21世纪高等学校本科计算机专业系列实用教材

面向对象

程序设计教程

◎ 冷英男　李文超　编著

電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

21 世纪高等学校本科计算机专业系列实用教材

面向对象程序设计教程

冷英男　李文超　编著

電子工業出版社
Publishing House of Electronics Industry
北京 • BEIJING

内 容 简 介

本书侧重于面向对象方法处理问题的观点和原理，用 C++程序设计语言作为描述工具，介绍面向对象的基本方法、实现机制、具体编程技术，并在各相关章节中介绍相应的软件开发方法。具体内容包括面向对象程序设计概论、从 C 语言到 C++语言、类与简单对象、类与复杂对象、继承和派生类、类成员的进一步使用、多态性和虚函数、运算符重载、模板、I/O 流、异常处理、面向对象软件开发方法。

本书可作为高等院校计算机科学与技术专业教材，也适用于高校教师、计算机科技人员及其他相关读者参考。

图书在版编目（CIP）数据

面向对象程序设计教程 / 冷英男，李文超编著. —北京：电子工业出版社，2007.6
（21 世纪高等学校本科计算机专业系列实用教材）
ISBN 978-7-121-04413-7
Ⅰ. 面… Ⅱ. ①冷… ②李… Ⅲ. 面向对象语言—程序设计—高等学校—教材 Ⅳ. TP312
中国版本图书馆 CIP 数据核字（2007）第 070971 号

责任编辑：刘海艳（lhy@phei com cn） 文字编辑：毕军志
印 刷：北京季蜂印刷有限公司
装 订：三河市万和装订厂
出版发行：电子工业出版社
北京市海淀区万寿路 173 信箱 邮编 100036
开 本：787×1092 1/16 印张：21 25 字数：544 千字
印 次：2007 年 6 月第 1 次印刷
印 数：5 000 册 定价：29 00 元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：（010）88254888。

质量投诉请发邮件至 zlts@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线：（010）88258888。

编委会名单

主任委员 庄燕滨

副主任委员 张永常 邵晓根 范剑波 沈振平 倪 伟 马正华 范兴南
华容茂

委　　员（以姓名笔画为序）

丁志云 丁海军 王 琳 石敏辉 刘玉龙 刘红玲 朱宇光
朱信诚 冷英男 闵立清 吴 胜 杨玉东 杨茂云 张宗杰
张碧霞 张献忠 查志琴 赵立江 赵 梅 郭小荟 徐煜明
唐土生 唐学忠 程红林 彭 珠 韩 雁

序　言

21 世纪是“信息”主导的世纪，是崇尚“创新与个性”发展的时代，体现“以人为本”、构建“和谐社会”是社会发展的主流。然而随着全球经济一体化进程的不断推进，市场与人才的竞争日趋激烈。对于国家倡导发展的 IT 产业，需要培养大量的、适应经济和科技发展的计算机人才。

众所周知，近年来，一些用人单位对部分大学毕业生到了工作岗位后，需要 1～2 年甚至多年的训练才能胜任工作的“半成品”现象反映强烈。从中反映出单位对人才的需求越来越讲究实用，社会要求学校培养学生的标准应该和社会实际需求的标准相统一。对于 IT 业界来讲，一方面需要一定的科研创新型人才，从事高端的技术研究，占领技术发展的高地；另一方面，更需要计算机工程应用、技术应用及各类服务实施人才，这些人才可统称“应用型”人才。

应用型本科教育，简单地讲就是培养高层次应用型人才的本科教育。其培养目标应是面向社会的高新技术产业，培养在工业、工程领域的生产、建设、管理、服务等第一线岗位，直接从事解决实际问题、维持工作正常运行的高等技术应用型人才。这种人才，一方面掌握某一技术学科的基本知识和基本技能，另一方面又具有较强的解决实际问题的基本能力，他们常常是复合性、综合性人才，受过较为完整的、系统的、有行业应用背景的“职业” 项目训练，其最大的特色就是有较强的专业理论基础支撑，能快速地适应职业岗位并发挥作用。因此，可以说“应用型人才培养既有本科人才培养的一般要求，又有强化岗位能力的内涵，它是在本科基础之上的以‘工程师’层次培养为主的人才培养体系”，人才培养模式必须吸取一般本科教育和职业教育的长处，兼蓄并顾。“计算机科学与技术”专业教学指导委员会已经在研究并指导实施计算机人才的“分类”培养，这需要我们转变传统的教育模式和教学方法，明确人才培养目标，构建课程体系，在保证“基础的前提”下，重视素质的养成，突出“工程性”、“技术应用性”、“适应性”概念，突出知识的应用能力、专业技术应用能力、工程实践能力、组织协调能力、创新能力和创业精神，较好地体现与实施人才培养过程的“传授知识，训练能力，培养素质”三者的有机统一。

在规划本套教材的编写时，我们遵循专业教学委员会的要求，针对“计算机工程”、“软件工程”、“信息技术”专业方向，以课群为单位选择部分主要课程，以计算机应用型人才培养为宗旨，确定编写体系，并提出以下的编写原则。

（1）本科平台：必须遵循专业基本规范，按照“计算机科学与技术”专业教学指导委员会的要求构建课程体系，覆盖课程教学知识点。

（2）工程理念：在教材体系编写时，要贯穿“系统”、“规范”、“项目”、“协作”等工程理念，内容取舍上以“工程背景”、“项目应用”为原则，尽量增加一些实例教学。

（3）能力强化：教学内容的举例，结合应用实际，力争有针对性；每本教材要安排课程实践教学指导，在课程实践环节的安排上，要统筹考虑，提供面向现场的设计性、综合性的实践教学指导内容。

（4）国际视野：本套教材的编写要做到兼长并蓄，吸收国内、国外优秀教材的特点，人才培养要有国际背景和视野。

本套教材的编委会成员及每本教材的主编都有着丰富的教学经验，从事过相关的工程项目（软件开发）的规划、组织与实施，希望本套教材的出版能为我国的计算机应用型人才的培养尽一点微薄之力。

编委会

前　言

面向对象程序设计（Object-Oriented Programming，OOP）和面向过程程序设计（Procedure-Oriented Programming）是两类主要的程序设计范型（Programming Paradigm）。面向过程程序设计基本上是面向机器（计算机）的。在这种程序设计范型中，程序以行为为基础，按照流程来组织。也就是说，人们必须从机器的观点出发来考虑程序的设计。因此，在本质上，它难以更好地描述现实世界；另外，其软件也较难实现重用。而面向对象程序设计强调“面向对象”的思想，它允许人们以问题域的结构为基础去设计程序，因而程序的构造过程更自然。与面向过程的程序设计方法相比，面向对象程序设计更符合人们观察和分析问题的习惯。而且，面向对象技术提供了较高的软件重用可能性，这使得面向对象程序设计技术得以迅速发展。当前，面向对象程序设计已经成为事实上流行的程序设计主流技术。专家们预测，面向对象程序设计思想将主导今后程序设计语言和程序设计方法的发展。

简单地说，“面向对象”的思想就是把软件结构建立在“对象”之上，而不是建立在行为之上。面向对象程序设计突出了软件重用的原则。“面向对象”实现程序设计使用的是封装、继承、多态、动态联编等具体技术。于是，什么是对象；如何确定对象；在程序中，对象是如何被操作的；对象间都有些什么关系、如何描述对象和实现这些关系等问题都是“面向对象”技术必须解决的问题。对这些问题的解决方法就形成了“面向对象程序设计”的主要内容。

程序设计必须在程序设计语言的支持下才能实现，所以如何更好地支持面向对象程序设计思想，就成了程序设计语言本身的研究必须考虑的问题。可以说，面向对象程序设计极大地影响了程序设计语言的发展。

但是，面向对象程序设计作为一种程序设计原理，应该独立于具体的程序设计语言。这意味着，面向对象程序设计思想也可以在传统的面向过程的程序设计语言环境中实现。例如，使用 C 语言、FORTRAN 语言或 Pascal 语言编写面向对象的程序。当然，在面向对象程序设计语言环境中进行面向对象程序设计，可以使面向对象思想得到更好的支持。所以，学习面向对象程序设计过程中，掌握程序设计语言的特征固然很重要，但掌握面向对象程序设计思想却是更本质的要求。

面向对象程序设计并没有否定传统的结构化程序设计原理，相反地，面向对象程序设计补充了这一原理，使得它比以往更加成熟和稳固。计算机执行程序在本质上是过程性的，因此，学习计算机程序设计不应完全割裂面向对象程序设计和面向过程程序设计这两个范型。使用面向对象程序设计语言对于大型软件的开发有明显的优势，但在小型软件的开发中这种做法尚有待商榷。

本书采用 C++语言作为介绍面向对象程序设计的描述工具。C++语言是 20 世纪 90 年代为了开发和维护复杂的应用软件而研制的。C++语言以 C 语言为基础，通过引入面向对象技术对其进行了扩充。因此，C++语言既支持 C 语言的机制，也支持面向对象的机制，在面向对象程序设计语言分类中，属于一种混合型面向对象程序的设计语言。

由于 C++语言既支持 C 语言的代码风格，又支持面向对象程序设计技术，所以，学习

C++语言对于初学程序设计的人来说，就可以同时熟悉传统的结构化程序设计和面向对象程序设计。这是我们选择 C++语言作为介绍面向对象程序设计技术的程序设计语言的主要原因。当然，学习面向对象程序设计更重要的是掌握面向对象程序设计思想，使它们在具体的语言实现机制中体现出来。

本书的重点是介绍面向对象程序设计方法，以 C++语言作为描述语言，所以，本书也可以作为学习 C++语言的教材。通常，偏重程序设计方法论的部分在大学的教学计划中都放在软件工程课程中讲述，程序设计或程序设计语言课程较少涉及设计方法论，两类课程一般间隔较长，这样，不利于理解程序的架构和体系。但在面向对象程序设计的学习中，理解程序的整体框架结构是非常重要的。所以，本书包含了有关方法论的内容。如果受学时的限制，或者以本书作为 C++语言的教材，那么这些内容可以作为学生自学和提高的内容而不必列入授课计划。

全书共分 12 章。第 1 章是面向对象程序设计概论，介绍了面向对象程序设计的基本原理和思想，便于在总体上理解面向对象技术。第 2 章主要介绍从 C 语言到 C++语言；第 3 章主要介绍类与简单对象；第 4 章主要介绍类与复杂对象；第 5 章主要介绍继承和派生类；第 6 章主要介绍类成员的进一步使用；第 7 章主要介绍多态性和虚函数；第 8 章主要介绍运算符重载；第 9 章主要介绍模板；第 10 章主要介绍 I/O 流；第 11 章主要介绍异常处理；第 12 章主要介绍面向对象软件开发方法。其中，第 2～11 章，每章都包含有上机实训，以便于读者通过理论学习和实践两个环节更好地掌握课程内容，提高编程能力。

本书的读者对象是大学本科计算机相关专业的教师和学生，本书也可以作为从事计算机相关领域工作的科学技术人员的参考书。

本书参考和引用了大量的书籍和文献资料，在此，向被引用文献的作者表示衷心的感谢，向给予本书帮助的所有人士表示衷心的感谢！

由于作者水平有限，不妥之处在所难免，欢迎读者批评指正。

编著者

2007 年 5 月

目　录

第1章

面向对象程序设计概论

本章要点

本章从范型的概念出发，主要介绍传统的结构化程序设计和面向对象程序设计两种程序设计范型的各自特点和它们的区别；回顾了结构化程序设计方法、面向过程的程序的结构和特点等面向过程程序设计的内容，重点介绍了面向对象程序设计的基本概念、原理和方法。其中，面向对象的抽象原则、类、对象、消息、继承、多态等内容是面向对象程序设计的核心概念。正确理解这些概念，了解面向对象程序的构造，以及面向对象程序设计语言的特点，对后续章节的学习是非常重要的。

1.1 程序设计范型的概念

范型（Paradigm），或者称为风范、风格，是程序设计领域内叫法不同、使用频繁的一个概念。其本意是指一种通用的、概念化的、关于客观世界的模型，这个模型提供一种观察和研究客观世界的视图。因此，不同的范型对同样的问题域提出的视图可能是不同的。一个范型规定了观察问题域的方法、所使用的抽象手段、所共享的概念集合。所以，范型是一个可用来对事物进行分类的概念。

程序设计范型就是程序设计过程使用的观察问题域的方法和抽象手段、共享的概念集及程序构成的某种规定。这些规定可以支持应用领域希望的程序设计风格。

程序设计范型在某种程度上和使用的程序设计语言相关。使用一种程序设计范型进行程序设计时，通常选择的编码语言都支持该程序设计范型，也就是说，所选择的程序设计语言具有该程序设计范型的特征。所以，从这个角度说，程序设计范型反映了程序设计语言的抽象特征，是一组程序设计语言特点的集合。程序设计范型体现了该组语言的共同风格。

程序设计概念几乎和现代计算机的概念同时诞生。现在大家都接受软件是计算机的灵魂这一观点，而程序是软件最重要的组成部分。因此，与程序设计相关的程序设计语言、程序设计方法学等领域历来都是计算机科学与技术学科的重要内容。许多专家学者经过长期的工作，在软件开发基础理论和具体开发技术两个方面作出了重要贡献，使得软件开发从无序的、手工操作方式进入了科学化、工程化和规范化的发展阶段。同时，也提出了许多行之有效的软件开发的理论和方法。这些不同的开发方法则规定了与之相应的不同的程序设计方法，在

一定程度上，它们也可以用程序设计范型分类。

一个规范化的软件开发过程一般由项目计划、系统分析、系统设计、编码、测试和维护几个阶段构成。程序设计一般指从系统设计的后半部分到编码这段工作。在软件开发过程中，软件开发方法论规定了开发过程中各阶段所使用的理论、过程、方法与工具，也就是说，它决定了在进行程序设计时人们观察问题域的角度，以及由此得到的关于客观世界的视图。所以，程序设计过程中所使用的观察问题域的方法、使用的抽象手段，共享的概念集和最终程序的构成方式都取决于程序设计范型。

下面是不同程序设计范型的程序段的示例。在一定程度上，从这些程序的外观可以看出程序设计范型对程序构造的影响。

【例 1.1】 面向过程程序的示例。

```
int push(stack *a, int data){
if(a->top<STACK_SIZE){
a->data[a->top++]=data;
return 1;
}else {
error("push(stack *, int):stack full!\n");
return 0;
}
}
```

【例 1.2】 面向对象程序的示例。

```
class Singleton{
public:
static inline T* instance( );
private:
Singleton(void){}
~Singleton(void){}
…
}
```

【例 1.3】 逻辑程序的示例。

```
simp(E, E):-atomic(E),!
simp(E, E):-var(E),!
simp(X-X, 0).
…
```

【例 1.4】 函数程序的示例。

```
(DE   EQUAL (X Y )
  (COND ((EQ X Y) T)
        ((ATOM   X) NIL)
        ((ATOM   Y) NIL)
        (( EQUAL (CAR X)   (CAR Y))
          (( EQUAL (CDR X)   (CDR Y))
```

范型最早用于程序设计语言分类。面向过程（Procedure-Oriented）的程序设计范型和面向对象（Object-Oriented）的程序设计范型是两类发展最成熟、使用最广泛的程序设计范型。例如，像 C 和 Pascal 这类程序设计语言归类于面向过程的程序设计范型，而将 C++、Java 这类程序设计语言归为面向对象的程序设计范型。通常，人们将使用这两类程序设计范型的程序设计方法分别称为面向过程的程序设计方法和面向对象的程序设计方法，因此，在本书中，将不去区别程序设计范型和程序设计方法这两个概念的差别，在不同的地方根据需要等同地使用它们。

1.2　面向过程程序设计方法

一般地，面向过程的程序设计方法中，编码语言都选择使用属于面向过程程序设计范型的程序设计语言，称为面向过程语言。面向过程语言的语句基本上是命令式的。命令构成了程序段，事先规定的语句顺序决定程序段执行的时序。这类语言以顺序、循环和选择机制作为基本的控制结构，过程或函数是程序的基本构成单元。所以，在这种范型中，程序设计的首要任务是设计过程。面向过程的程序设计范型与不同的软件分析和设计技术结合，形成了许多不同的软件开发方法，如 Jackson 方法、LCP 方法等。基于面向过程的程序设计范型和功能分解方法的软件设计技术形成了结构化软件开发方法学的基础。在本节，我们将主要介绍结构化程序设计方法、面向过程的程序结构及相关内容。

1.2.1　结构化程序设计方法

结构化程序设计（Structured Programming）的概念是在 20 世纪 70 年代提出来的，随之也出现了支持结构化程序设计方法的程序设计语言，如 Pascal、C、Ada 等。程序设计语言开始向模块化、形式化方向发展。不言而喻，这些程序设计语言都属于面向过程的程序设计范型。基于这些程序设计语言的支持，结构化程序设计方法逐渐成为 20 世纪七八十年代十分流行的程序设计方法。结构化程序设计方法强调程序结构的规范性，强调程序设计的自顶向下、逐步求精的演化过程。具体地说，结构化程序设计方法有以下几种基本原则。

1．自顶向下、逐步求精

这是 Niklaus Wirth 提出的设计策略。该原则把程序设计过程视为一个在不同层次间的演化过程。这些不同的层次对应了不同的抽象级别。当问题给定时，将所希望的软件结构按自顶向下的方式，对各个层次的过程细节和数据细节逐层细化，直到用程序设计语言的语句可以实现为止。这个过程是一个不断推敲，逐步迭代的过程。例如，开发一个高校学生教材购销系统，最初的抽象可以把整个系统作为一个整体，研究它和周围外部实体之间的数据交换；接下来的下层抽象可以把该系统分为学生教材销售和学生教材采购两个子系统；最后的抽象又可以把学生教材销售子系统分为有效性审查、账目处理、开发票、开领书单等部分，而学生教材采购子系统又可以分为缺书登记、进书通知等部分。当细化到一定程度时，就可以使用程序设计语言的语句实现了。