

21世纪高等学校规划教材 | 计算机科学与技术

程序设计与算法语言 —— C++程序设计基础

孔丽英 夏艳 徐勇 编著



清华大学出版社

内容简介

本书是“十二五”国家重点出版项目“计算机科学与技术”系列教材之一。本书以C++语言为工具，以程序设计基础为内容，以算法设计为灵魂，以应用为目的，力求做到概念清晰、重点突出、循序渐进、由浅入深、由易到难、由点到面、由理论到应用。本书可作为高等院校计算机专业及相关专业的教材，也可供从事计算机工作的工程技术人员参考。



程序设计与算法语言

—— C++程序设计基础

孔丽英 夏艳 徐勇 编著

清华大学出版社
北京

内 容 简 介

本书以程序设计为主线,通过案例教学引入数学模型的建立和算法的设计,并且详细地分析程序,从而达到培养学生分析程序和设计程序的能力。全书共分9章,第1章介绍利用计算机求解问题的步骤和算法设计以及计算机程序和C/C++语言;第2~7章是面向过程程序设计的基础,介绍数据类型和表达式、程序结构、控制结构程序设计、函数、构造数据类型和指针;第8章是面向对象程序设计的基础,介绍类和对象、构造函数、析构函数、对象指针、静态成员、友元、继承和多态性;第9章介绍文件、流类库以及通过文件流操作文件和输入/输出格式控制。

本书可作为大学本专科程序设计课程的教材,也可供广大读者自学参考。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

程序设计与算法语言: C++程序设计基础/孔丽英,夏艳,徐勇编著.--北京:清华大学出版社,2014
21世纪高等学校规划教材·计算机科学与技术
ISBN 978-7-302-36696-6

I. ①程… II. ①孔… ②夏… ③徐… III. ①C语言—程序设计—高等学校—教材 ②算法语言—程序设计—高等学校—教材 IV. ①TP312

中国版本图书馆CIP数据核字(2014)第117189号

责任编辑:刘向威 王冰飞

封面设计:傅瑞学

责任校对:时翠兰

责任印制:李红英

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦A座 邮 编:100084

社 总 机:010-62770175 邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

课 件 下 载: <http://www.tup.com.cn>, 010-62795954

印 装 者:北京国马印刷厂

经 销:全国新华书店

开 本:185mm×260mm

印 张:21.5

字 数:521千字

版 次:2014年11月第1版

印 次:2014年11月第1次印刷

印 数:1~2000

定 价:39.00元

产品编号:058321-01

出版说明

随着我国改革开放的进一步深化,高等教育也得到了快速发展,各地高校紧密结合地方经济建设发展需要,科学运用市场调节机制,加大了使用信息科学等现代科学技术提升、改造传统学科专业的投入力度,通过教育改革合理调整和配置了教育资源,优化了传统学科专业,积极为地方经济建设输送人才,为我国经济社会的快速、健康和可持续发展以及高等教育自身的改革发展做出了巨大贡献。但是,高等教育质量还需要进一步提高以适应经济社会发展的需要,不少高校的专业设置和结构不尽合理,教师队伍整体素质亟待提高,人才培养模式、教学内容和教学方法需要进一步转变,学生的实践能力和创新精神亟待加强。

教育部一直十分重视高等教育质量工作。2007年1月,教育部下发了《关于实施高等学校本科教学质量与教学改革工程的意见》,计划实施“高等学校本科教学质量与教学改革工程”(简称“质量工程”),通过专业结构调整、课程教材建设、实践教学改革、教学团队建设等多项内容,进一步深化高等学校教学改革,提高人才培养的能力和水平,更好地满足经济社会发展对高素质人才的需要。在贯彻和落实教育部“质量工程”的过程中,各地高校发挥师资力量强、办学经验丰富、教学资源充裕等优势,对其特色专业及特色课程(群)加以规划、整理和总结,更新教学内容、改革课程体系,建设了一大批内容新、体系新、方法新、手段新的特色课程。在此基础上,经教育部相关教学指导委员会专家的指导和建议,清华大学出版社在多个领域精选各高校的特色课程,分别规划出版系列教材,以配合“质量工程”的实施,满足各高校教学质量和教学改革的需要。

为了深入贯彻落实教育部《关于加强高等学校本科教学工作,提高教学质量的若干意见》精神,紧密配合教育部已经启动的“高等学校教学质量与教学改革工程精品课程建设工作”,在有关专家、教授的倡议和有关部门的大力支持下,我们组织并成立了“清华大学出版社教材编审委员会”(以下简称“编委会”),旨在配合教育部制定精品课程教材的出版规划,讨论并实施精品课程教材的编写与出版工作。“编委会”成员皆来自全国各类高等学校教学与科研第一线的骨干教师,其中许多教师为各校相关院、系主管教学的院长或系主任。

按照教育部的要求,“编委会”一致认为,精品课程的建设工作从开始就要坚持高标准、严要求,处于一个比较高的起点上。精品课程教材应该能够反映各高校教学改革与课程建设的需要,要有特色风格、有创新性(新体系、新内容、新手段、新思路,教材的内容体系有较高的科学创新、技术创新和理念创新的含量)、先进性(对原有的学科体系有实质性的改革和发展,顺应并符合21世纪教学发展的规律,代表并引领课程发展的趋势和方向)、示范性(教材所体现的课程体系具有较广泛的辐射性和示范性)和一定的前瞻性。教材由个人申报或各校推荐(通过所在高校的“编委会”成员推荐),经“编委会”认真评审,最后由清华大学出版

社审定出版。

目前,针对计算机类和电子信息类相关专业成立了两个“编委会”,即“清华大学出版社计算机教材编审委员会”和“清华大学出版社电子信息教材编审委员会”。推出的特色精品教材包括:

- (1) 21世纪高等学校规划教材·计算机应用——高等学校各类专业,特别是非计算机专业的计算机应用类教材。
- (2) 21世纪高等学校规划教材·计算机科学与技术——高等学校计算机相关专业的教材。
- (3) 21世纪高等学校规划教材·电子信息——高等学校电子信息相关专业的教材。
- (4) 21世纪高等学校规划教材·软件工程——高等学校软件工程相关专业的教材。
- (5) 21世纪高等学校规划教材·信息管理与信息系统。
- (6) 21世纪高等学校规划教材·财经管理与应用。
- (7) 21世纪高等学校规划教材·电子商务。
- (8) 21世纪高等学校规划教材·物联网。

清华大学出版社经过三十多年的努力,在教材尤其是计算机和电子信息类专业教材出版方面树立了权威品牌,为我国的高等教育事业做出了重要贡献。清华版教材形成了技术准确、内容严谨的独特风格,这种风格将延续并反映在特色精品教材的建设中。

清华大学出版社教材编审委员会

联系人:魏江江

E-mail: weijj@tp.tsinghua.edu.cn

前言

程序设计与算法语言是高等学校重要的专业基础课,它以编程语言为平台,介绍程序设计的思想和方法。通过该课程的学习,学生不仅要掌握程序设计与算法语言的理论知识,更为重要的是要掌握算法设计与程序设计的思路、方法,通过大量的练习,培养学生解决问题和编程的能力,熟悉上机的全过程及调试程序的基本方法与技巧,使学生能够利用所学知识解决一些科学计算及实际问题。

C++语言是使用最为广泛的程序设计与算法语言之一,它全面兼容C语言,全面支持面向对象程序设计,具有全面支持面向过程和面向对象的混合编程等特点,能够充分发挥两类编程技术的优势。C++语言不仅是一门基础课,而且是学习数据结构、操作系统等后续课程的重要基础。

目前,C++语言的教材很多,但大部分教材主要通过案例教学讲解如何用语言知识设计程序,很少讲述设计程序的思路,并缺少对程序的分析,造成大部分学生不理解程序的运行过程,更不懂得如何设计程序,使学生为了应试而背记程序,结果学生虽然通过了计算机水平考试,但设计程序的能力较低,思路单一,开拓创新不足,造成在后续课程学习中遇到了很大的困难。

根据多年来教学经验的积累,我们清楚地知道学生设计程序的能力需要大幅度提升,思路需要拓宽。学生在学习过程中忽略了算法设计,从而造成程序设计的能力差,难以胜任以后的工作。因此,我们认为C++程序设计的教材一定要以程序设计为主线,以案例为驱动,通过案例教学引入数学模型和算法设计以及分析程序的过程,使教材突出C++语言的特性,最终实现提高学生分析程序和设计程序能力的目标。另外,考虑到目前大部分高校的学生都参加计算机水平考试,作者根据多年的教学经验,结合计算机水平考试Ⅱ级的《C++程序设计》考试大纲的要求,在本书中的相关章节专门按考试大纲的题型编写了相应的习题与实验内容,让学生的学习更有针对性,从而达到事半功倍的效果,力求从根本上提高学生的计算机水平考试的通过率。

目前,C++语言的教材虽然很多,但有针对性地适应第二批高等院校的本科生的教材并不多,这类学生急需一本理论性不太强,但有实际操作性的C++语言教材,本书在编写上能满足这类学生的要求。

本书主要具有如下7个特点。

- (1) 内容组织符合学习规律:内容由简单到复杂,衔接紧密,使学生由浅入深地学习。
- (2) 内容精简:根据计算机水平考试内容、面授的对象、授课时间的安排以及后继课程的需要,着重在内容上与传统的教材相比进行了取舍,使学生易于掌握知识。
- (3) 分散难点:对于一些难学的知识点,分散在与其更密切关联的知识中,使学生易于学习。
- (4) 程序分析透彻:详细讲解程序运行和变量值的变化,使学生易于理解程序。

(5) 以案例培养设计程序的思维：通过案例教学，引入数学模型的建立和算法的设计，使学生易于理解设计程序的思路与方法。

(6) 习题多样化：每一章节的习题都有多种类型，而且有参考计算机水平考试的题型，使学生的学习更有针对性，促进学生考级能力的提高。

(7) 实验题设计科学：学习 C++ 语言不仅要有扎实的基础，更重要的是实践，本书中设计了大量的实验，突出重点知识，结构合理。

全书以程序设计为主线，通过案例教学引入数学模型的建立和算法的设计，详细地讲解程序运行的过程和变量值的变化，最终达到培养学生设计程序和分析程序能力的目标。

本书共有 9 章，可分为 4 个部分：

第一部分为第 1 章，程序设计概述。该部分介绍计算机求解问题的步骤和算法设计、计算机程序，并简要介绍 C/C++ 语言。

第二部分为第 2~7 章，面向过程程序设计基础。该部分介绍数据类型和表达式、程序结构、控制结构程序设计、函数、构造数据类型和指针。

第三部分为第 8 章，面向对象程序设计基础。该部分介绍类和对象、构造函数和析构函数、对象指针、静态成员、友元、继承和多态性。

第四部分为第 9 章，文件和流。该部分介绍文件、流类库、通过文件流（或指针）操作文件、输入/输出格式控制。

本书得到了数学与统计学院领导的支持，在此表示衷心的感谢。本书是作者根据多年的教学实践经验编写的，第 1~7 章由孔丽英编写，第 8~9 章由夏艳编写，徐勇对全书内容进行核查。本书可作为大学本专科程序设计课程的教材，也可供广大读者自学参考。由于作者水平有限，书中可能存在缺点或错误，敬请广大读者批评指正。

作者

2014 年 10 月

目 录

第 1 章 程序设计概述	1
1.1 计算机求解问题的步骤和算法	1
1.1.1 计算机求解问题的步骤.....	1
1.1.2 算法.....	2
1.2 计算机程序	6
1.2.1 程序设计语言.....	6
1.2.2 编译与解释.....	7
1.2.3 程序设计方法.....	7
1.3 C/C++语言简介	8
1.3.1 C语言简介.....	8
1.3.2 C++语言简介	8
习题 1	9
第 2 章 数据类型和表达式	10
2.1 基本数据类型.....	10
2.1.1 整数类型	10
2.1.2 实数类型	10
2.1.3 字符类型	11
2.1.4 布尔类型	11
2.1.5 空类型	11
2.2 C++的字符集	11
2.2.1 字符集	11
2.2.2 标识符	12
2.3 常量与符号常量.....	12
2.3.1 值常量	12
2.3.2 符号常量	15
2.4 变量与常变量.....	15
2.4.1 变量	15
2.4.2 常变量	17
2.5 表达式.....	17
2.5.1 运算符	18
2.5.2 表达式的运算规则	25

2.6 类型的转换	28
习题 2	29
第 3 章 程序结构	31
3.1 简单语句	31
3.1.1 表达式语句	31
3.1.2 空语句	31
3.1.3 复合语句	31
3.2 预处理命令	32
3.2.1 “文件包含”命令	32
3.2.2 宏定义	33
3.3 数据的输入/输出	34
3.3.1 标准输入/输出函数	34
3.3.2 格式化输入/输出函数	34
3.3.3 输入/输出流对象	36
3.4 C++ 程序结构	37
3.5 C++ 程序运行的步骤	38
习题 3	46
第 4 章 控制结构程序设计	50
4.1 顺序结构程序设计	50
4.2 选择结构程序设计	51
4.2.1 if 语句	51
4.2.2 switch 语句	56
4.3 循环结构程序设计	59
4.3.1 while 语句	60
4.3.2 do...while 语句	63
4.3.3 for 语句	66
4.3.4 循环语句小结	69
4.4 多重循环程序设计	70
4.5 常用转移语句	74
4.5.1 break 语句	74
4.5.2 continue 语句	74
习题 4	77
第 5 章 函数	84
5.1 引言	84
5.2 函数的定义与调用	85
5.2.1 函数的定义	85

881	5.2.2 函数的调用	86
881	5.3 参数传递方式	94
071	5.3.1 值传递	94
381	5.3.2 地址传递	95
	5.4 变量的作用域	98
881	5.4.1 局部变量	98
381	5.4.2 全局变量	99
181	5.5 变量的存储类型	100
181	5.5.1 自动变量	100
181	5.5.2 寄存器变量	101
381	5.5.3 静态局部变量	101
701	5.5.4 扩大或限制全局变量的作用域	102
108	5.6 嵌套与递归	105
608	5.6.1 嵌套	105
118	5.6.2 递归	106
818	5.7 有默认参数的函数	110
118	5.8 内联函数和函数重载	112
888	5.8.1 内联函数	112
888	5.8.2 函数重载	113
888	习题 5	114
	第 6 章 构造数据类型	121
888	6.1 数组	121
188	6.1.1 一维数组	122
188	6.1.2 二维数组	130
388	6.1.3 字符数组	136
888	6.1.4 数组与函数	144
988	6.1.5 字符串处理函数	150
888	6.2 结构体类型	155
188	6.2.1 结构体类型的定义	155
888	6.2.2 结构体变量的定义	156
888	6.2.3 结构体变量的使用	157
888	6.2.4 结构体变量的初始化	158
888	6.2.5 结构体数组	159
888	6.2.6 结构体类型的应用	161
888	6.2.7 结构体与函数	163
888	6.3 联合体类型	166
888	6.3.1 联合体类型的定义	166
888	6.3.2 联合体变量的定义	167

6.3.3	联合体类型数据的使用	168
6.3.4	联合体类型数据的应用	169
	习题 6	170
第 7 章 指针		182
7.1	指针的基本知识	182
7.1.1	指针的概念	182
7.1.2	指针的基本运算	184
7.2	指针与数组	191
7.2.1	指针与一维数组	191
7.2.2	指针与二维数组	193
7.2.3	指针数组	197
7.2.4	指针与字符串	201
7.3	指针与函数	205
7.4	指针与结构体	211
7.5	指向指针的指针	213
	习题 7	214
第 8 章 面向对象程序设计基础		223
8.1	基本概念	223
8.2	类和对象	225
8.2.1	类的定义	225
8.2.2	对象与对象数组	228
8.3	构造函数和析构函数	231
8.3.1	构造函数	231
8.3.2	析构函数	234
8.3.3	拷贝构造函数	236
8.4	对象指针	239
8.4.1	指向对象的指针	239
8.4.2	this 指针	241
8.5	静态成员	243
8.5.1	静态数据成员	243
8.5.2	静态成员函数	245
8.6	友元	247
8.6.1	友元函数	247
8.6.2	友元类	249
8.7	继承	250
8.7.1	继承的基本概念	250
8.7.2	继承方式	253

8.7.3 继承的构造函数和析构函数	256
8.8 多态性	259
8.8.1 多态性的概念和实现	259
8.8.2 运算符重载	260
8.8.3 虚函数	265
8.8.4 抽象类	268
习题 8	272
第 9 章 文件和流	283
9.1 文件的基本概念	283
9.2 流类库	284
9.2.1 C++输入/输出流	284
9.2.2 流类库的基本结构	285
9.2.3 iostream 头文件	285
9.2.4 文件流类	286
9.3 通过文件流操作文件	287
9.3.1 文件的打开与关闭	288
9.3.2 文件的输入/输出操作	290
9.4 输入/输出格式控制	296
9.4.1 使用 ios 成员函数控制格式	297
9.4.2 格式控制符	299
习题 9	301
实验指导	308
附录 A 常用 ASCII 码表	326
附录 B 常用的数学函数	327
附录 C 常用的关键字	328
参考文献	329

第1章

程序设计概述

本章学习目标

- 理解计算机求解问题的步骤
- 掌握算法设计的方法
- 了解程序设计的基本知识
- 了解 C++ 语言的特点

本章首先介绍计算机求解问题的步骤,然后着重讲解建立数学模型和算法设计的方法,最后简单介绍程序设计的基本知识和 C++ 语言的特点。

1.1 计算机求解问题的步骤和算法

计算机系统是按人的要求接收和存储信息,自动进行数据处理和计算,并输出结果的机器系统。计算机系统由硬件系统和软件系统组成。硬件系统是借助电、磁、光、机械等原理构成的各种物理部件的有机组合,是系统赖以工作的实体,例如显示器、硬盘、键盘等;软件系统是各种程序和文档,用于指挥系统按指定的要求进行工作,其中,程序是对计算任务的处理对象和处理规则的描述,文档是与软件研制、维护和使用有关的资料。在计算机中,一切信息处理都要受程序的控制,因此,求解任何问题最终要通过执行程序完成。

1.1.1 计算机求解问题的步骤

人类在解决一个问题时会根据不同的经验和环境采用不同的方法。用计算机解决现实中的问题,同样也有许多不同的方法,但解决问题的基本步骤是相同的。在计算机上运行的程序一般要经过分析问题、建立数学模型、设计算法、程序编码、测试和调试等步骤。

1. 分析问题

准确、完整地理解和描述问题是解决问题的关键。针对每个具体的问题,必须认真审查问题描述,理解问题的真实要求。分析问题就是明确要解决的问题,写出求解问题的规格说明,包括求解问题的数学模型或者对数据处理的需求、程序运行环境、用户要求输入/输出的数据及其形式等。

2. 建立数学模型

用计算机解决问题必须有合适的数学模型。数学模型是利用数学语言(符号、表达式与

图像)模拟现实的模型,把实际问题加以提炼构造数学模型的过程称为数学建模。

例 1-1 写出求 $1+2+3+\dots+100$ 的数学模型。

设 $s_{100}=1+2+\dots+100$,则对于任意 $k\in[1,100]$ 有 $s_k=1+2+\dots+k$,若令 $s_0=0$,则有 $s_k=s_{k-1}+k$,其中, $k=1,2,\dots,100$,因此,令 $s_0=0$,则数学模型为:

$$s_k = s_{k-1} + k, \quad k = 1, 2, \dots, 100$$

例 1-2 写出求两个正整数的最大公约数的数学模型。

设用 a, b 分别表示两个正整数,不妨假设 $a > b$,由辗转相除法得:

$$a = q_1 b + r_1,$$

$$b = q_2 r_1 + r_2,$$

$$r_1 = q_3 r_2 + r_3,$$

$$\vdots$$

$$r_{n-2} = q_n r_{n-1} + r_n,$$

$$r_{n-1} = q_{n+1} r_n$$

此时, r_n 是 a 与 b 的最大公约数。

假设在第 k 次辗转相除时,用 a_k 表示被除数,用 b_k 表示除数,用 r_k 表示余数,因此,令 $a_1 = a, b_1 = b$,则数学模型为:

$$r_k = a_k \bmod b_k, \quad a_{k+1} = b_k, \quad b_{k+1} = r_k, \quad k = 1, 2, 3, \dots$$

当余数 $r_k = 0$ 时,除数 b_k 是 a 与 b 的最大公约数。

3. 设计算法

设计算法是指把问题的数学模型或处理过程转化为计算机的解题步骤。

4. 程序编码

程序编码的主要任务是用某种程序设计语言将计算机的解题步骤设计为能在计算机上运行的程序。

5. 测试和调试

测试和调试的主要目的在于发现和纠正程序中的错误。

1.1.2 算法

1. 算法设计

算法是对特定问题求解步骤的一种描述,它是指令的有限序列,其中,每一条指令表示一个或多个操作。通俗点说,算法就是计算机解题的过程。在这个过程中,无论是形成解题思路还是编写程序,都是在实施某种算法,前者是推理实现的算法,后者是操作实现的算法。

设计算法是一件非常困难的工作,常用的算法设计技术有列举、穷举搜索、迭代、递归、回溯、递推、模拟、分治等。一个好的算法背后一般都有一个好的数学模型。

对于计算机科学来说,设计算法是至关重要的一步,设计的算法必须满足以下 5 个特性。

- (1) 有穷性: 算法在执行有穷个步骤后必须终止。
- (2) 确定性: 算法给出的每一个步骤都必须是精确定义, 无二义性。
- (3) 可行性: 算法中要执行的每一个步骤都可以在有限的时间内完成。
- (4) 输入: 有零个或多个外部数据作为算法的输入。
- (5) 输出: 算法产生一个或多个数据作为输出。

2. 算法描述

按照算法的执行顺序, 算法有顺序结构、选择结构和循环结构 3 种结构。顺序结构指算法按照书写步骤的顺序依次执行, 它是一种最基本、最简单的结构。选择结构是根据指定的条件进行判断, 根据判断的结果选择某些步骤的控制结构。循环结构是指在算法中需要重复执行一条或多条指令的控制结构, 即从某一条指令开始, 按照一定的条件反复执行某一处理步骤, 直到不满足条件时才结束, 反复执行的处理步骤称为循环体, 控制重复执行循环体的条件称为循环条件。

算法描述的方式主要有自然语言、流程图、盒图、PAD 图、伪代码和程序设计语言。

1) 自然语言

自然语言是人们日常使用的语言, 所描述的算法自然通俗易懂。

例 1-3 设计求两个数之和的算法。

用变量 a 、 b 分别表示这两个数, 用 c 表示 a 与 b 的和, 则数学模型为 $c=a+b$ 。

算法:

- (1) 输入 a 、 b 。
- (2) 计算 $c=a+b$ 。
- (3) 输出 c 。

显然, 该算法是顺序结构。

例 1-4 设计求两个数的最大值的算法。

若用变量 a 、 b 分别表示这两个数, 用 \max 表示 a 与 b 的最大值, 则数学模型为:

$$\max = \begin{cases} a & a > b \\ b & a \leq b \end{cases}$$

算法:

- (1) 输入 a 、 b 。
- (2) 如果 $a > b$, 则 $\max = a$, 否则 $\max = b$ 。
- (3) 输出 \max 。

显然, 第(2)步是选择结构, 其中, $a > b$ 是条件。

例 1-5 设计求 $1+2+3+\dots+100$ 的算法。

由例 1-1 可知, 令 $s_0=0$, 则数学模型为 $s_k=s_{k-1}+k, k=1, 2, \dots, 100$ 。

如果 s_0 、 s_{k-1} 、 s_k 都用变量 s 表示, 用 k 表示 $1 \sim 100$ 中的整数, 令 $s=0, k=1$, 则可以设计如下算法。

算法 1:

- (1) $k=1, s=0$ 。
- (2) 计算 $s=k+s, k=k+1$ 。

(3) 计算 $s=k+s, k=k+1$ 。

(4) 计算 $s=k+s, k=k+1$ 。

...

(100) 计算 $s=k+s, k=k+1$ 。

(101) 计算 $s=k+s$ 。

(102) 输出 s 。

算法 1 用了 102 步描述,步骤多,书写工作量大,因此是不可取的算法。

由算法 1 可知, $s=k+s$ 和 $k=k+1$ 重复执行了 100 次,因此可以用循环结构设计该数学模型。初值: $s=0, k=1$; 循环体: $s=s+k, k=k+1$; 循环条件: $k \leq 100$ 。

算法 2:

(1) $k=1, s=0$ 。

(2) 当 $k \leq 100$ 成立时转(3),否则转(6)。

(3) 计算 $s=s+k$ 。

(4) 计算 $k=k+1$ 。

(5) 转(3)。

(6) 输出 s 。

算法 2 中的第(2)~(5)步构成循环结构,其中,第(3)~(4)是循环体, $k \leq 100$ 是循环条件。第(2)步也可以改为当 $k > 100$ 成立时转(6)。

例 1-6 设计求两个正整数的最大公约数的算法。

由例 1-2 可知,令 $a_1=a, b_1=b$,则数学模型为:

$$r_k = a_k \bmod b_k, \quad a_{k+1} = b_k, \quad b_{k+1} = r_k, \quad k = 1, 2, 3, \dots$$

当余数 $r_k=0$ 时,除数 b_k 是 a 与 b 的最大公约数。

如果用 a 表示被除数 a_k 、用 b 表示除数 b_k 、用 r 表示余数 r_k ,显然先计算 $r=a \bmod b$,当 $r \neq 0$ 时重复做操作“ $a=b, b=r, r=a \bmod b$ ”,因此,可以用循环结构设计该模型。

初值:计算 $r=a \bmod b$; 循环条件: $r \neq 0$; 循环体: $a=b, b=r, r=a \bmod b$ 。当循环结束时,余数 $r=0$,这时 b 就是最大公约数。

算法:

(1) 输入 a, b 的值。

(2) $r=a \bmod b$ 。

(3) 如果 $r \neq 0$ 成立,转(4),否则转(8)。

(4) $a=b$ 。

(5) $b=r$ 。

(6) $r=a \bmod b$ 。

(7) 转(3)。

(8) 输出 b 。

在该算法中,第(3)~(7)步构成循环结构,其中,第(4)~(6)是循环体, $r \neq 0$ 是循环条件。第(3)步也可以改为如果 $r=0$ 成立,则转(8)。

2) 流程图

流程图采用一些图框表示各种操作,其形象直观、容易理解。流程图是描述算法的常用工具,一个流程图中包括表示相应操作的框、带箭头的流程线、框内/外有必要的说明文字,主要的流程图符号如图 1.1 所示。

顺序结构的描述如图 1.2 所示,选择结构的描述如图 1.3 所示,循环结构的描述如图 1.4 所示。

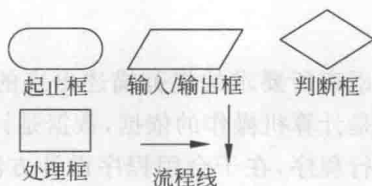


图 1.1 流程图符号

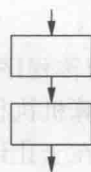


图 1.2 顺序结构

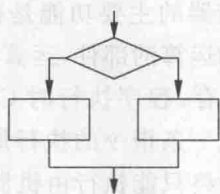


图 1.3 选择结构

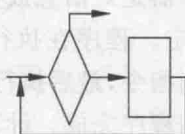


图 1.4 循环结构

例 1-7 求两个数之和。

求两个数之和的流程图如图 1.5 所示。

例 1-8 求两个数的最大值。

求两个数的最大值的流程图如图 1.6 所示。

例 1-9 求 $1+2+3+\dots+100$ 的值。

求 $1+2+3+\dots+100$ 的值的流程图如图 1.7 所示。

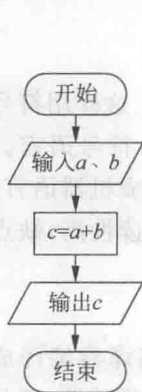


图 1.5 例 1-7

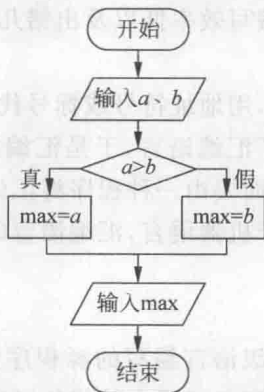


图 1.6 例 1-8

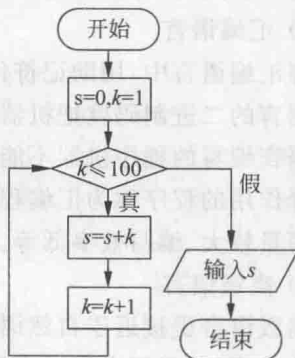


图 1.7 例 1-9

3) 程序设计语言

计算机不能识别自然语言、流程图等算法描述语言,程序设计语言是用于编写计算机程