

937

Deitel 编程金典

TP312C

D43a

C++ 编程金典

(第3版)

[美] H. M. Deitel, P. J. Deitel 著

周 靖 黄都培 译

杨小平 审校

清华大学出版社

(京)新登字 158 号

内 容 简 介

作为全球使用最广泛的 C++ 经典程序设计教材, 本书详细介绍了过程式与面向对象程序设计的基本知识与方法, 其中包括 C++ 的强大功能、最新特性和新增的 ANSI/ISO C++ 标准库。

本书通过面向一个大型对象(电梯模拟系统)的程序设计, 以数百个“活代码”示例程序, 重点突出了利用 UML 进行面向对象的设计。分布在各章的“对象思想”、“案例分析”、“常见编程错误”、“良好编程习惯”、“自测题”和“练习题”等特色部分非常具有实际指导意义, 不仅可让接触 C++ 的新手真实体验编程乐趣, 还可让有经验的程序员得到启发。

本书的读者对象为计算机软件、系统和网络编程人员, 也可作为大学计算机相关专业本科生和研究生的编程教材和参考书。

C++ How To Program

H. M. Deitel, P. J. Deitel

Copyright © 2001 by Prentice Hall, Inc.

Original English language edition published by Prentice Hall, Inc.

All right reserved.

No part of the contents of this book may be reproduced or transmitted in any form or by any means without the written permission of the publisher. For sale in the People's Republic of China Only.

本书中文简体版由 Prentice Hall, Inc. 授权清华大学出版社出版发行, 未经出版者书面许可, 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 01-2002-3036 号

版权所有, 翻印必究。

本书封面贴有清华大学出版社激光防伪标签, 无标签者不得销售。

图书在版编目(CIP)数据

C++ 编程金典/(美) H. M. 迪特尔, (美) P. J. 迪特尔著; 周靖, 黄都培译.

—北京: 清华大学出版社, 2002

(Deitel 编程金典)

ISBN 7-302-05785-0

I. C... II. ①H... ②P... ③周... ④黄... III. C 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字(2002)第 060128 号

出 版 者: 清华大学出版社(北京清华大学学研大厦, 邮编 100084)

<http://www.tup.tsinghua.edu.cn>

责任编辑: 文开棋

印 刷 者: 清华大学印刷厂

发 行 者: 新华书店总店北京发行所

开 本: 787×1092 1/16 印张: 64.75 字数: 1570 千字

版 次: 2002 年 9 月第 1 版 2002 年 9 月第 1 次印刷

书 号: ISBN 7-302-05785-0/TP·3422

印 数: 0001~4000

定 价: 118.00 元

《Deitel 编程金典》丛书序

计算机技术的日新月异,带动着软件行业的迅猛前进。软件几乎成为各行各业的核心,而编程语言又是其中最关键的因素。为使国内程序员了解编程语言的发展动态,使他们为以后的工作实践打好基础,我们特别针对国外著名的计算机编程教材,进行了广泛的调查研究和筛选,最后确定了 Deitel & Associates 公司的 How To Program 系列(即 **Deitel 编程金典**)。这套丛书是最受欢迎的编程教材,全球销量超过百万册。该丛书深入浅出,通俗易懂,包括 Perl, C, C++, XML, Python, Visual Basic, .NET 和 Java,结合最新的技术(如 UML),进行了深入的剖析。同时,还以 Deitel 父子独创的“活代码”形式(即在提供源代码的同时,给出实际的输出结果),提供了透彻的案例分析。此外,这套丛书还非常注重软件工程,强调良好的编程习惯,还辅之以条理清晰的提示、常见编程错误和难度不一的习题等。这不仅让刚接触编程的新手更容易理解,还能让经验丰富的程序员通过这套丛书的学习温故而知新。

Deitel & Associates 公司是一家成长迅速、享有国际盛誉的培训和出版公司,主要为全球各地的客户提供编程语言、因特网、万维网和对象技术的培训。他们提供有关各种编程语言和面向对象技术的培训课程。Deitel & Associates 公司的主管 Deitel 父子均毕业于麻省理工学院。他们的团队中,不乏素质良好的名校高材生和经验丰富、敬业严谨的高级技术人员,对技术和行业动态反应迅速,具有非常敏锐的洞察力。

为保持这套丛书的行文风格,又尽量贴近国内读者的需求,我们在筛选翻译人员时,要求非常严格(甚至于苛刻),最后选定了具有技术背景和丰富编程经验的资深译者。为尽量做到尽善尽美,我们对整套丛书的每一个环节(选题、翻译、编辑、版式设计、封面设计以及排版)都进行了细致的审查,期望能为读者献上丰盛的编程知识“套餐”。

这套丛书虽然经过长时间的雕琢,因篇幅较大,纰漏之处仍在所难免,我们期待着读者朋友提出宝贵意见。

清华大学出版社
2002 年 7 月

作者简介

H. M. Deitel 博士: Deitel & Associates 公司首席执行官,在计算机领域已有 40 年的工作经验,无论专业技术还是学校教育,均有非常高的造诣。他是全球知名的计算机科学导师以及培训班专业讲师。Deitel 博士拥有麻省理工学院的学士和硕士学位,以及波士顿大学的哲学博士学位。他参与过 IBM 和 MIT 的一系列领先于时代的虚拟内存操作系统项目,研究成果如今已广泛应用于 UNIX、Windows NT、OS/2 和 Linux 系统中。他从事大学教育 20 余年,与其子 P. J. Deitel 成立 Deitel & Associates 公司之前,一直担任波士顿大学计算机科学系主任。他创作及与他人合著了数十本书,并参与了多媒体产品的开发。更让人佩服的是,老当益壮的他,现在的出书量居然有增无减。多年来,他的作品已被翻译成简体中文、繁体中文、日语、俄语、西班牙语、朝鲜语、法语、波兰语以及葡萄牙语等,畅销全球。

P. J. Deitel: Deitel & Associates 公司执行副总裁,毕业于麻省理工学院的斯隆工商管理学院,主修信息技术。在 Deitel & Associates 公司,他负责向业内许多知名客户讲授 Java、C、C++、因特网/万维网课程,他们的客户有康柏、朗讯、Sun、NASA 肯尼迪航空中心、美国国家大风暴实验室(NSSL)、IBM 以及其他许多公司和机构。他负责为计算机机构联盟波士顿分部讲授 C++ 和 Java 课程。他目前正利用 Deitel & Associates、Prentice Hall 以及“美国技术教育网络”联合投资的一笔风险基金,利用卫星技术提供技术培训与课程。

Deitel 父子二人,先后参与了多本入门级大学计算机编程语言教材的编写工作,所有这些教材都非常畅销。

译者简介

周靖,资深计算机图书翻译作家,毕业于北京理工大学,性格严谨、认真、对待技术问题一丝不苟。对计算机有浓厚的兴趣,从事计算机类文章和书籍的翻译已有多年,早期主要以硬件为主,后来接触编程,转向对各种编程语言的研究与实践,C++ 和 Perl 是其常用的两种语言。翻译风格浅显易懂、用语恰当。

黄都培,中国政法大学计算机教研室主任、副教授,北京市计算机基础教育研究会理事、全国计算机基础教育研究会学术委员会委员、全国文科计算机基础教育研究会理事、计算机用户协会信息系统分会理事。从事计算机教学与应用工作 20 余年,主要方向为办公自动化及电子政务、信息管理与数据库应用、数据处理分析技术、计算机辅助管理、法学、社会分析统计与心理测试等,近两年发表有关论文十余篇。在国内多家出版社出版教材和专著多本。

审校者简介

杨小平,中国人民大学信息学院副院长、副教授,中国计算机用户协会信息系统分会副理事、秘书长,北京系统工程学会常务理事,国家教委文科计算机教学指导委员会委员。长期从事计算机信息系统教学与开发工作,主要专业领域为信息系统工程与软件工程。近年来,在专业学术刊物上发表数十篇专业论文,在上述领域出版专著、合著、译著、教材十余本。

因本书篇幅巨大,除上述主要翻译和审校人员外,参加翻译工作的还有李晋宏、姜昊、吴巧泉、肖林、尤晓东、林满山、沈金河、胡辉、李钦、陈华伟、查安霞、腾远方、林晓文、吉同路等人,参加审校的人员还有文开棋、潘旭燕、车立红、孟纯城、王松、管永川、另外,周迎春、徐海琳、王颖、张扬声也对全书的翻译做了很多协助工作,在此一并感谢!

前 言

欢迎进入 ANSI/ISO 标准C++ 的世界! 本书由一位老作者和一位年青作者撰写。老作者 H. M. Deitel 先生(1967 年毕业于麻省理工学院)已有 39 年的编程及教学经验。年青作者 P. J. Deitel 先生(1991 年毕业于麻省理工学院)已编了 18 年的程序,现已成为教学和写作的高手。老作者凭经验编程、教学;年青人则凭着好像永远用不完的旺盛精力努力工作。老作者希望程序有一个清晰的结构;年青人则希望程序具有强劲的性能。老作者追求程序的优雅与典范,年青人却更强调程序结果。两者结合在一起,本书便成为读者必备的工具书。我们的愿望很简单,希望读者在本书中找到有用的信息,既能感受到适度的挑战,又能获得无穷乐趣!

现在处于C++ 世界激动人心的变革时期,人们为了 ANSI/ISO C++ 标准的正式通过而欢呼雀跃。ANSI(美国国家标准协会)和 ISO(国际标准化组织)通力合作,为计算机世界制定了最重要的、具有里程碑意义的全球化标准之一。

在编写本书的第 2 版时,我们将它定位于大学的程序设计课程。当时,学生们主要学的是 Pascal 或 C,重点强调的是过程式编程思想。为计算机系的一二年级学生编写一本C++ 教科书,对我们而言无疑是一项艰巨的挑战。我们需要同时讲解两种编程思想,包括过程式编程(C++ 中仍然含有 C)和面向对象编程。这就要求我们在入门级别上,同时准备两份数量相当的材料。为此,我们采取的策略是在前 5 章中,用 C 风格的材料讲解基本数据类型、控制结构、函数、数组、指针、字符串和结构。然后在第 6 ~ 15 章中,讲解面向对象编程。

本书迅速成为全球使用最广的C++ 大学编程教材。而我们之所以推迟编写新版本,有以下两方面原因:

(1) 在这段时间内,C++ 正在经历活跃的开发过程,标准文档的新版草案频繁出现。但标准协会那一方,尚无明确的信号表明草案标准会在短时间内被接纳为正式标准。

(2) 我们一直在等候一个关键的信号,以便适时推出新版。1997 年 7 月,当 Bjarne Stroustrup 推出《The C++ Programming Language》的第 3 版时,那个信号终于到来了。Stroustrup 是C++ 之父,而他的书是C++ 语言最权威的著作。此时,C++ 的“新定义”对我们来说已经足够,完全可以出版第 3 版了。

后来又过了一段时间,我们转移了注意力,出版了 5 本有关 Java 书籍。但由于 ANSI/ISO C++ 草案标准的正式接纳时间迫在眉睫,所以我们的注意力又回到了C++ 身上。

《C++ 编程金典(第 3 版)》

我们对第 3 版进行了全面的审查,进行了上千处润色性的改动。我们还对书中的程序进行彻底的更新,使其更符合C++ 标准中“名称空间”的使用。

第 3 版最主要的新特点便是采用 UML(Unified Modeling Language,统一建模语言),实现

了一个完整的关于面向对象设计的案例分析。我们觉得,在现有的各种入门性编程教科书中,好像都缺乏对一个较大规模的面向对象设计项目的分析。这个案例分析是我们强烈推荐的,因为它能显著增强学生在第一年大学编程学习中的体验。这个案例分析为学生提供了一个绝佳的机会,可实际地接触一个上千行的C++ 程序。来自不同行业和学校的审稿人对其进行了严格的检查。

在本书以前的版本中,我们在第1~7章的末尾,包括了特殊的“对象思想”小节。这些小节将带领学生经历设计一个电梯系统软件模拟程序的全部步骤。我们要求学生完成所有这些步骤,并用自己编写的C++ 程序来实现这个设计。但在《C++ 编程金典(第3版)》中,我们完全修改了这个案例分析。在第1~7章末尾,以及在第9章末尾,我们会用“对象思想”小节展示一个循序渐进的、使用UML进行的面向对象设计。目前,UML是应用得最广泛的图形表示方案,特别适合面向对象的系统建模。UML是一种复杂的、具有丰富特性的图形化语言。在我们的“对象思想”小节中,会展示这些特性的一个简明的子集。然后,会利用这个子集,面向刚入门的面向对象设计者和编程人员,指导读者获得使用UML的初期设计经验。我们将采取一种完全解决好的格式,来展示这个案例分析。注意这并不是一个练习;相反地,它是一种面对面的学习过程,而且都有详细的、逐步指导的C++ 代码作为总结。

在前5章的每一章中,我们都将重点放在结构化编程的“常规”方法上,因为即将构建的对象至少一部分是用结构化编程的单元构建起来的。然后,我们会用一个“对象思想”小节结束每一章,在其中采取UML来介绍面向对象的理论。在这些“对象思想”小节,我们的目标是帮助学生建立起一种面向对象的思维方式。这样一来,从第6章开始,马上就能实际应用面向对象的编程概念。在第1章必读的“对象思想”小节中,我们讲解了基本概念(即“对象思想”)和术语(即“对象描述”)。在第2~5章可选的“对象思想”小节中,对更基本的问题进行了探讨,并准备利用面向对象设计(OOD)来解决一个颇具挑战性的问题。我们分析了一个典型的问题陈述,它要求构建一个系统,决定系统需要实现的对象,决定对象需要具有的属性,决定这些对象需要呈现的行为,并指定对象之间如何交互以满足系统要求。甚至在讲解如何编写面向对象的C++ 程序之前,就会做完上述全部事情。在第6章、第7章和第9章末的“对象思想”小节中,我们讨论了该面向对象系统的C++ 实现。

这个案例分析要比书中的其他项目大得多。读者走一遍这个完整的设计和实现过程,可获得可观的经验。项目使我们能合并讲解本书其他任何小节没有讨论到的一些主题,包括对象交互、有关句柄的深度讨论、使用引用和指针的原则以及使用提前声明来避免循环包含问题等等。掌握了这个案例,学生们将能更自信地迎接行业内各种实际项目的挑战。

“对象思想”小节

第2章,我们开始电梯模拟程序第一阶段的面向对象设计(OOD)——标识出实现模拟程序所需的类。此外还讲解了UML的用法例图、类图和对象图,以及关联、多重性、合成、参与者和链接的概念。

第3章探讨了实现电梯模拟程序所需的许多类属性。同时还讲解了UML状态图和活动图,讲解了事件和行动的概念,以及它们同这些图之间的关系。

第4章探讨了电梯模拟程序中类的多种操作(行为),介绍了UML顺序图,并讲解了在

对象之间发送消息的概念。

第5章探讨了多种合作关系(系统中各对象之间的交互),它们是实现电梯系统必需的,并用UML合作图来表示这种关系,还提供了—个因特网和Web资源列表,其中包含了UML 1.3规范以及其他参考材料、常规资源、教程、FAQ、文章、白皮书和软件等。

第6章用前面开发的UML类图来概括C++头文件,以便对所需的类进行定义,还讲解了系统中各对象使用的handle的概念,并开始学习如何用C++来实现handle。

第7章展示了一个用C++语言编写的、完整的电梯模拟程序(约有1000行代码),详细地、循序渐进地讲解了所有代码。这些代码直接基于以前小节创建的UML设计,并遵循我们讲解的所有良好编程习惯,包括使用静态和常量数据成员及函数等;还讨论了动态内存分配、通过handle进行合成和对象交互,以及如何利用提前声明来避免循环包容问题等等。

第9章,开始更新电梯模拟设计和实现,以便在其中集成“继承”的概念;还提供了进一步修改它的建议,使学生能利用以前小节展示的工具,自行设计和实现。

真心地希望这个最新的电梯模拟案例能同时为学生和老师提供—个富有挑战性、同时又极有意义的项目设计体验。我们采取—个精心开发的、循序渐进的面向对象过程,为我们的电梯模拟程序生成—个基于UML的设计。根据这个设计,我们采用—系列关键的编程记号法(包括类、对象、封装、可见性、合成和继承),生成了—个能实际工作的C++实现。

其他参考资料

我们经过艰苦工作,除写出—本极其出色的教科书之外,还精心制作了—大量参考资料,希望能对老师和学生们有用。有以下参考资料可供大家利用:

- 示例程序可从 www.deitel.com 下载。注意示例程序源代码已压缩成ZIP格式,必须使用像WinZip(www.winzip.com)或PKZIP(www.pkware.com)这样的ZIP工具把它们解到硬盘(注意,保留原来的目录结构)。选择解压的目标目录时,应指定—个单独的目录(比如cpphtp3e_examples)。
- 配套网站(www.prenhall.com/deitel)为老师和学生提供了大量有用的资源。为老师提供的资源包括教科书的附录(比如附录D“因特网和Web上的C++资源”)以及—个教学大纲管理器,以便对课程进行计划。至于为学生提供的资源,则包括每章学习目标、判断正误习题、每章重点、参考资料和—个消息板。
- 可自定义的PowerPoint Instructor Lecture Notes(PowerPoint教师讲义),提供许多完整的特性,包括每个程序的源码和讨论要点,以及主要的插图等等。这些讲义均可在www.deitel.com免费下载。

软件开发的革命

多年来,硬件以惊人的速度发展着,但由于这样或那样的原因,人们每一次试图让软件变得更快和更好时,总会遇到某些限制。如今,我们正处在—场轰轰烈烈的软件设计和编程革命的中期。这个革命基于标准化、可互换部件的记号法——这其实正是亨利·福特在T型福特车的年代所采用的方法。这些软件部件或组件即所谓的“对象”——更准确地说,就

是“类”，对象将在它们的基础上生成。

各种著名的面向对象语言中，最成熟的是施乐公司 Palo Alto 研究中心在 20 世纪 70 年代初开发成功的 Smalltalk。但目前使用最广泛的面向对象语言（用户群是 Smalltalk 的 10 倍）是 C++（该语言在 20 世纪 80 年代初，由 AT&T 公司的 Bjarne Stroustrup 等人开发成功）。在本书第 1 版和第 2 版的出版间隙中，另一个强大的竞争者问世了，它就是 Java 面向对象编程语言，该语言是 James Gosling 等人于 20 世纪 90 年代初在 Sun Microsystems 开发成功的。

为什么每隔 10 年，就会有一种意义重大的、新的面向对象编程语言问世呢？Smalltalk 作为一项研究成果，无疑颇为超前。C++ 则恰逢其时，可满足今天高性能系统编程和应用程序开发的需要。Java 则使开发者能创建具有高度可移植性的多媒体密集型和网络密集型因特网/Web 应用程序。

过程式编程、基于对象的编程、面向对象的编程和泛型编程

学习本书，您能掌握 C++ 的 5 个关键组件，以及 4 种同时期的编程思维模式：

(1) C 过程式编程——第 1~5 章和第 16~18 章：主要包括数据类型、控制结构、函数、数组、指针、字符串、结构、位处理、字符处理、预处理等等。

(2) C++ 在 C 基础上进行的过程式编程改进——3.15~3.21 节：主要包括内联函数、引用、默认参数、函数重载和函数模板。

(3) C++ 基于对象的编程——第 6~8 章：主要包括抽象数据类型、类、对象、封装、信息隐藏、成员访问控制、构造函数、析构函数、软件重用性、常量对象和成员函数、合成、友元、动态内存分配、static 成员、this 指针等等。

(4) C++ 面向对象编程——第 9~15 章，第 19 章和第 21 章：主要包括基类、单一继承、派生类、多重继承、虚拟函数、动态绑定、多态性、纯虚拟函数、抽象类、具体类、流输入/输出、类模板、异常处理、文件处理、数据结构、作为全功能对象的字符串、数据类型 bool、强制类型转换操作符、名称空间、运行时类型信息 (RTTI)、显式构造函数和 mutable 成员。

(5) C++ 泛型编程——第 20 章，是本书篇幅最长的一章；主要包括标准模板库 (STL)、模板化容器、顺序容器、关联容器、容器适配器、用于遍历模板容器的迭代器以及用于处理模板化容器元素的算法等等。

从“Pascal 和 C”向“C++ 和 Java”的进步

C++ 已经代替了 C，成为软件业首选的系统实现语言。但 C 编程仍是一项重要而有价值的技能，因为还有大量的 C 遗留代码需要维护。H. M. Deitel 博士在大学里已有 20 年教授编程入门课程的经验，他把教学重点放在如何设计思路清晰、结构良好的程序代码上。在这些课程中，他教得最多的便是编程的基本原理，强调控制结构和过程的有效运用。本书正是采用 H. M. Deitel 授课方式来撰写的。这样做有一定的局限性；但一旦碰到这些局限，我们会指出来，并解释有效处理方式。根据我们往来的经验，学生在使用本书的学习材料时，会遵循他们上 Pascal 语言课或 C 语言课时一样的规则。不过，这里却存在一点明显的差异：由

于他们现在要学习的是一种时髦的编程语言(C++),并采用一种时髦的编程思维模式(面向对象编程),所以会极大激发他们的学习兴趣。因为他们一旦离开校园,就能学以致用。由于C++比Pascal或C难学得多,所以兴趣的培养至关重要。

我们的目标很明确:面向刚跨入编程领域的学生(具备较少或根本没有编程基础),编写一本C++编程教材,作为大学计算机入门编程课程。与此同时,不管在理论上还是在实践上,这本书又不乏传统的、高级的C++课程所要求的深度与严密性。为达到这个目的,我们编写了这本比其他C++教材更厚的书——因为书中详细地讲解了过程式编程、基于对象的编程、面向对象编程和泛型编程的原理。全世界已有成千上万的人在使用本书。

直到20世纪90年代初,计算机专业课的重点仍然是过程式编程,采用的语言是Pascal和C。但自那时起,这些课程便在很大程度上逐渐变成了强调面向对象的编程,采用的语言也变成了C++和Java。Deitel & Associates公司的重点是为当今流行的编程语言编写高质量的教学材料。本书付印之后,我们的工作重心将转移到《Java程序设计(第4版)》、《高级C++程序设计》和《高级Java程序设计》。

从第1章开始介绍面向对象

策划本书写作提纲时,我们面临着一个艰巨的挑战。本书应当采取一种纯粹的面向对象的教学方式,还是采取“过程式编程”和“面向对象编程”兼而有之的形式。

许多采用本书教学的老师都同时讲授过程式编程(C或Pascal)。C++本身并非一种纯粹的面向对象语言。相反,它是一种混合型语言,同时支持过程式和面向对象编程。

所以,我们选择了以下的方式。本书前5章介绍用C++进行过程式编程。它们揭示了计算机概念、控制结构、函数、数组、指针和字符串。这些章包括C++原样保留的C部分,以及C++在C基础上进行的“过程式编程改进”。

为使前5章独具特色,我们进行了一些工作。每章末尾都有一个特别的“对象思想”小节。这些小节讲解了面向对象的概念和术语,帮助学生熟悉对象是什么,以及它们有哪些行为。

第1章的“对象思想”一节讲解了面向对象的概念和术语。第2~5章的“对象思想”小节则提供了一个实际的面向对象系统项目的需求规范,目的是构建一个电梯模拟系统。同时,精心引导学生经历面向对象设计过程的每个典型阶段。这些小节讨论了如何标识问题中的对象,如何指定对象的属性行为,以及如何指定对象之间的交互。当学生学完5章的内容后,便完成了电梯模拟系统的所有面向对象设计,并准备好开始用C++来编写电梯模拟程序。第6章和第7章讲解了数据抽象和类。这两章也有“对象思想”小节,带领学生完成用C++编写这个电梯模拟程序的各个阶段。第9章的“对象思想”小节则将C++继承的概念应用于该程序。

本书简介

《C++编程宝典(第3版)》提供了丰富的示例、习题和项目,它们选自多个不同的领域,

为学生提供解决有趣的实际问题的机会。本书重点讲解了如何才能良好地开发一个软件工程,并强调了清晰与严谨的编程思路的重要性。我们在讲授一个程序实例时,尽量避免采用某些晦涩难懂的术语和语法规范。

本书由教育界的专家写成。他们大部分时间都致力于讲授和编写与流行编程语言有关的知识。

本书针对学习目标进行了精心的设计。例如,针对每一个新的C++ 或面向对象编程概念,都会用一个完整的、能实际运行的C++ 程序来阐述。同时还附有程序的实际输出结果。读者在看这些程序时,感觉和在一台计算机上自行输入和运行它们无异。我们把它叫做“活代码(Live-Code)方式”。

本书中,其他辅助教学方式还包括每章开头的**学习目标**;每章正文中穿插的和在每章结束时总结的一系列**常见编程错误**、**良好编程习惯**、**性能提示**、**可移植性提示**、**软件工程知识**和**测试和调试提示**;每章末尾都附有**小结和术语**;每章都提供了大量的**自测题及答案**;以及所有C++ 课本都具备的大量**练习题**。

练习题包括简单的复习题、较长的编程题以及大型项目等等。在第3~21章的练习题中,老师能找到许多合适的问题作为学期实习项目使用。为了教师和学生,我们精心编写了练习题。

写本书时,我们使用过各种各样的C++ 编译器,努力保证本书的程序适用于所有ANSI/ISO编译器。

本书内容以“Accredited Standards Committee X3, Information Technology”及其“Technical Committee X3J16, Programming Language C++”开发的C++ 编程语言为基础。该语言已获国际标准组织(ISO)批准。欲知详情,请寄信到:

X3 Secretariat

1250 Eye Street NW

Washington DC 20005

严谨的程序员应当仔细阅读这些文档。这些文档不是教程,但严格而精确地定义了C++ 和 C,是编译器开发厂商和高级开发人员必备的参考读物。

我们已参照这些文档,仔细审阅了本书。由于本书属于入门级和中级课程,因此不打算讨论这些文档中提到的全部特性。

学习目标

每章开始都列有“学习目标”。其作用是介绍本章主题。而且在学完一章后,读者都应回过头去核查学习目标,看看自己是否已达到所有这些学习目标?这部分内容有助于树立读者的信心,并对后续的学习起到积极的促进作用。

268 个示例程序和 13 741 行代码,并附输出结果

我们用大量完整的、能实际运行的程序来讲解C++ 的各种特性;每个程序之后,都有一个窗口,其中包含了程序的运行结果——我们将这种方式称为“活代码方式”。采取这种做法,有助于学生查实程序是否按原先的设计要求运行。将输出结果同产生这些结果的程序

语句联系起来,对于概念的学习与巩固,将起到令人意想不到的作用。在我们的示例程序中,将通过具体运用C++各种特性来进行练习。只要仔细阅读本书,便会发现它的效果其实非常类似于在一台计算机上敲入并运行这些程序。

469 个插图/图表

本书包含众多插图和图表。第2章针对控制结构的讨论提供了精心绘制的流程图(注意:尽管书中并未将流程图作为一种程序开发工具正式介绍给大家,但在书中确实使用了一系列流程图,向读者指出C++控制结构的准确运作方式)。第15章则用线条图揭示了链接列表、队列、堆栈和二叉树的创建及维护。本书的其余部分也提供了其他形式的图。

625 条编程提示

在本书中,我们采用6大设计元素帮助学生将重点放在程序开发、测试和调试以及性能和可移植性的各个重要方面。我们采用以下形式,强调了数百个类似的提示,包括:良好编程习惯、常见编程错误、性能提示、可移植性提示、软件工程知识以及测试和调试提示等。所有这些提示和编程习惯是我们60年(几个人加在一起)的编程及教学经验的结晶。最近,我们的一名学生(数学硕士)告诉我们,她感觉这些“提示”就像数学教科书中的那些公理、定理和推论一样,牢记这些东西,有助于构建出最出色的软件。

112 条良好编程习惯

良好编程习惯将在正文中突出显示,目的是引导学生关注更好的编程技术。以前,我们向非程序员出身的人讲授入门级课程时,每堂课都在强调是“清晰性”。同时还指出,只要某种技术有助于写出结构更清晰、更易理解和更易维护的程序,便会对其重点介绍(本书中,则以“良好编程习惯”的形式表达出来)。

216 条常见编程错误

学生们在学习一种语言时(特别是上第一门程序课时),经常会犯某些错误。把注意力放在这些“常见编程错误”上,有助于避免再犯同样的错误。此外,这样也可大大减少学生们心中的疑问。

87 条性能提示

根据我们的经验,对于第一门程序课来说,教学生们写出清晰的、易于理解的程序仍然是迄今为止最首要的目标。然而,对年轻而又冲动的学生们来说,他们通常一开始便想编出速度奇快、占用资源最少以及敲击键盘次数最少的程序。事实上,他们最关心的是程序的“性能”。他们希望知道对自己的程序进行哪些调节之后,便可改善性能。尽管我们并不鼓励一开始便这样做,但由于确实有这方面的需求,所以本书仍然提供了一些“性能提示”,为部分学生提供改善程序性能的机会。

32 条可移植性提示

软件开发是一个复杂而又成本极高的过程。开发软件的公司通常需要开发不同版本的软件,以适应不同计算机、不同操作系统的需要。因此,如今的软件开发项目无一例外地对可移植性提出了严格要求。换言之,一个软件应该只需少量修改(或者根本不用修改),便能在其他平台上运行。

146 条软件工程知识

软件工程知识强调的是技术、结构以及设计等问题,均与软件工程有关。这些问题会大大影响软件系统的结构及构建机制(大型系统尤其如此)。这些知识往往会在以后学生们深造或正式工作时起到关键作用——一旦他们开始接触大型的、复杂的、现实世界的系统。

27 条测试和调试提示

严格地说,这部分内容恐怕算不上什么“提示”。我们决定将“测试和调试提示”编入本书时,曾讨论过这些“提示”其实应该算是一种建议,测试程序可参照这些建议找出并消除程序中的 Bug。事实上,大多数提示都是对C++ 功能与特性的一种“认识”,它们可从源头阻止 Bug 进入我们的程序。

小结

每章的最后一节是一个小结。目的是想帮助学生们复习和巩固一些关键概念。平均每章有 37 条小结。

术语

术语部分将按字母顺序,列出各章出现的重要技术词汇,有助于帮助学生进一步巩固各章介绍的知识。

提示、编程习惯和错误一览

我们列举当前章出现的所有良好编程习惯、常见编程错误、性能提示、可移植性提示、软件工程知识以及测试和调试提示。

554 个自测题及答案

所有自测题和答案均针对自学而精心设计。学习这部分内容,学生有机会建立起深入学习的信心,并进一步巩固所学知识。完成这些自测题之后,参加正式考试时也会胸有成竹!

877 个练习题(此为分类合并后的数据;共计 1 431 个练习题)

在每一章最后,我们提供了一系列练习题,目的是帮助大家回忆重要的术语和概念、编写独立的C++ 语句、编写少数C++ 函数和类、编写完整的C++ 函数/类/程序以及编写重大的学期实习项目等等。根据这些练习题,教师可对自己的课程进行调整,以适应不同层次学生的需要。同时,每学期也可以相应地改变授课计划。以这些练习题为基础,教师可以灵活地、有针对性地安排家庭作业、随堂考以及大型测验。

550 页“教师手册”,附练习题答案

练习题的解决方案放在“教师手册”光盘上,并只可由教师向 Prentice Hall 出版社索取(注意,不要向我们写信索取教师手册。只有将本书作为教材使用的大学教师才有权获得。符合这一条件的读者,可向 Prentice Hall 的销售代表索取)。

本书概述

本书第 1~5 章对C++ 的过程式编程进行了全面讲解,内容涉及数据类型、输入和输出、控制结构、函数、数组、指针和字符串。第 1~5 章末尾的“对象思想”小节讲解了对象技术,展示了一个有趣而富有挑战性的可选案例分析,重点是设计和实现一个实际的面向对象

系统。

第6~8章实际讲解了类数据抽象、对象和操作符重载。这部分或许应该叫做“用对象编程”。第6章和第7章的“对象思想”小节开发和展示了一个1 000行以上的C++程序,用于实现第2~5章展示的设计。

第9章和第10章讲解了继承、虚拟函数和多态性——均为真正的面向对象编程所依赖的基本技术。第9章末尾的“对象思想”在电梯模拟程序的设计和实现中集成了“继承”的概念。

第11章和第14章展示了C++风格、面向流的输入和输出,需要随键盘、屏幕、文件和字符数组使用流I/O;同时讨论了顺序文件处理和直接访问(即随机访问)文件处理。

第12章和第13章讨论了C++中新增加的两部分内容,即模板和异常处理。模板也叫做“参数化类型”,它极大地促进了软件的重用能力。利用异常,程序员可以开发出健壮的、容错的、事务关键和任务关键的系统。

第15章全面探讨了动态数据结构,比如链接列表、队列、堆栈和树。本章的内容如果与第20章介绍的标准模板库(STL)配合,可更全面地理解数据结构,并可作为传统计算机科学数据结构和算法课程的一个良好的C++补充使用。

第16~18章讨论了一系列全面的主题,包括位、字符和字符串处理;预处理程序和其他主题。

第19~21章针对ANSI/ISO C++标准中对于C++和C++标准库进行的最新改进而专门设计。我们在此讨论了string类、字符串流处理、标准模板库以及针对C++进行的其他各种更新。

在本书最后,列出了一系列参考资料,它们是对正文的补充,其中包括操作符优先级、ASCII字符集、数值系统(二进制、十进制、八进制和十六进制)以及因特网和万维网上的C++资源等。接下来详细介绍各章。

第1章——计算机和C++编程概述:介绍计算机是什么,它们是如何工作的,以及如何为它们编程。其中讲解了结构化编程概念,并解释了这一系列技术为什么引发了程序编写方式的革命。本章简要介绍了编程语言的开发历史,从最开始的机器语言,一直发展到汇编语言,再发展到高级语言。讨论了C++编程语言的起源。本章提供了对一个典型C++编程环境的简介,并简要解释了C++程序的编写。针对在C++中如何作出决策以及算术操作符的问题,本章也进行了详细介绍。学完本章后,读者即可掌握如何编写简单而又完整的C++程序。我们讨论了因特网的爆炸性增长,并介绍了万维网和Java编程语言的起源。我们讨论了名称空间和using语句,以便读者使用同C++标准相容的编译器。我们采用了新样式的头文件。可能需要几年的时间,才能完全“消除”目前仍在广泛使用的较旧的编译器。在最后的“对象思想”小节,读者马上就能接触到面向对象的概念,其中讲解了对象技术的基本术语。

第2章——控制结构:介绍了用于解决问题的算法(过程)概念。解释了有效使用控制结构的重要性,这样做的目的是生成易于理解、调试、维护以及首次便可大致正确运行的程序。讲解了顺序结构、选择结构(if,if/else和switch)和重复结构(while,do/while和for)。我们详细解释了重复的概念,并对比了由计数器控制的循环和由标记(又称哨兵值)控制的循

环。还解释了自上而下求精法;利用它,可生成正确结构化的程序。同时,我们在本章还讲解了流行的程序设计辅助手段——伪代码。第2章讲解的方法和手段可保证控制结构在任何程序设计语言中的有效使用——而非仅仅在C++中。本章将帮助学生建立良好的编程习惯,以便在本书其余部分从容地应付更为实际的编程任务。本章最后还讨论了逻辑操作符&&(逻辑AND)、|| (逻辑OR)和!(逻辑NOT)。关键字表格已用ANSI/ISO C++标准中新的C++关键字进行了改进。我们介绍了新样式的static_cast操作符。与使用C++从C中继承的原有的强制类型转换相比,它显得更安全。我们增加了“Peter Minuit”练习题,让学生们认识到复利的奇迹——为了能让他们体验到这个奇迹,计算机可谓功不可没!我们针对for循环中的循环计数器,讨论了新的范围规则。在“对象思想”小节,我们将启动电梯模拟程序面向对象设计(OOD)的第一个阶段——标识出实现模拟程序所需的类。还讲解了UML用例、类和对象图,同时解释了关联、多重性、合成、参与者和链接的概念。

第3章——函数:讨论了程序模块的设计和构建。C++的函数相关功能包括标准库函数、程序员定义的函数、递归、按值调用和按引用调用功能。为了生成正确结构化的程序,特别是那些较大的程序和软件(由系统程序员和应用程序开发者在现实环境中开发),第3章讲解的技术是至关重要的。“分而治之”策略在此得到了讲述,它是解决复杂问题的有效手段,做法是将复杂的问题分解成较简单的、相互作用的组件。学生们都喜欢学习随机数和模拟方面的知识,他们会喜欢本章对于掷骰子程序的讨论,该程序良好地运用了控制结构。本章为递归的学习奠定了扎实的基础,并提供了一张表格,总结了会在本书其余部分讲解的大量递归例子和练习。有的教材喜欢将递归安排在最后一章;但我们感觉,这个主题最好均匀地分布于全书各处进行讲解。在本章末尾的60个练习题中,包括了几个经典的递归问题,比如汉诺塔。本章讨论了所谓的“C++对于C的增强”,包括内联函数、引用参数、默认实参、一元作用域分辨符、函数重载和函数模板。头文件已进行了修改,在其中包括了许多新的头文件,这些文件是读者贯穿整本书经常需要用到的。请务必完成练习题3.54,为掷骰子程序加入赌注功能。在“对象思想”小节,我们定下了大量类属性,它们是实现电梯模拟程序所需的。在介绍了UML状态图和活动图,还解释了同这些图关联的事件和行动的概念。

第4章——数组:讨论如何将数据结构化为“数组”,它是同类型的各个相关数据项的一种组合。本章展示了单下标和双下标数组的多个实例。人们公认,在开发正确结构化的程序时,“对数据进行正确的结构化”和“有效运用控制结构”具有同等的重要性。本章的各个例子探讨了各种标准数组操作、直方图的打印、数据排序、将数组传给函数以及问卷调查数据分析(使用简单的统计数据)。本章的特色内容是讨论了基本排序和搜索技术,并揭示了二元搜索与线性搜索相比所表现出来的巨大的性能提升。最后94个练习题包括了大量有趣而富有挑战性的问题,比如改进的排序技术,设计一个航班订票系统,海龟图概念入门(自LOGO语言而著名)以及骑士旅行和八皇后问题(它们引入了启发式编程的概念,目前已在人工智能领域得到了广泛采用)。在最后的练习题中,总结了许多递归问题,包括选择排序、回文、线性搜索、二元搜索、八皇后、打印一个数组、逆向打印字符串以及找出数组中的最小值等。本章仍然使用C风格的数组。正如第5章所述,它们实际上是指针,指向内存中的数组内容。我们肯定能将数组当作全功能的对象使用。第8章将利用操作符重载技术,加工一个有价值的Array类,以便通过它创建Array对象;而这种对象比第4章介绍的数组更

健壮,也更好用。第20章“标准模板库(STL)”中,还会介绍 STL 的 `vector` 类。如果将它与迭代器和算法结合使用,会将数组创建成一种全功能的对象。第4章的“对象思想”小节,解释了电梯模拟系统中的类的“操作”(行为)。另外,还介绍了 UML 顺序图以及在对象之间互传消息的概念。

第5章——指针和字符串:开始讲解C++语言中功能最强、但又最难掌握的特性之一,即“指针”。本章详细解释了指针操作符、按引用调用、指针表达式、指针算术、指针和数组的关系、指针数组和函数指针等等。在C++中,指针、数组和字符串存在密切的关系。所以,我们讲解了基本的字符串操作概念,并讨论了一些最流行的字符串处理函数,其中包括 `getline` (输入一行文本)、`strcpy` 和 `strncpy` (复制一个字符串)、`strcat` 和 `strncat` (连接两个字符串)、`strcmp` 和 `strncmp` (比较两个字符串)、`strtok` (将一个字符串“标记”为不同的部分)以及 `strlen` (计算字符串长度)等等。在49个练习题中,包括对经典龟兔赛跑问题的模拟、洗牌和发牌问题算法、递归快速排序和递归走迷宫等等。另外还包括一个特色部分,名为“建立自己的计算机”。本部分解释了机器语言编程,并实际进行一个项目,目的是设计和实现一个计算机模拟器,让读者编写和运行机器语言程序。对那些希望理解计算机的真正工作原理的读者来说,本书的这个特色格外有用。事实上,我们的学生非常喜欢这个项目,并经常对其进行改进,许多改进已体现在习题中。第15章中的另一个特色是指导读者自行构建一个编译器;编译器生成的机器语言会在第7章生成的机器语言模拟器上执行。在编译器和模拟器之间,信息将通过第14章讨论的顺序文件实现通信。第5章的第二个“特色部分”中,包含一个颇具挑战性的字符串操作练习,它涉及到文本分析、字处理、用各种格式打印日期、支票保护、写入支票金额的大写形式、莫尔斯码和公制-英制转换等等。读者可在完成了第19章中对于 `string` 类的学习后,回头复习这些字符串操作练习。许多人觉得,在入门性的编程课程中,指针无疑是最难学的。在C和“原始C++”中,数组和字符串其实根本就是指针,它们指向内存中的数组和字符串内容。即使函数名,同样也是指针。通过仔细研读本章,你能深入地理解指针的复杂主题。同样地,在本书的后面部分,我们会讨论如何将数组和字符串看作一种全功能的成熟的对象。在第8章,我们会利用操作符重载,加工自定义的 `Array` 和 `String` 类。在第19章,我们会讨论标准库的 `string` 类,并展示如何操作 `string` 对象。第20章,会讨论 `vector` 类,它能实现任何数组对象。第5章充满大量富有挑战性的练习题。请务必尝试一下“特色部分:建立自己的计算机”。在“对象思想”一节,我们会探讨实现电梯系统所需的许多“合作”(系统中各对象之间的交互),并用 UML 合作图来表示这些合作。还提供了一系列因特网和万维网资源,以便你参考 UML 1.3 规范以及其他辅助材料、常规资源、教程、FAQ、文章、白皮书和软件等等。

第6章——类和数据抽象(一):开始引入“基于对象编程”。本章是讲授“数据抽象”的绝好机会——通过一种语言(C++),明确地实现抽象数据类型(ADT)。近几年,数据抽象已经成为初级计算机课程的一个重要主题。第6~8章对数据抽象进行了全面的讲解。其中,第6章讨论了将ADT实现为 `struct`;将ADT实现为C++风格的 `class`,它为什么要优于 `struct`;对 `class` 成员的访问;如何将接口同实现分离;使用 `access` 函数和 `utility` 函数;用构造函数初始化对象;用析构函数删除对象;通过默认的“以成员为单位副本”进行赋值;以及软件的重用性。本章的习题要求学生为复数、有理数、时间、日期、矩形、大整数(`HugeInteger`)

和三连棋游戏分别开发类。学生通常都喜欢游戏程序。在“对象思想”小节,会要求你为电梯模拟程序中的每个类都编写一个类的头文件。那些爱好数学的读者会喜欢创建 Complex 类(复数)、Rational 类(有理数)和 HugeInteger 类(任意大的整数)的习题。在“对象思想”小节,我们用以前开发的 UML 类图概括 C++ 头文件,以便定义我们的类。另外还讲解了对象 handle(句柄)的概念,以及如何在 C++ 中实现 handle。

第 7 章——类和数据抽象(二):继续讨论类和数据抽象相关主题。本章讨论了声明和使用常量对象;常量成员函数;合成——构建类,它用其他类的对象作为自己的成员;友元函数和友元类,它们拥有对类的 private 和 protected 成员的特殊访问权限;this 指针,允许一个对象知道自己的地址是什么;动态内存分配;静态类成员,用于包含和操作类范围内的数据;流行抽象数据类型的例子(数组、字符串和队列);容器类和迭代器等等。在本章的习题中,要求学生开发一个储蓄账户(SavingsAccount)类,并用一个类来容纳整数集合。在对常量对象的讨论中,简要提到了 mutable 关键字。如同第 21 章所述,它的用法非常微妙,用于在常量对象中对“不可见”的实现进行修改。我们讨论了用 new 和 delete 进行的动态内存分配。一旦 new 失败,它会返回一个 0 指针(在 C++ 标准化之前)。我们会在第 7~12 章一直采用这种“标准化之前”的风格。第 13 章开始讨论新式的 new 失败。届时,new 会“抛出一个异常”。为了激发读者对静态类成员的兴趣,我们采用了一个基于视频游戏的示例。我们在本书以及以往的专业培训中,会一直强调在类的用户面前隐藏实现细节的重要性。然后,我们在类头文件中展示了 private 数据,它肯定会揭示实现细节。我们讨论了代理类,它是在类用户面前隐藏 private 数据的有效手段。“对象思想”小节要求你在电梯模拟程序中集成动态内存管理和合成功能。学生会喜欢创建 IntegerSet 类的练习。通过它,可以极大地刺激在第 8 章对操作符重载的学习兴趣。在“对象思想”小节,我们展示了一个完整的电梯模拟程序(约有 1 000 行代码),并逐步讲解所有代码。注意代码是直接根据以前各“对象思想”小节的 UML 设计建立起来的,同时还遵循了所有良好的编程习惯,其中包括使用静态和常量数据成员及函数。另外,还讨论了动态内存分配、合成和通过 handle 进行的对象交互,另外还讲解了如何利用“提前声明”,以避免循环包容问题。

第 8 章——操作符重载:这是 C++ 课程中最流行的主题之一。学生们通常会对本章的内容抱有浓厚的兴趣。他们会发现,本章内容同第 6 章和第 7 章讨论的抽象数据类型紧密相连。通过操作符重载,程序员可告诉编译器如何将现有的操作符应用于新类型的对象。C++ 已知如何将这操作符用于内建类型(比如整型、浮点型和字符型)的对象。但是,假定我们新建一个字符串类,那么在字符串对象之间使用加号时,它是什么意思呢?许多程序员将加号(+)用于字符串时,意思是“连接”。第 8 章将描述如何使用“重载”加号,使其在一个表达式中出现在两个对象之间时,编译器会生成对一个“操作符函数”的函数调用,以便连接两个字符串。本章讨论了操作符重载的基本知识,操作符重载时的限制,为类成员函数和非成员函数使用重载的对比,重载一元和二元操作符,以及不同类型间的转换等等。本章的一个特点是收集了许多实际的案例分析,包括一个数组类、一个字符串类、一个日期类、一个大整数类以及一个复数类(后两者及其完整源码出现在练习题中)。许多爱好数学的学生会喜欢在练习中创建多项式类。这里采取的做法不同于大多数程序语言和课程中的做法。操作符重载是一个复杂的主题,但同时也是一个重要的主题。巧妙运用操作符重载,可对你的