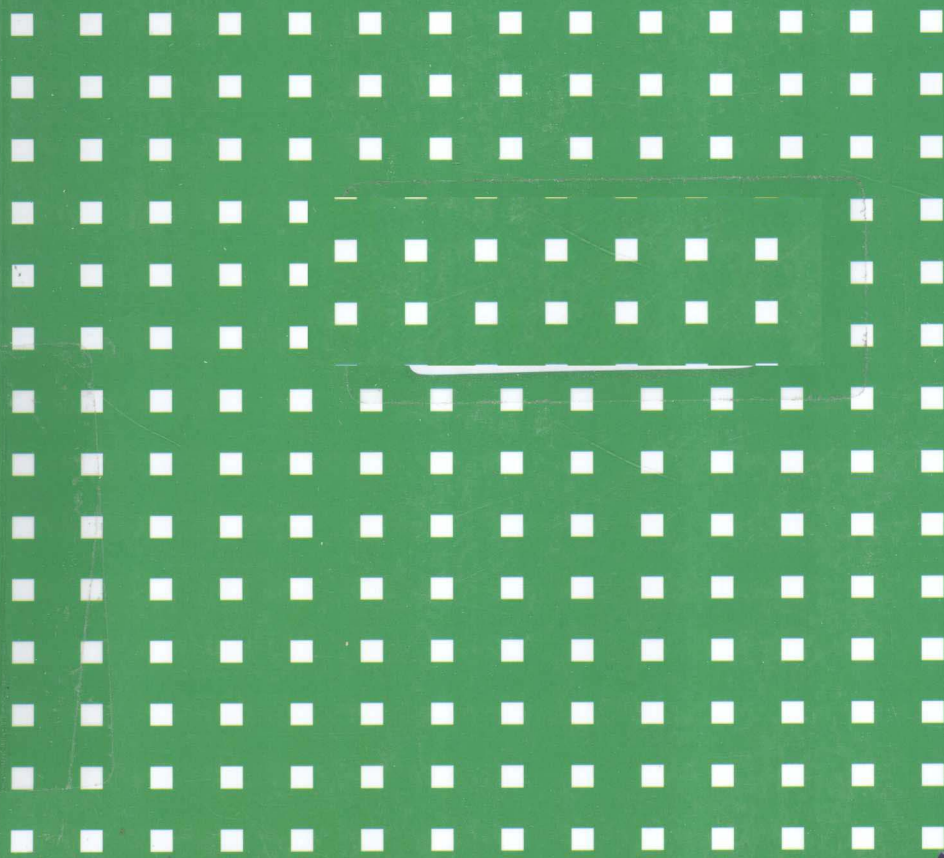


高等学校计算机专业教材精选·算法与程序设计

# C/C++程序设计

张树粹 孟佳娜 刘迎军 编著



清华大学出版社

高等学校计算机专业教材精选·算法与程序设计

# C/C++程序设计

张树粹 孟佳娜 刘迎军 编著

清华大学出版社  
北京

## 内 容 简 介

C/C++ 语言是国际上广泛应用的计算机程序设计语言。以其功能强大、表达灵活、代码效率高和可移植性好而著称。因此,它广泛应用于编写各种系统软件和应用软件。

本教材以面向过程程序设计为基础,引进了C++ 新增的运算符、函数重载、类和对象的封装性等,介绍了C++ 对 C 的改进,目的是使读者在具备基本编程能力的基础上循序渐进地向面向对象程序设计过渡。

基于多年的教学经验,编者注重 C/C++ 本身的系统性与认知规律的结合,针对初学者的特点,在写法上,勿求深入浅出,通俗易懂;在结构和内容上,勿求准确定位,合理取舍,精选例题,强化实验。全书共分 9 章,各章均精选了国家各类计算机考试的习题和实验,并采用目前用得最多的 Visual C++ 6.0 编译系统。

本书适合作为普通高等院校、高职高专、各类成人教育院校作为开设程序设计基础课程的教材,也可作为编程人员和参加计算机考试(C /C++ 模块)自学者的参考书。

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

C/C++ 程序设计 / 张树粹等编著. —北京:清华大学出版社,2010.3  
ISBN 978-7-302-21863-0

I. ①C… II. ①张… III. ①C 语言—程序设计 IV. ①TP312

中国版本图书馆 CIP 数据核字(2010)第 011219 号

责任编辑:战晓雷

责任校对:白 蕾

责任印制:何 芊

出版发行:清华大学出版社

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

社 总 机:010-62770175

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

印 装 者:北京国马印刷厂

经 销:全国新华书店

开 本:185×260

印 张:21

字 数:477 千字

版 次:2010 年 3 月第 1 版

印 次:2010 年 3 月第 1 次印刷

印 数:1~3000

定 价:32.00 元

地 址:北京清华大学学研大厦 A 座

邮 编:100084

邮 购:010-62786544

本书如存在文字不清、漏印、缺页、倒页、脱页等印装质量问题,请与清华大学出版社出版部联系调换。联系电话:010-62770177 转 3103 产品编号:036330-01

# 前言

C/C++ 语言是国际上得到广泛应用的计算机程序设计语言,以功能强大、表达灵活、代码效率高和可移植性好而著称,因此广泛应用于编写各种系统软件和应用软件。

编者的《C 语言程序设计》自 2005 年出版以来,得到了广大师生的认可,经过几年的教学检验后被评为山东省高等院校优秀教材。本次修订,笔者对全书内容做了认真细致的调整,并扩充了有关 C++ 的部分内容,力求以通俗、简练的语言叙述 C/C++ 语言中的概念、语法和程序设计方法,使学生能够快速入门,熟练掌握程序设计的基本方法。本书选用了 VC++ 6.0 编译系统,使得 C/C++ 程序的调试更加直观、方便。

清华大学自 1990 年开始将计算机教学纳入基础课的范畴,作为校重点课程进行建设和管理,并按照“计算机文化基础”、“计算机技术基础”和“计算机应用基础”三个层次的课程体系组织教学。现在,全国各高等院校也参照清华大学的做法,推广三个层次的课程体系教学。“C/C++ 程序设计”是计算机科学与技术专业大多数学科分支的入门课程,当前,高等院校在众多计算机程序设计语言中都选择 C/C++ 语言作为学科教学的先修课程。近年来,高等院校还普遍对非计算机专业开设了 C/C++ 程序设计课程,培养各专业学生的计算机使用能力。同时,在全国各类计算机考试中,越来越多的学生选择 C/C++ 程序设计模块。学习 C/C++ 已经成为广大学生的迫切要求。但由于 C/C++ 涉及的概念多,数据类型多,并且由于其表达灵活的特性,初学者很容易出错,所以会觉得 C/C++ 难学,迫切希望有一本易于快速入门的 C/C++ 教材。为此,作者根据多年的程序设计语言课程教学经验,组织课程组的教师编写了本教材。

本教材主要讲述 C/C++ 程序设计的基本原理和基本思想方法,在 C 语言的基础上扩充了 C++ 的相关内容,使 C 语言和 C++ 有机地结合在一起,使学生在编写面向过程程序时,能够运用 C++ 的新增功能简化程序,提高效率。

本书具有以下主要特点。

1. 定位准确,取舍合理。本书是针对应用型本科院校和高职高专院校计算机及相关专业的程序设计基础课而编写的,同时充分兼顾了各类高校非计算机专业 C/C++ 语言课程教学的需要。教师根据不同层次的教学要求,可以对本书的内容进行灵活取舍。

2. 精选例题,通俗易懂。为了使 C/C++ 程序设计的基本概念、基本理

论叙述得更加通俗易懂,使学生能够熟练掌握 C/C++ 程序设计的基本方法,本书选用了作者多年积累的大量教学实例,并由作者在 TC 和 VC++ 6.0 环境下分别作了精心调试。实例讲解的重点始终放在分析求解问题的思想方法上。

3. 合理设计,综合实例。程序设计是一门实践性很强的课程,不仅要讲授基本概念和基本理论,更要着力培养学生的设计和编程能力。为此,每一章最后都选编了与教学内容紧密相关的实验题和综合实验,以方便教学,循序渐进地培养学生的综合应用能力。

4. 循序渐进,向面向对象程序设计过渡。本教材以面向过程程序设计为主,介绍了 C++ 对 C 的改进,引进了 C++ 的运算符、函数重载,同时,对类和对象的封装性进行了叙述。例题和实验题均采用可视化编程的 Visual C++ 编译器,为学生进一步学习面向对象程序设计打下基础。

5. 本书提供教师教学使用的电子教案、例题源代码、习题参考答案等。

书中标有“\*”号的内容可根据教学要求进行取舍。根据编者学校的教学实践,建议教学计划 78 学时(计算机专业),其中讲授 48 学时,实验 30 学时。如果还选带有“\*”部分的内容,建议教学计划 66 学时(非计算机专业),其中讲授 34 学时,实验 32 学时。

本书共分 9 章,其中第 1、2、3、8、9 章由张树粹编写,第 4、5、6、7 章由孟佳娜编写,刘迎军参加了第 5、6 章的编写。本书统稿由张树粹、孟佳娜和胡潇琨共同完成。胡潇琨、李海军、卢云宏、谭征、刘培华等为本书的编写给予了热情帮助,在此表示最诚挚的谢意!

本书配套电子教案及书中相关源程序代码均可从清华大学出版社网站(<http://www.tup.com.cn>)本书页面中下载。

由于编者水平有限,疏漏之处在所难免,请各位读者不吝指正。

编者

2009 年 12 月

# 目 录

<b>第 1 章 概述</b>	/1
1.1 程序设计与高级语言	/1
1.1.1 程序与程序设计	/1
1.1.2 高级语言	/2
1.2 算法	/3
1.2.1 算法的特性	/3
1.2.2 算法表示	/3
1.3 C/C++ 的发展史与特点	/5
1.3.1 C/C++ 的发展史	/5
1.3.2 C/C++ 语言的特点	/5
1.4 C 程序结构及书写规则	/7
1.4.1 C 程序的基本结构	/7
1.4.2 程序的书写规则	/8
1.5 C 的基本词法	/8
1.5.1 C 使用的字符集	/8
1.5.2 保留字	/9
1.5.3 预定义标识符	/10
1.5.4 标识符	/10
1.5.5 C 的词类	/11
1.6 C 的基本语句	/11
1.7 标准输入/输出函数	/12
1.7.1 格式化输入/输出函数	/13
1.7.2 C++ 的输入/输出	/22
1.7.3 非格式化字符输入/输出函数	/24
1.8 C 程序的编辑、编译、连接和执行	/25
1.9 Visual C++ 的上机环境介绍	/26
1.9.1 Visual C++ 的安装和启动	/27
1.9.2 输入和编辑源程序	/27
1.9.3 编译、连接和运行	/30
1.9.4 建立和运行包含多个文件的程序的方法	/35
习题	/42

实验1 C程序调试初步 /44

## 第2章 基本数据类型及运算符 /46

2.1 C语言的数据类型 /46

2.2 常量 /47

2.2.1 整型常量 /47

2.2.2 实型常量 /47

2.2.3 字符常量 /47

2.2.4 符号常量 /48

2.2.5 字符串常量 /49

2.3 变量 /49

2.3.1 变量的数据类型及其定义 /49

2.3.2 变量的存储类型及其定义 /51

2.3.3 变量的初始化 /54

2.3.4 基本数据类型的使用 /54

2.4 运算符及表达式 /57

2.4.1 算术运算符和算术表达式 /58

2.4.2 关系运算符和关系表达式 /59

2.4.3 逻辑运算符和逻辑表达式 /60

2.4.4 赋值运算符和赋值表达式 /62

2.4.5 逗号运算符和逗号表达式 /63

2.4.6 变量的自增、自减(++，--)运算符 /64

2.4.7 长度运算符 /65

2.4.8 混合运算和类型转换 /66

2.5 综合运算举例 /67

习题 /68

实验2 认识数据类型 /73

实验3 使用运算符与表达式进行计算 /74

## 第3章 程序控制结构 /76

3.1 顺序结构程序设计 /76

3.2 选择结构程序设计 /77

3.2.1 if语句的3种形式 /78

3.2.2 条件运算符“?:” /84

3.2.3 switch语句实现多分支选择结构 /85

3.3 循环结构程序设计 /88

3.3.1 当循环结构 /88

3.3.2 直到循环结构 /91

3.3.3	次数循环结构	/92
3.3.4	循环嵌套与多重循环结构	/94
3.3.5	3种循环语句的比较	/95
3.4	循环体内使用 break 语句和 continue 语句	/95
3.4.1	break 语句	/95
3.4.2	continue 语句	/97
3.5	goto 语句及标号语句	/98
3.6	综合举例	/99
	习题	/102
实验 4	顺序结构程序设计	/108
实验 5	选择结构程序设计	/108
实验 6	循环结构程序设计	/110

## 第 4 章 数组 /113

4.1	一维数组	/113
4.1.1	一维数组定义	/113
4.1.2	一维数组的存储形式	/114
4.1.3	一维数组元素的引用	/114
4.1.4	一维数组的初始化	/115
4.1.5	一维数组程序设计举例	/116
4.2	二维数组及多维数组	/119
4.2.1	二维数组及多维数组定义	/119
4.2.2	二维数组及多维数组的存储形式	/119
4.2.3	二维数组元素的引用	/120
4.2.4	二维数组的初始化	/121
4.2.5	二维数组程序设计举例	/122
4.3	字符数组与字符串	/124
4.3.1	字符数组与字符串	/124
4.3.2	字符数组的输入与输出	/125
4.3.3	字符串处理函数	/128
4.3.4	字符数组程序设计举例	/130
4.4	C++ 字符串类与字符串变量	/132
4.5	数组程序举例	/133
	习题	/136
实验 7	数组及其应用	/140

## 第 5 章 函数 /143

5.1	函数概述	/143
-----	------	------



5.2	C函数的定义及构成	/144
5.3	函数的调用	/146
5.3.1	函数的调用	/146
5.3.2	C++中函数形参默认值	/150
5.4	C++中的函数重载	/151
5.5	函数间的数据传递	/153
5.5.1	值传递方式	/153
5.5.2	地址传递方式	/154
5.5.3	返回值方式	/156
5.5.4	全局变量传递方式	/157
5.5.5	C++中访问全局变量	/160
5.6	递归调用与递归函数	/160
5.6.1	递归函数的特点	/160
5.6.2	递归函数的设计	/162
5.7	内部函数和外部函数	/163
5.7.1	内部函数	/163
5.7.2	外部函数	/164
5.8	函数应用程序举例	/164
	习题	/169
	实验8 函数及其应用	/174

**第6章 指针** /177

6.1	地址、指针和指针变量的概念	/177
6.2	指针变量的定义、赋值和引用	/178
6.2.1	指针变量的定义	/178
6.2.2	指针变量的赋值	/179
6.2.3	指针的引用	/179
6.3	指针的运算	/181
6.3.1	指针的赋值运算和算术运算	/181
6.3.2	指针的关系运算	/182
6.4	指针与一维数组	/183
6.5	指针变量作函数参数	/186
6.6	指针与二维数组	/190
6.6.1	二维数组的指针表示方式	/190
6.6.2	行指针变量	/192
6.7	指针数组	/194
6.7.1	指针数组的引用	/194
6.7.2	行指针和指针数组的比较	/195

6.7.3	指针数组处理字符串	/197
* 6.7.4	指针数组作 main 函数的参数	/198
* 6.8	返回指针值的函数	/200
6.9	指向指针的指针	/201
6.9.1	二级指针的定义	/201
6.9.2	二级指针的使用	/202
* 6.10	指向函数的指针变量	/204
6.10.1	指向函数的指针的概念	/204
6.10.2	指向函数的指针的应用	/205
6.11	指针程序举例	/206
	习题	/210
	实验 9 指针及其应用	/214

## 第 7 章 构造数据类型 /218

7.1	结构体类型	/218
7.1.1	结构体类型的定义	/218
7.1.2	结构体类型变量的定义和初始化	/219
7.1.3	结构体类型变量成员的引用	/222
7.1.4	结构体类型数组的定义和初始化	/223
7.1.5	结构体类型数组元素的引用	/225
7.2	指向结构体类型数据的指针变量	/225
7.2.1	指向结构体类型变量的指针	/225
7.2.2	指向结构体类型数组元素的指针	/228
7.2.3	函数间结构体类型数据的传递	/229
7.3	动态分配和撤销内存空间	/231
* 7.4	结构体类型的应用——链表及其操作	/234
7.4.1	链表	/234
7.4.2	简单链表	/235
7.4.3	建立动态链表	/236
7.4.4	遍历链表	/237
7.4.5	链表的插入操作	/238
7.4.6	链表的删除操作	/239
7.5	共用体类型	/239
7.5.1	共用体类型变量的定义	/239
7.5.2	共用体类型变量的引用	/241
7.6	枚举类型	/243
7.6.1	枚举类型的定义	/243
7.6.2	枚举类型变量的引用	/244

7.7	C++ 中类类型的简单介绍	/245
7.7.1	类的定义	/246
7.7.2	类的对象变量	/247
7.7.3	对象的公有成员的访问	/247
7.7.4	构造函数和析构函数	/249
7.7.5	指向对象的指针变量	/251
*7.8	用 typedef 定义类型的别名	/252
7.9	综合程序设计举例(学籍管理程序)	/254
	习题	/258
	实验 10 结构体类型及其应用	/265

## 第 8 章 文件 /268

8.1	文件概述	/268
8.1.1	磁盘文件名	/268
8.1.2	文件缓冲区	/269
8.1.3	磁盘文件的打开与关闭	/269
8.1.4	磁盘文件的数据格式分类	/269
8.1.5	磁盘文件的读写格式分类	/270
8.1.6	设备文件	/270
8.2	文件类型及文件指针	/271
8.3	文件的打开与关闭函数	/271
8.3.1	打开文件函数	/272
8.3.2	关闭文件函数	/274
8.3.3	标准设备文件的打开与关闭	/274
8.4	文件的读写函数	/274
8.4.1	文件尾测试函数	/275
8.4.2	字符读写函数	/275
8.4.3	字符串读写函数	/277
8.4.4	数据读写函数	/279
8.4.5	格式读写函数	/282
*8.5	文件处理的其他常用函数	/284
8.5.1	文件头定位函数	/284
8.5.2	文件随机定位函数	/285
8.5.3	错误处理函数	/287
8.6	文件应用程序举例	/288
	习题	/289
	实验 11 文件操作	/292

<b>第 9 章 编译预处理</b>	/294
9.1 宏定义	/294
9.1.1 不带参数的宏定义	/294
9.1.2 带参宏的定义和引用	/297
* 9.2 文件包含处理	/299
* 9.3 条件编译	/302
习题	/305
实验 12 C 编译预处理	/307
<b>附录 A ASCII 字符编码表</b>	/309
A.1 标准 ASCII 字符集	/309
A.2 扩充 ASCII 字符集	/310
<b>附录 B C 运算符的优先级和结合性</b>	/311
<b>附录 C 常用的 C 库函数</b>	/313
C.1 数学函数(要求在源文件中包含 math. h)	/313
C.2 字符和字符串函数(要求在源文件中包含 string. h 和 ctype. h)	/314
C.3 I/O 函数(要求在源文件中包含 stdio. h)	/315
C.4 字符屏幕函数(要求在源文件中包含 conio. h)	/316
C.5 图形屏幕函数(要求在源文件中包含 graphics. h)	/316
C.6 动态存储分配函数(要求在源文件中包含 alloc. h 和 stdlib. h)	/318
C.7 类型转换函数(要求在源文件中包含 stdlib. h)	/318
<b>参考文献</b>	/319

## 概 述

本章主要介绍程序和程序设计的基本概念,算法与程序基本结构,C 的特点及发展史,C 的字符集、基本词类和基本句类。另外,还介绍 C/C++ 语言标准输入/输出函数,以及使用 Visual C++ 6.0 调试 C/C++ 程序的方法和步骤。

## 1.1 程序设计 with 高级语言

## 1.1.1 程序与程序设计

程序是使用计算机语言解决某个问题的方法和步骤的描述。计算机程序设计是在某一程序语言环境下编写出能够使计算机理解并执行的程序代码。程序的特点是有始有终,每个步骤都能操作,所有步骤执行完后则对应问题得到解决。

例如:求两个整数和的方法和步骤如下:

第一步 获取两个整数  $a$  和  $b$ ;

第二步 计算  $c=a+b$ ;

第三步 输出  $c$ ;

第四步 结束。

如果将上述问题转化成用计算机语言编写的程序,例如用 C 语言描述,则程序清单如下。

**【例 1.1】** 求两数和。

```
#include <stdio.h>
main()
{
    int a,b,c;
    scanf("%d,%d",&a,&b);
    c=a+b;
    printf("sum=%d\n",c);
}
/* 包含头函数 */
/* 主函数 */
/* 主函数开始 */
/* 定义 a,b,c 为整数 */
/* 给整型变量 a 和 b 赋值 */
/* 计算 c=a+b */
/* 输出计算结果 c */
/* 程序结束 */
```

该程序包括 3 部分,一是对变量  $a, b, c$  进行数据类型说明,二是确定了求两数和的计算方法,最后将计算结果打印输出。

从上述例子,我们可以看到一个 C 语言程序的基本形式是:

```

main()
{
    一组程序语句
}

```

其中,main 是主函数名,一对大括号是函数体,包括一组程序语句,每条语句以分号结束,程序中类似“/\* 输出计算结果 c \*/”的字符串是注释语句。注释语句不是程序语句,它的作用是增强程序的可读性。在 C 语言中使用形如“/\* …… \*/”的形式进行注释,VC++ 中又增加了形如“//……”的方法。

### 1.1.2 高级语言

在程序设计的发展过程中,出现了各种计算机语言,早期用二进制代码编写程序,称为“机器语言”,又称为“低级语言”。很快又出现了汇编语言,汇编语言是用符号来代表二进制代码,所以又称为“符号语言”。机器语言和汇编语言都是“面向机器的语言”,它的特点是很难脱离硬件,难以记忆。

由于程序设计的关键是将解决问题的算法过程描述出来,所以,一种能方便地描述算法过程,同时又脱离了对机型的要求,并且能够面向问题的计算机程序设计语言很快研制成功,就是现在使用的“高级语言”。目前被广泛使用的高级语言有 C、Pascal、FORTRAN 以及 VC、VB 等。每一种高级语言都有各自人为规定的专用符号、英文单词、语法规则和语句结构。高级语言与自然语言(英语)更接近,而与硬件功能相分离,便于广大用户掌握和使用。高级语言的通用性强,兼容性好,便于移植。高级语言的发展经历了从无序化的程序设计到面向过程的程序设计、面向对象的程序设计的发展过程。面向过程的语言的特点是,程序员要告诉计算机“做什么”和“怎么做”。FORTRAN、Pascal 和 C 等都是面向过程的高级语言。面向对象的语言的特点是面向具体的应用功能,即面向“对象”,其方法就是软件的集成化,将数据和处理数据的过程作为一个整体来处理。采用类的封装、数据隐藏、继承性和多态性等方法,便于将实际问题分解和抽象,使代码易于维护,并保持较高的复用性。典型的面向对象的高级语言有 C++、Java 等。

使用计算机语言编写的程序叫源程序。源程序不能在计算机上直接运行,计算机只能接受 0 和 1 组成的二进制程序(又称二进制机器指令),高级语言源程序必须通过系统软件将其翻译成二进制程序后才能执行。翻译程序有两种执行方式:一种是通过“解释程序”将源程序翻译一句执行一句,这种执行方式称为“解释执行”方式;另一种是通过“编译程序”将源程序全部翻译成二进制程序后再执行,这种执行方式称为“编译执行”方式。大多数高级语言采用“编译执行”方式,C 就是其中之一。从源程序到在计算机上得到运行结果,其操作过程(以 C 为例)如图 1.1 所示。

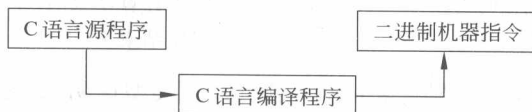


图 1.1 C 程序编译过程

## 1.2 算法

学习计算机程序设计语言的目的,是要用语言作为工具,设计出计算机能够运行的程序。设计一个程序首先需要做两方面的工作:一是组织合理结构的数据,二是设计解决问题的算法。这样,就可用任何一种计算机语言编写程序,也就是常说的计算机源程序。因此,人们用以下公式来表示程序:

$$\text{程序} = \text{算法} + \text{数据结构}$$

### 1.2.1 算法的特性

算法是指为了解决某个特定问题而采用的确定且有效的步骤。计算机算法可分为两大类:数值运算和非数值运算。数值运算的目的是求数值解,例如,求方程的根、求圆的面积、求  $n$  的阶乘等,都属于数值运算。非数值运算包括的领域十分广泛,主要用于事务管理,例如人事管理、图书管理、学籍管理等。一个算法应当有以下五个特性。

#### (1) 有穷性

一个算法应当包含有限个操作步骤。也就是说,在执行若干个操作之后,算法将结束,而且每一步都在合理的时间内完成。

#### (2) 确定性

算法中的每一条指令必须有确切的含义,不能有二义性,对于相同的输入必须能得出相同的结果。

#### (3) 可行性

算法中的每一步都应当有效执行,通过基本运算后能够实现目标。

#### (4) 有零个或多个输入

在计算机上实现算法,所需的数据大多数情况下要在程序执行时通过输入得到,但也有些程序不需要输入数据。

#### (5) 有一个或多个输出

算法的目的是求解(结果),结果要通过输出得到。

### 1.2.2 算法表示

算法可以用各种描述方法进行描述,常用的有自然语言、伪代码、传统流程图和 N-S 流程图。使用流程图(又称框图)等将算法描述出来,然后,根据流程图编写程序代码是计算机程序设计常常采用的方法。

#### 1. 用伪代码表示算法

伪代码是用介于自然语言和计算机语言之间的文字和符号来描述算法的。

**【例 1.2】** 用伪代码描述输出  $x$  绝对值的问题。

```
IF x is positive THEN
```

```

    print x
ELSE
    print -x

```

也可以中英混用,将上例改写成:

```

若 x 为正
    打印 x
否则
    打印 -x

```

### 2. 用传统流程图表示算法

传统流程图也是描述算法的很好的工具,传统流程图中使用的符号如图 1.2 所示。



图 1.2 传统流程图符号

其中,流程线包括 4 个方向线。即:  $\rightarrow$   $\leftarrow$   $\downarrow$   $\uparrow$ 。

**【例 1.3】** 求两个整数之和的算法的传统流程图如图 1.3 所示。

### 3. 用 N-S 流程图表示算法

随着结构化程序设计方法的出现,1973 年美国学者 I. Nassi 和 B. Shneiderman 提出了一种新的流程图形式。这种流程图去掉了流程线,算法的每一步都用一个矩形框来描述,一个完整的算法就是按用户设计的执行顺序连接起来的一个大矩形。人们把这种流程图称为 N-S 流程图,如图 1.4 所示。

例如,求两个整数之和的算法的 N-S 流程图如图 1.5 所示。

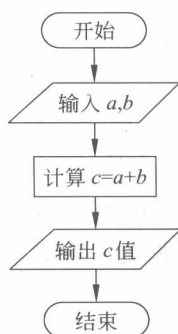


图 1.3 求两数和的传统流程图

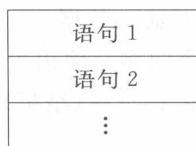


图 1.4 N-S 流程图

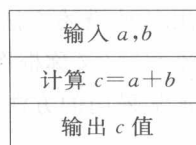


图 1.5 求两数和的 N-S 流程图



## 1.3 C/C++ 的发展史与特点

### 1.3.1 C/C++ 的发展史

C语言是目前国际上广泛流行的一种结构化的程序设计语言,计算机专业人员使用它来开发系统软件,软件开发人员使用它来编写应用软件。近些年来,广大计算机爱好者也越来越青睐C语言。

C语言的前身是ALGOL语言。ALGOL语言是1960年开发出的一种面向问题的高级语言,用ALGOL来描述算法很方便,但它的缺点是离计算机硬件系统比较远,不宜用来编写系统程序。1963年英国剑桥大学在ALGOL语言的基础上增添了处理硬件的能力,推出了CPL(Combined Programming Language)语言。CPL语言比ALGOL语言接近硬件一些,但由于规模较大,学习和掌握困难,没有流行开来。1967年英国剑桥大学的Martin Richards对CPL语言进行了简化,推出了BCPL语言。1970年美国贝尔实验室(Bell Laboratories)的K. Thompson对BCPL语言又进行了进一步简化,设计出了既简单又接近硬件系统的B语言(以BCPL的第一个字母命名),同时,使用B语言编写出了UNIX操作系统,并在PDP-7上实现。1972年美国贝尔实验室的Dennis M. Ritchie在B语言的基础上设计出C语言(以BCPL的第二个字母命名)。C语言在保留BCPL语言和B语言的强大的硬件处理功能的基础上,扩充了数据类型,恢复了通用性。1973年Dennis M. Ritchie和K. Thompson合作将UNIX操作系统用C重写了一遍(即UNIX V5)。系统的代码量比以前的版本大了三分之一,加进了多道程序设计功能,特别是整个UNIX操作系统(包括C编译程序本身)都建立在C语言的基础之上,而UNIX V5奠定了UNIX操作系统的基础。随着UNIX的日益广泛使用,C语言也得到迅速推广,可以说C语言和UNIX在发展过程中相辅相成。C语言早已成为世界上公认的应用最广泛的计算机语言之一。

1977年K. Thompson和Dennis M. Ritchie撰写了《C程序设计语言》一书,对C语言进行了规范化的描述,成为当时的标准,称为K&R标准。随着微型机的普及,出现了不同的C语言标准版本,为了统一标准,美国国家标准学会(ANSI)于1983年制定了一套ANSI C标准,此后又相继推出87 ANSI、C99标准。

C++是由C发展而来的,与C兼容,C++是C的超集。C++语言既可用于面向过程的结构化程序设计,又可用于面向对象的程序设计,是一种功能强大的程序设计语言。C++保留了C的风格和特点,同时对C的不足做了大量的改进,增加了面向对象的机制。改进后的C++与C相比,在数据类型方面更加严格,使用更加方便。

### 1.3.2 C/C++ 语言的特点

#### 1. C语言的特点

##### (1) 结构化语言

C语言是结构化程序设计语言,面向过程编程。每一类语言都有它的特点。结构化