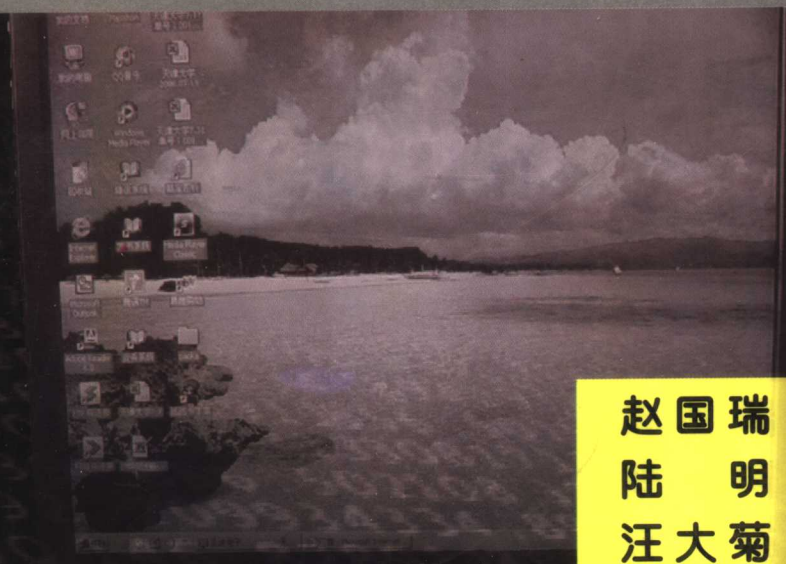
普通高等教育“十一五”国家级规划教材

C++程序设计与数据结构基础教程



赵国瑞
陆　明
汪大菊　主编

天津大学出版社
TIANJIN UNIVERSITY PRESS

普通高等教育“十一五”国家级规划教材

C++程序设计与数据结构基础教程

赵国瑞　陆明　汪大菊　主编

天津大学出版社
TIANJIN UNIVERSITY PRESS

内 容 提 要

本书由三部分组成:第一部分介绍 C++ 程序设计语言的基本成分,详细阐明了使用 C++ 进行面向过程的结构化程序设计方法;第二部分介绍 C++ 的面向对象的特征,同时给出了使用 C++ 进行面向对象的程序设计方法;第三部分介绍了基本的数据结构知识以及使用 C++ 的标准模板库(STL)实现各种基本数据结构的方法。

本书内容由浅入深、重点突出、概念清晰、通俗易懂,并含有大量的程序设计例题和各种练习题,以供读者自学。本书适全 60~80 讲授学时使用。同时出版了《C++ 程序设计与数据结构基础实验指导 习题 答案》一书与之配套使用。

本书主要为普通高等学校非计算机专业学生学习 C++ 程序设计课程而编写的,还可供各类计算机软件人员和软件开发人员、程序设计爱好者和工程技术人员参考。

图书在版编目(CIP)数据

C++ 程序设计与数据结构基础教程/赵国瑞,陆明,汪大菊主编.—天津:天津大学出版社,2006.9

ISBN 7-5618-2306-1

Ⅰ.C... Ⅱ.①赵...②陆...③汪... Ⅲ.①C 语言-程序设计-高等学校-教材 ②数据结构-高等学校-教材 Ⅳ.TP312②TP311.12

中国版本图书馆 CIP 数据核字(2006)第 082376 号

出版发行	天津大学出版社
出 版 人	杨欢
地　　址	天津市卫津路 92 号天津大学内(邮编:300072)
电　　话	发行部:022-27403647　邮购部:022-27402742
网　　址	www.tjup.com
短信网址	发送"天大"至 916088
印　　刷	迁安万隆印刷有限责任公司
经　　销	全国各地新华书店
开　　本	185mm × 260mm
印　　张	29.25
字　　数	730 千
版　　次	2006 年 9 月第 1 版
印　　次	2006 年 9 月第 1 次
印　　数	1-4 000
定　　价	45.00 元

前　言

C++是当今最流行的一种高级程序设计语言，它具有极高的灵活性、强大的功能和非常高的效率，常用于专业应用程序的开发，应用十分广泛。数据结构是程序设计的重要理论技术基础，掌握数据结构知识对于有效地开发计算机程序非常必要。程序设计语言和数据结构是程序设计的基础。程序设计能力是培养21世纪信息化时代科技人才的一个重要方面。

本书将C++和数据结构综合在一起，全书共分三部分：第一部分是C++结构化程序设计，主要介绍了数据类型、程序控制、指针、函数和结构化程序设计方法等内容；第二部分是C++面向对象程序设计，主要介绍了类、继承、重载、多态性、输入/输出流类、异常处理和面向对象程序设计方法等内容；第三部分是数据结构基础，介绍了线性表、栈、队列、串、数组、树、二叉树、图等基本数据结构以及相应的算法和简单应用。

本书从面向对象和应用的角度，按照1998年通过的C++国际标准ISO/IEC 14882较为全面地介绍了C++编程语言。书中既包括相当于传统C语言的内容，又包括对C的扩展内容。这些扩展分为三个方面，即增强传统C语言的功能、面向对象程序设计语言和标准模板库(STL)。阅读本书将获得编写C++应用程序的必要知识。本书基本上不需要读者具有任何编程语言知识，只需要了解基本的计算机数据表示和编程概念(相关内容在第1章中介绍)就可以读懂。在数据结构部分，我们选择了最基本、最实用的内容，对重要的算法都用C++语言进行了描述，并给出了详细的解释。通过学习数据结构，读者不仅可以掌握数据结构的基本内容、典型算法和使用方法，而且还可以训练其应用程序设计的能力。标准模板库(STL)是C++语言的一个重要部分，它包含了广泛应用的数据结构和各种通用的算法，是支持数据结构与算法的重要工具。本书从应用的角度对STL进行了简单的介绍。书中内容的覆盖面较广，特别是介绍了进行结构化和面向对象程序设计的主要原理和实现方法。这一方面能够使读者对相关内容有较为全面的了解，另一方面也为学有余力的读者提供了更多的学习内容。

本书各部分都提供了丰富的例题，读者可以从这些例题中学习程序设计的技巧和有效使用数据结构求解问题的方法。各章后设有习题，通过各种类型习题的练习，不仅能加深读者对基本概念和定义的理解，而且还能通过上机，提高编程能力和调试程序的能力。

建议使用本书的课程在一学年完成。第一学期推荐授课学时为32学时，上机实验24学时，课外上机实验32学时，授课内容为1至6章；第二学期推荐授课学时为32学时，上机实验16学时，课外上机实验32学时，授课内容为7至14章。考虑到不同专业对计算机编程技术的不同要求以及不同的学时安排，在使用本教材时，教师应根据具体情况对教学内容进行适当剪裁。

本书第1章到第4章由汪大菊编写，第5章到第11章由赵国瑞编写，第12章到第14章由陆明编写。赵国瑞教授审阅了全书。

由于作者水平有限，书中不足之处在所难免，敬请读者批评指正。

编者

2006年5月

目 录

第一部分 C++结构化程序设计

第一部分　C++ 结构化程序设计

第 1 章　C++ 程序设计基础

本章主要介绍C++ 语言及程序设计的基本概念，包括C++ 简单程序格式，C++ 基本数据类型、运算符及表达式和简单的输入输出。通过本章的学习，可以掌握C++ 语言的基础知识并能编写顺序结构的简单C++ 程序，为后续的学习打下基础。

1.1　C++ 语言概述

1.1.1　C++ 语言与程序设计

语言是人类交流思想的工具。在人和计算机打交道的时候，要让计算机按人们预先安排的步骤进行工作，就要解决人和计算机进行交流的问题。人和计算机进行交流的语言，称为计算机语言。

最早的计算机语言称为机器语言。机器语言是一条条二进制代码的指令，每一条二进制指令表示一个功能，例如取数、加运算等。计算机可以直接执行机器语言而不需要转化。由计算机专业人员用指令编写的机器语言程序难读、难写、难修改。为简化机器语言，人们用符号代替二进制代码，这种便于记忆的符号语言称为汇编语言。用汇编语言写出的程序称为汇编源程序。汇编源程序上机运行时必须通过一个“汇编程序”将汇编源程序翻译成二进制指令机器才能执行。汇编语言的出现，为程序设计人员提供了很大方便。但用汇编语言编写程序同样是一件繁琐的工作，它需要程序设计人员了解计算机硬件的细节，因而影响了计算机的推广、应用。高级语言的出现为广大非计算机专业人员应用计算机提供了极大的方便。目前常用的高级语言有 BASIC、FORTRAN、PASCAL、C、C++ 及 JAVA 等。用高级语言编写的程序称为源程序。计算机不能直接执行源程序，必须经过“编译程序”或“解释程序”将源程序翻译成机器指令，机器才能执行。不同的高级语言有不同的编译程序或解释程序。因为计算机语言是程序设计使用的语言，所以又称为程序设计语言。

程序设计就是将解决某个问题的过程用程序设计语言描述出来，计算机按这个描述去逐步实现。不同高级语言的程序设计方法不同。因此，从程序设计方法的角度看，高级语言中的FORTRAN、PASCAL、C 等都是面向过程的结构化程序设计语言，而C++ 、JAVA 等在面向过程语言的基础上增加了面向对象的语言内容，常称为面向对象的程序设计语言。面向对象的程序设计方法被认为是最有希望、最有前途的方法。它是为适应计算机发展，特别是为操作系统等软件资源的发展而产生的。面向对象的程序设计方法是对面向过程的结构化程序设计方法的

一次革命。这两者的根本区别是:在面向过程的结构化程序设计中,程序设计人员把重点放在解决某个问题的过程上;而在面向对象的程序设计中,设计人员把着眼点放在解决"什么"问题上,而不是问题的解决过程上,也就是只需关心一个对象能做什么,而不必关心对象的内部构成,从而使程序设计人员以更开阔的视野来观察问题、解决问题,使计算机的求解过程更接近人的思维活动,从而可以更充分发挥计算机系统的潜在能力。

C++是在C语言基础上为支持面向对象的程序设计研制的程序设计语言。C语言的前身是1967年英国剑桥大学Matin Richards推出的BCPL语言。1970年美国贝尔实验室的Ken Thompson以BCPL为基础,设计出简单而又很接近硬件的B语言。1973年贝尔实验室的D.M.Ritchie在B语言的基础上保留了它简练、更接近硬件等特点,设计出了C语言。特别是用C语言成功地改写了UNIX操作系统,从而使C语言迅速得到推广,先后被移植到大、中、小型机及微机上。C语言是一种优秀的程序设计语言。它既有高级语言的优点,也有低级语言的特点,多年来受到普遍欢迎。C++是C语言的扩充,它保留了C语言高效灵活、易于理解、可移植性好等优点,又克服了C语言类型检查不严格以及程序规模较大时很难查找和排除错误等缺点,同时增加了面向对象的特征,并扩充了许多新功能,使得C++成为目前最流行的程序设计语言。

由于C++与C兼容,从而使许多C程序代码和用C语言编写的大量的库函数不经修改就可以在C++中使用。正是由于C++是一个C的扩充而不是一个纯正的面向对象的语言,所以C++既支持面向过程的结构化程序设计,又支持面向对象的程序设计。从1980年由美国贝尔实验室的Bjarne Stroutstrup提出C++以后,几经修改完善,并于1998年颁布了ISO/MSI C++标准。目前,C++语言仍在不断发展之中。

1.1.2 C++语言和面向对象的程序设计

C++与C的最大区别是前者支持面向对象的程序设计方法。面向对象的程序设计方法简称OOP(object oriented programming),是现代程序设计的里程碑,也是结构化程序设计以及模块化、数据抽象等方法的发展。因此,作为程序设计者,既要学习结构化程序设计方法,又要学习面向对象的程序设计方法。

在面向对象的程序设计中首先要理解"对象"一词的含义。客观世界中的任何事物都可称为对象。这些事物小到一个螺丝钉、一本书,大到一个工厂、一个学校。对象是具有某种特性和某种功能的东西。对于使用者来说,选择一个对象主要不是考虑对象内部是怎样构成的,而是考虑对象能做什么,也就是说它的功能是什么。就如同用户购买电视机时,只要知道怎样操纵各按钮就行了,至于内部构造不必关心。面向对象是一种崭新的方法,用这种方法观察世界,将客观世界看成由许多不同种类的对象组成,每种对象有特定的特征和操作,不同对象的相互作用构成了完整的客观世界。把具有共同特征(属性)和操作(方法)的对象抽象出来,形成了类。类是面向对象的程序设计中最重要的概念。显然,对象是类的具体化,是某个类的一个实例。面向对象的程序主要是由对象和类建立起来的,具有封装性、继承性和多态性。

在面向对象的程序概念中,将数据和与这些数据有关的操作结合在一起,形成一个有机的整体,称为"封装"。封装是一种信息隐藏技术,通过封装将数据和其有关的行为隐蔽起来,而将一部分行为作为对外的接口。从用户或应用人员的角度看,对外的接口就是为其提供的操作功能,即为一组服务,而服务的具体实现,即对象的内部却被隐藏着。正如一台电视机给人们提

供的是外部操作，而它内部的电路被隐藏起来了。在C++程序设计中封装是由类实现的。

“继承”是面向对象的程序设计中最重要的特征。继承所表达的是一种类之间的相互关系。它使某个类除了具有特有的属性和方法外，还可以继承其他类的属性和方法。在实际工作中，一个特定问题的认识和处理，往往前人已经进行过较为深入的研究和探讨，他们的研究成果后人可以利用，并且在此基础上有所发展和创造，这正是社会发展的过程。C++ 提供的类的继承机制允许程序设计人员在保持原有类的基础上，通过继承进行扩充成为新的类。

“多态”也是面向对象的程序设计中的一个重要特征。在C++ 中，程序设计人员用向一个对象发送消息来完成一系列动作。而多态性是指不同的对象接收到相同的消息时，会产生不同的动作。C++ 语言支持两种多态性，即编译时的静态多态性和程序运行时的动态多态性。静态多态性通过函数重载实现，动态多态性通过虚函数机制实现。

1.2　计算机中的数据与编码

计算机中的信息可以分为两大类，即控制信息和数据信息。控制信息是控制计算机完成各种操作的指令和控制字。计算机加工的对象是数据信息。这里的数据包括数字信息、文字信息以及各种图形、图像、声音等。无论是哪种数据，在计算机内部都要转换成二进制数后，计算机才能进行传送、存储和加工处理。

1.2.1　进位计数制

在进行程序设计时，除使用十进制外，八进制和十六进制也经常使用。一般来说，如果数制仅采用 R 个基本符号，称为 R 进制。其中 R 称为基数。任意一个 R 进制数 N 都可以用一个式子展开，即

$$N = m_{n-1}R^{n-1} + m_{n-2}R^{n-2} + \cdots + m_0R_0 + m_{-1}R^{-1} + m_{-2}R^{-2} + \cdots$$

这里 m、n 是正整数。m 是每个数位上的值，取值范围 $0 \sim R-1$。R^i 是第 i 位的权值。该式又称按权展开式。例如，十进制数 32343.43 可以展开为：

$$3\times10^4 + 2\times10^3 + 3\times10^2 + 4\times10^1 + 3\times10^0 + 4\times10^{-1} + 3\times10^{-2}$$

二进制数 101.01 展开后为：

$$1\times2^2 + 0\times2^1 + 1\times2^0 + 0\times2^{-1} + 1\times2^{-2}$$

几种常用进位计数制的表示见表 1.1。

表 1.1　常用进位计数制

进制	基数	进位	使用符号
二进制	2	逢 2 进位	0、1
八进制	8	逢 8 进位	0、1、2、3、4、5、6、7
十进制	10	逢 10 进位	0、1、2、3、5、5、6、7、8、9
十六进制	16	逢 16 进位	0、1、2、3、4、5、6、7、8、9、A、B、C、D、E、F

1.2.2　不同数制之间的转换

使用计算机时，输入或输出数据往往采用十进制，而在计算机内部采用的是二进制。这就

需要了解各种不同进制之间的转换。

1. R 进制转换为十进制

转换方法是:将要转换的数按权展开,各位数字与它的权相乘,最后求的和就是十进制数。例如:

$$(1101.01)_2 = 1\times2^3 + 1\times2^2 + 0\times2^1 + 1\times2^0 + 0\times2^{-1} + 1\times2^{-2} = (13.25)_{10}$$

$$(205.4)_8 = 2\times8^2 + 0\times8^1 + 5\times8^0 + 4\times8^{-1} = (133.5)_{10}$$

$$(AF.8)_{16} = 10\times16^1 + 15\times16^0 + 8\times16^{-1} = (175.5)_{10}$$

其中二进制到十进制的转换是最简单也是最常用的。

2. 十进制转换为 R 进制

将十进制转换为 R 进制时,将十进制数的整数部分和小数部分分别转换,然后再拼接起来。

整数部分是连续除以 R 进制的 R,直到商为 0,将每次除以 R 的余数自下而上排列,即构成 R 进制的整数部分。

小数部分是连续乘以 R 进制的 R,每次取积的整数,并将其自上而下排列,直到小数部分为 0 或规定精度为止。

例如,将$(237.625)_{10}$转换为二进制的过程如下:

整数部分除以 2 取余	余 数		整数	小数部分乘以 2 取整
2 | 2 3 7	1			0.625
2 | 1 1 8	0			× 2
2 | 5 9	1	取值方向 ↑	1	1.250
2 | 2 9	1			0.25
2 | 1 4	0			× 2
2 | 7	1		0	0.50
2 | 3	1			× 2
2 | 1	1		1	1.00
0				

转换结果为:$(237.625)_{10} = (11101101.101)_2$

又如,将$(237.625)_{10}$转换为八进制的过程如下:

整数部分除以 8 取余	余数		整数	小数部分乘以 8 取整
8 | 2 3 7	5			0.625
8 | 2 9	5	取值方向 ↑		× 8
8 | 3	3		5	5.000
0				

转换结果为:$(237.625)_{10} = (355.5)_8$

再如,将$(237.625)_{10}$转换为十六进制的过程如下:

整数部分除以 16 取余	余数		整数	小数部分乘以 16 取整
16 | 2 3 7	D(13)	取值方向		0.625
16 | 1 4	E(14)			× 16
0			A	10.000

转换结果为：$(237.625)_{10} = (ED.A)_{16}$

3.二、八、十六进制之间的转换

计算机内部表示的二进制数书写起来比较长，不太方便，因此在程序中用到二进制时，往往采用八进制和十六进制。每位八进制数可以用 3 位二进制表示($2^3 = 8$)；每位十六进制数可用 4 位二进制表示($2^4 = 16$)。

将二进制转换成八进制时，以小数点为界，左右分别按 3 位一组划分，两头不够 3 位时补 0，将划分的 3 位数按八进制数读出即可。例如将二进制 10100110101.0111 转换成八进制：

010	100	110	101	.	011	100
2	4	6	5	.	3	4

因此：$(10100110101.0111)_2 = (2465.34)_8$

将八进制转换成二进制时，将每位八进制数用 3 位二进制表示出来即可。例如，将八进制数 7325 转换成二进制：

7	3	2	5
↓	↓	↓	↓
111	011	010	101

因此：$(7325)_8 = (111011010101)_2$

将二进制转换成十六进制时，以小数点为界，左右分别按 4 位一组划分，两头不够 4 位时补 0，将划分的 4 位数按十六进制数读出即可。例如，将二进制 11110110101.101 转换成十六进制：

0111	1011	0101	.	1010
7	B	5	.	A

因此：$(11110110101.101)_2 = (7B5.A)_{16}$

同样，将十六进制转换成二进制时，将每位十六进制数用 4 位二进制表示出来即可。

1.2.3　二进制信息的存储单位

在计算机内存储二进制形式的信息通常采用的存储单位有位、字节、字。

位(bit)是指一位二进制信息，是计算机中数据处理的最小单位。

字节(Byte)是由 8 位二进制数组成的(1 Byte = 8 bit)。字节是信息存储最常用的基本单位。计算机存储器的容量通常是以字节表示的。常用来描述容量的单位有：

千字节(kB)，1 kB = 1024 bit

兆字节(MB)，1 MB = 1024 kB

千兆字节(GB)，1 GB = 1024 MB

字(Word)是一个独立的信息处理单位，又称为计算机字。它的长度(位数)取决于计算机的类型。一个字可以是一个字节，也可以是多个字节。常用的字长有 8 位、16 位、32 位、64 位等。如某类计算机的字长为 32 位，相应的计算机称为 32 位机。

1.2.4 二进制数据的编码

一个数在计算机内的表示形式称为“机器数”,它代表的数值称为“真值”。

数据信息在计算机内采用二进制形式,并有正、负之分,一般用“0”表示正号,用“1”表示负号,符号位一般放在数的最高位。例如8位二进制数 A = +1100101 和 B = -1100101,在机器中可以表示为:

A: 0 1 1 0 0 1 0 1

B: 1 1 1 0 0 1 0 1

↑

符号位

为了简化运算及计算机的硬件结构,人们研究出了带符号数的多种二进制编码。常用的编码有原码、反码和补码。

1.原码

将符号位用0或1表示,数的绝对值与符号一起编码,称为原码。例如,用一个字节存放一个整数的原码表示如下:

$X1 = +1100101 \qquad [X1]_{原} = 01100101$

$X2 = -1100101 \qquad [X2]_{原} = 11100101$

这里$[X1]_{原}$和$[X2]_{原}$为机器数,而 $X1$ 和 $X2$ 为真值。原码对0(零)的表示不是唯一的,有正0和负0之分:

$[+0]_{原} = 00000000$ (正0)

$[-0]_{原} = 10000000$ (负0)

采用原码表示的数,编码简单直观,转换方便,但在做加减运算时符号处理比较麻烦。

2.反码

反码的正数与原码完全相同。负数的反码符号位与原码相同,仍用1表示,数值部分由原码数值各位取反得到。例如:

$X1 = +1100101 \qquad [X1]_{原} = 01100101 = [X1]_{反} = 01100101$

$X2 = -1100101 \qquad [X2]_{原} = 11100101 \quad [X2]_{反} = 10011010$

用反码表示零也有两种形式,即

$X1 = +0000000 \qquad [X1]_{原} = 00000000 = [X1]_{反} = 00000000$

$X2 = -0000000 \qquad [X2]_{原} = 10000000 \quad [X2]_{反} = 11111111$

计算机中一般不采用反码存储和运算,反码常作为求补码的一种中间码使用。

3.补码

正数的补码与原码和反码相同。例如:

$X1 = +1100101 \qquad [X1]_{原} = [X1]_{反} = [X1]_{补} = 01100101$

负数的补码是求该数的反码并在最末位加1得到。例如,$X2 = -1100101$ 的原码、反码、补码如下:

$[X2]_{原} = 11100101$

$[X2]_{反} = 10011010$

$[X2]_{补} = [X2]_{反} + 1 = 10011010 + 1 = 10011011$

补码的特点之一就是零的表示是唯一的。当 $X = 0$ 时，$[+0]_{补} = [-0]_{补}$，以 n 位数为例：

$$[+0]_{补} = \underbrace{0000\cdots0}_{n个0}$$

$$[-0]_{补} = [-0]_{反} + 1 = \underbrace{1111\cdots1}_{n个1} + 1 = \boxed{1}\underbrace{00000\cdots0}_{n个0}$$

机器自动丢掉

上述运算中，由于受机器字长 n 位的限定，最高位前面的 1 表示不出来而丢失，从而使 $[+0]_{补} = [-0]_{补}$。

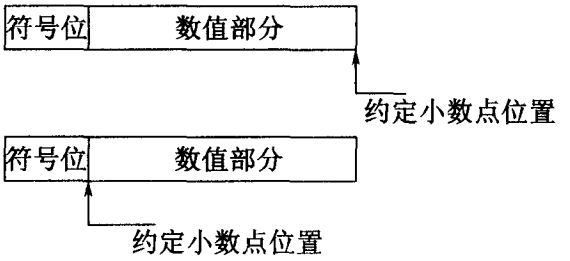
补码的另一个特点就是符号位可以作为数值参加运算，符号位无需单独处理，得到结果的最高位仍是正确的符号位。采用补码进行运算时，减法运算可以转换为加法运算，从而简化了运算电路。因此，补码在计算机中得到广泛应用。

1.2.5　定点数和浮点数

机器中处理的数除有正、负之分外，还有整数和小数之分。为了能正确地表示小数点的位置，一般采用两种方式：一是约定小数点的位置固定不变，这样表示的机器数称为定点数；另一种是允许小数点浮动，这样表示的机器数称为浮点数。

1. 定点数

所谓定点数即小数点的位置是固定的，通常采用把小数点固定在最低位的右边或是最高位的前面，即将参加运算的数化成纯整数或纯小数形式处理。以下是这两种定点数示意：



2. 浮点数

一个数 N 用浮点形式表示，可以写成：

$$N = \pm M \times R^{\pm E}$$

其中，R 表示基数，一般是 2；E 表示 2 的次幂，称为数 N 的阶码。阶码确定了数 N 小数点的位置，其位数决定浮点数表示数的范围，是个带符号的整数。M 是有效数字，称为尾数，通常是纯小数，其位数反映了数的精度。

阶码和尾数都是带符号的数，可以采用不同的码制表示。例如，阶码常用补码表示；尾数常用原码或补码表示。

浮点数的具体格式随机器的不同而不同。例如，设某机器字长 32 位，用左边的 8 位作为阶码，右边的 24 位作为尾数，符号位都在最高位，格式见图 1.1。

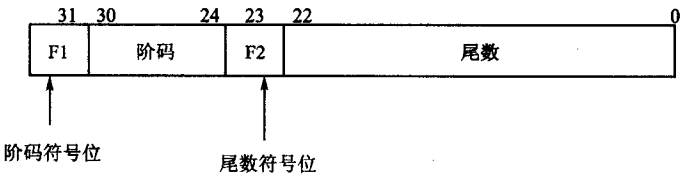


图 1.1 浮点数格式

1.2.6 数的表示范围

计算机中数的表示范围与机器的位数和所用的编码有关。设机器数为 N 位(包括 1 位符号位),定点数表示的范围如下:

①表示整数时,原码和反码能表示的最大数为 $2^{N-1}-1$,最小数为 $-(2^{N-1}-1)$。若用补码表示,最大数为 $2^{N-1}-1$,最小数为 -2^{N-1}。

②表示小数时,若采用原码和反码表示,所表示的数的范围是 $-(1-2^{-N})\sim(1-2^{-N})$。采用补码表示时,数的范围是 $-1\sim(1-2^{-N})$。

浮点数所能表示的数的范围由阶码的位数和尾数的位数决定。例如,某计算机用 32 位表示一个浮点数,其中阶码用 8 位补码定点整数表示,尾数用 24 位补码定点小数表示(规格化)。格式见图 1.1。它所能表示的数值的范围是:

$$-1\times 2^{2^7-1}\sim +(1-2^{-23})\times 2^{2^7-1}$$

显然,它比 32 位定点数所能表示的数的范围($-2^{31}\sim +(2^{31}-1)$)大得多。

1.2.7 非数值数据的表示

计算机内使用的非数值数据也用二进制进行编码,最常用的编码是 ASCII(American Standard Code for Information Interchange)码。ASCII 用一个字节中的 7 位二进制位表示一个字符,每个字符唯一对应一个 ASCII 码,最多可表示 $2^7=128$ 个字符。有些机器系统利用没有用到的最高位作为校验码位,以提高机器系统存储和传输的可靠性。

ASCII 码包括:数字 0~9,编码是 30H~39H(H 表示是 16 进制);英文大写字母 A~Z,编码是 41H~5AH;英文小写字母 a~z,编码是 61H~7AH。共有 95 个可打印字符和 33 个控制字符。具体对应关系见附录。

汉字在计算机内部也采用二进制的数字编码。由于汉字的数据量大,显然用一个字节是不够的。目前应用比较广泛的是 GB 2312—80(《国家标准信息交换用汉字编码》),简称国标码。国标码采用二字节编码,每个字节仅用低 7 位,最高位为 0,共编码了 7 445 个图形字符,其中 6 763 个汉字和 682 个非汉字符号。

1.3 C++程序开发过程

一个C++系统通常由程序开发环境、语言和C++标准库等部分组成。

在C++程序开发环境中,一个C++程序要经过编辑(edit)、预处理(preprocess)、编译(compile)、连接(link)、装入(load)和执行(excute)等六个阶段。以使用 Windows 下的集成程序开发