是对问题分而治之。工程上常用的分治法是减半递推技术。所谓“减半”,是指将问题的规模减半,而问题的性质不变;所谓“递推”,是指重复“减半”的过程。

5. 算法分类

计算机算法分为数值运算和非数值运算两大类。数值运算的目的是求数值解,如求方程的根、求定积分等;非数值运算包括的范围较广,如人事管理、图书检索等。由于数值运算有现成的模型,可以运用数值分析方法,因此人们对数值运算的算法研究比较深入,有许多较成熟的算法可供选用,常常把这些算法汇编成册(写成程序形式),或将这些程序存放在磁盘或光盘等存储介质中,供用户调用。因此计算机的算法研究主要对一些典型的非数值运算算法进行较深入的研究。

6. 算法描述

描述算法有多种工具,自然语言、传统流程图、N-S流程图、判定表、判定树、伪码等。下面介绍几种常见的算法描述方法。

1) 自然语言

用自然语言表示算法,通俗易懂,特别适用于对顺序结构算法的描述。

例如: $s=1+2+3+4+\cdots+N$ 求和问题使用自然语言描述如下。

- (1) 输入 N 的值。
- (2) 设 i 的值为 1; s 的值为 0。
- (3) 如果 $i \leq N$, 则执行 (4), 否则转到 (7) 执行。
- (4) 计算 $s+i$, 并将结果赋给 s 。
- (5) 计算 $i+1$, 并将结果赋给 i 。
- (6) 重新返回到 (3) 开始执行。
- (7) 输出 s 的结果。

使用自然语言描述算法的不足之处表现在下面几个方面:

- 易出现二义性;
- 难以清晰表达出分支、循环结构;
- 其描述的算法冗长。

2) 传统流程图

传统流程图四框一线,符合人们的思维习惯,用它表示算法,直观形象,易于理解。常用的框图符号如图 1-1 所示。

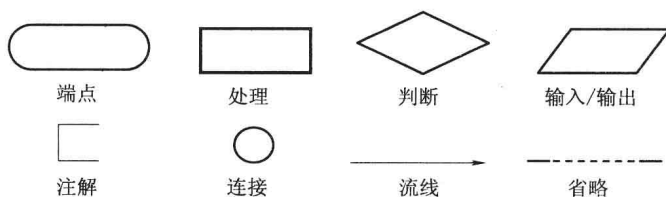


图 1-1 传统流程图常用符号

例如， $s=1+2+3+4+\dots+N$ 求和问题使用流程图描述如图 1-2 所示。

流程图采用简单规范的符号，直观形象、结构清晰地显示算法各步骤之间的逻辑关系，使得算法容易理解。但是，当算法比较复杂时，占用篇幅较多。另外，由于流程图中流程线的使用没有严格限制，可以使流程随意地转来转去，使流程图变得毫无规律，阅读时难以理解算法的逻辑，从而使算法的可靠性和可维护性难以得到保证。

3) N-S 图

1973 年美国学者提出了一种新型流程图：N-S 流程图。这种流程图描述顺序结构如图 1-3 (a) 所示，选择结构如图 1-3 (b) 所示，当型循环结构如图 1-3 (c)、直到型循环结构如图 1-3 (d) 所示。

N-S 流程图比较容易描述较复杂的选择结构和循环结构。

(1) 顺序结构：程序执行完 A 语句后接着执行 B 语句。

(2) 选择结构：当条件 P 成立时，则执行 A 语句，否则执行 B 语句。

(3) 当型循环结构：当条件 P 成立时，则循环执行 A 语句。

(4) 直到型循环结构：循环执行 A 语句，直到条件 P 成立为止。

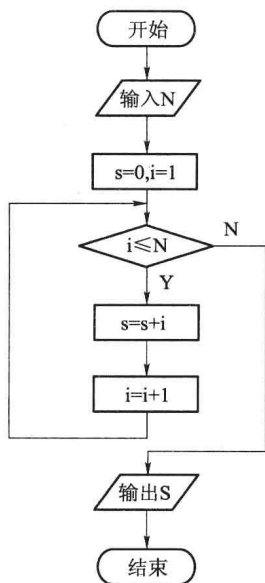


图 1-2 流程图算法描述

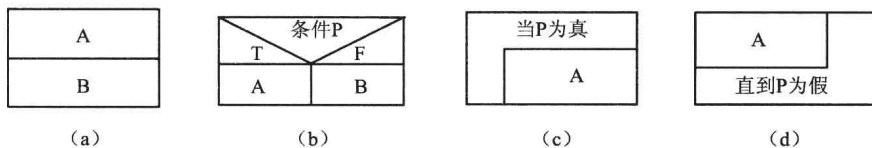


图 1-3 N-S 图控制结构描述

(a) 顺序结构；(b) 选择结构；(c) 当型循环；(d) 直到型循环

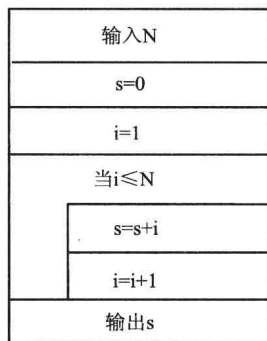


图 1-4 N-S 图算法描述

例如， $s=1+2+3+4+\dots+N$ 求和问题使用 N-S 图描述如图 1-4 所示。

7. 算法分析

对算法的分析主要是对算法的时间复杂度和空间复杂度的分析，以求改进算法的效率。

1) 时间复杂度

一个算法的时间复杂度是该算法的时间耗费，即算法执行过程中所需要的基本运算次数。

一个算法所消耗的时间等于算法中每条语句的执行时间之和。

每条语句的执行时间等于语句的执行次数乘以语句执行一次所需时间。算法转换为程序后，每条语句执行一次所需的时间取决于机器的指令性能、速度以及编译所产生的代码质量等因素。

2) 空间复杂度

一个算法的空间复杂度定义为该算法在执行过程中所需要的存储空间。

算法的时间复杂度和空间复杂度合称为算法的复杂度。

1.2.2 程序设计

一个算法若用计算机语言来书写，则它就是一个程序。计算机程序设计是为计算机规划、安排解题步骤的过程，一般来说由以下 4 个步骤组成。

(1) 分析问题：在着手解决问题之前，要通过分析来充分理解问题，明确原始数据、解题要求、需要输出的数据及形式等。

(2) 设计算法：首先进行算法的总体规划，然后逐层降低问题的抽象性，逐步充实其细节，直到最终把抽象的问题具体转化为算法。

(3) 编码：用计算机语言表示算法的过程称为编码。程序是用计算机语言编码的解题算法。

(4) 调试程序：调试过程包括编译和连接等。编译程序对源程序进行语法检查，程序员根据编译错误信息的提示，查找并改正错误后再编译，直到没有语法错误为止，编译将源程序转换成目标程序。大多数语言还要用连接程序把目标程序与库文件连接成可执行文件。在连接过程中若程序使用了错误的内部函数名，则会引起连接错误。对运行顺利结束并得到结果的程序，还要对所得结果进行分析。只有得到正确结果的程序才是正确的程序。

综上所述，程序设计就是分析问题、设计算法、编码、调试与测试的过程。

注意：在程序设计过程中算法是程序设计的灵魂。

1.2.3 程序设计方法

1. 早期的程序设计

从第一台计算机问世至今，计算机硬件技术得到了飞速发展，相应地，软件开发工具经历了机器语言、汇编语言、高级语言到面向对象程序设计语言等阶段，程序设计方法也经历了早期的手工作坊式的程序设计、面向过程的结构化程序设计到面向对象的可视化程序设计等阶段。

20 世纪 50 年代，人们用机器语言编写代码，程序设计工作十分繁重，但后期逐渐被使用符号指令的汇编程序设计所代替。汇编语言是为特定计算机或计算机系列设计的。汇编语言程序比机器语言程序易读、易检查、易修改，同时又保持了机器语言执行速度快、占用存储空间少的优点。汇编语言也是面向机器的一种低级语言，不具备通用性和可移植性。1954 年提出了第一个高级语言——FORTRAN 语言，大大简化了程序设计，高级语言是由各种意义的词和数学公式按照一定的语法规则组成的，它更容易阅读、理解和修改，编程效率高。高级语言不是面向机器的，而是面向问题的，与具体机器无关，具有很强的通用性和可移植性。