

经 典 原 版 书 库

面向对象方法 原理与实践

(英文版·第3版)

OBJECT- ORIENTED METHODS

PRINCIPLES & PRACTICE
THIRD EDITION

IAN GRAHAM



(英) Ian Graham 著



机械工业出版社
China Machine Press



TP311.52
Y14

经典原版书库

面向对象方法 原理与实践

(英文版·第3版)

Object-Oriented Methods
Principles & Practice

江苏工业学院图书馆
藏书章

(英) Ian Graham 著

北方工业大学图书馆



00730401



机械工业出版社
China Machine Press

Ian Graham: Object-Oriented Methods: Principles & Practice, Third Edition (ISBN: 0-201-61913-X).

Copyright © Pearson Education Limited 2001.

This edition of Object-Oriented Methods: Principles & Practice, Third Edition is published by arrangement with Pearson Education Limited. Licensed for sale in the mainland territory of the People's Republic of China only, excluding Hong Kong, Macau, and Taiwan.

本书英文影印版由英国Pearson Education培生教育出版集团授权出版。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。

此影印版只限在中国大陆地区销售(不包括香港、澳门、台湾地区)。

版权所有,侵权必究。

本书版权登记号: 图字: 01-2003-2314

图书在版编目(CIP)数据

面向对象方法: 原理与实践(英文版·第3版)/(英)格雷厄姆(Graham, I.)著. -北京: 机械工业出版社, 2003.5

(经典原版书库)

书名原文: Object-Oriented Methods: Principles & Practice, Third Edition

ISBN 7-111-11982-7

I. 面… II. 格… III. 面向对象语言-程序设计-英文 IV. TP312

中国版本图书馆CIP数据核字(2003)第036048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: 杨海玲

北京中加印刷有限公司印刷·新华书店北京发行所发行

2003年5月第1版第1次印刷

787mm×1092mm 1/16·53.75印张

印数: 0 001-3 000册

定价: 79.00元

凡购本书, 如有倒页、脱页、缺页, 由本社发行部调换

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中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、研究及收藏。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 | | | |

This book is dedicated to my son, Robert Graham Miller.



PREFACE TO THE THIRD EDITION

He threw down his book, stretched his legs towards the embers in the grate, and clasped his hands at the back of his head, in that agreeable afterglow of excitement when thought lapses from examination of a specific object into a suffusive sense of its connexions with all the rest of our existence.

George Eliot (*Middlemarch*)

When the first edition of this book appeared in 1991, there was a great deal of excitement and interest in object-oriented techniques in the IT industry and in academia, with public seminars held almost weekly. New journals and regular conferences were being established, and the membership of special interest groups was growing rapidly. By the time the second edition came out interest had peaked and many commercial organizations had already had their first experiences of the technology for better or worse. Still the fuss continues and, while the languages have stabilized a little, the furore over methods and life-cycle issues has, if anything intensified even further – even though the emergence of the Unified Modelling Language (UML) has largely settled disputes over notation. Current concern now focuses on cross-enterprise application integration and component-based development. Since the second edition appeared the field of object technology (OT) has, perhaps, trebled in its size and scope: published work and topics covered. It was a daunting prospect to begin to expand this volume, whose original aim had been to provide a comprehensive survey, without making the book excessively large. On the other hand the character of many of the new developments has been that of variations on themes that were around in 1994: the popularity of Jim Coplien's C++ idioms was the first inkling of the huge explosion of interest in design patterns; early object request brokers based on the Object Management Group (OMG) architecture model have matured and been put into production; new, better object-oriented programming languages have emerged; object-oriented databases are now in everyday commercial use – though still on a limited scale. Beside all this there has been a complete shake-out in the area of analysis and design methods. *Plus ça change, plus c'est la même chose.* The result has had to be a very substantial rewrite.

**SUBJECT
MATTER**

This book is essentially a survey of the whole area of object technology. It covers object-oriented programming, object-oriented design, object-oriented analysis, object-oriented databases and concerns several related technologies. There is a number of good books on object technology covering specific languages and methods. More general coverage is provided in these books only incidentally. They give a high level overview of the philosophy and benefits of object-orientation in general but trouble the reader with a great deal of material specifically about programming and are dependent on the syntax of particular languages. At the other extreme there are now some very good 'management surveys' available, but these are not generally of sufficient technical depth for practitioners or students. The reader seeking a reasonably detailed understanding of those aspects of object-orientation not related to programming has to turn to the research literature, conference proceedings, massive monographs or collections of highly technical papers. If such a reader wants to gain a general understanding of the whole field rapidly or evaluate the future rôle of object technology, there are few coherent and comprehensive sources. My aims are therefore to address a gap in the literature in the following ways.

- Providing a single source, comprehensive, language-independent introduction covering all aspects of object technology from the perspectives of both the developer and management.
- Placing much more emphasis on the viewpoint of the conceptual modeller, compared to that of the programmer or designer; and upon the practical issues surrounding the use of object-oriented techniques in commercial environments.
- Propagating the view that object-orientation, artificial intelligence and data modelling together (rather than separately) are required to address the central issues of IT.
- Providing an introduction to and evaluation of object-oriented languages, middleware, databases and methods; and relating them each to conventional technology. In particular, providing the first concise explanation of the powerful CatalysisTM method for component-based development (D'Souza and Wills, 1999).
- Attempting to explode some of the myths surrounding object technology while retaining a genuinely optimistic evaluation of its practical use.
- Supplying sufficient depth and reference material to guide students and practitioners entering the field.

The further objective of this book, which it shares with both previous editions, is to state in a clear manner the answers to the following questions.

- What are object-oriented methods?
- What are the benefits, pitfalls and likely costs?
- What languages, methods and tools are available, and how may they be evaluated?
- What has to be done to get started with adoption?

- What is the rôle of object-oriented analysis and design methods?
- How does one capture requirements for OO systems?
- How can object-orientation be managed?
- What special skills are required?
- What are the links to other areas of Information Technology (IT)?
- What are suitable applications?

Incidentally to the above aims the book exposes some of my own original work on object-oriented conceptual modelling using the idea of rulesets, requirements engineering and development process – collectively referred to as SOMA (Semantic Object Modelling Approach). I have interwoven SOMA with Catalysis and used the UML notation throughout the chapters on method. In reading this material the reader should be aware of the distinctive feature of my approach, which interprets object modelling as a general form of knowledge representation, rather than just a way to describe computer programs.

MAJOR CHANGES IN THE THIRD EDITION

This edition is a very substantial revision and extension of the Second Edition, reflecting major changes in the field since the first two editions. The rapid acceptance of object technology since I wrote the first edition has astounded me along with even the most fervent propagandists for it. The rapid changes in the technology which have occurred during the past three years or so are less surprising. Not only have the products changed and the number of methods grown, but the conclusions that a practitioner or, indeed, a careful observer can draw are quite different from what they were in 1991 or 1994. One of the most significant changes has been the acceptance, by the industry at large, of the OMG and their publication of various standards for object technology. On the other hand there is much that has not changed. Thus, while the objective of the book remains the same, the means of achieving it must be substantially different. This edition brings all the definitional material up to date and into conformity with newly emerging standards. It modernizes the descriptions of products and methods available and draws new conclusions based on the new facts.

The major changes are as follows. Chapter 1 has been modified slightly to reflect greater clarity and standardization of terminology than was present in the industry when the book was first written. It has benefited from what I hope are better, more mature pedagogical techniques that I have developed in the course of lecturing about and teaching the subject to many people. A new Chapter 4 on middleware and migration strategies has been included, with more material on OMG standards. The material on object-oriented databases in Chapter 5 has been completely updated to reflect new and increasingly mature products in this area. The biggest change is to the material on object-oriented analysis and design and their management. The survey of the 50 or more methods that were around six years ago has been relegated to an appendix, which will be of chiefly historical interest. The book now uses UML throughout and Chapters 6 and 7 describe best practice for object-oriented analysis and design, based mainly on the insights of Catalysis and SOMA. A new appendix summarizes the UML notation. Chapter 7 is new and

covers software architecture, patterns and component-based development. Chapter 8 describes the SOMA approach to requirements engineering in detail. Chapter 9, on management, is substantially reorganized for greater clarity of exposition and to give a far more definite prescription of the recommended development process. It now includes guidance on user interface design. All other chapters and Appendix A have undergone slight revision and improvement to reflect new developments in the field and correct any errors which remained in the Second Edition and which I was aware of.

I have added exercises at the end of most chapters to assist the substantial number of educational users of whom I became gradually aware over recent years. Selected answers can be found on the TriReme web site; where I know the answers, that is. The Bibliography is substantially expanded, to reflect the general growth in the volume of the literature as well as the new material in this edition, and the Glossary has been updated and improved.

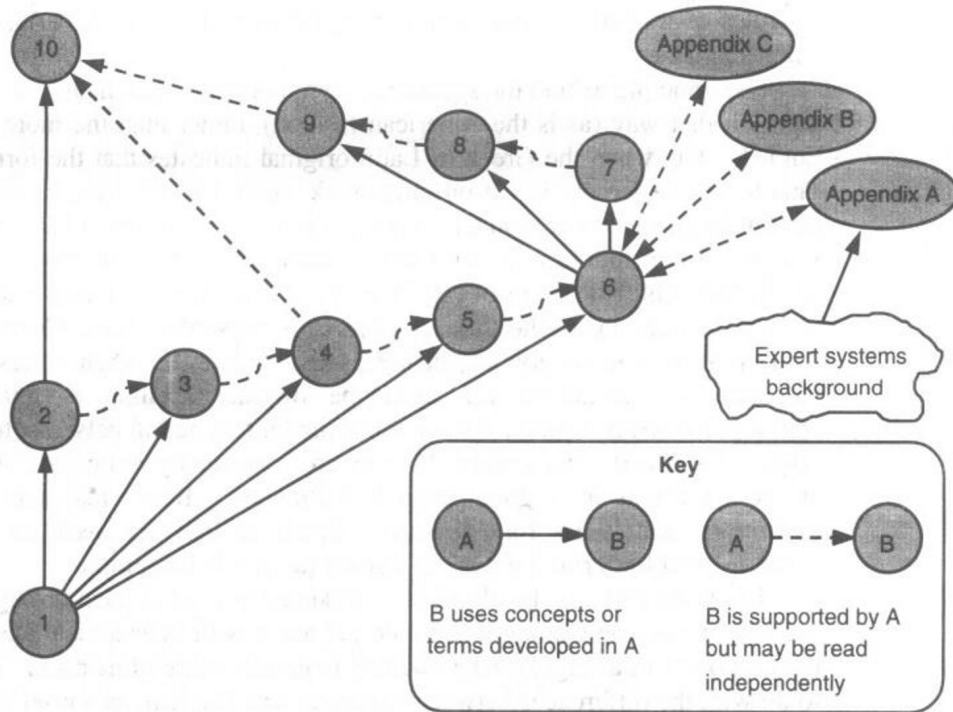
Despite these drastic changes the essential purpose of the book remains unchanged and I hope it is merely a more comprehensive, detailed, up-to-date and accurate survey of object-oriented methods than it would have been without the alterations.

**INTENDED
READERSHIP**

The book is intended to be accessible to readers who do not have deep specialist knowledge of theoretical computer science or advanced mathematics, but at the same time it attempts to treat the important issues accurately and in depth. It provides advice on how best to exploit object technology in practice.

The primary audience I had in mind for earlier editions was the IT and DP profession; software engineers and, generally, people who work with computers whether in user organizations, consultancies, universities or manufacturers. Although this is still the case with this edition, it has become clear that the book now has a loyal following in universities where the book is often used as a text for introductory undergraduate courses in object technology within an information technology or software engineering curriculum, perhaps complementing another course on object-oriented programming. It will therefore be of interest to teachers of Computer Science, Business Systems Analysis and possibly Artificial Intelligence. Researchers will be interested in the book as a survey and for the original contributions. They may also find some of the commentary scattered through the book thought provoking or even controversial. Managers and project planners will read it to gain an understanding of how the technology will affect their business practices and to be able to plan more effectively for change. Consultants, project managers, systems analysts and designers will read it to evaluate and stay abreast of the technology and, I hope, use the techniques explained in their day-to-day work. Programmers will read it to broaden their horizons in this area.

The material in this book, as it evolved, has been presented to very many audiences at various conferences, public seminars and in-house training courses.

READING
MAP

There are a number of optional reading paths through the material presented in this book. The two prime routes through the text are illustrated in the reading map. Managers and those wishing for a high level view of the subject may take the high road through Chapters 1, 2 and 10. Project managers might also include Chapter 9. People interested in analysis methods must take the low road, as Chapters 6–8 build on all previous chapters. However, Chapter 1 should be read even by people familiar with object-oriented programming since it introduces terminology which may differ from that of other works, stressing – as I have indicated – a conceptual modelling viewpoint. Specialist topics and significant digressions are indicated by the diversion icon shown to the left of this paragraph (which is not a digression).

I hope the book tells a story if read sequentially. There are certainly some key themes to look out for. These are the differentiation of the viewpoints of the system designer from the conceptual modeller, the need for object-orientation to absorb techniques and ideas from other areas of computing and the need to add semantic richness to the more well-understood benefits of object-orientation: reusability and extensibility.

**A NOTE ON
LANGUAGE
AND
SPELLING**

Since many of us are prone to strong opinions on language and spelling, and since some of my habitual usage has attracted comment from some referees of this and previous editions of this work, I feel that it is worthwhile clarifying the principles that I have applied in this respect in this text. Also, the conventions of spelling and

the rules of transformational syntax vary between English, American English and other 'Englishes'.

The principle behind the spelling is etymological. Thus many words ending -IZE are spelt that way (as is the American fashion), rather than the more usual modern English -ISE, when the Greek or Latin original indicates that the former is correct. This has to be largely based on guesswork, since I have never formally learnt any foreign languages other than Chinese and German. However, I have discovered that a useful rule of thumb is to ask whether there can be an -IZATION. If not, as with ADVERTISE, CIRCUMCISE or DEVISE, then the S is correct. The only exception I know of (and the only one in the *Concise Oxford*) is IMPROVISATION. CONNEXION is spelt with an X for similar etymological reasons. However, when words from foreign languages or quotations are used the original spelling is retained. Thus, CONNECTIONISM is used for the school of thought on neural nets, due to its American origin. Latin and other imported words are consistently italicized. All such words are to be found in a good English dictionary. Bold characters are used for **definitions** and italics for *emphasis*. Small capitals are used for the names of 'patterns' and for terms we wish to discuss (as in this paragraph).

The word DATA is the plural of DATUM and is used as such throughout. I fail to see why writers on Computing should get away with their almost universal error of treating DATA as a singular when writers in no other discipline do so. This is nothing to do with the differences between American and English, as a brief glance at books on Geography, Statistics, Medicine, Management and so on from both sides of the Atlantic will soon demonstrate. Words with two dictionary definitions such as BIMONTHLY and BILLION are proscribed.

The word METHODS is used in three different ways in this book:

- methods for doing software engineering in general (in the title);
- particular methods for doing software engineering and modelling (as with the Catalysis and SOMA methods); and
- methods (programs) implementing the behaviour of objects.

I have resisted the non-word METHODOLOGIES on the same grounds that I would reject Physicses, or Chemistries. The OED has it that Methodology (and the capitalization is necessary) is 'the science of method' and Chambers adds 'within a science or discipline'. The plural could only be used if there had been a 'paradigm shift' in the sense of Thomas Kuhn (a solecism in itself¹) in our perception of the way of doing things. Since many people believe that object technology is such a paradigm shift, we could correctly write the sentence: 'There is a difference between the Structured and the Object-Oriented Methodologies.'; meaning that the whole approach to computer science has altered. It is pure barbarism to write: 'There is a difference between the Yourdon and Jackson methodologies.' I try to avoid words like PARADIGM too.

¹PARADIGM really means a model or especially typical example.

Sexist assumptions in writing – as in life – should be avoided. My preference is for a plural personal pronoun or s/he (pronounced ‘she’) instead of the ugly and intrusive ‘he or she’. However, I distinguish between sexual and grammatical gender; so that MAN is merely short for MANKIND and means the same as HUMANITY. Thus the term MAN-DAY is to be taken generically; the gender being grammatical rather than sexual. Otherwise, the French would have to start saying ‘le ou la personne’ (people are feminine in most European languages) and the English speaker would be obliged to use WOPERSON for WOMAN. Political correctness is achieved by overthrowing evils, not by tinkering with language.

Split infinitives are usually avoided, based on personal preference and my belief that readability is enhanced when adverbs appear late in sentences. No attempt to write in short, journalistic sentences has been made, except where clarity so dictates. I have tried to employ the *correct* word, rather than a word that everybody will know. Particularly obscure (to my referees) words, or words where the dictionary would not be adequate due to nuance, have been added to the glossary or defined in a footnote. The dictionaries used were the *Oxford English Dictionary (Compact Edition)* (OED) and *Chambers English Dictionary*. A certain number of words that are not in the dictionary are inevitable in a book on a technical subject of this kind. An example is WORKSTATION, as opposed to WORK STATION. The latter seems distinctly wrong in the context of a local computer, where the emphasis is on its capability rather than its location. I have taken such liberties fairly thoughtfully though. Language, after all, is a living thing.

These are the theoretical principles applied. In practice there may be several grammatical and spelling errors.

ACKNOWLEDGEMENTS

Although it contains much original material, this book is largely a survey of other people’s work and could not have been written without that work. I would like to acknowledge the contribution of these other authors. Also, many of the ideas contained are the result of personal interaction between the author and a number of his colleagues, clients and friends.

Special thanks are due to Akmal Chaudri for his contribution to Chapter 5. The remarks of several anonymous referees were very helpful too. Alan O’Callaghan of De Montfort University co-wrote Chapter 7. Alan Cameron Wills co-wrote Chapters 6 and 7. The latter deserves special mention: many of the figures in these chapters were copied from his original artwork as contained in TriReme’s UML and Catalysis course material. Alan is still teaching me a great deal about the subject and I acknowledge a great debt to him for this and his encouragement during the preparation of this text. This should not be taken to mean that either of my co-authors agrees with all the assertions I make or stances I adopt. The responsibility for any errors that remain in the text is entirely mine.

With the long history of this text, there is not space to acknowledge the nature of every individual contribution to this and earlier editions. I want, however, to thank all the following people and organizations for their help: Mark Addison, Nigel Backhurst, Sav Bhoja, Steve Birkbeck, Julia Bischof, Grady Booch, Franco Civello,

Dave Clark, Alistair Cockburn, Graham Collins, Larry Constantine, Cos Constantinou, John Cresswell, Robin Crewe, John Daniels, Sally Davies, Malcomb Dick, Faramarz Farhoodi, Armando Ferreira, Don Firesmith, Martin Fowler, Florence Froidevaux, Stuart Frost, Tim Gee, Thomas Grotehen, Brian Henderson-Sellers, Kevlin Henney, Chris Harris-Jones, Benedict Heal, Tim Lamb, David Harvey, Jane Hillston, Michael Jackson, Margaret James, Peter Jones, Fiona Kinnear, David Lee, Chris Lees, Mark Lewis, Nick Lukic, Margaret Lyall, Neil Maiden, Sonia Math, Clive Menhinick, Sally Mortimore, Derek Pearce, Colin Prosser, Rob Radmore, Al-Noor Ramji, Dan Rawsthorne, David Redmond-Pyle, Elaine Richardson, René Schwarb, Richard Seed, Tony Simons, Richard Smith, Gail Swaffield, John Taylor, Brian Thal, Ulrika Thyssen, Richard Walker, Rose Watson, John Welch, David Welton, Kerry Williams, Benoit Xhenseval, Jennifer Yates, Marcus Zemp. BIS Applied Systems, BIS Banking Systems, Chase Manhattan Bank, De Montfort University, Eric Leach Marketing, Equitable Life, Swiss Bank Corporation (now UBS), TriReme International and the Expert Systems, OOPS and Requirements Engineering Specialist Groups of the British Computer Society. All the staff at Pearson Education demonstrated their usual splendid professionalism and especial thanks are due to my editor, Alison Birtwell.

Thanks go too to the various audiences who endured my seminars and classes during the period when these ideas were being developed and refined and who provided valuable feedback, and occasionally – and most usefully – asked hard questions. I would also like to thank the thousands of people scattered throughout the world, living and dead, whose labour, by hand or by brain, created the word processing system I am now using. Without it no edition of this book would ever have been written.

Lastly, it's about time that I came clean and acknowledged the contribution of The Grove: the hostelry where much of all my books has been written and corrected. It is a Victorian pub with an interior that deserves to be listed and nearly always empty enough to guarantee me a table to work at; which is perhaps why some of the locals – who come from all walks of life from car mechanics to criminologists – call it The Grave. The other writers that rely on it tend to refer to The Balham Reading Rooms instead; to my knowledge at least two books on the culture of the Solomon islands, one novel and a song (The Ballad of Ken Livingstone – copies on request) also had their origins there. It also provides an occasional venue for me to join the resident Wandle Delta String Band and inflict my bodhrán playing on the public. See you there!

Ian Graham
Balham, November 2000
(ian@trireme.com)



华章教育 赤诚奉献

计算机科学丛书

- | | |
|-----------------------------------|----------------------------|
| 计算机文化 (原书第4版) | Parsons / 龚波 / 50.00 |
| 离散数学及其应用 (原书第4版) | Rosen / 袁崇义 / 75.00 |
| 组合数学 (原书第3版) | Brualdi / 冯舜玺 / 38.00 |
| 程序设计实践 | Kernighan / 裘宗燕 / 20.00 |
| 程序设计语言概念和结构 (原书第2版) | Sethi / 裘宗燕 / 45.00 |
| 编码的奥秘 | Petzold / 伍卫国 / 24.00 |
| C语言解析教程 (原书第4版) | Pohl / 麻志毅 / 48.00 |
| C程序设计教程 | Deitel / 薛万鹏 / 33.00 |
| C程序设计语言 | Kernighan / 徐宝文 / 28.00 |
| C++程序设计语言 (特别版) | Stroustrup / 裘宗燕 / 78.00 |
| 《C++程序设计语言》题解 | Vandevoorde / 裘宗燕 / 23.00 |
| C++面向对象开发 (原书第2版) | Lee / 麻志毅 / 45.00 |
| C++编程思想 (原书第2版) | Eckel / 刘宗田 / 59.00 |
| C++精髓: 软件工程方法 | Shtern / 李师贤 / 85.00 |
| C++语言的设计和演化 | Stroustrup / 裘宗燕 / 48.00 |
| C++程序设计教程 | Deitel / 薛万鹏 / 22.00 |
| C++编程思想 (原书第1版) | Eckel / 袁兆山 / 39.00 |
| Java语言导学 (原书第3版) | Campione / 马朝晖 / 59.00 |
| Java编程思想 (原书第2版) | Eckel / 侯捷 / 99.00 |
| Java编程思想 (原书第1版) | Eckel / 京京工作室 / 60.00 |
| Java程序设计教程 (原书第3版) 上册 (附光盘) | Deitel / 袁兆山 / 55.00 |
| Java程序设计教程 (原书第3版) 下册 (附光盘) | Deitel / 袁兆山 / 69.00 |
| Visual Basic.NET 程序设计专家指南 (原书第2版) | Deitel / 龚波 / 118.00 |
| 面向对象程序设计——Java语言描述 (附光盘) | Kalin / 孙艳春 / 55.00 |
| 面向对象程序设计——C++语言描述 | Johnsonbaugh / 谢君英 / 48.00 |
| 标准C++与面向对象程序设计 | Wang / 李健 / 39.00 |
| Java面向对象程序设计教程 | Kafura / 袁晓华 / 49.00 |
| 面向对象程序设计——图形应用实例 | Laszlo / 何玉洁 / 35.00 |
| 数据结构, 算法与应用——C++语言描述 | Sahni / 王广芳 / 49.00 |
| 编译原理及实践 | Louden / 冯博琴 / 39.00 |
| Linux操作系统内核实习 (附光盘) | Nutt / 陆丽娜 / 29.00 |
| UNIX操作系统设计 | Bach / 陈葆钰 / 35.00 |
| UNIX环境高级编程 | Stevens / 尤晋元 / 55.00 |
| UNIX编程环境 | Kernighan / 陈向群 / 24.00 |
| 现代操作系统 | Tanenbaum / 陈向群 / 40.00 |
| 并行计算机体系结构 (原书第2版) | Culler / 李晓明 / 78.00 |

| | |
|--|-------------------------|
| 结构化计算机组成 | Tanenbaum/刘卫东/46.00 |
| 可扩展并行计算: 技术、结构与编程 | Hwang/陆鑫达/49.00 |
| 并行程序设计 | Wilkinson/陆鑫达/43.00 |
| 数据库系统概念 (原书第4版) | Silberschatz/杨冬青/69.00 |
| 数据库原理、编程与性能 (原书第2版) | O'Neil/周傲英/55.00 |
| 数据库设计 | Stephens/何玉洁/35.00 |
| 数据库系统导论 | Date/孟小峰/66.00 |
| 数据库系统实现 | Garcia-Molina/杨冬青/45.00 |
| 数据库系统概念 | Silberschatz/杨冬青/49.00 |
| 数据挖掘: 概念与技术 | Han/范明/39.00 |
| 数据仓库 (原书第3版) | Inmon/黄厚宽/29.00 |
| 数据仓库 | Inmon/黄厚宽/25.00 |
| 信息系统原理 | Reynolds/张靖/42.00 |
| 计算机信息处理 | Mandell/尤晓东/38.00 |
| 数据通信与网络 (原书第2版) | Forouzan/吴时霖/68.00 |
| 神经网络设计 | Hagan/戴葵/49.00 |
| 高性能通信网络 (原书第2版) | Walrand/史美林/55.00 |
| 数据广播 | Tvede/徐良贤/28.00 |
| ISDN、B-ISDN与帧中继和ATM (原书第4版) | Stallings/程时端/48.00 |
| 计算机网络 (原书第2版) | Peterson/叶新铭/49.00 |
| 计算机网络实用教程 (附光盘) | Dean/陶华敏/65.00 |
| 计算机网络实用教程实验手册 | Dean/陶华敏/15.00 |
| 计算机网络与因特网 (附光盘) | Comer/徐良贤/40.00 |
| TCP/IP详解 卷I: 协议 | Stevens/范建华/45.00 |
| TCP/IP详解 卷II: 实现 | Wright/陆雪莹/78.00 |
| TCP/IP详解 卷III: TCP事务协议、HTTP、NNTP和UNIX域协议 | Stevens/胡谷雨/35.00 |
| 数据通信与网络 (第1版) | Forouzan/吴时霖/48.00 |
| 数据通信与网络教程 | Shay/高传善/40.00 |
| 最新网络技术基础 | Palmer/严伟/20.00 |
| Internet技术基础 | Comer/袁兆山/18.00 |
| 分布式操作系统: 原理与实践 | Galli/徐良贤/38.00 |
| 分布式计算的安全原理 | Bruce/李如豹/35.00 |
| 分布式系统设计 | Wu/高传善/30.00 |
| 软件工程 (原书第6版) | Sommerville/程成/49.00 |
| 软件工程: 实践者的研究方法 (原书第5版) | Pressman/梅宏/59.00 |
| 设计模式: 可复用面向对象软件的基础 | Gamma/吕建/35.00 |
| 软件工程: 实践者的研究方法 (原书第4版) | Pressman/梅宏/48.00 |
| 软件工程: Java语言实现 | Schach/袁兆山/38.00 |
| 计算机图形学的算法基础 (原书第2版) | Rogers/石教英/55.00 |
| 专家系统原理与编程 (附光盘) | Giarratano/印鉴/49.00 |
| 人工智能 | Nilsson/郑扣根/30.00 |
| 机器学习 | Mitchell/曾华军/35.00 |
| 数字逻辑: 应用与设计 | Yarbrough/李书浩/49.00 |
| 嵌入式计算系统设计原理 (附光盘) | Wolf/孙玉芳/65.00 |
| 计算理论导引 | Sipser/张立昂/30.00 |