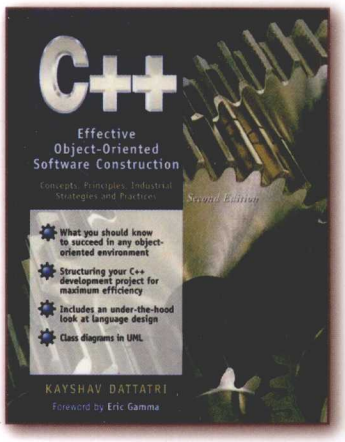
PEARSON

C和C++实务精选
品味岁月积淀，读享技术菁华

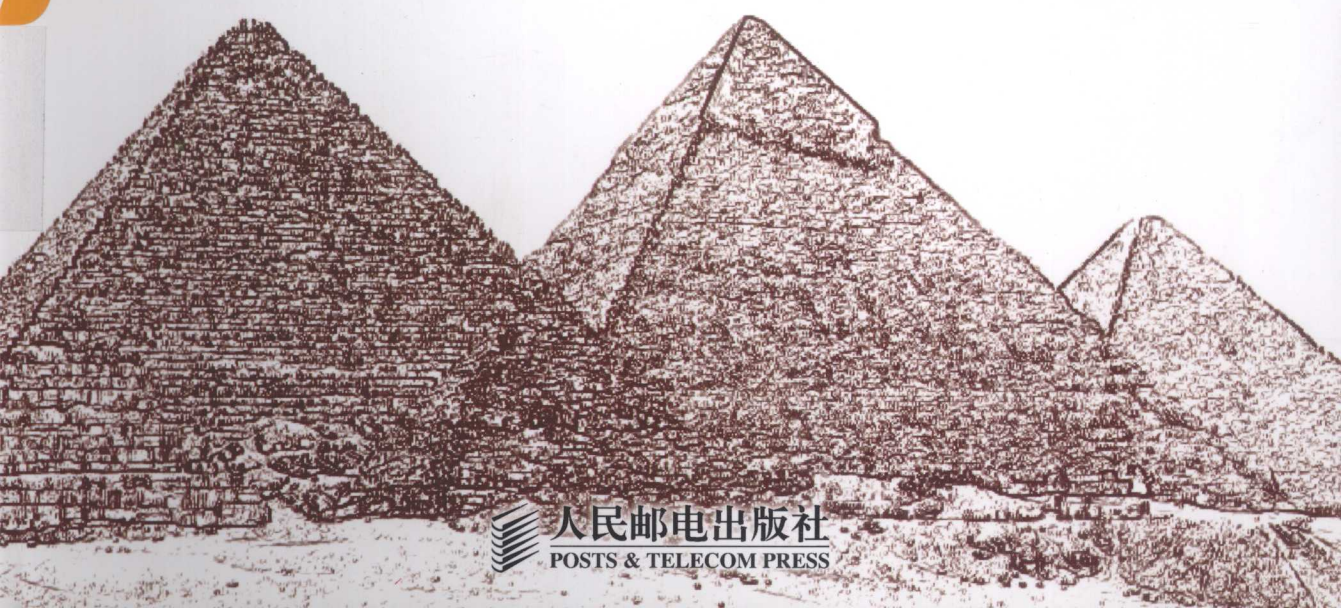
C++面向对象高效编程（第2版）

[美] Kayshav Dattatri 著　　叶尘 译

C++: Effective Object-Oriented Software Construction (2nd Edition)



- 比肩*Thinking in C++*、*The C++ Programming Language*等经典著作；
- *Design Patterns*作者Erich Gamma博士为本书作序；
- 国内知名技术专家孟岩、方舟联袂推荐。



人民邮电出版社
POSTS & TELECOM PRESS

C++
面向对象高效编程
（第2版）

[美] Kayshav Dattatri 著　　叶尘 译

人民邮电出版社
北京

图书在版编目（CIP）数据

C++面向对象高效编程 : 第2版 / (美) 达特特里(Dattatri,K.) 著 ; 叶尘译. -- 北京 : 人民邮电出版社, 2013.10
ISBN 978-7-115-32934-9

Ⅰ. ①C… Ⅱ. ①达… ②叶… Ⅲ. ①C语言－程序设计 Ⅳ. ①TP312

中国版本图书馆CIP数据核字(2013)第197290号

版权声明

◆ 著　[美] Kayshav Dattatri
译　叶　尘
责任编辑　傅道坤
责任印制　程彦红　杨林杰

◆ 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　电子邮件　315@ptpress.com.cn
网址　http://www.ptpress.com.cn
北京艺辉印刷有限公司印刷

◆ 开本：787×1092　1/16
印张：49
字数：1156 千字　2013 年 10 月第 1 版
印数：1- 3 000 册　2013 年 10 月北京第 1 次印刷
著作权合同登记号　图字：01-2011-7819 号

定价：118.00 元

读者服务热线：(010)67132692　印装质量热线：(010)67129223
反盗版热线：(010)67171154
广告经营许可证：京崇工商广字第 0021 号

内容提要

本书以帮助读者掌握C++面向对象高效编程范式为目的，详细介绍了C++编程中的各种概念和应用技巧。全书共分为两部分，第一部分（第 1 章至第 10 章）介绍面向对象编程的基础和应用，如数据抽象、继承、泛型类型、异常处理等内容；第二部分（第 11 章至第 13 章）深入探讨如何建立抽象及其策略，并研究了C++对象模型。书中包含大量的代码实例，读者不仅能从理论上得以提高，而且还能轻松地在实践中应用。

本书适用于C++程序员，也可供对面向对象程序设计感兴趣的编程人员及大专院校计算机专业师生参考。

序

当今软件系统的复杂程度剧增，面向对象编程为开发人员提供了强大的概念和工具。然而，为了解决棘手的问题，很容易写出极其复杂的面向对象程序。C++丰富的特性确实能让人事半功倍。但事实上，我见过的许多C++程序，更像是一个编译器测试套件，而不是一个实际问题的解决方案。对于小型系统而言，这样的程序不会出现问题，但是在不断运行的大型系统中，你将无法应付层出不穷的各种复杂问题。

如果因此而责备C++是不公平的。有些C++程序非常简洁明了、功能强大。开发人员必须根据实际情况明智地选择C++的不同特性，才不至于滥用。你不仅要理解每种特性的语义和性能开销，还要深入了解特性的优点、缺点，以及它的危险之处。

Kayshav Dattatri 在这方面有极其丰富的经验。在C++出现的早期，我就已经认识Kayshav了。我们在1990年的USENIOX C++会议上相识，当时会议上的许多论文都是AT&T（美国电话电报公司）发表的，估计一半的参与者都是AT&T的内部人员，而程序委员会也都是AT&T的人。稍后，我们发现自己都在Taligent公司工作。当时，Taligent已将C++应用于大规模的问题中，将其作用发挥到极限。这里是用C++构建实际应用软件解决问题的绝佳之地。

本书的所有内容都是经验之谈。书中的练习都建立在Kayshav多年的C++经验基础上。Kayshav不仅详尽地解释了面向对象的概念以及从理论上介绍C++的语言特性，还介绍了继承、mixin类、模板类和异常这些方面的实践经验，探讨了模板实例化、共享库、线程安全性和许多其他问题。

虽然本书从基本概念开始介绍C++和面向对象编程。然而，我确信，无论是新手还是经验丰富的C++程序员，都能从实践角度在本书中学到新的知识。

Erich Gamma 博士
Object Technology Internation 公司 技术主管

译者序

第一次拿到这本书时，我的第一印象是挺厚实。离开学校以后，总是很难静下心来啃“大部头”。没时间的借口已经毫无新意，实际上是多看几页就觉得睡眠不足。如果你喜欢幽默风趣的行文，就不用费心思在这本书里找乐子了。没有什么比讲解概念，介绍理论更让人昏昏欲睡。起初我甚至可以想象自己边译边睡的情景，但万万没想到的是，经常从书房出来，天已经黑了。要说是作者文笔精彩绝伦，不如说是他的教授引人入胜。作者的文笔循规蹈矩，通篇教科书范儿。但是，跟着作者的思路，针对实际问题分析出解决方案的过程，引人入胜。

相信很多读者对教科书的印象很深。严格的定义、枯燥的理论、呆板的例证。然而，本书与那些沉闷的基础理论书籍大为不同。这是一本真正将理论结合实际的书，书中介绍的示例几乎都与现实生活息息相关。小至玩具书系统、文件处理软件，大至金融系统资金管理、核电站控制程序、卫星发射系统。正如作者所言，将 C++应用于解决实际的问题上，将 OOP 的潜力发挥到极限。作者在讲解某个知识点或介绍设计方案时，先通过简化后的示例让读者了解一些基本情况，引导读者思考关键的问题。读者通过阅读不断地积累知识，有助于理解作者在后面介绍的更好更简洁的解决方案。整本书的知识结构循序渐进、由浅入深。

虽然本书几乎从零开始介绍面向对象编程，对于 C++的相关概念也从最基础的部分讲起，但你仍然需要具备一些基本的编程经验。本书不仅深入分析 C++的相关知识点，还将其与设计策略相结合。作者讲解得如此细致和透彻，让读者知其然，知其所以然，真正做到了“授人以渔”。值得一提的是，书中介绍的很多解决方案都与设计模式相关。如果对面向对象不太熟悉，可能很难体会到其中的精髓。只有真正理解了编程思想，才能举一反三，在适当的场景运用相应的策略。否则，滥用一些技巧只会弄巧成拙，让程序更加不堪重负。

本书包含大量的代码示例和讲解，除了代码本身是英文，作者在后面的讲解会用到很多类名、对象名、异常类名、数据成员名、成员函数名、关键字等，有时英文在一页上占很大篇幅。阅读不是记忆测试，为了减轻读者的识辨负担，我在文字排版上采用不同的字体来区分（如语言 **SmallTalk**，一般类名 TStudent，异常类名

NToAccessToMachine，函数名 IsEqual，关键字 static，等）。另外，相信不少读者都有看到脚注却找不到文中标注的尴尬。为此，我特意选择明显的字体标出（如类的对象[9]）。对一些比较重要以及尚未统一译名的术语，中英并陈，英文在括号中以斜体标出（如数据封装（*data encapsulation*））。

另外，还需提醒读者注意的是，作者为简洁起见（突出重点），在书中所列的大部分代码只是框架，仅为了体现相应的编程思想和设计策略。如果读者要在编译器中调试，需要自行补充一些必备的函数及其实现。另外，书中的代码仅符合当时的 C++编码风格。C++发展至今，语言细节上发生了许多变化。如果读者照抄进最新的编译器，可能会无法编译或报错。然而，瑕不掩瑜。这的确是一本夯实基础、拓展思维的好书。

本书与原书页页对译，得以保留英文索引。文字、代码和图片的位置也与原书相同。

感谢傅道坤编辑，不仅热心解答我提出的问题，还一再容忍我延期交稿。特别感谢 Gott，不仅给我提供了大量的相关英文原版书籍，还认真地阅读我的译稿，提出了许多宝贵的意见和建议。感谢父母的支持和理解，让我能安逸地在电脑前敲键盘。

追求完美的痛苦是永远都不完美，要不是交稿日期一拖再拖，我可能无限期地拖下去。无论校对多少次，再看的时候总会有所改动。本书的翻译、作图、文字排版都是我。如果因为翻译或排版让读者不快，喷我好了。在阅读本书的过程中，如果发现任何错误或译文不妥之处，请发邮件至：yesenaaron@gmail.com。

作为译者，我从本书中收获颇多，希望所有的读者都能从中受益。本书无论是详尽的理论剖析，还是丰富的示例辅助说明，都可称得上是丰盛的知识盛宴。求知若渴的朋友们，请尽情地享用吧！

叶尘

2013 年 6 月 5 日

前　言

面向对象软件开发已逐渐成为开发软件的首选。优秀的面向对象软件开发人员、设计人员、系统架构师对其需求与日俱增。要想成为一名成功的面向对象编程（OOP）人员必须忘却（摈弃）多年来面向程序编程的习惯，从新的角度分析问题。

面向对象编程要求程序员和设计者非常熟悉一些基本范式或概念。理解这些范式是在面向对象软件领域打下牢固基础的基本要求。支持 OOP 的语言都必须支持这些基本范式。换言之，学习 OOP，简单地说，就是学习许多语言（如 **C++**，**Eiffel**，***SmallTalk***，**Java** 等）所支持的强大范式。本书的第一个目标是，让你在不过分深入语言语法要素的前提下，理解面向对象编程最重要的概念和原则。第一部分——概念、实践和应用，将涵盖这方面的内容。

掌握支持 OOP 的语言语法和学习 OOP 的概念不一样。对基本 OOP 范式一无所知的人，也能成为 **C++**或 **Java** 的佼佼者。但是，理解 OOP 基本概念的人可以在任何支持 OOP 的语言中有效地使用这些概念。而且，他/她还知道何时加入特定的概念。任何掌握链表概念的人都会发现，它是在 **Pascal**、**C** 或 **Modula-2** 中实现链表的基础。比方说，如果你知道如何游泳，就可以在湖泊、池塘或游泳池中游泳。语言只是一个帮助你实现最终目标的载体。

学习 OOP 概念，仅仅是漫漫长路的第一个里程碑，不是程序员和设计人员的终极目标。你应该用这些概念去解决自己专业领域中的问题。财政计划人员想开发一个对象框架，以管理个人资金；商店想开发应用程序，以管理存货。但是，要把 OOP 的原则应用于不同的领域实属不易。例如，解决玩具课本的问题可能很琐碎，而解决某个专业领域的问题和构建系统则非常复杂和困难。学习专业领域的特定例子（例如文件系统、汽车代理管理系统、桌面排版系统、航班计划系统等）将会有所帮助。显然，我不是这些领域的专家，大部分读者也不熟悉这些相关领域，对它们也不感兴趣。这是在进行面向对象设计时，很多设计者和程序员所面临的典型问题。然而，无论涉及什么领域，总会有一些编程或设计方面（与相关领域无关）的经验原则，对软件专业人士解决复杂问题非常有用。专业人士必须对何

时使用什么工具了如指掌。本书的第二部分——构建强大的面向对象软件，将通过许多简单的例子阐明一些高级 OOP 技术。这部分主要介绍有效使用面向对象设计的强大策略。专业开发人员并不满足于只学习一些诀窍和技巧，他们还希望深入了解每种技巧的利弊、替换方法，以及对移植和效率的影响等诸如此类的问题。本书详细讨论了各种必要和重要的技巧，学习完本书的全部内容后，不谙编程的新手也能成为该领域的专家。

尽管书中涉及一些语言议题，但本书的重点仍是概念不是语言的语法。本书在所有的主要问题中，都会讨论 **C++**和其他面向对象语言之间的区别。将 **C++**与其他 OOP 语言的特性做对比，能更好地理解 OOP 的概念（有时，可以帮助理解语言设计）。这些内容深入且广泛地介绍了 OOP 领域。尽管书中的例子都用 **C++**编写，但并不意味着 **C++**是唯一的选择。除 **C++**外，还有许多同样优秀的 OOP 语言。深入了解不同语言的设计意图会让你受益匪浅。考虑到熟悉其他 OOP 语言（非 **C++**）的读者，本书从 **C++**的角度介绍主要概念的细节，随后再讨论 **SmallTalk** 和 **Eiffel**。

随着 **C++**日益流行（因为它有丰富的特性设置、严格的静态类型检查，而且支持编写工业级的软件），理想情况下，你应该精通 **C++**和 OOP 的概念。当然，只学习 **C++**语法比较乏味，我们要瞄准更高的目标。为了利用 **C++**挖掘出 OOP 的所有潜力，以适应面向对象编程的开发者们需要用到很多特殊的模式、技术和技巧。目前，**C++**中还有许多时间测试技术，它时常能帮助我们明白为什么某些特性会以特定的方式存在于语言中，能让程序员更好地理解语言（从而更尊重语言的设计者），更加高效地使用语言。本书第一部分的后几章，将介绍这方面的内容。本书的例子展示了 **C++**功能强大的策略。

初学者应该从第一部分开始阅读，该部分介绍范式、理论和应用程序。第一部分根据知识的相关性和难度组织章节。第 1 章初步介绍面向对象的基本概念；第 2 章详细讨论数据抽象，本章最后简要介绍了统一建模语言（UML）和 Booch 方法论；第 3 章和第 4 章进一步探讨对象模型和良好的接口设计；第 5 章和第 6 章详细介绍继承及其特性；第 7 章和第 8 章讨论一些简单的问题；第 9 章介绍泛型类型；最后，第 10 章详细讨论异常管理。在第二部分中，第 11 章阐述构建抽象的策略；第 12 章介绍如何高效使用继承；最后，第 13 章深入研究 **C++**对象模型。

你可以随意阅读任意章节，但是，由于某些例子源于前 4 个章节，选读可能会破坏阅读的连贯性。第二部分的章节是独立的。

大多数实现代码都异常乏味。因此，为了突出重点保持简洁，本书尽量减少枯燥的实现代码，仅在确实对讨论有所帮助时，才在相应的位置添加一些。

本书可作为面向对象编程的入门书或高级读本，对初次接触 OOP 的程序员和学生有很

大帮助。谙熟 OOP 概念的程序员会从本书的第二部分中受益良多。熟悉其他 OOP（除 C++ 外）语言的程序员，建议先阅读第一部分，再重点阅读第二部分。本书的第一部分可作为本科生 OOP 课程的教案，而第一部分和第二部分一起可作为研究生关于 OOP 和 C++的研究生课程用书。本书的绝大多数内容不仅能极大地帮助初学者、初级/中级程序开发和设计人员，而且也能让资深的专家从中获益。在阅读大部分章节后，本书还可以成为你桌上的一本很好的参考书。

本书假设你已熟悉基本的 C++语法，且具备普通的编程经验（仅在课堂做过练习就已足够）。关于语言语法方面，本书对 C++语言更复杂的部分（例如模板、异常和继承）做了详尽的说明。

本书第 2 版更正了第 1 版中发现的错误，更新了一些代码示例，以符合 ANSI C++标准。除此之外，为了更加清楚地说明某些概念，本书第 2 版还重新编写了一些内容。

致　谢

本书是加州大学伯克利分校（University of California Berkeley）教学（和多家公司）的推广计划。感谢参与我所教授课程的数百名学生，我和他们中的许多人已成为好朋友。这些学生们让我从不同的角度观察事物，时常意识到自己对一些问题的简单想法是不对的。他们问我的许多问题，答案都在本书中。正是他们让我意识到，哪些是学生在学习 OOP 和 C++经常难以理解的地方。非常感谢 Jack Grimes 博士，他在百忙之中认真审阅本书，提供了很多反馈材料，他是本书创作灵感的源泉。感谢 Rajiv Maheshwari、Christine Lu 和 Sumathi Kadur 审阅书稿，感谢他们的辛勤付出。他们帮助我消除了很多 bug，保证了内容的准确。感谢 Rampalli Narasimhan 提出的诚恳意见。特别感谢 Erich Gamma 博士与我分享他丰富的面向对象知识，他宝贵的建议和对 Smalltalk 的真知灼见是无价的。感谢 Brooks Applegate 博士带我进入 Smalltalk 的世界。

要特别提到 Taligent 有限公司的好友们。他们是对象技术专家中的杰出人士，他们与我分享的知识财富是无价的，不胜感激。与他们共事非常愉快，丰富了我的阅历。特别感谢 Taligent 设计原则的架构师（本书遵循 Taligent 使用的编码风格要素）David Goldsmith。

向我的妻子 Narmada Iyer 致以最真诚的感谢，感谢她的支持和爱。没有她不断的鼓励（和偶尔的不耐烦）我不可能完成本书的写作。她激励我不断前行，在我废寝忘食时提醒我注意时间。感谢她容忍我所有的癖好（很抱歉错过了假期）。

感谢 Prentice Hall 的优秀团队，特别是，Paul Becker 容忍我一再地延期交稿。特别感谢 Nicholas Radhuber 耐心地接受我在最后阶段无数次地修改。没有他孜孜不倦地努力和付出，本书不会是现在这样。

非常感谢以下热心的读者指出第 1 版的错误：Ojvind Bernandar、Christoph Sadil、Darstoph Wallach、Raza Muzaffar、Rakesh Garg、Anuradha Natarajan、Tom Zal、Fred Campbell、Uma Gopalan、William Ray、Nickolay Baturin、Steve Beckert、Margon Otway。特别感谢 Luther Baker 和 Kiran Prabhakar 在修正错误时提供的个人帮助。

没有 Prentice Hall 杰出团队的帮助，本书的第 2 版不可能面市。再次感谢 Jeffrey Pepper 和他的团队。特别感谢 Nicholas Rdhuber 编辑本书的图片和文字。还要感谢 Penny Baker 在编辑过程中的贡献，你们真的很棒。

目　录

第一部分　概念、实践和应用