

<<双语版C++程序设计>>

图书基本信息

书名：<<双语版C++程序设计>>

13位ISBN编号：9787121107979

10位ISBN编号：712110797X

出版时间：2010-6

出版时间：电子工业出版社

作者：保罗·凯利

页数：388

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

前言

21世纪初的5至10年是我国国民经济和社会发展的关键时期，也是信息产业快速发展的关键时期。在我国加入WTO后的今天，培养一支适应国际化竞争的一流IT人才队伍是我国高等教育的重要任务之一。

信息科学和技术方面人才的优劣与多寡，是我国面对国际竞争时成败的关键因素。

当前，正值我国高等教育特别是信息科学领域的教育调整、变革的重大时期，为使我国教育体制与国际化接轨，有条件的高等院校正在为某些信息学科和技术课程使用国外优秀教材和优秀原版教材，以使我国在计算机教学上尽快赶上国际先进水平。

电子工业出版社秉承多年来引进国外优秀图书的经验，翻译出版了“国外计算机科学教材系列”丛书，这套教材覆盖学科范围广、领域宽、层次多，既有本科专业课程教材，也有研究生课程教材，以适应不同院系、不同专业、不同层次的师生对教材的需求，广大师生可自由选择 and 自由组合使用。这些教材涉及的学科方向包括网络与通信、操作系统、计算机组织与结构、算法与数据结构、数据库与信息处理、编程语言、图形图像与多媒体、软件工程等。

同时，我们也适当引进了一些优秀英文原版教材，本着翻译版本和英文原版并重的原则，对重点图书既提供英文原版又提供相应的翻译版本。

在图书选题上，我们大都选择国外著名出版公司出版的高校教材，如Pearson：Education培生教育出版集团、麦格劳-希尔教育出版集团、麻省理工学院出版社、剑桥大学出版社等。

撰写教材的许多作者都是蜚声世界的教授、学者，如道格拉斯·科默（Douglas E. Comer）、威廉·斯托林斯（William Stallings）、哈维·戴特尔（Halvey M. Deitel）、尤利斯·布莱克（Uyless Black）等。

为确保教材的选题质量和翻译质量，我们约请了清华大学、北京大学、北京航空航天大学、复旦大学、上海交通大学、南京大学、浙江大学、哈尔滨工业大学、华中科技大学、西安交通大学、国防科学技术大学、解放军理工大学等著名高校的教授和骨干教师参与了本系列教材的选题、翻译和审校工作。

他们中既有讲授同类教材的骨干教师、博士，也有积累了几十年教学经验的老教授和博士生导师。

在该系列教材的选题、翻译和编辑加工过程中，为提高教材质量，我们做了大量细致的工作，包括对所选教材进行全面论证；选择编辑时力求达到专业对口；对排版、印制质量进行严格把关。对于英文教材中出现的错误，我们通过作者联络和网上下载勘误表等方式，逐一进行了修订。

此外，我们还将与国外著名出版公司合作，提供一些教材的教学支持资料，希望能为授课老师提供帮助。

今后，我们将继续加强与各高校教师的密切联系，为广大师生引进更多的国外优秀教材和参考书，为我国计算机科学教学体系与国际教学体系的接轨做出努力。

<<双语版C++程序设计>>

内容概要

本书由在计算机程序设计方面有着丰富教学和实践经验的中外作者合作编写。

共分14章内容，由浅入深全面介绍C++程序设计方法。

本书通俗易懂，例子贴近生活，尤其强调读者的亲自参与意识。

所有实例经过精心挑选。

每章都为初学者提供了常见错误分析，每章结尾有很多有趣的习题，可以提高读者上机编程的兴趣。

本书是国内首次出版的中英文对照混排式双语版C++程序设计教材，既方便初学者熟悉相关概念和内容，也便于英文非母语的读者熟悉英文专业词汇。

本书可作为高等学校计算机相关专业或软件学院的C++程序设计双语教材，也可供程序员和编程爱好者参考使用。

<<双语版C++程序设计>>

书籍目录

Chapter One Introduction (绪论) 1.1 What is a computer program? (什么是计算机程序?) 1.2 Developing a computer program (开发计算机程序) 1.3 Learning C++ (学习 C++) 1.4 Web site for this book (本书的网站) 1.5 Brief history of C++ (C++简史) 1.6 ANSI/ISO C++ standard (ANSI/ISO C++标准) Chapter Two Beginning to Program in C++ (C++编程入门) 2.1 Constants (常量) 2.2 Variables (变量) 2.3 Simple output to the screen (简单的屏幕输出) 2.4 Comments (注释) 2.5 Data types (数据类型) 2.6 Data type sizes (数据类型的大小) 2.7 Operators (运算符) 2.8 Operator precedence (运算符的优先级) 2.9 Type conversions and casts (类型转换和强转) Programming pitfalls Quick syntax reference Exercises Chapter Three Keyboard Input and Screen Output (键盘输入和屏幕输出) 3.1 Simple keyboard input (简单的键盘输入) 3.2 Manipulators (流操纵符) 3.3 Single-character input and output (单个字符的输入和输出) Programming pitfalls Quick syntax reference Exercises Chapter Four Selection and Iteration (选择与循环) 4.1 Selection (选择) 4.2 Iteration (循环) Programming pitfalls Quick syntax reference Exercises Chapter Five Arrays and Structures (数组和结构体) 5.1 Arrays (数组) 5.2 Structures (结构体) 5.3 The typedef statement (typedef语句) 5.4 Arrays of structures (结构体数组) 5.5 Enumerated data types (枚举数据类型) Programming pitfalls Quick syntax reference Exercises Chapter Six Strings (字符串) 6.1 C-strings (C风格字符串) 6.2 C-string input and output (C风格字符串的输入和输出) 6.3 Accessing individual characters of a C-string (访问C风格字符串中的单个字符) 6.4 C-string functions (C风格字符串函数) 6.5 C++ strings (C++字符串) 6.6 Arrays of strings (string类型的数组) 6.7 Character classification (字符分类) Programming Pitfalls Quick Syntax Reference Exercises Chapter Seven Functions (函数) Chapter Eight Objects and Classes (对象和类) Chapter Nine Pointers and Dynamic Memory (指针和动态内存分配) Chapter Ten Operator Overloading (运算符重载) Chapter Eleven Inheritance (继承) Chapter Twelve Polymorphism (多态性) Chapter Thirteen Templates (模板) Chapter Fourteen Files and Streams (文件和流) Appendix A List of C++ Keywords Appendix B Precedence and Associativity of C++ Operators Appendix C ASCII Character Codes Appendix D Fundamental C++ Built-in Data Types Appendix E Common iomanip Manipulators Appendix F Escape Sequences Appendix G The C++ Preprocessor

章节摘录

第五步：测试程序。

程序在运行时有可能没有像预期的那样执行。

一个程序没有编译时错误并不能保证这个程序就能按照要求运行。

举个例子，编程人员可能错误地在一条指令中把0作为除数。

这种错误称为运行时错误，它会导致程序在完成它的任务之前中止。

虽然程序执行完毕，但是结果却是错误的，或者结果显示的屏幕位置是错误的。

这类错误称为逻辑错误或者俗称为“bugs（缺陷）”。

缺陷比编译时错误更难发现和修正。

某些bug仅在特定的条件下才出现，例如仅在特定的数据集上运行程序时才出现。

定位和修正程序错误的过程，称为调试。

第六步：调试程序。

bug一旦被确认，紧接着就要定位其在源代码中的位置。

很多编译器都提供了可用于帮助定位缺陷的工具。

改正一个缺陷后要返回第二步，但是希望不要返回到更前面去。

如果返回到第一步甚至是分析和设计阶段，就好像要求建筑师重新设计一座正在建造中的房子一样。

一般情况下，应该尽可能早地发现错误。

.....

<<双语版C++程序设计>>

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>