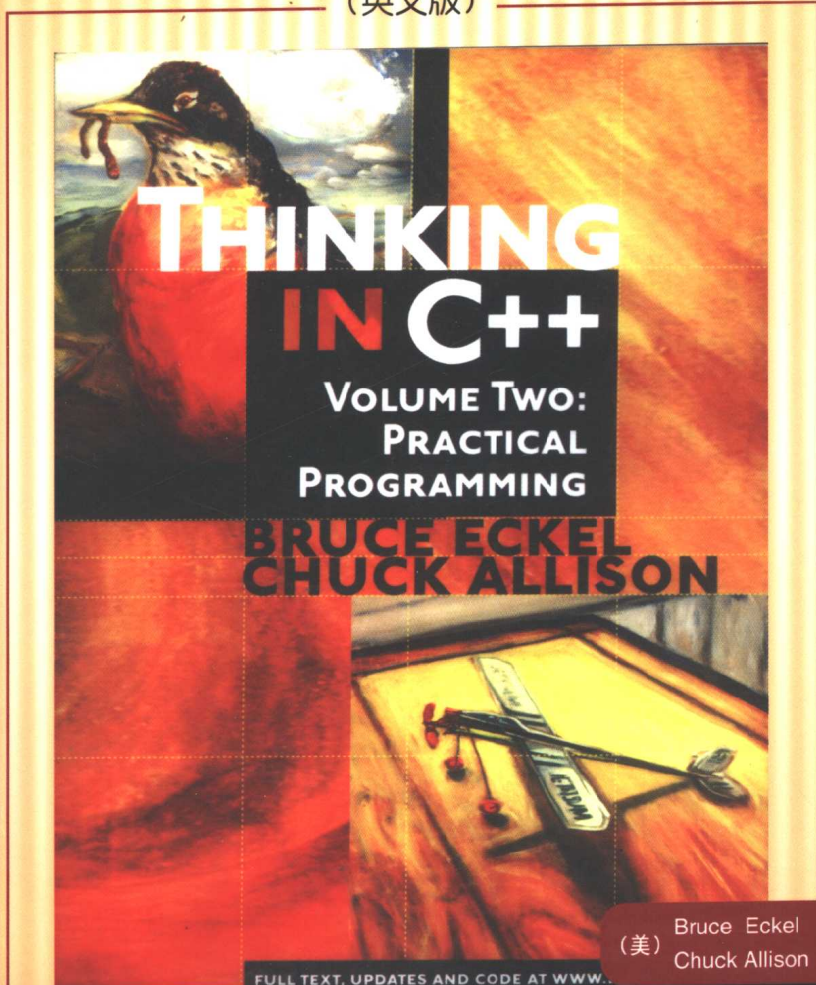


经 典 原 版 书 库

C++ 编程思想

第2卷 实用编程技术

(英文版)



FULL TEXT, UPDATES AND CODE AT WWW.



机械工业出版社
China Machine Press



TP312
Y188

经典原版书库

C++ 编程思想

第2卷 实用编程技术

(英文版)

Thinking in C++

Volume 2: Practical Programming

(美) Bruce Eckel 著
Chuck Allison

北方工业大学图书馆



00731283



机械工业出版社
China Machine Press

English reprint edition copyright © 2004 by Pearson Education Asia Limited and China Machine Press.

Original English language title: *Thinking in C++, Volume 2: Practical Programming* (ISBN: 0-13-035313-2) by Bruce Eckel and Chuck Allison, Copyright © 2004.

All rights reserved.

Published by arrangement with the original publisher, Pearson Education, Inc., publishing as Prentice Hall, Inc.

For sale and distribution in the People's Republic of China exclusively (except Taiwan, Hong Kong SAR and Macau SAR).

本书英文影印版由Pearson Education Asia Ltd.授权机械工业出版社独家出版。未经出版者书面许可,不得以任何方式复制或抄袭本书内容。

仅限于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地区)销售发行。

本书封面贴有Pearson Education培生教育出版集团激光防伪标签,无标签者不得销售。

版权所有,侵权必究。

本书版权登记号:图字:01-2004-0316

图书在版编目(CIP)数据

C++编程思想 第2卷:实用编程技术(英文版)/(美)埃克尔(Eckel, B.)等著.-北京:机械工业出版社,2004.2

(经典原版书库)

书名原文:Thinking in C++, Volume 2: Practical Programming

ISBN 7-111-12188-0

I. C… II. 埃… III. C语言-程序设计-英文 IV. TP312

中国版本图书馆CIP数据核字(2003)第124331号

机械工业出版社(北京市西城区百万庄大街22号 邮政编码 100037)

责任编辑:迟振春

北京中加印刷有限公司印刷·新华书店北京发行所发行

2004年2月第1版第1次印刷

850mm×1168mm 1/32·26印张

印数:0 001-4 000册

定价:49.00元

凡购本书,如有缺页、倒页、脱页,由本社发行部调换
本社购书热线:(010) 68326294

出版者的话

文艺复兴以降，源远流长的科学精神和逐步形成的学术规范，使西方国家在自然科学的各个领域取得了垄断性的优势；也正是这样的传统，使美国在信息技术发展的六十多年间名家辈出、独领风骚。在商业化的进程中，美国的产业界与教育界越来越紧密地结合，计算机学科中的许多泰山北斗同时身处科研和教学的最前线，由此而产生的经典科学著作，不仅擘划了研究的范畴，还揭橥了学术的源变，既遵循学术规范，又自有学者个性，其价值并不会因年月的流逝而减退。

近年，在全球信息化大潮的推动下，我国的计算机产业发展迅猛，对专业人才的需求日益迫切。这对计算机教育界和出版界都既是机遇，也是挑战；而专业教材的建设在教育战略上显得举足轻重。在我国信息技术发展时间较短、从业人员较少的现状下，美国等发达国家在其计算机科学发展的几十年间积淀的经典教材仍有许多值得借鉴之处。因此，引进一批国外优秀计算机教材将对我国计算机教育事业的发展起积极的推动作用，也是与世界接轨、建设真正的世界一流大学的必由之路。

机械工业出版社华章图文信息有限公司较早意识到“出版要为教育服务”。自1998年开始，华章公司就将工作重点放在了遴选、移译国外优秀教材上。经过几年的不懈努力，我们与Prentice Hall, Addison-Wesley, McGraw-Hill, Morgan Kaufmann等世界著名出版公司建立了良好的合作关系，从它们现有的数百种教材中甄选出Tanenbaum, Stroustrup, Kernighan, Jim Gray等大师名家的一批经典作品，以“计算机科学丛书”为总称出版，供读者学习、研究及度藏。大理石纹理的封面，也正体现了这套丛书的品位和格调。

“计算机科学丛书”的出版工作得到了国内外学者的鼎力襄助，国内的专家不仅提供了中肯的选题指导，还不辞劳苦地担任了翻译和审校的工作；而原书的作者也相当关注其作品在中国的传播，有的还专诚为其书的中译本作序。迄今，“计算机科学丛书”已经出版了近百个品种，这些书籍在读者中树立了良好的口碑，并被许多高校采用为正式教材和参考书籍，为进一步推广与发展打下了坚实的基础。

随着学科建设的初步完善和教材改革的逐渐深化，教育界对国外计

计算机教材的需求和应用都步入一个新的阶段。为此，华章公司将加大引进教材的力度，在“华章教育”的总规划之下出版三个系列的计算机教材：除“计算机科学丛书”之外，对影印版的教材，则单独开辟出“经典原版书库”；同时，引进全美通行的教学辅导书“Schaum's Outlines”系列组成“全美经典学习指导系列”。为了保证这三套丛书的权威性，同时也为了更好地为学校和老师服务，华章公司聘请了中国科学院、北京大学、清华大学、国防科技大学、复旦大学、上海交通大学、南京大学、浙江大学、中国科技大学、哈尔滨工业大学、西安交通大学、中国人民大学、北京航空航天大学、北京邮电大学、中山大学、解放军理工大学、郑州大学、湖北工学院、中国国家信息安全测评认证中心等国内重点大学和科研机构在计算机的各个领域的著名学者组成“专家指导委员会”，为我们提供选题意见和出版监督。

这三套丛书是响应教育部提出的使用外版教材的号召，为国内高校的计算机及相关专业的教学度身订造的。其中许多教材均已为M. I. T., Stanford, U.C. Berkeley, C. M. U. 等世界名牌大学所采用。不仅涵盖了程序设计、数据结构、操作系统、计算机体系结构、数据库、编译原理、软件工程、图形学、通信与网络、离散数学等国内大学计算机专业普遍开设的核心课程，而且各具特色——有的出自语言设计者之手、有的历经三十年而不衰、有的已被全世界的几百所高校采用。在这些圆熟通博的名师大作的指引之下，读者必将在计算机科学的宫殿中由登堂而入室。

权威的作者、经典的教材、一流的译者、严格的审校、精细的编辑，这些因素使我们的图书有了质量的保证，但我们的目标是尽善尽美，而反馈的意见正是我们达到这一终极目标的重要帮助。教材的出版只是我们的后续服务的起点。华章公司欢迎老师和读者对我们的工作提出建议或给予指正，我们的联系方式如下：

电子邮件：hzedu@hzbook.com

联系电话：(010) 68995264

联系地址：北京市西城区百万庄南街1号

邮政编码：100037

专家指导委员会

(按姓氏笔画顺序)

尤晋元	王 珊	冯博琴	史忠植	史美林
石教英	吕 建	孙玉芳	吴世忠	吴时霖
张立昂	李伟琴	李师贤	李建中	杨冬青
邵维忠	陆丽娜	陆鑫达	陈向群	周伯生
周立柱	周克定	周傲英	孟小峰	岳丽华
范 明	郑国梁	施伯乐	钟玉琢	唐世渭
袁崇义	高传善	梅 宏	程 旭	程时端
谢希仁	裘宗燕	戴 葵		

秘 书 组

武卫东

温莉芳

刘 江

杨海玲

Dedication

To all those who have worked tirelessly
to develop the C++ language

"I'd like to congratulate the both of you for a very impressive work! Not only did I find your book to be an enjoyable and rewarding read ... I was astounded by the accuracy both in terms of technical correctness and use of the language ... I believe that you have attained a level of craftsmanship that is simply outstanding."

Bjorn Karlsson
Editorial Board, C/C++ Users Journal

"This book is a tremendous achievement. You owe it to yourself to have a copy on your shelf."

Al Stevens
Contributing Editor, Doctor Dobbs Journal

"Eckel's book is the only one to so clearly explain how to rethink program construction for object orientation. That the book is also an excellent tutorial on the ins and outs of C++ is an added bonus."

Andrew Binstock
Editor, Unix Review

"Bruce continues to amaze me with his insight into C++, and *Thinking in C++* is his best collection of ideas yet. If you want clear answers to difficult questions about C++, buy this outstanding book."

Gary Entsminger
Author, *The Tao of Objects*

"*Thinking in C++* patiently and methodically explores the issues of when and how to use inlines, references, operator overloading, inheritance and dynamic objects, as well as advanced topics such as the proper use of templates, exceptions and multiple inheritance. The entire effort is woven in a fabric that includes Eckel's own philosophy of object and program design. A must for every C++ developer's bookshelf, *Thinking in C++* is the one C++ book you must have if you're doing serious development with C++."

Richard Hale Shaw
Contributing Editor, PC Magazine

DID THE BOOK YOU'RE HOLDING
GET YOU THINKING? THEN TRY...



WINNER:

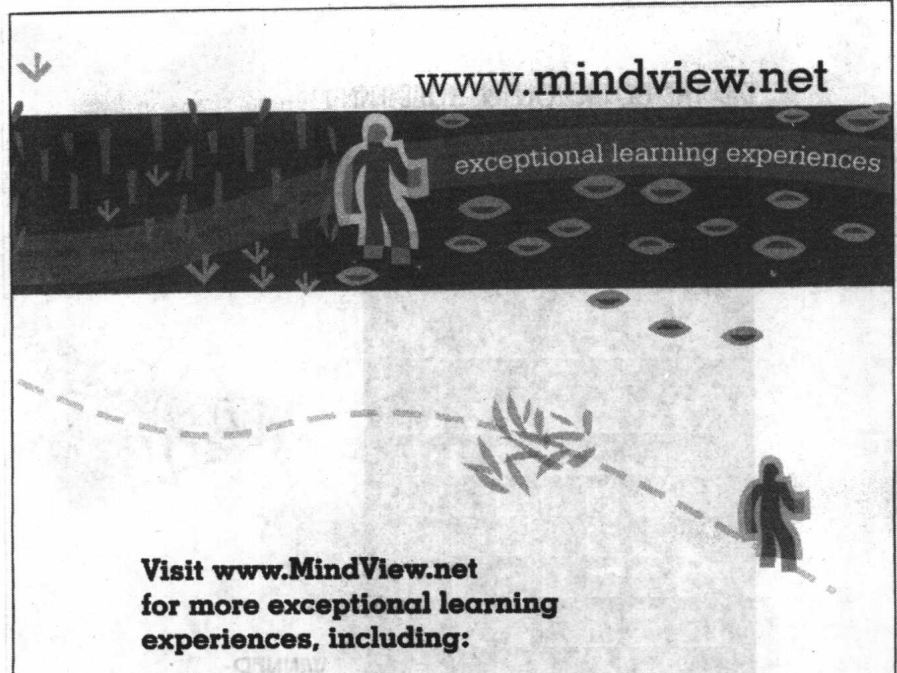
Software Development
Magazine Jolt Award for
Best book of 2002

JavaWorld Editor's Choice
Award for Best Book, 2001

JavaWorld Reader's Choice
Award for Best Book, 2001

Software Development
Magazine Productivity
Award, 1999

Java Developer's Journal
Editor's Choice Award for
Best Book, 1998, 2003



www.mindview.net

exceptional learning experiences

**Visit www.MindView.net
for more exceptional learning
experiences, including:**

- Seminars: public, or onsite at your company
- Seminars on CD Rom: Java, C++, and more
- Consulting and mentoring
- Downloadable electronic books, including this one
- Exercise solution guides
- Audio interviews with luminaries in the software field
- Bruce Eckel's weblog on software development topics

What's inside...

Introduction

1

Goals.....	1
Chapters	2
Exercises.....	5
Exercise solutions	5
Source code	5
Compilers.....	7
Language standards.....	9
Seminars, CD-ROMs & consulting	9
Errors	10
About the cover	10
Acknowledgements.....	10

I: Building Stable Systems

13

1: Exception Handling

15

Traditional error handling.....	16
Throwing an exception	18
Catching an exception	20
The try block	20
Exception handlers.....	20
Termination and resumption	22
Exception matching	23
Catching any exception.....	25
Rethrowing an exception.....	26
Uncaught exceptions	26
Cleaning up	28
Resource management	30

Making everything	
an object	32
auto_ptr	35
Function-level try blocks.....	36
Standard exceptions	38
Exception specifications.....	40
Better exception	
specifications?	45
Exception specifications	
and inheritance.....	46
When not to use	
exception specifications	47
Exception safety.....	48
Programming with exceptions.....	52
When to avoid exceptions	52
Typical uses of exceptions....	54
Overhead	58
Summary	60
Exercises.....	61

2: Defensive Programming

63

Assertions	66
A simple unit test framework	70
Automated testing	71
The TestSuite Framework	75
Test suites	79
The test framework code	81
Debugging techniques.....	87
Trace macros	87
Trace file	88
Finding memory leaks	90
Summary	96
Exercises.....	97

3: Strings in Depth

What's in a string?	104
Creating and initializing	
C++ strings	106
Operating on strings	109
Appending, inserting, and	
concatenating strings	110
Replacing string	
characters	112
Concatenation using nonmember	
overloaded operators	117
Searching in strings	117
Finding in reverse	123
Finding first/last of	
a set of characters	124
Removing characters	
from strings	126
Comparing strings	129
Strings and	
character traits	134
A string application	140
Summary	145
Exercises	146

4: Iostreams

Why iostreams?	151
Iostreams to the rescue	156
Inserters and extractors	156
Common usage	161
Line-oriented input	164
Handling stream errors	165
File iostreams	168
A File-Processing	
Example	169

Open modes.....	171
Iostream buffering.....	173
Seeking in iostreams.....	175
String iostreams.....	179
Input string streams.....	180
Output string streams.....	182
Output stream formatting.....	186
Format flags.....	186
Format fields.....	188
Width, fill, and precision	190
An exhaustive example.....	191
Manipulators.....	194
Manipulators with	
arguments	196
Creating manipulators	199
Effectors	201
Iostream examples.....	203
Maintaining class library	
source code.....	204
Detecting compiler errors....	208
A simple data logger	211
Internationalization.....	216
Wide Streams.....	216
Locales	218
Summary	221
Exercises.....	222

5: Templates in Depth

227

Template parameters	227
Non-type	
template parameters.....	228
Default template	
arguments	230
Template template	
parameters	232
The typename keyword	238
Using the template	
keyword as a hint	240

Member Templates.....	242
Function template issues	245
Type deduction of function	
template arguments	245
Function template	
overloading	249
Taking the address	
of a generated	
function template.....	251
Applying a function	
to an STL sequence	255
Partial ordering of	
function templates	259
Template specialization	260
Explicit specialization.....	261
Partial Specialization	263
A practical example.....	265
Preventing template	
code bloat.....	268
Name lookup issues	273
Names in templates	273
Templates and friends	279
Template programming idioms ...	285
Traits	285
Policies	291
The curiously recurring	
template pattern.....	294
Template metaprogramming	297
Compile-time	
programming	298
Expression templates	308
Template compilation models	315
The inclusion model.....	315
Explicit instantiation	316
The separation model	319
Summary	320
Exercises.....	321

A first look	325
Predicates	329
Stream iterators	331
Algorithm complexity	333
Function objects	335
Classification of	
function objects	336
Automatic creation of	
function objects	338
Adaptable function objects ..	341
More function	
object examples	343
Function pointer adaptors ..	351
Writing your own	
function object adaptors	358
A catalog of STL algorithms	362
Support tools for	
example creation	365
Filling and generating	368
Counting	370
Manipulating sequences	372
Searching and replacing	377
Comparing ranges	385
Removing elements	389
Sorting and operations	
on sorted ranges	393
Heap operations	403
Applying an operation to	
each element in a range	405
Numeric algorithms	413
General utilities	417
Creating your own	
STL-style algorithms	419
Summary	420
Exercises	421

Containers and iterators	429
STL reference	
documentation	431
A first look	432
Containers of strings	438
Inheriting from	
STL containers.....	440
A plethora of iterators	442
Iterators in	
reversible containers	445
Iterator categories	446
Predefined iterators.....	448
The basic sequences:	
vector, list, deque	454
Basic sequence operations .	454
vector.....	457
deque.....	465
Converting between	
sequences.....	467
Checked random-access	470
list	471
Swapping sequences	477
set	479
A completely	
reusable tokenizer.....	482
stack	487
queue	491
Priority queues.....	496
Holding bits	506
bitset<n>	507
vector<bool>	511
Associative containers	513
Generators and fillers	
for associative containers...	518
The magic of maps.....	521
Multimaps and	
duplicate keys	523