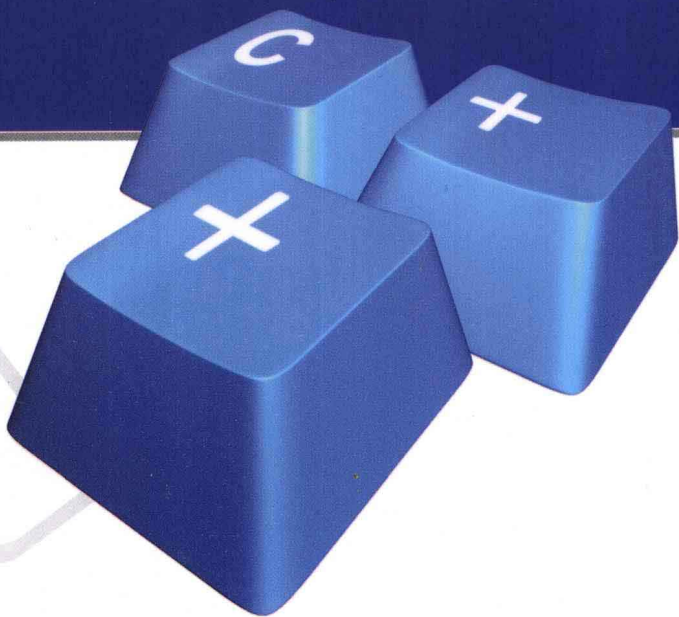


21 世纪面向工程应用型
计算机人才培养规划教材

孔丽英 夏艳 徐勇 等 编著

程序设计与算法语言 —— C++ 程序设计基础



清华大学出版社

程序设计与算法语言 ——C++ 程序设计基础

孔丽英 夏 艳 徐 勇 等编著

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,详细分析程序,以培养读者分析程序和设计程序的能力。全书共分9章。第1章介绍计算机求解问题的步骤和算法设计、计算机程序和C/C++语言。第2~7章是面向过程程序设计基础,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、通过文件指针(或流)操作文件和输入输出格式控制。

本书可作为高校非计算机专业的本科教材和自学教材以及高职各专业的教材,也可供广大软件工作者参考。

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,徐勇等编著. —北京:清华大学出版社,2011.9

(21世纪面向工程应用型计算机人才培养规划教材)

ISBN 978-7-302-26288-6

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(2011)第141689号

责任编辑:索梅薛阳

责任校对:梁毅

责任印制:杨艳

出版发行:清华大学出版社

地 址:北京清华大学学研大厦A座

<http://www.tup.com.cn>

邮 编:100084

社 总 机:010-62770175

邮 购:010-62786544

投稿与读者服务:010-62795954,jsjic@tup.tsinghua.edu.cn

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

印 装 者:北京国马印刷厂

经 销:全国新华书店

开 本:185×260 印 张:19.5 字 数:487千字

版 次:2011年9月第1版 印 次:2011年9月第1次印刷

印 数:1~3000

定 价:29.50元

产品编号:043106-01

程序设计与算法语言是高等学校重要的专业基础课,它是以为编程语言为平台,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,而且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,掌握问题求解和编程的技能,熟悉上机的全过程及调试程序的基本方法与技巧,并能利用所学知识解决一些科学计算及其实际问题。

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,又全面支持面向对象程序设计,并且具有全面支持面向过程和面向对象的混合编程等特点,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专业基础。

目前C++语言的教材很多,但大部分教材主要是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,而且缺乏程序分析,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等级考试,但设计程序的能力比较缺乏,思路单一,开拓创新不足,使学生在后续课程学习中遇到了很大的困难。

根据多年的教学经验,我们清楚地知道学生设计程序的能力比较缺乏,思路单一。学生学习过程中忽略算法设计,从而造成程序设计的能力差、很大程度上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,最终实现学生具备分析程序和设计程序的能力为目标。另外,考虑目前大部分高校的学生都要参加计算机水平考试,作者根据多年教学经验,结合计算机水平考试II级的《C++程序设计》考试大纲内容的要求,在本书中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,让学生学习更有针对性,达到事半功倍的目的,力求从根本上提高学生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,这批学生急需要一本理论性不太强,但有实用操作性的C++语言教材,本教材在编写上能满足这批学生的要求。

本书的特点有: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:根据计算机等级考试内容、面授的对象、授课时间的安排以及后续课程的需要,着重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，而且有参考计算机等级考试的题型，使学生学习更有针对性，促进学生考级能力的提高。⑦实验题设计科学：C++语言的学习不仅要有扎实的基础，更重要的是实践，教材中设计大量的实验，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。

全书共9章，分为四部分：

第1部分为第1章，程序设计概述。介绍计算机求解问题的步骤和算法设计、计算机程序和C/C++语言。

第2部分为第2~7章，面向过程程序设计基础。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3部分为第8章，面向对象程序设计基础。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4部分为第9章，文件和流。介绍文件、流类库、通过文件指针(或流)操作文件和输入输出格式控制。

本书的编写得到作者所在学校领导的大力支持，在此表示衷心的感谢。本书是作者根据多年教学实践的经验编写的。第1~7章由孔丽英编写，第8章由夏艳编写，第9章由徐勇编写，实验指导由夏艳编写。本书适合作为大学第二批本科非计算机专业和高职计算机专业的程序设计课程的教材，也可供读者自学参考。由于作者水平有限，书中可能存在缺点或错误，恳求广大读者批评指正。

作者

2011年8月

目 录

contents

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 算法设计	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	6
1.2.3 程序设计方法	7
1.3 C/C++ 语言简介	7
1.3.1 C 语言简介	7
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	14
2.4 变量与常变量	15
2.4.1 变量	15
2.4.2 常变量	16
2.5 表达式	17
2.5.1 运算符	17

2.5.2 表达式运算规则	22
2.6 类型的转换	24
习题 2	25
第 3 章 程序结构	27
3.1 简单语句	27
3.1.1 表达式语句	27
3.1.2 空语句	27
3.1.3 复合语句	28
3.2 预处理命令	28
3.2.1 “文件包含”命令	28
3.2.2 宏定义	29
3.3 数据的输入输出	30
3.3.1 标准输入输出函数	30
3.3.2 格式化输入输出函数	30
3.3.3 输入输出流对象	32
3.4 C++程序	33
3.4.1 程序运行的步骤	33
3.4.2 C++程序结构	37
习题 3	38
第 4 章 控制结构程序设计	41
4.1 顺序结构程序设计	41
4.2 选择结构程序设计	42
4.2.1 if 语句	42
4.2.2 switch 语句	47
4.3 循环结构程序设计	50
4.3.1 while 语句	50
4.3.2 do-while 语句	53
4.3.3 for 语句	56
4.3.4 循环语句小结	59
4.4 多重循环程序设计	60
4.5 常用转移语句	63
4.5.1 break 语句	63
4.5.2 continue 语句	64
习题 4	66
第 5 章 函数	73
5.1 引言	73

5.2	函数的定义与调用	74
5.2.1	函数的定义	74
5.2.2	函数调用方式	75
5.3	参数传递方式	81
5.3.1	值传递	81
5.3.2	地址传递	82
5.4	变量的作用域	84
5.4.1	局部变量	84
5.4.2	全局变量	85
5.5	变量的存储类别	86
5.5.1	自动变量	87
5.5.2	寄存器变量	87
5.5.3	静态局部变量	88
5.5.4	外部变量的声明	89
5.6	嵌套与递归	91
5.6.1	嵌套	91
5.6.2	递归	92
5.7	有默认参数的函数	95
5.8	内联函数和函数重载	98
5.8.1	内联函数	98
5.8.2	函数重载	98
	习题 5	99
第 6 章	构造数据类型	105
6.1	数组	105
6.1.1	一维数组	106
6.1.2	二维数组	114
6.1.3	字符数组	121
6.1.4	数组与函数	128
6.1.5	字符串处理函数	134
6.2	结构体类型	138
6.2.1	结构体类型的定义	138
6.2.2	结构体变量的定义	139
6.2.3	结构体变量的使用	140
6.2.4	结构体变量的初始化	141
6.2.5	结构体数组	142
6.2.6	结构体类型的应用	143
6.2.7	结构体与函数	145
6.3	联合体类型	147

6.3.1	联合体类型的定义	148
6.3.2	联合体变量的定义	148
6.3.3	联合体类型数据的使用	149
6.3.4	联合体类型数据的初始化	150
6.3.5	联合体类型数据的应用	151
	习题 6	152
第 7 章	指针	159
7.1	指针的基本知识	159
7.1.1	地址和指针	159
7.1.2	指针的基本运算	161
7.2	指针与数组	169
7.2.1	指针与一维数组	169
7.2.2	指针与二维数组	171
7.2.3	指针数组	174
7.2.4	指针与字符串	177
7.3	指针与函数	181
7.4	指针与结构体	185
7.5	指向指针的指针与常指针	187
	习题 7	189
第 8 章	面向对象程序设计基础	195
8.1	基本概念	195
8.2	类和对象	197
8.2.1	类的定义	197
8.2.2	对象与对象数组	200
8.3	构造函数和析构函数	202
8.3.1	构造函数	202
8.3.2	析构函数	205
8.3.3	拷贝构造函数	207
8.4	对象指针	210
8.4.1	指向对象的指针	210
8.4.2	this 指针	212
8.5	静态成员	213
8.5.1	静态数据成员	213
8.5.2	静态成员函数	215
8.6	友元	217
8.6.1	友元函数	217
8.6.2	友元类	218

8.7 继承	220
8.7.1 基本概念	220
8.7.2 继承方式	223
8.7.3 继承的构造函数和析构函数	225
8.8 多态性	228
8.8.1 基本概念	228
8.8.2 虚函数	231
8.8.3 抽象类	234
习题 8	236
第 9 章 文件和流	245
9.1 基本概念	245
9.1.1 文件	245
9.1.2 文件指针	246
9.2 流类库	247
9.2.1 基本结构	247
9.2.2 预定义流	247
9.2.3 文件流类	248
9.3 通过文件指针操作文件	249
9.3.1 文件打开与关闭	249
9.3.2 文本文件操作	251
9.3.3 二进制文件操作	253
9.3.4 文件的随机读写	256
9.4 通过文件流操作文件	257
9.4.1 文件打开与关闭	257
9.4.2 文本文件操作	259
9.4.3 二进制文件操作	261
9.4.4 文件的随机读写	264
9.5 输入输出格式控制	268
9.5.1 使用 ios 成员函数控制格式	268
9.5.2 格式控制符	272
习题 9	273
实验指导	278
附录 A 常用 ASCII 码	298
附录 B 常用的数学函数	299
附录 C 常用关键字	300
参考文献	301

程序设计概述

本章学习目标

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

本章先向读者介绍计算机求解问题的步骤,并通过实例讲解建立数学模型的思路和设计循环体的方法以及设计算法的过程,然后介绍程序设计语言的分类和语言编译与解释的方式,最后介绍 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,如显示器、硬盘、键盘等;软件系统是各种程序和文档,用于指挥全系统按指定的要求进行工作,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,因此任何问题求解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,根据不同的经验、不同的环境会采用不同的方法,用计算机解决现实中的问题,同样也有很多不同的方法,但解决问题的基本步骤是相同的。在计算机上运行的程序,一般都要经过分析问题、建立数学模型、设计算法、编写程序代码、测试程序等步骤。

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。如题目要求得到什么结果?题目中有哪些假定?是否有潜在的信息?等等。

问题分析的目的就是明确要解决的问题,写出求解问题的规格说明,其主要内容包括用

户要求输入输出的数据及其形式、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而构造数学模型的过程称为数学建模,数学建模通常采用的方法就是归纳法。

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 $k=1, 2, \dots, 100$, 因此数学模型为

$$\begin{aligned} s_0 &= 0 \\ s_k &= s_{k-1} + k, \quad k = 1, 2, \dots, 100 \end{aligned}$$

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 和 b 分别表示两个正整数($a > b$), 由辗转相除法得

$$\begin{aligned} a &= q_1 b + r_1 \\ b &= q_2 r_1 + r_2 \\ r_1 &= q_3 r_2 + r_3 \\ &\vdots \\ r_{n-2} &= q_n r_{n-1} + r_n \\ r_{n-1} &= q_{n+1} r_n \end{aligned}$$

此时, r_{n-1} 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, b_k 表示除数, r_k 表示余数, 则数学模型为

$$\begin{aligned} a_1 &= a, \quad b_1 = b \\ r_k &= a_k \bmod b_k, \quad a_{k+1} = b_k, \quad b_{k+1} = r_k, \quad k = 2, 3, \dots \end{aligned}$$

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

3. 算法设计

算法设计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 算法设计

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。通俗地说, 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。

算法设计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、贪心等。一个好的算法一般背后都有一个好的数学模型来描述。

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 5 个特性:

- (1) 有穷性。一个算法在执行有穷个步骤后必须终止。
- (2) 确定性。一个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三种结构。顺序结构指算法按照书写步骤的顺序依次执行,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,由判断的结构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,反复执行处理的步骤称为循环体,重复执行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所用的语言,所描述的算法自然也通俗易懂。

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,则该问题的数学模型是 $c=a+b$ 。

算法:

- ① 输入 a , b 的值;
- ② 计算 $c=a+b$;
- ③ 输出 c 的值。

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, \max 表示 a 与 b 的最大值,则数学模型是

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

算法:

- ① 输入 a , b 的值;
- ② 如果 $a > b$ 则 $\max = a$; 否则 $\max = b$;
- ③ 输出 \max 的值。

显然,该算法的第②步是选择结构, $a > b$ 是条件。

例 1-5 设计求 $1+2+3+\dots+100$ 的算法。

由例 1-1 知,求 $1+2+3+\dots+100$ 的数学模型为

$$\begin{aligned} s_0 &= 0 \\ s_k &= s_{k-1} + k, \quad k = 1, 2, \dots, 100 \end{aligned}$$

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

算法 1:

第 1 步: $k=1, s=0$;

第 2 步: 计算 $s=k+s, k=k+1$

第 3 步: 计算 $s=k+s, k=k+1$

第 4 步: 计算 $s=k+s, k=k+1$

⋮

第 99 步: 计算 $s=k+s, k=k+1$

第 100 步: 计算 $s=k+s$

第 101 步: 输出 s 的值。

算法 1 用了 101 步描述, 步骤多, 书写的工作量大, 因此算法 1 是不可取的算法。

由算法 1 可知, $s=s+k, k=k+1$ 重复执行了 100 次, 因此可以把这个算法设计成循环结构。开始时令 $s=0, k=1$, 循环体是 $s=s+k, k=k+1$, 循环条件 $k \leq 100$, 由此得到算法 2。

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②。

⑥ 输出 s 的值。

第②~⑤步构成了循环结构, 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, 则转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 知, 求两个正整数的最大公约数的数学模型为

$$a_1 = a, \quad b_1 = b$$

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

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

若对于每一个 k , 被除数 a_k 用整型变量 a 表示, 除数 b_k 用整型变量 b 表示, 余数 r_k 用整型变量 r 表示, 则 $r = a \bmod b, a = b, b = r$ 被重复执行, 因此可以用循环结构设计算法。开始时, 计算 $r = a \bmod b$, 循环条件是 $r \neq 0$, 循环体是 $a = b, b = r, r = a \bmod b$ 。当余数 $r = 0$ 时, a 就是所要求的最大公约数。因此得到该问题的算法。

算法:

① 输入 a, b 的值。

② $r = a \bmod b$ 。

③ 如果 $r \neq 0$ 成立, 则转④, 否则转⑧。

④ $a = b$ 。

⑤ $b = r$ 。

⑥ $r = a \bmod b$ 。

⑦ 转③。

⑧ 输出 a 的值。

算法中, 第③~⑦步构成了循环结构, 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,形象直观,易理解。它是描述算法的常用工具。一个流程图包括: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。主要的流程图符号见图 1.1,顺序结构的描述见图 1.2,选择结构的描述见图 1.3,循环结构的描述见图 1.4。

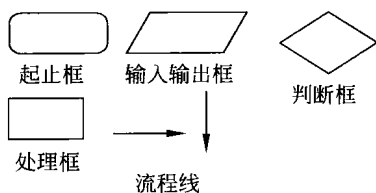


图 1.1 流程图符号

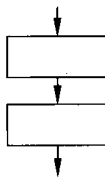


图 1.2 顺序结构

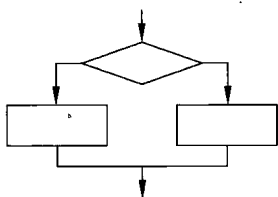


图 1.3 选择结构

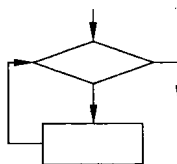


图 1.4 循环结构

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。

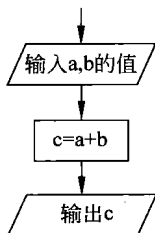


图 1.5 例 1-7

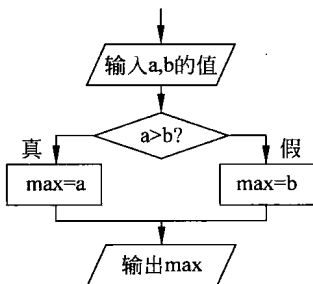


图 1.6 例 1-8

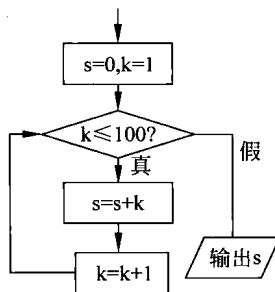


图 1.7 例 1-9

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序的语言,它是人与计算机交流的工具,因此我们要用程序设计语言描述需要解决的问题。

1.2 计算机程序

1.2.1 程序设计语言

1. 计算机程序

计算机程序(简称程序)是用程序设计语言所要求的规范描述出来的一系列动作,它表示程序员要求计算机执行的操作。程序是计算机操作的依据,数据是计算机操作的对象。学习程序设计语言在于让计算机准确地执行程序,在于会用程序设计方法去实现动作序列的表达。

计算机硬件系统由运算器、控制器、存储器、输入设备和输出设备 5 个部分组成,其中,运算器和控制器组成中央处理器(简称 CPU)。存储器主要功能是存放以二进制数据表示的指令和数据,运算器是对信息或数据进行处理和运算的部件,运算器每次执行的操作由当前指令的操作码确定。程序在执行前必须装入内存,程序执行时 CPU 负责从内存中逐条取出指令,分析识别指令,最后执行指令,从而完成了一条指令的执行周期。CPU 就是这样周而复始地工作,直到程序完成。计算机硬件系统最终只能执行由机器指令组成的程序。

2. 程序设计语言的分类

程序设计语言是用于编写计算机程序的语言。按照语言级别,程序设计语言可以分为低级语言和高级语言。低级语言又包括机器语言和汇编语言。

1) 机器语言

机器语言面向机器指令集,书写形式为二进制代码,是不需要进行任何翻译的编程语言。采用机器语言编写程序,书写的内容为机器指令,即程序员书写的源文件就是机器可以执行的指令集合。对机器来讲,机器语言的优点是执行速度极高,对程序员来讲,机器语言的缺点是不直观、编写代码量大、编写效率低以及出错几率大。

2) 汇编语言

在汇编语言中,用助记符代替操作码,用地址符号或标号代替地址码。这样用符号代替机器语言的二进制码,就把机器语言变成了汇编语言,于是汇编语言亦称为符号语言。使用汇编语言编写的程序,机器不能直接识别,要由一种程序将汇编语言翻译成机器语言,这种起翻译作用的程序叫汇编程序。相对机器语言,汇编语言的优点是可读性提高,它的缺点是代码书写量较大、编写效率低等。

3) 高级语言

高级语言更接近于自然语言。使用高级语言编写的源程序需要经过编译器编译成机器语言才能执行,但用高级语言编写的源程序不但具有较好的重用性和移植性,而且可读性和可维护性好。与机器语言相比,它的缺点是执行速度慢,需要翻译。目前 C++ 和 Java 等都是广泛使用的高级语言。

1.2.2 编译与解释

用高级语言编写的程序称为源代码或源程序。计算机是不能够直接执行源程序的,必须把源程序翻译成计算机能解读、运行的低级机器语言程序,即目标代码。翻译的方式有编

译和解释两种。

1. 编译

将一种语言编写的程序转换成等效(等价)的另一种语言的过程称为翻译,将源程序翻译成低级语言的过程称为编译。翻译的过程一般由程序来完成,将完成编译功能的程序称为编译程序(编译器),如图 1.8 所示。

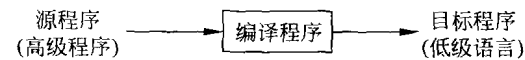


图 1.8 编译器

编译程序把一个源程序翻译成目标程序的工作过程分为 5 个阶段:词法分析、语法分析、语义检查和中间代码生成、代码优化、目标代码生成。编译程序以源程序作为输入,编译后产生目标程序。在计算机上运行的是目标程序。如 C、C++、Visual Basic 等都是采用编译方式的程序设计语言。

2. 解释

翻译的另一方式是解释运行方式,它是按照源程序中语句的动态顺序,直接地逐句进行分析解释,并立即执行。与编译方式相同的是,解释的过程是由程序完成的,称为解释程序。解释程序对源代码中的程序进行逐句翻译,翻译一句执行一句,翻译过程中并不生成可执行文件。

1.2.3 程序设计方法

用高级语言编写程序的过程称为程序设计。目前,程序设计方法可分为面向过程的程序设计和面向对象的程序设计方法。

在面向过程的设计中,程序设计者必须指定计算机执行的具体步骤,程序设计者不仅要考虑程序要“做什么”,还要解决“怎么做”的问题。例如,C 语言是面向过程的程序设计语言。面向过程的程序设计语言采用结构化程序设计的方法,按结构化程序设计的要求,程序在设计中应当采用“自顶向下、逐步求精”和“模块化”原则。

对于小规模软件,用面向过程设计的程序非常适用。但是,当软件规模相当大时,用面向过程设计的程序就出现可修改性和可重用性差的缺点。为了克服这些困难,出现了面向对象的程序设计方法。在面向对象的程序设计中,其设计思路和人们日常生活中处理问题的思路相似,它能够有效地改进结构化程序设计中存在的问题。例如,C++ 语言是面向对象的程序设计语言。

1.3 C/C++ 语言简介

1.3.1 C 语言简介

1. C 语言发展史

C 语言的起源可以追溯到 1960 年出现的 Algol 60。1963 年,剑桥大学将 Algol 60 语言发展成为 CPL(Combined Programming Language)语言。1967 年,剑桥大学的 Martin Richards 对 CPL 语言进行了简化,于是产生了 BCPL(Basic Combined Programming Language)语言。1970 年,美国贝尔实验室的 Ken Thompson 将 BCPL 进行了修改,并为它起了一个有趣的名字“B 语言”。并且他用 B 语言写了第一个 UNIX 操作系统。而在 1972 年,