



软件工程技术丛书

实现系列

面向对象软件开发原理

(原书第2版)

Principles of
Object-Oriented Software
Development *Second Edition*



(荷) Anton Eliëns 著

袁兆山 等译



机械工业出版社
China Machine Press

软件工程技术丛书

实践系列

面向对象软件开发原理

(原书第2版)

Principles of
Object-Oriented Software
Development *Second Edition*

(荷) Anton Eliëns 著

袁兆山 等译



机械工业出版社
China Machine Press

本书分成四个部分,共12章、10个附录。第一部分面向对象系统设计(第1~4章),介绍OO软件开发的基础知识。第二部分面向对象的语言和系统(第5~7章),比较各种OO语言,讨论开发多语言系统及解决方案。第三部分面向对象的建模基础(第8~10章),介绍实现抽象数据模型方法和类型的相关概念,提出程序验证的断言逻辑,用形式化的方法提供表示对象行为集合特征的概要。第四部分面向对象应用框架(第11~12章),讨论商务处理再设计,对象技术的Web应用及其在计算、智能、移动agent方面的新趋势。附录部分介绍Smalltalk、Eiffel、C++、Java和逻辑程序设计语言DLP,也介绍UML和CORBA IDL(用Orbacus的CORBA应用程序设计指导),提出开发中小型学期项目的建议。

本书基本内容安排紧凑,并引用了许多研究文献,适合不同类型的读者,包括学生(作为课本或研究论文和项目论文的补充读物),软件工程师(作为面向对象软件开发的参考书),专业教师(作为面向对象软件开发教程的现成教材)。本书也适合其他读者,例如研究人员、程序员等,并可作为大专院校OO课程的主要教材,或者自学参考书。

Anton Eliëns: Principles of Object-Oriented Software Development, Second Edition(ISBN 0-201-39856-7).

Original edition copyright © 2000 by Addison Wesley. All rights reserved.

Chinese edition published by arrangement with Addison Wesley Longman, Inc. Chinese simplified language edition published by China Machine Press. All rights reserved.

本书中文简体字版由美国Addison Wesley公司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。

版权所有,侵权必究。

本书版权登记号: 图字: 01-2001-4886

图书在版编目(CIP)数据

面向对象软件开发原理(原书第2版)/(荷)艾伦(Eliëns, A.)著;袁兆山等译. - 北京:机械工业出版社,2003.1

(软件工程技术丛书)

书名原文: Principles of Object-Oriented Software Development, Second Edition
ISBN 7-111-11188-5

I. 面… II. ①艾… ②袁… III. 面向对象语言-软件开发 IV. TP311.52

中国版本图书馆CIP数据核字(2002)第088415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:李伯民

北京忠信诚胶印厂印刷·新华书店北京发行所发行

2003年1月第1版第1次印刷

787mm×1092mm 1/16·23.75印张

印数:0 001-4 000册

定价:48.00元(附光盘)

凡购本书,如有倒页、脱页、缺页,由本社发行部调换

译者序

面向对象(OO)技术,其本质是强调从客观世界固有的事物出发来构造系统,用人类习惯的思维方式来认识、理解和描述客观事物,强调最终建立的软件系统能够映射问题域,软件系统中的对象及对象之间的关系能够如实反映问题域中事物及其关系。这需要理论的指导。尽管OO思想在某种程度上已经非常成功,OO技术风靡全球,从分析、设计到编码,以至整个软件工程,C++,VB,VC,J++等OO语言正取代早期的结构化语言,但是,在很多的领域,OO技术仍然不够成熟,OO技术的理论落后于实践。从OOPL开始就缺乏形式化描述工具;已经发布的OOSD可用的规范,例如UML等,还需进一步完善;OOA的许多问题还有待研究和解决。我们认为,只有对OO技术的来龙去脉有一个彻底的研究,才能从真正意义上对OO技术有一个全面的了解。

《面向对象软件开发原理》(第2版)是荷兰Vrije大学的Anton Eliëns从软件开发角度全面介绍OO软件开发原理的著作。该书是他在OO的研究和教学实践基础上编著的教科书,用于Vrije大学高年级本科生的OO程序设计课程。他在OO软件开发方面造诣颇深,并具有较丰富的实践经验。本书的内容丰富,把OO技术基本原理与软件工程实践联系起来,覆盖面广,可读性好,是一本协调理论与实践及多方面应用的书。

本书的特色是架起了语言与软件工程之间的桥梁,协调理论与实践;介绍了OO程序设计,包括设计、语言和语义的基础,行为求精、模型、组件、软件体系结构,OO建模的标准UML和CORBA;注重可重用、设计模式、分布式对象技术、Web应用中面向对象技术的应用等关键技术问题。清晰表明了软件工程实践中的指导原则与方法之间、指导原则与理论研究中的约束条件和观点之间的相互联系;注重基础问题,为设计者在选择OO语言实现其思想方面,提供了实用的解决方案;提供了大量的Java和C++的实例,包括OO技术的GUI开发、商务处理再设计,多媒体信息系统和Web应用,以及说明概念和问题所用的短小的程序段实例。本书对高级的领域也有很广的覆盖面;以附录形式简要介绍Smalltalk, Eiffel, C++, Java, UML和CORBA;以“幻灯片”形式组织内容,重点概念和观点突出,方便预览;提供联机的教学指导和习题答案。

本书分成四个部分,共12章。

第一部分面向对象系统设计(第1~4章):介绍OO软件开发的领域、OO生命周期和OO程序设计;说明语言的习惯和模式;讨论软件开发过程和分析、设计的各种软件工程观点,获取对象类之间关系的工具——契约,以及基于契约的确认和测试方法;用一个完整的应用阐明设计和实现中的问题,表明如何从形式化说明书到最终实现的过程。

第二部分面向对象的语言和系统(第5~7章):比较各种OO语言,包括Smalltalk、Eiffel、C++、Java以及Self、Javascript,讨论授权动态继承、同步活动对象和基于类语言的元级体系结构等问题;讨论组件技术、分布式对象技术和微软的COM/ActiveX、JavaBeans、Java RMI 和CORBA等解决方案;讨论开发多语言系统,并举例说明如何用Java使本地接口嵌入到现存的框架之中。

第三部分面向对象的建模基础(第8~10章):介绍抽象数据类型、结构化语言和OO语言在实现抽象数据模型方法的不同;讨论类型概念的关联、继承性、子类型关系、多态性、类属和重载的类型理论化的处理方法,获取隐藏和自引用数据的类型计算;提出程序验证的断言逻辑,讨论基于跟踪的对象行为验证的操作模型,对判定语法子类是否满足行为细化的关系给出准确的指导方法,并用形式化的方法提供表示对象行为集合特征的概要。

第四部分面向对象应用框架(第11~12章):讨论IT在商务处理中的重设计,说明商务处理建模把模拟作为获取逻辑结构的工具,通过在Web上布署显示和发布,说明如何建立可以用于决策的模型;讨论对象技术如何应用到Web中及如何应用CORBA,并展望了在计算、智能和移动agent方面的新趋势。

本书有10个附录。附录部分简要介绍Smalltalk、Eiffel、C++、Java和分布式逻辑编程语言DLP,同时包含对UML和CORBA IDL的概述,用Orbacus的CORBA应用程序设计指导,对中小型项目的建议以及各章习题的答案。

本书适合不同类型的读者,包括学生(作为课本或作为研究或项目论文的补充读物),软件工程师(作为面向对象软件开发的参考书),专业教师(作为面向对象软件开发教程的现成教材)。本书也适合其他读者,例如研究人员,程序员等。本书可作为大专院校OO课程的主要教材,或者自学参考书。本书内容安排紧凑,并引用了其他许多研究工作文献。

本书是自成体系的。但是阅读本书的读者需要具备计算机科学的基础知识(计算机语言、最基础的数据结构知识)。要完全理解基础问题或形式化方面,也要具有一些初等数理逻辑知识。

本书由合肥工业大学计算机与信息学院袁兆山教授为主翻译。参加本书翻译工作的有合肥工业大学的胡学钢、王浩、袁晓靖、刘珊珊、袁晓辉、张晶、王骋、张健、李心科、刘宗田、李向上、苗沛荣、张玉红和杨静等。参加审校工作的有袁晓辉、武鹏程、程勇和张艳明,并由袁兆山教授审校定稿。

本书翻译过程中得到了华章图文信息有限公司的大力支持,在此表示诚挚的谢意。

限于译者水平和时间仓促,译文难免有不足之处,欢迎广大读者不吝指正。

译者

2001年12月于合肥

序 言

这是多么不寻常的一本书!

我看过许多关于面向对象软件开发方面的书籍,甚至其中有一些书籍包含相似的内容,但是, Anton Eliëns的书却是完全不同的一种类型。正如这个领域的很多书籍一样,这本书也是以讲义为蓝本。然而,在从讲义到成书的过程中, Eliëns却表现出令人惊奇的不同之处。他不是简单地充实笔记、重组内容和编写传统的教科书,而是把原讲义的精华生动的再现出来。

在整本书中以“幻灯片”作为随处可见的提示和小结,把要点浓缩到幻灯片中,这使读者的感受会非常不同。跟随着这些幻灯片,读者能得到快捷的线索:这正是刚刚入门的学生求之不得的。然后,读者可以进行进一步地阅读,把精力集中在对Eliëns的全文说明的掌握上。对书中重要部分的解释很透彻,而对大量无需深究的细节的解释又很浅显。阅读这本书并通过实践来加深我们的理解,这是令人愉快的。

为了了解本书的结构,我阅读了第1版的前言和序言。然而,值得指出的是Eliëns对本书的第1版已经作了重大改进。组件是我所关注的主题,已经编排到书中。软件体系结构,是与此紧密联系且日益重要的主题,这在书中也有所涉及。

全书的论题十分广泛,使本书所体现的风格和教学质量的要求相符。生活在这个瞬息万变、学科极其多样化的时代,我们不得不重视为数不多的好书,这些书涵盖范围很广,综合性很强。毕竟,使用像Eliëns的这一类书,其中重构和综合了大量的领域之外的边缘知识,这对读者来说是重大的挑战。然而,在Eliëns这本书的帮助下,读者完全有机会去攻克这个难以逾越的目标。教师和讲授者也同样会感受到这本书给教学带来的益处。

最后,对于未来的软件开发者和当今的软件开发者或关注软件开发的计算机科学家来说,需要了解更多的分支领域的知识,以便真正地把握软件开发。为了达到这个目的,我们需要了解从建模和设计技术、体系结构和组件到用某种特定的编程语言实现细节的全过程及其中的一切。本书是从面向对象观点去完成这些事情的很好的起点。(当然也要关注其他的观点和方法!)

Clemens Szyperski

1999年11月

第1版序言

本书是对面向对象技术的巨大贡献，它在语言与软件工程之间架起了一座桥梁。本书的内容包括了语言设计的相关问题：继承（inheritance）、类型（type）、多态性（polymorphism）、活动对象（active object）、以及对象建模技术（object modeling technique, OMT）、模型视图控制器（model-view-controller, MVC）、责任驱动设计（responsibility-driven design）等软件设计的范例。本书的主题分成四个部分：面向对象的设计，语言和系统，面向对象的基础，应用框架（这使实践和理论得到很好的平衡，它同时包括实用的设计技术和基本模型）。本书以C++作为主要的应用语言，Smalltalk和Eiffel语言作为辅助语言，因而可作为主流面向对象语言的程序设计教程。

本书引人入胜的风格以及对广泛主题的最新处理方法，与总体的平衡和观念浑然一体。每一章后面都有概念性的习题（附录中有答案），其中有一些是很有情趣的。例如，为了让读者检验C++语言一种结构的价值，提问“你为什么需要朋友？”在回答中指出，要权衡效率和安全性，最终告诫你要“善待朋友”。

面向对象程序设计最初是作为单用户系统的语言框架，但是它逐渐走向成熟，发展为着眼于交互操作性（interoperability）和用异构模块集成的异构分布式网络系统。面向对象程序设计结构的概念，与结构化程序设计的结构是类似的，但要复杂得多。本书的内容反映了从单用户到分布式系统技术逐步成熟的过程，架起了一座从单用户程序设计的面向对象概念到分布软件设计概念的桥梁。

基本的面向对象概念是从设计观点提出来的，因此，通过它在软件生命周期中担当的角色促进了程序设计语言概念的发展。前四章介绍基本概念，即使对于经验丰富的读者，也有意想不到的价值。第1章讨论编程范例，提出一种独特的软件生命周期的面向对象观点。第2章给出C++，讨论其优点和缺陷，并与Smalltalk和Eiffel加以比较。第3章关于面向对象设计，深入讨论模型、契约（contract）和规格说明（specification），对设计的概念基础的替代方法进行了比较和综合。第4章着重于软件验证的测试和度量的讨论，并与前面三章中的重点概念进行对照。

前四章的内容是基础，和后面的主题有着紧密联系。语言设计原理一章是继承和委托（delegation）模型的回顾；关于并发性的一章深入讨论继承异常、并发对象模型和分布式程序设计的基本原理；组成和协作一章，解释回调（callback）、窗口管理和事件驱动计算。关于基础的三章探讨抽象数据类型的代数模型，计算类型的多态性，通过子类型进行行为细化。最后两章讨论大量的互操作性、标准、库设计、需求工程、超媒体链接和异构系统。

本书涵盖了广泛的主题，非常容易阅读，它在理论和实践的平衡以及多方面的展开是非常独特的。Anton Eliëns展示了其对最新文献的精通，以及对不同的软件研究和开发领域的工作进行比较、评估和综合的杰出能力。在异构分布式领域，他以其在设计一个顺序解释线程中所表现出的技能和多才多艺而备受赞扬。

Peter Wegner

1994年10月

前 言

这是一本有关面向对象软件开发的书，它是基于在阿姆斯特丹的Vrije大学高年级本科生面向对象程序设计课程的内容。

以上是第一版前言的开始部分。这句话依然是正确的。然而，面向对象（OO）是一个迅速发展的领域。第1版是1994出版的，到如今可能已经有些过时。例如，刚出版后不久，模式成为公众兴趣的焦点。再如，Java的大潮也扑面而来。有鉴于此，需要及时再版本书，增加这样或那样的主题，以及如在CORBA的情形增加更多的细节。

另一个原因，是在于OO领域本身已经相当成熟。接受UML作为建模语言标准就是一个例证。在重要的商务应用中利用CORBA的应用迅速增加，这是一个信号，说明（分布式）对象技术具有足够的健壮性。

由于许多的出版物已经包含了这些内容，内容本身并不能证明第2版的合理性。你只需想象一下大量的Java方面的书……。以我之见，既然可以通过书中采用的叙述方法来区别于其他同类书，那么，修订的第二版可以证明其合理性。通过“幻灯片”组织起一系列的讲座，这本书包含了很多主题，有些很有深度，有些却较浅显。从教学的观点来看，这种方法的好处是直接使用教材，包括在课堂上演示幻灯片。而且，对于大多数读者来说，“幻灯片”能起到很好的帮助理解和回忆的作用。

最后，推出修订版还有个人原因，在OO的研究和教学中，我已经积累了丰富的经验，甚至可以说，我的OO思想在某种程度上已经非常成熟。特别是，正如在第1版的第12章所介绍的，我的课题组已经开发出多范例OO框架，并应用于诸如Web的商务处理的再设计和在Web上的集体即兴创作（collective improvisation）等很多方面。虽然我没打算过多地讨论这些内容，但是它已经为许多例子提供了基础，而且在随附的光盘上可获得这些材料（包括论文、软件和例子）。

本书的特色

- 介绍了面向对象程序设计，包括设计、语言、基础问题。注重重用（reuse）、组件（component）技术、设计模式等问题，尤其是Web应用中面向对象技术的应用。
- 包含开发面向对象应用程序的准则。除了实例，提供了开发方法的概要，以及介绍面向对象建模的标准UML。尤其是将设计模式（design pattern）作为反复涉及的问题，并将从这个角度讨论例子和解决方案。
- 分布式对象技术将会成为一个主要题目。介绍了CORBA，使学生获得开发CORBA应用的亲身经验。同时讨论了相互竞争的技术，特别是详细阐明组件技术和分布式对象的区别。用Java和C++编写了例子程序。
- 本书的另一个主要题目，是精确地建立软件工程实践中的指导原则与方法之间的关系，以

及建立理论研究中的约束条件和观点之间的联系。注重基础问题，也为设计者在选择面向对象语言实现其思想方面，提供实用的解决方案。

- 介绍的许多概念和讨论的许多问题都用短小的程序段加以说明，大部分用Java编写，还有用C++编写。实例包括GUI开发、商务处理二次工程（reengineering）和Web应用。由于附录中已经简要介绍了一些面向对象程序设计语言，包括C++、Smalltalk、Eiffel和Java，所以就没有过多地涉及到语言知识。
- 内容都以“幻灯片”的形式组织起来。书中的幻灯片采用简略的格式，但是也可用Powerpoint和Netscape Presentation格式。每一张幻灯片提炼出相关章节中所论述的重点概念和观点。具有联机“教师指南”，提供演示幻灯片的提示和习题答案。
- 整本书，包括例子中的软件和“教师指南”，可从本书附带的光盘和Internet上得到电子版。电子版也包括对因特网上其他材料的链接。电子版可用幻灯片格式在课堂上演示。

读者对象

本书主要是针对高等院校的读者或IT专业人员。但是，自从我愈来愈多地和商业伙伴进行合作研究，以及介入院校课程以外的开发，与IT有关的商业实例自然也被吸收进来。特别是，包括了商务处理重设计的面向对象模拟的推广一节，以及使用对象技术实现商业数据3D显示一节。

本书可作为OO课程的主要教材，或者自学参考书。可供以下不同类型的读者使用：

- 学生——作为课本，或作为主题研究、项目论文的参考读物
- 软件工程师——作为面向对象软件开发的参考书
- 专业教师——作为面向对象软件开发教程的现成教材

自然，这并不意味着其他读者不能使用。比如，研究人员会发现书中解决基础问题方法的用处。程序员能从提示和用Java和C++编写的程序例子中得到帮助。使用本书的另一个原因是对所熟悉的材料安排紧凑，并引用了其他（通常是研究）方面的文献。

本书自成体系。不过，读者仍需要掌握计算机科学的基础知识（至少要了解计算机语言和数据结构）。要完全理解讨论基础问题或形式化方面的各节，也要具备一些初等数理逻辑的知识。

组织

本书分成四个部分。每一部分从不同的角度对面向对象程序设计中涉及的问题进行讨论，分别是软件工程和设计，语言和系统开发，抽象数据类型和多态性，应用和框架。

第一部分：面向对象系统设计

第1章 简介

这一章介绍面向对象软件开发的领域。给出面向对象生命周期的全局观点，讨论对象作为程序设计范例的定位。讨论一些成为公众关注焦点的趋势和技术，阐明它们与‘对象定位’的关系。

第2章 习惯用法和模式*

这一章介绍语言的习惯用法和设计模式，作为获取面向对象程序设计的递归结构和解决方案的手段。区分与某种特定语言相联系的解决方案之间的不同习惯用法，以及作为合理产品设计之间的不同模式。本章包含大量的Java实例。

第3章 软件工程观点

这一章讨论软件开发过程，以及在分析和设计中各种各样的建模观点。解释获得一个合理对象模型涉及的问题，介绍契约（contract）的概念，它被作为获取对象类之间关系的工具。另外，提出一个基于契约的验证和测试方法。

第4章 应用开发*

在这一章，介绍我们开发的一个完整的应用程序，并讨论设计和实现中的问题。提供（个体）类设计的准则，给出了一个例子，表明如何从形式化的规格说明导出最终实现的过程。

第二部分：面向对象语言和系统

第5章 面向对象程序设计语言*

这一章比较各种面向对象语言，包括Smalltalk、Eiffel、C++ 和Java。进一步讨论了其他语言，例如Self和Javascript，其中每一种都有自己的对象模型，论述了诸如委派动态继承、同步活动对象、基于类语言的元级（meta-level）体系结构等问题。

第6章 组件技术*

这一章讨论组件技术和分布式对象技术之间的关系，给出从市场可以获取的解决方案的概况，其中包括微软的COM/ActiveX、JavaBeans、Java RMI 和CORBA。同时提供一个简单的工作组应用，以及集成CORBA和一个现存的软件库的例子。

第7章 软件体系结构*

这一章说明软件体系结构是如何影响设计和实现的。寻求用于分布式对象系统的设计模式，讨论开发多语言系统的技术问题。举出一个例子，表明如何用Java使本机接口嵌入到已存在的框架之中。

第三部分：面向对象建模的基础

第8章 抽象数据类型

这一章从类型约束的角度考虑抽象数据类型，提出可将对象当作一种数学问题来处理的代数方法。而且，解释过程语言和面向对象语言在实现抽象数据模型的典型方法之间的差异。同时讨论更实际的类型概念所隐含的内容。

第9章 多态性

这一章从声明的角度讨论继承性，给出子类型关系的精确特征。进一步剖析多态性的各种特点，提出类属(genericity)和重载(overloading)的类型理论化的处理方法。同时，给出获取数据隐藏和自引用（self-reference）的类型计算。用Eiffel、C++和Java实现多态性过程中，这些都是相关联的。

第10章 行为求精

这一章把类型概念扩展为包括行为属性在内的类型约束。提出程序验证的断言逻辑，讨论基于跟踪的对象行为验证的操作模型。对判定语法子类是否满足行为求精关系，给出了明确的准则。最后，概述表示对象集合行为特征的形式化方法。

第四部分：面向对象应用框架

第11章 商务处理再设计*

这一章我们讨论IT在商务处理再设计中所提供的机会。特别是，我们注意把面向对象的模拟作为商务处理建模获取逻辑特征的工具，此外，考察如何通过Web上布署显示和发布，就可以建立能够用于决策的模拟模型。

第12章 Web应用*

这一章介绍对象技术如何应用到Web中。将考察客户端的扩展和服务端端的解决方案。尤其是，除了Web技术，还考虑采用CORBA的系统。我们也大致展望了在计算、智能、移动agent方面的新趋势，而且我们认为agent是直接来源于对象技术。

附录

附录包括Smalltalk、Eiffel、C++、Java和分布式逻辑编程语言DLP的简要介绍。其中也包括UML和CORBA IDL的概要，用Orbacus的CORBA应用程序设计指导，以及对中、小型项目的建议。

路径

对于面向对象程序设计课程，这本书提供了几条路径以供选择，取材如下表所示。同时，指明包括高级材料的各节。

	常规部分	扩充部分	高级部分
程序设计	2、4、5、12	6、11	7、8
软件工程	1、3、4、11	8.1-2、10.1	9.1-3、10.2
理论	1、3、8	5、9.1-4	9.5-6、10

程序设计课程路径包括第2、4、5章和第12章，可参看附录和第6、11章。软件工程课程路径包括第1、3、4章和第11章，可增补理论课程中的材料。理论课程路径，包括第8、9章和第10章，也需要从其余两种课程增补更多有关OOP一般信息的材料。

与第一版的不同之处

为清楚起见，我在作重大修改的各章上标注了星号*。

一方面，增添了新的内容，另一方面，必然要删除一些部分。我删去了原来的以下几章：C++（第2章）、软件工程问题（第4章）、C++的并发（第6章）、组合机制（第7章）、软件库（第11章）以及超媒体（第12章）。其中，有些部分，如超媒体（第12章）、组合机制（第7章）、软件工程问题（第4章）将在其他地方再现。然而，有些部分是孤立的，而另外一些在课堂中作用

不大，最好删掉，以便为其他题材留出更大的空间。

背景和动机

我对面向对象语言和软件开发的兴趣来源于对DLP语言的研究，这是一种具有面向对象特征和并行性的语言（Eliëns, 1992）。当我想寻求一本面向对象程序设计的教材时，我发现没有一本书能够做到足够重视基础和形式化方面。大多数教材只是从OOP的某个角度着手编写，这并不完全符合我的目的。在某种程度上，我想要的东西只能从研究论文中找到。所以，我从为数不多的论文中选取了一部分，组编了这本OOP教程，我认为，这些论文是里程碑性质的，它们以其提出某些重要的概念和非凡的见识而著名。但是，基于OOP论文的教程具有明显的不足之处，即显然缺乏一个统一的观点和用法一致的术语。这种方法的优势在于，促使学生通过批判地比较论文表述的不同观点，去评价每篇论文的贡献，以形成他们自己的见解。我个人更愿意使用原文，因为可以从中清楚地看出那些思想最初是如何提出的。经过多次润色后的文字经常缺乏这种“发现”的品质。

利用“幻灯片”组织这本书的想法是很自然的，因为我不断制作了很多幻灯片，同时希望通过幻灯片来保持叙述的紧凑性。

材料的选择反映了我个人对基础问题的侧重点，换句话说，论文更注重的是概念而不是实践。我的兴趣在分布式超媒体系统和Web上，这体现在教材内容的选择上。在某种程度上，这也与我过去所从事的分布式逻辑程序设计工作有关。虽然这本书没有以语言结构为重点，但是建模问题和基础问题与现存的或可以想象的语言结构是普遍相关的，都可以使用为这种目的而开发的实例来说明（只要可能的话）。

选择Java作为书中的程序段或例子的开发工具，只是因为人们普遍使用Java。另外一些例子是用C++写的，因为我相信C++仍旧是面向对象软件开发的有效的编程语言。然而，我也相信，在不久的将来，多种范例的方法（扩展Java和C++）将扮演更重要的角色。

本书所体现的特色可以说就是抽象，注意力主要集中在概念上而不是在解决方案或实现语言的细节上。很偶然，为了准备课堂讨论，我在字典中查阅了抽象（abstract）这个词。我惊奇地发现，它还有一个意思是偷窃（to steal/to take away dishonestly）。我觉察到，这个词义使抽象数据类型具有了不同的含义，但从更深层我认识到，本书所提出思想，在某种程度上得益于其他人最初提出的思想。然而，我用更抽象的方式表达这些思想，并不意味着用不诚实的方式剽窃他们的成果，而是把这些思想融入到合适的环境和框架中，这包括了面向对象计算的理论和实用性两个方面。字典中还有抽象（abstract）一词的形容词含义abstruse（难解的，深奥的）。这不用多说，我将尽我所能，努力避免难以理解。

最后，在本书的取材中，我尽量保持足够程度的客观性。然而，只要带有个人判断，就会引起争论而不会成为定论。

信息

电子版可从网址<http://www.cs.vu.nl/~eliens/online/oo>获得。有任何问题或建议，请与作者联系，电子邮件地址eliens@cs.vu.nl，通信地址Dr A. Eliëns, Vrije Universiteit, Faculty of

Sciences, Division of Mathematics and Computer Science, De Boelelaan 1081, 1081 HV Amsterdam, The Netherland.

光盘内容

光盘包括本书完整的英文电子版, 以及附带的讲义、软件和链接到因特网上的资源。电子版可在课堂上演示, 用Netscape Presentation格式, 由Netscape Navigator 4.x或更高版本以及IE 4.x或更高版本支持。光盘也提供PowerPoint格式的演示, 可供教师使用。详细信息请参看光盘上的README文件。

致谢

在本书编写过程中, 得益于很多学生和同事的热情帮助和中肯批评。在最近几年, Jacco van Ossenbruggen、Bastiaan Schönhage和Martijn van Welie都成为我的博士研究生, 他们坚持不懈地为概念的清晰提出建议、批评和意见。

本书有些材料取自合写的相关论文。Jacco van Ossenbruggen负责了第2章中的句柄/主体用法。John Caspers编写第6章的工作组应用程序。第7章的体系结构的样式与风格一节由Bastiaan Schönhage执笔。第11、12章是在Jacco van Ossenbruggen、Bastiaan Schönhage、Martijn van Welie、Frank Niessink、Dirk Bolier、Sam Megens和Paul Nash的密切合作下完成的。我也要感谢我的同事Hans de Bruin为我提供了软件体系结构方面的材料。感谢ASZ研究开发主管Thiel Chang和ASZ/旧金山开发小组, 包括Arne Bultman、Joris Kuipers、Ard van der Scheer、Remco van de Woestijne和Irmien de Jong, 他们在一个令人感兴趣的项目中给予大力合作。最后, 也同样感谢我的学生们, 他们在学期论文中对面向对象提出了新颖的见解。

对第1版, 我非常感谢Hans van Vliet、Henri Bal、Dick Grune、Cees Visser、Ira Pohl、John Caspers和Matthijs van Doorn (排名不分先后)。同时, 我对Chris Dollin表示感谢, 他为第1版做了详细的和有建设性的评论。最后, 要感谢各位编辑: Sally Mortimore同我签订了第2版合同; 当Sally离开之后, Keith Mansfield接替了他留下的工作; Alison Birtwell和Karen Sutherland给予我大力协助; Hedwig van Lier (供职于Pearson Education) 给我提供了很多好书; Julie Knight在出书过程中自始至终给予我指导; Karen Mosman和我讨论了在一系列Web会议中使用联机版本的可能性; 我也不能忘记Simon Plumtree和Andrew McGettrick的工作, 因为他们真正地懂得“灵活处理”。

面向对象程序设计中蕴含的概念, 第一眼看上去会令人误认为很简单 (不需要C++的复杂性和类型理论的分析)。然而, 在开发面向对象的模型和应用中, 仍旧存在内在的难点问题, 而对这些我们没有确切的答案, 甚至可能需要丰富的专业知识和技术才能求得部分解决方案。

Anton Eliëns

2000年1月

目 录

译者序	
序言	
第1版序言	
前言	

第一部分 面向对象系统设计

第1章 简介	1
1.1 主题与变化	1
1.1.1 对象术语	3
1.1.2 面向对象的计算	4
1.1.3 按契约设计	7
1.2 编程范例	10
1.2.1 过程化程序设计	11
1.2.2 数据抽象	12
1.2.3 面向对象编程	12
1.3 面向对象的软件生命周期	13
1.3.1 需求分析	15
1.3.2 设计	17
1.3.3 实现	18
1.4 超越面向对象吗	20
1.5 小结	22
习题	23
进一步读物	23
第2章 习惯用法和模式	24
2.1 多态性	24
2.1.1 Java中的继承和委托	25
2.1.2 C++中的多态性	27
2.1.3 C++中的断言	28
2.1.4 规范类的用法	30
2.2 hush的用法	31
2.2.1 句柄/主体用法	34
2.2.2 虚拟自引用	36

2.2.3 动态角色转换	37
2.2.4 hush编程的技巧	39
2.3 设计模式的分类	41
2.3.1 创建性模式	42
2.3.2 结构模式	43
2.3.3 行为模式	44
2.4 事件驱动的计算	46
2.4.1 reactor模式	46
2.4.2 抽象事件系统	48
2.5 小结	51
习题	52
进一步读物	53
第3章 软件工程的观点	54
3.1 软件开发方法	54
3.1.1 建模	56
3.1.2 需求工程——综合法	57
3.1.3 分析和设计方法——比较研究	58
3.2 标识对象	60
3.2.1 启发式建模	61
3.2.2 分派责任	63
3.2.3 对象角色和交互	65
3.3 契约	66
3.3.1 说明契约化责任	66
3.3.2 契约的求精	69
3.3.3 运行时一致性检查	71
3.4 形式化方法	71
3.5 小结	72
习题	74
进一步读物	74
第4章 应用开发	75
4.1 drawtool应用	75
4.2 设计准则	86

4.2.1 单个类的设计	87
4.2.2 继承和不变性	87
4.2.3 风格的客观意义	88
4.3 从规格说明到现实	90
4.3.1 结构封装与行为封装	90
4.3.2 基于模型的规格说明	91
4.3.3 抽象系统和事件	95
4.4 小结	98
习题	99
进一步读物	99

第二部分 面向对象的语言和系统

第5章 面向对象程序设计语言	101
5.1 对象范例	101
5.1.1 面向对象语言的分类	103
5.1.2 替代对象模型	105
5.1.3 Lisp, C和Prolog的对象扩展	105
5.1.4 脚本语言——与Java的集成	107
5.2 Smalltalk, Eiffel, C++和Java的比较	108
5.2.1 比较的标准	109
5.2.2 语言特征	110
5.3 面向对象语言的设计维度	113
5.3.1 基于对象与面向对象	113
5.3.2 关于正交化方法——类型扩展	114
5.3.3 多范例语言——逻辑程序设计	116
5.3.4 主动对象——同步Java / C++	118
5.4 原型-委托与继承的比较	120
5.4.1 共享的替代形式	120
5.4.2 实现技巧——Self	122
5.5 元级体系结构	123
5.6 小结	125
习题	127
进一步读物	127
第6章 组件技术	128
6.1 对象与组件	128
6.1.1 定义	128
6.1.2 技术矩阵	129

6.1.3 组件神话	130
6.2 互操作性的标准	131
6.2.1 对象链接与嵌入——COM	132
6.2.2 对象请求代理——CORBA	132
6.2.3 持久对象——ODMG	134
6.3 Java平台——技术革新	136
6.4 一个基于Internet的工作组应用程序	138
6.5 crush——用CORBA扩展hush	139
6.6 小结	148
习题	149
进一步读物	149

第7章 软件体系结构	150
7.1 体系结构的元素	150
7.2 实例研究——多媒体特征检测	152
7.3 交叉边界	158
7.3.1 嵌入式逻辑——跨范例边界	159
7.3.2 本机对象——跨语言边界	160
7.3.3 Java和C++的结合	164
7.4 体系结构的模式与风格	168
7.4.1 从技术到风格	168
7.4.2 实例研究——可视化的角度	169
7.5 交叉平台开发	171
7.6 小结	172
习题	173
进一步读物	174

第三部分 面向对象建模的基础

第8章 抽象数据类型	175
8.1 抽象和类型	175
8.1.1 编程语言中的抽象	175
8.1.2 基本观点——类型作为约束条件	176
8.1.3 OOP类型的目标	178
8.2 代数规范说明	179
8.2.1 特征——生成器和观察器	180
8.2.2 等式——规定约束条件	182
8.2.3 初始代数语义	184
8.2.4 作为代数的对象	187

8.3 分解——模块与对象	190
8.3.1 抽象接口	191
8.3.2 表示和实现	192
8.3.3 添加新生成器	194
8.3.4 添加新观察器	196
8.4 类型与类	198
8.5 小结	201
习题	202
进一步读物	202
第9章 多态性	203
9.1 抽象继承	203
9.2 子类型关系	205
9.2.1 作为集合的类型	205
9.2.2 子类型细化关系	206
9.2.3 作为记录的对象	208
9.3 多态性的风格	210
9.4 类型抽象	212
9.4.1 简单类型演算	214
9.4.2 交叉类型	217
9.4.3 受限多态性	220
9.5 存在类型——隐藏	223
9.6 自引用	225
9.7 小结	230
习题	232
进一步读物	232
第10章 行为求精	233
10.1 作为行为的类型	233
10.2 检验行为属性	235
10.2.1 状态转换	235
10.2.2 断言逻辑	236
10.3 行为概念	238
10.4 作为行为类型的对象	241
10.4.1 抽象和表示	241
10.4.2 对应关系	244
10.5 说明行为组成	246
10.6 小结	250
习题	251

进一步读物	252
-------------	-----

第四部分 面向对象应用框架

第11章 商务处理再设计	253
11.1 商务对象——San Francisco框架	253
11.2 商务处理模型	256
11.2.1 基于逻辑的模型	256
11.2.2 商务处理模拟	257
11.2.3 实例——贷款需求	259
11.3 面向对象模拟	260
11.4 可视化支持	267
11.4.1 有多少用户、就有多少视角	268
11.4.2 DIVA——分布式可视化体系结构	269
11.5 从遗留应用中迁移	271
11.6 小结	272
习题	273
进一步读物	274
第12章 Web应用	275
12.1 对象和Web	275
12.1.1 趋势和技术	276
12.1.2 Object Web——CORBA/Java与微软 对抗	278
12.2 Web程序设计——API的探索	280
12.2.1 计算模型	280
12.2.2 智能代理	282
12.3 DejaVU的经验——Web上的音乐演奏	284
12.4 软件体系结构回顾	289
12.5 小结	291
习题	292
进一步读物	292

附 录

附录A Smalltalk语言	293
附录B Eiffel语言	298
附录C C++语言	302
附录D Java语言	309
附录E DLP语言	315

附录F 统一建模语言	320	附录I 软件开发项目	336
附录G 接口定义语言——IDL	324	附录J 习题答案	338
附录H Hello (CORBA) Universe	327	参考文献	345