



国外经典教材·计算机科学与技术

Mc
Graw
Hill Education

数值方法(C++描述)

Numerical Methods With
Computer Programs In C++

本书涵盖了本科数值方法的全部内容。

从应用的角度，描述数值方法在科学工程中的应用。

C++语言实现各种数值方法。

PALLAB GHOSH 著
徐士良 等译



清华大学出版社

国外经典教材·计算机科学与技术

数值方法(C++ 描述)

Pallab Ghosh 著

徐士良 葛 兵 徐 艳 等译

徐士良 审校

清华大学出版社

北 京

Pallab ghosh

Numerical Methods With Computer Programs In C++

EISBN: 8120329872

Copyright © 2008 by The McGraw-Hill Companies, Inc.

Original language published by The McGraw-Hill Companies, Inc. All Rights reserved. No part of this publication may be reproduced or distributed by any means, or stored in a database or retrieval system, without the prior written permission of the publisher.

Simplified Chinese translation edition is published and distributed exclusively by Tsinghua University Press under the authorization by McGraw-Hill Education (Asia) Co., within the territory of the People's Republic of China only, excluding Hong Kong, macao SAR and Taiwan. Unauthorized export of this edition is a violation of the Copyright Act. Violation of this Law is subject to Civil and Criminal Penalties.

本书中文简体翻译版由美国麦格劳-希尔教育出版(亚洲)公司授权清华大学出版社在中华人民共和国境内(不包括中国香港、澳门特别行政区和中国台湾)独家出版发行。未经许可之出口,视为违反著作权法,将受法律之制裁。未经出版者预先书面许可,不得以任何方式复制或抄袭本书的任何部分。

北京市版权局著作权合同登记号 图字 01-2008-0571 号

本书封面贴有 McGraw-Hill 公司防伪标签,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

数值方法(C++描述)/(印)高希(Ghosh, P.)著;徐士良等译. —北京:清华大学出版社, 2008.9

书名原文: Numerical Methods With Computer Programs In C++
(国外经典教材·计算机科学与技术)

ISBN 978-7-302-17643-5

I. 数… II. ①高… ②徐… III. ①电子计算机—数值计算 ②C语言—程序设计
IV. TP301.6 TP312

中国版本图书馆 CIP 数据核字(2008)第 073313 号

责任编辑:龙啟铭 李玮琪

责任校对:徐俊伟

责任印制:杨 艳

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

地 址:北京清华大学学研大厦 A 座

邮 编:100084

邮 购:010-62786544

印 刷 者:北京国马印刷厂

装 订 者:三河市新茂装订有限公司

经 销:全国新华书店

开 本:185×260 印 张:30.75

字 数:744 千字

版 次:2008 年 9 月第 1 版

印 次:2008 年 9 月第 1 次印刷

印 数:1~3000

定 价:59.00 元

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:026433-01

译者序

数值方法原属于数值计算的问题,现在又引入了计算机。本书从应用的角度来描述数值方法,又直接用计算机来实现这些方法,这不仅对于学生,而且对于广大工程技术人员来说,是很有帮助的。有了本书,对数学不很精通的人也能解决工程中的数值计算问题,因为书中有了现成的程序;而对计算机不很熟悉的人来说,也能用计算机解决数值计算的问题,也是因为书中有了现成的程序。这就是本书的最显著的特点。

本书不仅可以作为高等院校理工科专业《数值方法》等课程的教材或参考书,也是广大工程技术人员在进行数值计算时的一本很好的工具书。

担任本书主要翻译工作的有徐士良、葛兵、徐艳,同时参加翻译工作的还有马尔呢、白小玲、徐娟、刘庆元、张苹、彭明静、卢谦、王瑄等。

因水平有限,译文中有错误或不妥之处,恳请读者批评指正。

译者

前 言

本书是一本关于使用计算机程序语言 C++ 求解数值问题的书籍,它展示了计算机在解决科学和工程问题中所扮演的重要角色。计算机除了速度优势外,它们还具有准确性、灵活性和多功能性。

你可以找到很多相关的书籍,这些书可以让你很好地认识数值方法和如何应用它们去解决问题。所以问题来了:我们还需要另外一本这方面的书吗?当我让同项目中的同事去写一本关于数值方法的书时,他们抱怨说:“什么?另一本书?”我确信,一本能在提供计算机语言的同时,清晰示范出如何使用它们解决科学工程问题的数值方法的书是必要的。

有很多理由来支持这一观点,其中最重要的是,近些年,C++ 语言已经成为软件开发者的首选,因为它有很多吸引人的特征,很多科学工程学科的学生毕业后都选择了软件开发的职业。

市场上使用这一强大语言的数值方法的书籍并不多,本书在第一章对 C++ 语言作了简短的介绍。现在,学校里会讲授 C 程序设计语言。在很多的工科院校,工科新生都要学习高级 C 和 C++ 语言。所以,程序设计语言不会成为计算机实现数值方法的障碍。

目前,一门关于数值方法的课程通常会达到一周 2~3 小时的授课以及 3~4 小时的计算机实验部分。上机时学生可以在计算机上实践算法。本书就是针对这样的课程而写的。它主要用于大学工科课程。不必将 15 章的内容在一个学期内全部讲完,根据学生的需求,教师可以挑出部分章节进行讲授。本书也适用于那些需要计算机数值方法实现的专业人士。

本书由三个主要部分组成:(1)数学基础(包括一些定理和推论);(2)科学工程中的方法应用;(3)计算机实现。这三部分涵盖了本科生的整个课程,很少有一本书能包括全部内容。在市场上能够看到很多优秀的书籍,这些书籍都提供了数值方法的理论。本书从应用的角度来描述数值方法。那些简单的问题可以通过手算得到结果,但重点是如何使用计算机来解决问题。就像计算机手册一样,很多已解决的问题会在所有的章节出现,便于概念的理解。学生需要对每章最后给出的问题进行实践。并详细参考在每章结尾处列出的参考书籍,对本章中的主题进行深入学习。在本书的最后给出了很多不错的参考书籍,建议学生阅读。

在学习了一个程序如何工作以及如何开发这样一个程序后,你足以理解一个数值算法是如何工作的了。此时你可能还能开发出一个简洁的程序。然后你就可以开始使用像 MATLAB、Mathematica、Polymath、NAG 和 IMSL 这些商业软件了。

我要感谢 IIT Guwahati 的主任和代理主任的支持,并出版了本书。我的学生们一直希望有一本这样的计算机程序书,他们给出了有价值的反馈和建议,为此我表示感谢。准备本书期间,我得到了姐姐 Kakali 以及双亲一贯的支持。我的同事很热情、亲切友好并且很配合工作,我同样要感谢他们。我要特别感谢我以前的同事——Anupam Shukla 教授对我写这本书的支持。最后,我想感谢出版者 Prentice-Hall of India,特别是经理和编辑与发行组,感谢他们在出版本书过程中的紧密合作。

如果本书能激发大家对数值方法及其计算机实现的兴趣,我会感到很高兴。欢迎广大读者对本书内容提出宝贵意见。

Pallab Ghosh

IIT Guwahati

目 录

第 1 章 C++ 与面向对象的程序设计	1	第 3 章 求解联立线性代数方程	50
1.1 引言	1	3.1 引言	50
1.2 C++ 程序的组成	1	3.2 几个矩阵定义	51
1.3 C++ 标识符与关键字	4	3.3 解的唯一性	52
1.4 头文件	5	3.4 顺序 Gauss 消去法	53
1.5 C++ 数据类型	6	3.5 全选主元 Gauss 消去法	59
1.6 常量声明	8	3.6 带有正向与反向代入的 LU 分解	63
1.7 运算符	10	3.7 Cholesky 分解	69
1.8 循环结构	13	3.8 Gauss-Jordan 消去法	74
1.9 判断语句	16	3.9 Gauss-Jordan 法求矩阵的逆	81
1.10 C++ 函数	20	3.10 求解三对角线方程组	87
1.11 C++ 文件处理	24	3.11 Gauss-Seidel 迭代	91
1.12 数组	27	3.12 Gauss-Seidel 松弛迭代	96
1.13 构造函数与析构函数	32	3.13 病态系统	100
总结	34	总结	101
关键字	35	关键字	102
练习	36	练习	102
文献	38	文献	106
第 2 章 数值计算的准确性与稳定性	39	第 4 章 求解非线性方程	107
2.1 引言	39	4.1 引言	107
2.2 有效数字	39	4.2 图解法	108
2.3 误差定义	40	4.3 用反复试验法求解	108
2.4 舍入误差	41	4.4 归类与开放法	108
2.5 数值运算的有效数字规则	43	4.5 迭代法的收敛速度	109
2.6 级数的截断误差	43	4.6 对分法	109
2.7 误差的传播与计算不稳定性	45	4.7 试位法	115
2.8 病态系统	47	4.8 逐次代入法	118
总结	47	4.9 Newton-Raphson 法	122
关键字	47	4.9.1 Newton-Raphson 法的收敛 速度	125
练习	47	4.9.2 初值的重要性	126
文献	49		

4.10 割线法	126	分布	223
4.11 求解非线性方程组的方法	129	6.2.4 累积分布函数的计算	224
4.12 确定多项式方程的根	138	6.2.5 特征函数	228
4.12.1 多项式方程的求根 方法	139	6.2.6 集中趋势测度	229
4.12.2 多项式值和它的导数 的计算	140	6.2.7 联合分布和协方差	231
4.12.3 综合除法	141	6.3 数据抽样	233
4.12.4 Graeffe 根平方法	143	6.4 平均值与方差的计算	234
4.12.5 求复根的 Lin 法	151	6.5 抽样分布的性质	237
4.12.6 求复根的 Newton-Raphson 法	157	6.6 置信区间	238
总结	160	6.7 假设检验	239
关键字	161	6.7.1 错误类型	239
练习	161	6.7.2 置信度	240
文献	166	总结	242
第 5 章 矩阵的特征值与特征向量	167	关键字	242
5.1 引言	167	附录 6A	243
5.2 Cayley-Hamilton 定理	170	附录 6B	244
5.3 特征向量的正交性与规格化正 交性	171	附录 6C	246
5.4 确定特征值与特征向量的方法	172	附录 6D	247
5.5 Faddeev-Leverrier 法	172	练习	249
5.6 乘幂法	177	文献	252
5.7 对称矩阵特征值的计算	186	第 7 章 曲线拟合	253
5.7.1 Jacobi 法	186	7.1 引言	253
5.7.2 Householder 法	197	7.2 线性回归	254
5.7.3 确定三对角矩阵的特 征值	203	7.3 曲线回归	265
总结	206	7.4 多元回归	272
关键字	206	7.5 使用正交多项式的回归	279
附录 5A 涉及向量和矩阵的一些普通 运算	207	总结	281
练习	214	关键字	282
文献	217	练习	282
第 6 章 数据统计分析	218	文献	285
6.1 引言	218	第 8 章 数据排序	286
6.2 初等概率论	218	8.1 引言	286
6.2.1 概率分布	218	8.2 冒泡排序	287
6.2.2 分布的矩	220	8.3 Shell 排序	289
6.2.3 正态分布与对数正态 分布	223	8.4 快速排序	294
		8.5 冒泡排序、Shell 排序和快速排序的 比较研究	298
		总结	298
		关键字	299
		练习	299
		文献	300

第 9 章 函数逼近	301	11.2 Newton-Cotes 闭合积分公式	362
9.1 引言	301	11.3 Richardson 外推法	372
9.2 Chebyshev 逼近	302	11.4 Romberg 求积法	373
9.3 Padé 逼近	311	11.5 Gauss 求积法	378
9.4 误差函数	314	11.6 数据的积分	381
9.5 Beta 和 Gamma 函数	316	11.7 开放积分公式	383
9.6 Bessel 函数	319	总结	384
总结	324	关键字	384
关键字	324	练习	384
附录 9A 某些公共函数的 Maclaurin 级数	324	文献	387
附录 9B	325	第 12 章 数值微分	388
附录 9C	326	12.1 引言	388
附录 9D 第一类 0 阶和 1 阶 Bessel 函数	327	12.2 向前、向后和中心差分公式	388
附录 9E 第二类 0 阶和 1 阶 Bessel 函数	329	12.3 Richardson 外推法	393
附录 9F $J_0(x)$ 、 $J_1(x)$ 、 $Y_0(x)$ 和 $Y_1(x)$ 的前 10 个零点	330	12.4 Lagrange 微分公式	396
练习	330	总结	397
文献	332	关键字	397
第 10 章 插值法	333	练习	397
10.1 引言	333	文献	399
10.2 多项式插值	334	第 13 章 求解常微分方程: 初值问题	400
10.3 Newton 向前差分公式	335	13.1 引言	400
10.4 Newton 向后差分公式	337	13.2 Taylor 级数展开法	401
10.5 Gauss 中心差分公式	340	13.3 Euler 法	402
10.6 Newton 差商公式	341	13.3.1 改进 Euler 法: 预报-校正 公式	403
10.7 Lagrange 插值公式	344	13.3.2 中点法	407
10.8 样条插值	347	13.4 Runge-Kutta 法	408
10.8.1 线性样条函数	348	13.5 Runge-Kutta-Fehlberg 法	412
10.8.2 二次样条函数	348	13.6 用 Runge-Kutta 法求解常微分 方程组	416
10.8.3 三次样条函数	350	13.7 刚性微分方程	419
总结	357	13.8 数值解法的稳定性	421
关键字	357	13.9 隐式 Euler 法和 Crank-Nicolson 法	422
附录 10A 由式[36]计算二阶导数的 三对角算法	357	13.10 多步法	423
练习	358	13.10.1 多步 Euler 预报-校正 法	424
文献	360	13.10.2 Adams 公式的推导	427
第 11 章 数值积分	361	13.10.3 Milne 预报-校正法	428
11.1 引言	361	13.10.4 Adams 预报-校正法	430
		13.11 向后微分公式(BDFs)	432

13.12 微分-代数方程组	433	第 15 章 偏微分方程数值解	461
总结	434	15.1 引言	461
关键字	435	15.2 偏微分方程的类型	461
练习	435	15.3 初始与边界条件的详细说明	462
文献	439	15.4 有限差分法求解	463
第 14 章 求解常微分方程: 边值问题	440	15.4.1 显式法求解	465
14.1 引言	440	15.4.2 隐式法求解	468
14.2 边界条件类型	441	15.4.3 Crank-Nicolson 法	468
14.3 试射法	441	15.4.4 双曲线型偏微分方程 求解	471
14.4 有限差分法	445	总结	472
14.5 正交配置法	448	关键字	472
14.6 特征值问题	454	练习	473
总结	455	文献	475
关键字	456	附录 常用的微分和积分公式	476
附录 14A	456	参考文献	480
练习	457		
文献	460		

C++ 与面向对象的程序设计

1.1 引言

C++ 是比较好的 C 语言版本。它最主要的一个特点就是支持面向对象程序设计 (OOP)。然而, C++ 不是唯一支持 OOP 的语言, 像 Smltalk、Ruby 和 Eiffel 这些语言也支持 OOP。但有很多原因使得 C++ 比其他语言更受欢迎。当 C++ 刚问世的时候, 它被当作“C 语言的扩展集”。有很多 C 语言的用户要写各种各样的应用程序, 他们中的很多程序员都看中了 C++ 语言的优点和易用性而改用这种语言了。工程应用程序传统上是使用 FORTRAN 开发的, 但 C++ 语言提供了面向对象方法和程序的可移植性。另外, 它还提供了开发有趣图形的特性。这使软件开发成为一种乐趣。

C++ 既可用于科学计算, 也可用于系统程序设计。现在, 有很多非常不错的编译器可供选择, 如 Microsoft、Borland、GNU 和 Intel C++ 编译器。程序员们就是用这些很令人兴奋的工具来开发软件的。本书中的程序是使用 Microsoft Visual C++ 编译器 (6.0 版本), 操作系统为 Windows 98/2000/XP。然而, 所有这些程序在做较小修改后也可以在 UNIX 系统下运行。

C++ 是一种很丰富的语言, 它涵盖了内容本来就很丰富的 C 语言。在本章中不打算让大家学习全部 C++ 语言, 只为那些打算使用本语言开发数值计算程序的工程人员做个 C++ 语言的介绍。它不能替代大量优秀的 C++ 程序书籍, 本章的“文献”中列出了不少这样的书籍。这里介绍了该语言的一部分, 并给出例子进行阐述。这可以让读者理解所有本书中给出的程序。假设读者已经有了初步的计算机知识, 就可以启动开发软件, 并使用软件菜单编译、运行程序。C 语言知识并不是学习 C++ 所必需的, 然而如果读者已经学习了 C 语言, 他会发现 C++ 与之很相似。

1.2 C++ 程序的组成

一个面向对象的 C++ 程序包含用户定义的类 (数据和成员函数在其中), 类的对象和一个 main 函数。下面是本书的第一个 C++ 程序。

程序 P1.1

```

//P1_1.CPP
#include <iostream.h>
#include <math.h>
class program1
{
    private:
        double a, result;
    public:
        void square_root()
        {
            cout << "\n 输入一个数 ";
            cin >> a;
            result = sqrt(a);
            cout << "\n 数的平方根是 " << result << endl;
        }
};
//主函数
void main()
{
    program1 p1;
    p1.square_root();
}

```

这个程序以//P1_1.CPP 开始。双斜杠//代表本行包含注释语句。任何以双斜杠开头的将被视为注释,它在程序中不会被执行。编译器会忽略注释,它不会增加程序大小和执行时间。你可以在注释中写任何东西(当然也包括有意义的声明)。接下来是两个#include命令。这些预处理命令告诉编译器在程序中添加两个括号<>中的文件(例如 iostream.h 和 math.h)的内容。为什么要这样做?因为这些文件中写入了我们刚才开发的程序所需要一些声明(输入、输出和数学运算)。这些文件通常称为头文件,作为习惯它们有后缀.h。

程序中的下一个声明指明了一个名为“program1”的类。该声明包含了关键词 class,后面跟着类名 program1。该行后面的花括号中界定了类体。这个类中包含了两个关键字 private 和 public,每一个关键字的后面都有个冒号。那些不可在类外部访问(但可在类内部访问)的数据或者函数都放在 private 下,而那些可以在类外部访问的数据或函数则放在 public 下。这样做是为了保护程序中的数据,在程序设计术语中,称之为数据隐藏。数据项称为类的数据成员,类似地,函数称为类的成员函数。面向对象程序设计的主要思想是将数据和函数放置在一个体中(如类)。类的描述如图 1.1 所示。

在本书中,我们会将数据置于“private”中,同时将函数放在“public”下。在上一个例子中,数据项 a 和 result 都是双精度类型。它们通过下面的声明语句来定义。

```

class [类名称]
{
    private:
        [私有函数和数据项];
    public:
        [公有函数和数据项];
};

```

图 1.1 C++ 类的描述

```
double a, result;
```

注意, C++ 语句总是以分号结束的。C++ 编译器将忽略空白区域(如空格、换行、回车、造纸和制表符)。例如, 下面的语句是一样的。

语句 1:

```
double a, result;
```

语句 2:

```
double a,  
result;
```

但第一个语句的可读性要比第二个语句好。

成员函数通常在“public”下定义。在本例中, 函数 `square_root` 就定义在“public”下, 它的返回类型为 `void`。一些函数会返回一个值, 而另一些不会, 当一个函数不返回任何值时, 可以使用 `void` 关键词。函数体用花括号进行限定。在本例中给出的函数是在类内部定义的一个小函数。有时函数体会很长, 这时更通常的做法是在类内部声明, 而在程序的其他地方进行定义。函数的定义如图 1.2 所示。当函数名后的括号为空则表明该函数没有参数(参数是指当函数被调用时传递给它的数据)。如果没有变量, 也可以在括号内填入 `void`。然而多数 C++ 程序更倾向省略掉 `void`。

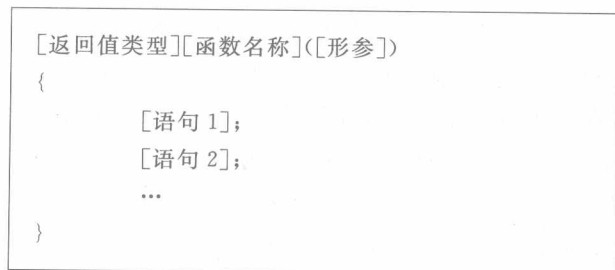


图 1.2 函数定义

函数定义的第一行称为声明, 在程序 P1.1 中, 声明是

```
void square_root()
```

这个函数的目的是请求用户输入一个数, 计算它的平方根, 并在计算机屏幕上显示该平方根的值。函数中的第一个语句是:

```
cout << "\n 输入一个数";
```

该语句在屏幕上显示了引号内除了 `\n` 以外的内容。`\n` 调用了一个转义字符, 这个转义字符的作用是换行, 确保下一个字符打印在新的一行。标识符 `cout` (读作 C-out) 是用来在屏幕上显示结果和语句的。只有包含了头文件 `iostream.h` 时才能使用它。跟在其后面的操作符 `<<` 称为插入操作符。在这个语句中, 它将引号内的字符递给 `cout`, 后者将字符发送到显示设备。在下一个输出语句

```
cout << "\n 数的平方根是" << result << endl;
```

中,字符串常量将显示出来,后面带有数据成员 `result` 的值。在此操作符 `<<` 被串联在一起。操作符 `endl` 称为操纵器,它将调用回车符,其作用与 `\n` 相同。这样的操纵器经常用于修改程序输出显示的方式。它们与插入操作符共同使用。

上面讨论的第一句输出语句要求用户输入一个数,输入可以通过下面的语句来实现:

```
cin>>a;
```

标识符 `cin` 读作 C-in,它必须包含头文件 `iostream.h`。操作符 `>>` 称为提取操作符。用户输入的值将被存储在 `a` 中。就像插入操作符一样,这个操作符也可以串联在一起多次使用。

在输入一个数值后,可以通过下面的语句计算该数值的平方根。

```
result= sqrt(a);
```

这里的“`sqrt`”是库函数,在 `math.h` 文件中定义。它将括号中“`a`”当成参数并返回平方根值。注意 C++ 语言是区分大小写的,如果你用“`Cout`”代替 `cout`,编译器将立刻提示错误。

重要提示: 每个 C++ 语句都以分号结束,每个类也以分号结束。在类的指定和函数声明符后没有分号。然而,一个函数声明(原型)需要以分号结束。

虽然程序(即计算平方根)的主要部分已经编写好了,但程序到这里并没有结束,每个 C++ 程序必须有一个 `main` 函数。在这个程序中,`main` 函数的声明如下:

```
void main()
```

该函数的主体包含在括号内。在 C++ 程序中第一个执行的语句是 `main` 函数开始的语句,而不是程序中的函数和类。`main` 函数可以写在程序的开头也可以写在程序的结尾,或者程序中的任何地方。在这个例子中,`main` 函数中定义了类 `program1` 的对象 `p1`。“对象”是类的实例,一旦定义了一个对象,那么内存中就会为它开辟出一个空间。当定义一个类时,并没有产生一个对象。定义只是说明了当对象被创建时它看起来是什么样的。一旦建立了对象,就可以使用它来调用类的成员函数,语句如下:

```
p1.square_root();
```

对象和类间的点“.”代表了对象和成员函数之间的联系,类的成员函数只可以被该类的对象访问。

1.3 C++ 标识符与关键字

标识符是由一系列字符(字母和数字)组成,用来表示如对象名或数据项、类名、类成员、函数、标签或宏这些项目。第一个字符必须是字母,这里将下划线“`_`”视为字母。所以类名 `program1` 是一个标识符。但字符序列 `007` 不能作为一个标识符。同样,字符序列 `a.b.c` 或 `a c` 也不是合法的标识符。程序员可以为了方便起见来选择用户定义的标识符。这里建议使用有意义的标识符(可以反映它们的目的),并在以后一直沿用该统一的风格。标识符的长度依赖于支持的编译器。

一些标识符是预定义或预留的,它们有特殊的意义。我们不能使用这些名字做我们自己的标识符。这样的标识符称为关键字。下面列出 C++ 中的关键字。

C++ 关键字

asm	dynamic_cast	namespace
auto	else	new
bool	enum	operator
break	explicit	private
case	extern	protected
catch	false	public
char	float	register
class	for	reinterpret_cast
const	friend	return
const_cast	goto	short
continue	if	signed
default	inline	sizeof
delete	int	static
do	long	static_cast
double	mutable	struct
switch	typedef	virtual
template	typeid	void
this	typename	volatile
throw	union	wchar_t
true	unsigned	while
try	using	

另外,编译器的开发者会保留他们自己的关键字,这些会列在他们手册里。注意,那些标识符和关键字都是区分大小写的。所以,标识符 `class` 是关键字,而 `Class` 不是。

1.4 头文件

标准输入输出、库函数、各种格式化说明和过程处理的声明都在头文件中定义。有时也称它们为包含文件。从历史观点上说,这样的文件叫做标题文件,可以以任何顺序包含它们,必须在任何使用头文件声明之前包含该头文件。表 1.1 中列出了本书中常用到的一些头文件。

表 1.1 一些 C++ 头文件和它们的应用

头 文 件	用 途
<code>fstream. h</code>	包括用于处理输入和输出文件的声明
<code>iomanip. h</code>	包括输入/输出操作(即用于改变显示数据路径的运算符)的声明
<code>iostream. h</code>	包括用于 C++ 程序标准输入输出操作的声明
<code>math. h</code>	声明数学函数和宏
<code>process. h</code>	声明进程管理的符号和结构
<code>stdio. h</code>	声明输入和输出函数、类型和宏
<code>stdlib. h</code>	声明如数字转换和存储分配等操作函数

这些头文件中有一部分来源于 C 语言,但它们同样也可用于 C++ 程序。

1.5 C++ 数据类型

C++ 有 6 个主要的数据类型,它们的关键字是: char(字符)、int(整型)、long int(长整型)、float(浮点型)、double(双精度浮点型)和 long double(长双精度浮点型)。字符数据类型存储值范围从 -128~127 的整数。它只占用一个字节的内存,通常用于存储 ASCII 字符。当使用一对单引号将一个字符括起来,如 'A',它就变成一个字符常量。编译器将它翻译成相应的 ASCII 码并存储。例 1.1 给出了字符数据类型的应用。

例 1.1 编写一个程序,输出与数值 127 对应的 ASCII 符号,同时示范定义字符常数并在屏幕上输出。

解: 下列程序实现描述的任务。

程序 P1.2

```
//P1_2.CPP
#include <iostream.h>
class program2
{
    private:
        char a, b;
    public:
        void output()
        {
            a=127;
            b='A';
            cout << "a=" <<a <<endl;
            cout << "b=" <<b <<endl;
        }
};
//主函数
void main()
{
    program2 p2;
    p2.output();
}
```

程序的输出是:

```
a = △
b = A
```

注意,数值 127 的 ASCII 符号是△。

特殊的一组字符常数被用作转义字符。这些常数之所以被称为转义字符,是因为它们的输出与正常情况使用的字符类型不同。在表 1.2 中列出了一些转义字符。

表 1.2 转义字符

转义字符	结 果	转义字符	结 果
\a	哔哔声	\t	制表符
\b	退格	\\	反斜线符号
\f	换页	\'	单引号
\n	换行	\"	双引号
\r	回车		

整型(关键字: int)表示整数(例如,1,2,3,...)。计算离散量如迭代计数就使用这些数值。整数类型的有效数值范围为-32 768~32 767。long int 类型的有效数值范围为-2 147 483 648~2 147 483 647。在数值计算中,有时你可能必须使用 long int 类型。这个类型有时只写成 long。

其他整数类型有 short 类型,在此我们不讨论它。在某些操作系统中,它与 int 类型是一样的;但在某些环境中,它要比 int 类型的数值范围小一些。

重要提示: C++ 提供了一种说明数据类型的灵活性,即在靠近首次使用的数据项处说明其类型。有时这种方法使程序容易理解,因为读者不必去别处搜索其类型。

数值计算主要包括具有小数部分的数值。在 C++ 中,定义了 3 种类型具有小数部分的数据: float、double 和 long double。通常称它们为浮点数。在表 1.3 中列出了它们的范围和精度。

表 1.3 C++ 浮点数

类 型	范 围	精度(位数)
float	$3.4 \times 10^{-38} \sim 3.4 \times 10^{38}$	7
double	$1.7 \times 10^{-308} \sim 1.7 \times 10^{308}$	15
long double	$3.4 \times 10^{-4932} \sim 3.4 \times 10^{4932}$	19

浮点数可以用指数表示。即 3.4×10^{18} 在程序中写成 3.4e18。在过去,当计算资源不允许时采用 float 类型,但严重时往往导致数值错误。在本书中,一般用 double 类型来处理浮点数。当需要特别精确时,例如处理很小的数,可以使用 long double 类型。在下面的例子中说明整型和浮点数的使用。

例 1.2 编写一个使用 int、long 和 double 类型数据的示范程序。

解: 程序 P1.3 说明了整型数据和浮点数的处理。

程序 P1.3

```
//P1_3.CPP
//整型和浮点型数据的使用说明程序
#include <iostream.h>
#include <math.h>
class program3
{
    private:
        int a, b;
        long int c;
```