



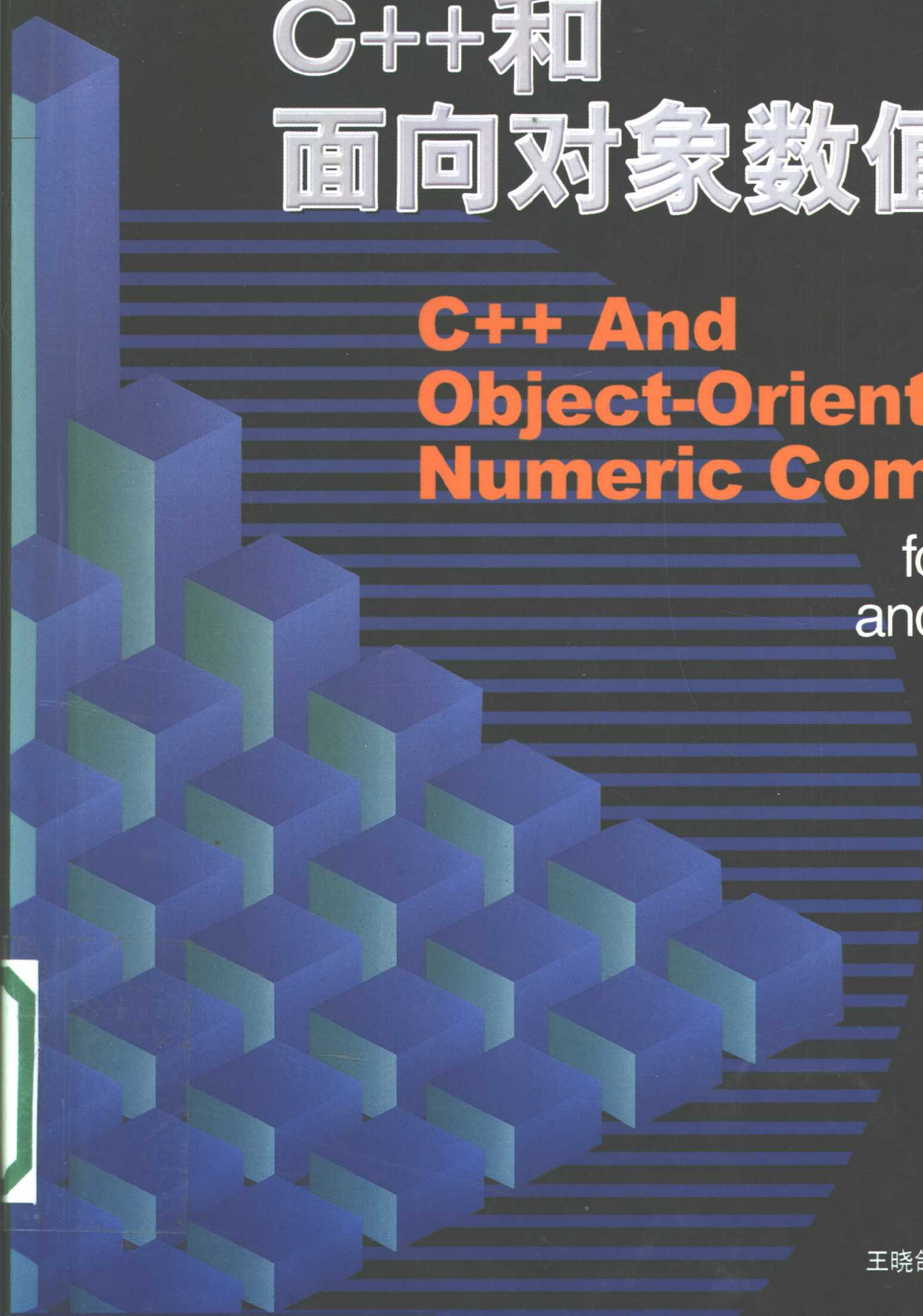
C 和 C++ 实务精选



# C++和 面向对象数值计算

**C++ And  
Object-Oriented  
Numeric Computing**

for Scientists  
and Engineers



|     |     |    |     |    |
|-----|-----|----|-----|----|
| 王晓鸽 | 盛佳  | 武勃 | 杨道奇 | 著  |
|     | 杨道奇 | 梁磊 |     | 译  |
|     |     |    |     | 审校 |

人民邮电出版社  
POSTS & TELECOMMUNICATIONS PRESS

C和C++实务精选

# C++和面向对象数值计算

---

杨道奇 著

王晓鸽 盛佳 武勃 译

杨道奇 梁磊 审校

人民邮电出版社

## 图书在版编目(CIP)数据

C++和面向对象数值计算 / 杨道奇, 王晓鸽编著. —北京: 人民邮电出版社, 2003.1

ISBN 7-115-10626-6

I. C... II. ①杨...②王... III. C 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字 (2002) 第 084973 号

## 版 权 声 明

Translation from the English language edition:

C++ and Object-Oriented Numeric Computing for Scientists and Engineers by Daoqi Yang

Copyright© 2001 Springer-Verlag New York, Inc.

Springer-Verlag is a company in the BertelsmannSpringer publishing group.

All Rights Reserved.

本书中文简体字版由 Springer-Verlag 授权人民邮电出版社出版。未经出版者书面许可, 不得以任何方式复制或抄袭。

版权所有, 侵权必究。

C 和 C++实务精选

### C++ 和面向对象数值计算

---

- ◆ 著 杨道奇  
译 王晓鸽 盛佳武 勃  
审校 杨道奇 梁磊  
责任编辑 陈冀康
- ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号  
邮编 100061 电子函件 315@ptpress.com.cn  
网址 <http://www.ptpress.com.cn>  
读者热线 010-67132705  
北京汉魂图文设计有限公司制作  
北京顺义振华印刷厂印刷  
新华书店总店北京发行所经销
- ◆ 开本: 800×1000 1/16  
印张: 29.5  
字数: 710 千字 2003 年 1 月第 1 版  
印数: 1-4 000 册 2003 年 1 月北京第 1 次印刷

著作权合同登记 图字: 01 - 2002 - 1540 号

ISBN 7-115-10626-6/TP · 3083

---

定价: 52.00 元

本书如有印装质量问题, 请与本社联系 电话: (010) 67129223

---

# 内 容 提 要

---

本书教你学习 C++ 编程和面向对象的数值计算方法。

本书包括 11 章，分别从类型、表达式、语句、文件、类、操作符重载、模板、例外等方面探讨了 C++ 编程的基本特点以及在数值计算方面的应用方法，最后一章专门讨论了线性方程组求解的问题。

本书适合学习 C++ 编程的计算机科学或工程专业的学生及科研人员阅读，对于有一定基础的 C++ 程序员通过本书可以了解 C++ 的高级特性和在数值计算方面的应用。

# 中文版序

当我写本书英文版时，我没有想到它要被翻译成其他语言。今天，我很高兴看到它被翻译成中文——我的母语。

作为一种计算机语言，C++有很多优点。很多复杂的算法和设计可以比较容易地用 C++ 来实现。同时它的计算速度可以与 FORTRAN 和 C 相媲美(有时比 FORTRAN 和 C 更快)。当前，C++已经被普遍地应用于科学技术和日常生活的各个领域。

本书除了介绍 C++语言的基础知识和高级技巧，还阐明了很多计算方法。另外，本书包括一些实例来说明怎样提高程序的计算速度。在写本书时，我力争把这些内容写得简明扼要，通俗易懂，而又比较完整。这样它既可以作为 C++初学者的教材，又可以供给有经验的 C++程序员参考。本书特别适用于各类科学技术专业的学生和工作人员，因为它包含很多常用的计算方法及其 C++源程序。

总之，希望本书中文版的读者能够有所收获并喜欢它，同时敬请读者多多指教。

最后，感谢清华大学王晓鸽副教授和清华大学的盛佳和武勃，感谢他们辛勤的翻译工作，感谢福特汽车公司梁磊先生对译稿的校对工作和人民邮电出版社陈冀康先生的编辑工作。

杨道奇

2002 年 10 月于美国密西根州

# 作者简介

杨道奇，山东省郯城县人。1980 年考入山东大学数学系。1987 年获计算数学硕士学位。从 1987 年 8 月到 1990 年 8 月任教于青岛大学数学系，后就读于美国普度大学(Purdue University)，并在 1994 年获计算机科学硕士学位，1995 年获应用数学博士学位(导师为 Jim Douglas, Jr.)。

1995 年 8 月他任教于美国密西根州韦恩州立大学(Wayne State University)数学系。他从 1996 年 9 月到 1997 年 8 月在 Minnesota University 当博士后。之后返回韦恩大学任助理教授。从 2000 年底以来，他在工业界从事软件开发工作。目前他在美国密西根州 Mount Clemens General Hospital 任高级系统分析师。

杨道奇曾发表科技论文 30 余篇，其中一篇于 1995 年获英国数学及其应用协会(Institute of Mathematics and its Applications)的 Leslie Fox Prize 二等奖。他和妻子王秀娟、女儿杨笑笑、儿子杨成林现住在美国密西根州 Troy 市。



# 前言

本书希望为从事科学和工程的学生和专业人员提供对ISO/ANSI C++编程语言的通俗、简明而又比较完全的介绍，并着重对面向对象的数值计算进行了讲解。本书中对于语言的描述是与平台无关的，只要有标准的C++编译器的支持，它适合于不同的操作系统，如Unix、Linux、MacOS、Windows 以及DOS。阅读本书的前提只需基本的微积分和线性代数知识。不过，如果只是为了讲授C++而略过所有数值计算的内容的话，可以不用满足以上这些要求。对其他的编程语言，如FORTRAN、C、Ada、Pascal、Maple或Matlab等有经验的话肯定会对阅读本书有所帮助，但这不是必需的。

书中对C++的所有特点都以通俗易懂的方式进行介绍，利用函数、复数、向量、矩阵、积分等这些对于每一位从事工程和科学研究的学生或专业人员都很熟悉的概念来讲解。在最后一章中，对一些在FORTRAN、C、Ada以及Matlab中没有的高级特性的介绍融汇在线性方程组的迭代算法，比如带预条件的共轭梯度(CG)方法和广义最小残差(GMRES)方法的实现中。对于CG、GMRES和其他预处理技术不需要作事先的了解，书中会作详细的解释。矩阵能够以3种方式存储：密集存储（所有的元素都被存储）、带状存储（只存储沿主对角线方向上的一个带状区域之内的元素以节省内存）、稀疏存储（只保存非零元素）。对于CG或GMRES来说只需要有一种定义就可以适用于3种存储方式。这和其他基于过程的语言，比如FORTRAN、C、Ada或Matlab需要给出三种不同定义相比有很大的不同。这是面向对象编程的一个突出的特点，被称为继承性。CG和GMRES方法定义成一个基类，并能够被三种矩阵继承。如果以后需要再添加一种对称矩阵的存储方式（对称矩阵中只需要存储一半元素以节省空间），那么CG和GMRES的代码可以重用，不需要重新编译。

另一个值得注意的特性是通过模板实现的泛型编程，它可以定义一个函数，根据不同的调用条件采用不同类型的参数。比如，CG和GMRES的同样的定义可以处理单精度、双精度、扩展双精度甚至是实部和虚部具有不同数据类型的复数的矩阵。同样，使用C和FORTRAN 需要定义该函数的不同版本来处理不同的数据类型，当日后函数需要作出改进来提高健壮性或效率时，所有的版本都必须更新，这是非常枯燥的，而且容易出错。C++的运算符重载特性使得矩阵和向量使用+和\*来相加或相乘就像普通的数值计算一样。运算符是可以由用户定义的，所以比Matlab更加灵活。比如，运算符+和\*可以用来相加、相乘三种不同的矩阵。这些特性以及C++的其他特性，比如信息隐藏、封装、多态、错误处理和标准函数库等，都将在书中作详

细介绍。随着数值算法的复杂性增加，很多复杂的数据结构用C++比用FORTRAN和C实现相对容易一些，这是C++一个十分重要的特性。C++编译器比FORTRAN、C、Ada、Pascal和其他许多语言的编译器纠错能力更强，这使得C++使用更安全。

然而，凡事都是有利有弊的。比如，模板会造成编译时的额外负担，而继承（和动态类型绑定）会造成运行时的额外负担。这些特性会使得程序在编译和运行时变慢。好的C++编译器可以尽可能地减少这些额外开销，而程序员就可以在享受C++带来的开发便利的同时不需要过多地为编译运行速度担心。这一点可以从C++在工业界，特别是在对速度要求非常高的领域，如数据库、计算机图形学和电子通信等方面的流行得到印证。当人们不断的优化编译器，C++就变得越来越快。在有限元分析中作的一些性能比较表明C++已经可以在速度上和FORTRAN 77和C抗衡了。另一方面，C++具有超越FORTRAN 77和C的潜力，因为它有很多内置的复合运算符和模板机制，使得它能够避免比如由于传递函数参数带来的负担，并且许多高性能的函数库（像valarray）可以在不同的体系结构上优化。这种潜力在许多测试中已是现实。

本书由三个部分组成：第一部分（第1章到第4章）是对C++基本特性的介绍，有些和FORTRAN 90和C有类似的地方。如果读者只了解这一部分，他可以做一个FORTRAN 90或C程序员可以做的任何事。同时也对如何与Makefile文件、使用调试器、建立库和如何对程序计时和配置一个程序作了介绍。第二部分（第5章到第9章）是关于C++的面向对象和通用编程特性的。这些在FORTRAN 90或C中是没有的。许多编写高效率的C++程序的技术也作了介绍，参见§6.5、§7.2、§7.6、§7.2.4、§7.7和§8.6节。正是这一部分使得许多人认为C++“太复杂”，为了能够使得普通的科学与工程人员和学生易于理解，我使用了复数、向量、矩阵和积分的基本概念和运算。这些应该是能够容易为一个具有微积分和线性代数基本知识的人所接受的。第三部分（第10章和第11章）描述了C++容器类的标准库（比如链表、集合、向量、映射、栈和队列），算法（比如线性排序、搜索算法等），以及对一个由用户定义的数值线性代数库，它包含CG和GMRES算法，可以适用于用各种方式存储的实数和复数矩阵。这可以加深读者对于第一、第二部分的理解。并且，如果读者想创建自己的库的话，它可以节省很多时间。在书中，我很仔细地将容易理解和在数值计算中经常使用的特性首先介绍。书中的一些例子是取自标准的数值计算函数库，也可以用于建立数值计算函数库。而另外的例子则是为C++特性的解释和说明而构造的。

从数值计算角度来看，本书包含了有关以下数值方法的讨论和程序：多项式求值(§3.12)，数值积分技术(§3.13, §5.1和§7.7)，向量和矩阵计算(§6.3和§7.1)，非线性方程的迭代算法(§4.7)，多项式插值(§7.8)，实域和复域上线性方程组的迭代和直接算法(§7.8, §11.3和§11.4)，求解常微分方程的Euler和Runge-Kutta方法(§5.9)以及求解偏微分方程的有限差分方法(§11.5)。这些数值算法还远远称不上完全，而只是想让读者对数值方法有一个基本的了解，并且可以看出C++是如何有效地应用于这些问题的。对于学习如何编写数值计算程序的读者可以仔细阅读这些部分，而如果只想了解C++，则略过或浏览一下就可以了。

C++不是一个完美的语言，但它似乎拥有了一个程序员所能想到的所有特征和标准库。我的经验是使用它编程比FORTRAN和C容易许多，因为后两者运行时有太多的错误和太少的特征及标准库。许多运行时错误是难以调试的，但却可以被C++编译器轻松地发现。



总而言之，C++是这样一种记号的集合，人们不需要完全了解它的所有特性，就可以用它来编写优美和有用的程序。希望读者能喜欢！

#### 如何使用本书：

本书可以作为教材，也可以作为自学者的参考教材。

1. 本书的主要目的是为科研人员和学生同时教授C++和数值计算。首先介绍C++，然后是数值算法。
2. 它也可以为只想了解基本和高级C++特性的读者使用。这样有关数值计算的章节就可以省略了，并且不需要微积分和线性代数的知识。但是读者还是需要知道什么是向量和矩阵，知道向量和矩阵上的基本运算，如向量求和、标量一向量相乘以及矩阵一向量相乘等等。
3. 本书可以作为学习数值方法的参考教材，因为书中有很多的数值计算源程序。初次接触数值方法的学生实现算法时往往会遇到困难。书中的代码可能会对他们有帮助并能加深他们的理解。
4. 对于熟练的C++程序员来说，本书也可以作为参考。它包括了ISO/ANSI C++的所有特性，以及书写高效C++程序的技巧。比如，在大多数介绍C++的书上介绍的对向量运算采用标准的运算符重载的方法，可能会比用C或FORTRAN慢几倍。而在本书中给出了一些技巧，使得C++的代码不比相应的C或FORTRAN慢。同时，还给出了使用静态多态来代替某些纯虚函数的例子，以提高运行效率。其他的高级技巧还包括表达式模板和模板元程序。

#### 致谢：

本书已经作为韦恩州立大学（Wayne State University）的本科生的C++和数值计算课程的教科书。作者愿借此机会向他的学生们表示感谢。感谢他们就本书所开展的讨论和对本书提出的建议。作者还感谢编辑和技术审稿人，是他们认真的审阅和建议使得本书的质量得以提高。作者也欢迎读者通过电子邮件对本书提出批评和建议。作者的电子邮件地址是dyang@na-net.ornl.gov。

# 目录

|                  |    |
|------------------|----|
| 第 1 章 基本类型       | 1  |
| 1.1 一个例程         | 1  |
| 1.2 类型与声明        | 6  |
| 1.3 基本类型         | 7  |
| 1.3.1 整数类型       | 7  |
| 1.3.2 字符类型       | 9  |
| 1.3.3 浮点类型       | 11 |
| 1.3.4 布尔类型       | 13 |
| 1.3.5 Void类型     | 13 |
| 1.4 数值极限         | 14 |
| 1.5 标识符和关键词      | 17 |
| 1.5.1 标识符        | 17 |
| 1.5.2 关键词        | 19 |
| 1.6 练习           | 19 |
| 第 2 章 表达式和语句     | 25 |
| 2.1 作用域和存储分类     | 25 |
| 2.1.1 局部变量和全局变量  | 25 |
| 2.1.2 外部变量和寄存器变量 | 27 |
| 2.2 表达式          | 28 |
| 2.2.1 算术表达式      | 28 |
| 2.2.2 关系表达式      | 29 |
| 2.2.3 逻辑表达式      | 30 |
| 2.2.4 位运算表达式     | 31 |
| 2.2.5 逗号表达式      | 32 |
| 2.3 语句           | 33 |
| 2.3.1 声明和初始化     | 33 |
| 2.3.2 赋值语句       | 34 |

|              |             |           |
|--------------|-------------|-----------|
| 2.3.3        | 复合赋值语句      | 34        |
| 2.3.4        | 增值减值语句      | 35        |
| 2.3.5        | 复合语句        | 35        |
| 2.3.6        | 条件语句        | 36        |
| 2.3.7        | 循环语句        | 40        |
| 2.4          | 斐波纳契数       | 44        |
| 2.5          | 练习          | 47        |
| <b>第 3 章</b> | <b>派生类型</b> | <b>51</b> |
| 3.1          | 常量和宏        | 51        |
| 3.2          | 枚举类型        | 52        |
| 3.3          | 数组          | 54        |
| 3.4          | 结构          | 56        |
| 3.5          | 联合和比特域      | 57        |
| 3.6          | 指针          | 60        |
| 3.6.1        | 指针运算        | 64        |
| 3.6.2        | 多重指针        | 64        |
| 3.6.3        | 偏移量指针       | 67        |
| 3.6.4        | 常量指针        | 69        |
| 3.6.5        | Void 指针和空指针 | 70        |
| 3.6.6        | 结构指针        | 71        |
| 3.6.7        | 字符指针        | 72        |
| 3.6.8        | 指针和数组       | 73        |
| 3.7          | 引用          | 73        |
| 3.8          | 函数          | 74        |
| 3.8.1        | 函数声明和定义     | 74        |
| 3.8.2        | 函数重载        | 75        |
| 3.8.3        | 参数传递        | 76        |
| 3.8.4        | 返回值         | 79        |
| 3.8.5        | 递归函数        | 82        |
| 3.8.6        | 内联函数        | 82        |
| 3.8.7        | 缺省参数        | 83        |
| 3.8.8        | 函数类型        | 84        |
| 3.8.9        | 静态局部变量      | 85        |
| 3.8.10       | main 函数     | 86        |
| 3.9          | 程序的运行空间     | 88        |
| 3.10         | 运算符概要及优先级   | 89        |
| 3.11         | 标准数学函数库     | 93        |

|                            |            |
|----------------------------|------------|
| 3.12 多项式求值 .....           | 96         |
| 3.13 梯形公式和Simpson公式 .....  | 99         |
| 3.14 练习 .....              | 102        |
| <b>第 4 章 命名空间和文件</b> ..... | <b>113</b> |
| 4.1 命名空间 .....             | 113        |
| 4.1.1 使用声明和指令 .....        | 117        |
| 4.1.2 多重接口 .....           | 119        |
| 4.1.3 命名空间别名 .....         | 120        |
| 4.1.4 无名命名空间 .....         | 121        |
| 4.1.5 名称查找 .....           | 122        |
| 4.2 包含文件 .....             | 122        |
| 4.2.1 包含标准库文件 .....        | 122        |
| 4.2.2 用户自定义头文件 .....       | 126        |
| 4.2.3 条件包含指令 .....         | 128        |
| 4.2.4 文件包含 .....           | 129        |
| 4.3 源文件和连接 .....           | 130        |
| 4.3.1 独立编译 .....           | 130        |
| 4.3.2 外部连接和内部连接 .....      | 130        |
| 4.3.3 与其他语言连接 .....        | 134        |
| 4.4 一些有用的工具 .....          | 134        |
| 4.4.1 给程序计时的方法 .....       | 135        |
| 4.4.2 编译选项和调试器 .....       | 137        |
| 4.4.3 创建库 .....            | 139        |
| 4.4.4 Makefile .....       | 140        |
| 4.5 字符串标准函数库 .....         | 143        |
| 4.5.1 声明和初始化 .....         | 143        |
| 4.5.2 操作 .....             | 144        |
| 4.5.3 C语言字符串 .....         | 145        |
| 4.5.4 输入输出 .....           | 145        |
| 4.5.5 C字符串函数库 .....        | 146        |
| 4.6 流标准函数库 .....           | 147        |
| 4.6.1 整数格式化输出 .....        | 147        |
| 4.6.2 浮点数格式化输出 .....       | 149        |
| 4.6.3 输出宽度 .....           | 150        |
| 4.6.4 文件的输入输出 .....        | 151        |
| 4.6.5 字符的输入输出 .....        | 154        |
| 4.6.6 字符串流 .....           | 158        |

|              |              |            |
|--------------|--------------|------------|
| 4.7          | 非线性方程的迭代解法   | 160        |
| 4.7.1        | 二分法          | 161        |
| 4.7.2        | 牛顿法          | 167        |
| 4.8          | 练习           | 170        |
| <b>第 5 章</b> | <b>类</b>     | <b>175</b> |
| 5.1          | 类的声明与定义      | 175        |
| 5.2          | 拷贝构造函数和拷贝赋值  | 182        |
| 5.3          | 友元           | 187        |
| 5.4          | 静态成员         | 188        |
| 5.5          | 常量和可变成员      | 190        |
| 5.6          | 类的对象作为成员     | 192        |
| 5.7          | 类的数组         | 194        |
| 5.8          | 成员指针         | 196        |
| 5.9          | 常微分方程的数值解法   | 197        |
| 5.10         | 练习           | 201        |
| <b>第 6 章</b> | <b>运算符重载</b> | <b>205</b> |
| 6.1          | 复数           | 205        |
| 6.1.1        | 初始化          | 208        |
| 6.1.2        | 缺省拷贝构造和赋值    | 208        |
| 6.1.3        | 转换和混合模式运算    | 209        |
| 6.2          | 运算符函数        | 209        |
| 6.3          | 向量和矩阵        | 213        |
| 6.4          | 显式和隐式类型转换    | 224        |
| 6.5          | 效率和运算符重载     | 227        |
| 6.6          | 共轭梯度算法       | 230        |
| 6.7          | 练习           | 233        |
| <b>第 7 章</b> | <b>模板</b>    | <b>235</b> |
| 7.1          | 类模板          | 235        |
| 7.1.1        | 成员和友元定义      | 237        |
| 7.1.2        | 模板实例化        | 238        |
| 7.1.3        | 模板参数         | 239        |
| 7.1.4        | 类型等价         | 239        |
| 7.1.5        | 用户定义的特化      | 239        |
| 7.1.6        | 特化顺序         | 244        |
| 7.2          | 函数模板         | 244        |
| 7.2.1        | 函数模板参数       | 246        |

|              |                            |            |
|--------------|----------------------------|------------|
| 7.2.2        | 函数模板重载                     | 246        |
| 7.2.3        | 特化                         | 248        |
| 7.2.4        | 类模板作为函数模板参数                | 249        |
| 7.2.5        | 成员函数模板                     | 251        |
| 7.2.6        | 友元函数模板                     | 251        |
| 7.3          | 模板源代码组织                    | 253        |
| 7.4          | 标准复数库                      | 254        |
| 7.5          | 标准 <code>valarray</code> 库 | 257        |
| 7.5.1        | <code>valarray</code> 类型   | 257        |
| 7.5.2        | 分段数组                       | 259        |
| 7.5.3        | 广义分段数组                     | 261        |
| 7.5.4        | 掩码数组和间接数组                  | 263        |
| 7.6          | 数值算法标准函数库                  | 264        |
| 7.6.1        | 累加                         | 265        |
| 7.6.2        | 内积                         | 268        |
| 7.6.3        | 部分和                        | 269        |
| 7.6.4        | 临近差分                       | 270        |
| 7.7          | 数值积分的高效技术                  | 270        |
| 7.7.1        | 函数对象方法                     | 271        |
| 7.7.2        | 函数指针作为模板参数                 | 272        |
| 7.7.3        | 使用点积和模板表达式                 | 273        |
| 7.7.4        | 采用点积和模板元程序                 | 278        |
| 7.8          | 多项式插值                      | 280        |
| 7.8.1        | 拉格朗日形式                     | 281        |
| 7.8.2        | 牛顿形式                       | 282        |
| 7.9          | 练习                         | 287        |
| <b>第 8 章</b> | <b>类的继承</b>                | <b>289</b> |
| 8.1          | 派生类                        | 289        |
| 8.1.1        | 成员函数                       | 291        |
| 8.1.2        | 构造函数和析构函数                  | 292        |
| 8.1.3        | 拷贝                         | 292        |
| 8.1.4        | 类层次结构                      | 293        |
| 8.1.5        | 虚函数                        | 294        |
| 8.1.6        | 虚析构函数                      | 295        |
| 8.2          | 抽象类                        | 297        |
| 8.3          | 访问控制                       | 301        |
| 8.3.1        | 访问成员                       | 301        |

|               |                 |            |
|---------------|-----------------|------------|
| 8.3.2         | 基类访问            | 302        |
| 8.4           | 多重继承            | 305        |
| 8.4.1         | 去除二义性           | 306        |
| 8.4.2         | 重复基类            | 308        |
| 8.4.3         | 虚基类             | 309        |
| 8.4.4         | 多重继承中的访问控制      | 310        |
| 8.5           | 运行时的类型信息        | 312        |
| 8.5.1         | 动态投影机制          | 312        |
| 8.5.2         | 类型标识机制          | 314        |
| 8.5.3         | 运行时负担           | 315        |
| 8.6           | 用静态多态代替虚函数      | 315        |
| 8.7           | 练习              | 323        |
| <b>第 9 章</b>  | <b>异常处理</b>     | <b>325</b> |
| 9.1           | 抛出和捕获           | 325        |
| 9.2           | 派生异常            | 329        |
| 9.3           | 捕获异常            | 331        |
| 9.3.1         | 重抛出             | 332        |
| 9.3.2         | 捕获所有异常          | 332        |
| 9.3.3         | 处理函数的顺序         | 333        |
| 9.4           | 在函数中指定异常        | 333        |
| 9.5           | 标准异常            | 335        |
| 9.6           | 练习              | 337        |
| <b>第 10 章</b> | <b>容器和算法标准库</b> | <b>339</b> |
| 10.1          | 标准容器            | 339        |
| 10.1.1        | 向量              | 340        |
| 10.1.2        | 链表              | 347        |
| 10.1.3        | 映射与集合           | 349        |
| 10.1.4        | 栈和队列            | 353        |
| 10.2          | 标准算法            | 354        |
| 10.2.1        | 排序、复制和替换算法      | 354        |
| 10.2.2        | 搜索和遍历算法         | 362        |
| 10.2.3        | 集合、排列和堆算法       | 368        |
| 10.3          | 标准函数对象和适配器      | 373        |
| 10.3.1        | 算术函数对象          | 373        |
| 10.3.2        | 关系函数对象          | 374        |
| 10.3.3        | 逻辑函数对象          | 375        |

|                               |            |
|-------------------------------|------------|
| 10.3.4 标准适配器 . . . . .        | 375        |
| 10.4 练习 . . . . .             | 376        |
| <b>第 11 章 线性方程组求解法</b>        | <b>379</b> |
| 11.1 矩阵存储格式 . . . . .         | 379        |
| 11.1.1 满矩阵 . . . . .          | 380        |
| 11.1.2 带状矩阵 . . . . .         | 380        |
| 11.1.3 稀疏矩阵 . . . . .         | 382        |
| 11.2 矩阵类层次 . . . . .          | 383        |
| 11.3 迭代算法 . . . . .           | 394        |
| 11.3.1 共轭梯度方法 . . . . .       | 394        |
| 11.3.2 广义最小残差法 . . . . .      | 398        |
| 11.3.3 预处理技术 . . . . .        | 407        |
| 11.4 高斯消元法 . . . . .          | 410        |
| 11.4.1 LU分解 . . . . .         | 411        |
| 11.4.2 高斯消元法 . . . . .        | 415        |
| 11.4.3 主元高斯消元法 . . . . .      | 417        |
| 11.5 求解偏微分方程的有限差分方法 . . . . . | 423        |
| 11.6 练习 . . . . .             | 434        |
| <b>参考文献</b>                   | <b>437</b> |
| <b>索引</b>                     | <b>439</b> |



# 第 1 章

## 基本类型

本章首先给出一个C++示例程序，然后引出整数和浮点数基本数据类型和两个特殊类型：*bool*和*void*。接下来，将介绍类型的数值极限，比如在特定的计算机上，C++程序能够实现的最大整数和最小双精度数。本章最后将对标识符和关键词加以讨论。

### 1.1 一个例程

一个完整的C++程序必须有一个名为*main*的函数，这个函数的语句包含在花括号{}之间。每条语句可以扩展为多行，但是每条语句必须以分号结尾。

注释部分应该包含在/\*和\*/之间，或者是以//开头的一行。上述第一种方式可以用来加入多行注释，而第二种方式用来加入单行注释。一种方式可以在另一种方式的作用范围之外提供注释功能。注释部分和空行会被编译器忽略，它们只是增加程序的可读性。

以下是一个计算两个给定的整数之间所有整数和的简单程序。运行程序后，用户会被提示通过标准输入流（通常是键盘）输入两个整数，*cin*就代表标准输入流。程序的运行结果会通过标准输出流（通常是显示器终端）输出，*cout*就代表标准输出流。输入输出流*cin*和*cout*在C++标准函数库<*iostream*>中定义。例程如下：

```
/* 一个用来说明C++的一些基本特征的简单程序。这个程序计算两个给定整数之间所有整数的和，并将结果输出到屏幕。*/
```

```
#include <iostream>          // 包含标准输入输出函数库
using namespace std;
main() {
    int n, m;                // 声明整型变量n和m
```