21世纪高等学校计算机教育实用规划教材

面向对象技术（C++）

年福忠 庞淑侠 朱红蕾 编著

清华大学出版社

21世纪高等学校计算机教育实用规划教材

面向对象技术（C++）

年福忠 庞淑侠 朱红蕾 编著

清华大学出版社
北京

内 容 简 介

本书主要介绍面向对象程序设计(C++)的基本概念、基本理念以及编程思路与技巧等,内容包括类和对象、继承与派生、多态性、模板与异常处理、输入输出流以及 ODBC 与数据库编程等内容;并在最后一章,从课程设计的角度,通过一个精心选择的实例介绍了面向对象的应用程序开发方法和开发过程。本书内容全面,重点突出,从读者的角度出发,由简入深,围绕精心选择的例子,力求深入浅出,举一反三,融会贯通;每章都配有习题,以指导读者深入地进行学习。

本书既可以作为高等学校计算机软件技术课程的教材,也可以作为管理信息系统开发人员的技术参考书。

图书在版编目(CIP)数据

面向对象技术:C++/年福忠等编著. —北京:清华大学出版社,2015
21 世纪高等学校计算机教育实用规划教材
ISBN 978-7-302-39557-7

Ⅰ. ①面… Ⅱ. ①年… Ⅲ. ①C 语言－程序设计－高等学校－教材 Ⅳ. ①TP312

中国版本图书馆 CIP 数据核字(2015)第 046767 号

责任编辑:付弘宇　薛　阳
封面设计:常雪影
责任校对:梁　毅
责任印制:何　芊

出版发行:清华大学出版社
网　　址:http://www.tup.com.cn, http://www.wqbook.com
地　　址:北京清华大学学研大厦 A 座　　邮　　编:100084
社 总 机:010-62770175　　邮　　购:010-62786544
投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn
质 量 反 馈:010-62772015, zhiliang@tup.tsinghua.edu.cn
课 件 下 载:http://www.tup.com.cn,010-62775954

印 装 者:北京国马印刷厂
经　　销:全国新华书店
开　　本:185mm×260mm　　**印　　张**:23　　**字　　数**:576 千字
版　　次:2015 年 5 月第 1 版　　**印　　次**:2015 年 5 月第1次印刷
印　　数:1～2000
定　　价:44.50 元

产品编号:063309-01

出版说明

随着我国高等教育规模的扩大以及产业结构调整的进一步完善，社会对高层次应用型人才的需求将更加迫切。各地高校紧密结合地方经济建设发展需要，科学运用市场调节机制，合理调整和配置教育资源，在改革和改造传统学科专业的基础上，加强工程型和应用型学科专业建设，积极设置主要面向地方支柱产业、高新技术产业、服务业的工程型和应用型学科专业，积极为地方经济建设输送各类应用型人才。各高校加大了使用信息科学等现代科学技术提升、改造传统学科专业的力度，从而实现传统学科专业向工程型和应用型学科专业的发展与转变。在发挥传统学科专业师资力量强、办学经验丰富、教学资源充裕等优势的同时，不断更新教学内容、改革课程体系，使工程型和应用型学科专业教育与经济建设相适应。计算机课程教学在从传统学科向工程型和应用型学科转变中起着至关重要的作用，工程型和应用型学科专业中的计算机课程设置、内容体系和教学手段及方法等也具有不同于传统学科的鲜明特点。

为了配合高校工程型和应用型学科专业的建设和发展，急需出版一批内容新、体系新、方法新、手段新的高水平计算机课程教材。目前，工程型和应用型学科专业计算机课程教材的建设工作仍滞后于教学改革的实践，如现有的计算机教材中有不少内容陈旧(依然用传统专业计算机教材代替工程型和应用型学科专业教材)，重理论、轻实践，不能满足新的教学计划、课程设置的需要；一些课程的教材可供选择的品种太少；一些基础课的教材虽然品种较多，但低水平重复严重；有些教材内容庞杂，书越编越厚；专业课教材、教学辅助教材及教学参考书短缺，等等，都不利于学生能力的提高和素质的培养。为此，在教育部相关教学指导委员会专家的指导和建议下，清华大学出版社组织出版本系列教材，以满足工程型和应用型学科专业计算机课程教学的需要。本系列教材在规划过程中体现了如下一些基本原则和特点。

(1) 面向工程型与应用型学科专业，强调计算机在各专业中的应用。教材内容坚持基本理论适度，反映基本理论和原理的综合应用，强调实践和应用环节。

(2) 反映教学需要，促进教学发展。教材规划以新的工程型和应用型专业目录为依据。教材要适应多样化的教学需要，正确把握教学内容和课程体系的改革方向，在选择教材内容和编写体系时注意体现素质教育、创新能力与实践能力的培养，为学生知识、能力、素质协调发展创造条件。

(3) 实施精品战略，突出重点，保证质量。规划教材建设仍然把重点放在公共基础课和专业基础课的教材建设上；特别注意选择并安排一部分原来基础比较好的优秀教材或讲义修订再版，逐步形成精品教材；提倡并鼓励编写体现工程型和应用型专业教学内容和课程体系改革成果的教材。

(4) 主张一纲多本,合理配套。基础课和专业基础课教材要配套,同一门课程可以有多本具有不同内容特点的教材。处理好教材统一性与多样化,基本教材与辅助教材,教学参考书,文字教材与软件教材的关系,实现教材系列资源配套。

(5) 依靠专家,择优选用。在制订教材规划时要依靠各课程专家在调查研究本课程教材建设现状的基础上提出规划选题。在落实主编人选时,要引入竞争机制,通过申报、评审确定主编。书稿完成后要认真实行审稿程序,确保出书质量。

繁荣教材出版事业,提高教材质量的关键是教师。建立一支高水平的以老带新的教材编写队伍才能保证教材的编写质量和建设力度,希望有志于教材建设的教师能够加入到我们的编写队伍中来。

21 世纪高等学校计算机教育实用规划教材编委会
联系人:魏江江 weijj@tup.tsinghua.edu.cn

前　言

面向对象程序设计兴起于20世纪90年代，它是不同于传统程序设计的一种新的程序设计范型。面向对象程序设计是在结构化程序设计的基础上发展起来的，它吸取了结构化程序设计中最为精华的部分，有人称它是“被结构化了的结构化程序设计”。这种新的程序设计思想更接近于人的思维活动，这使得人们在程序设计与开发中能够更加灵活，也极大地降低了程序的维护成本。因此，面向对象技术从它诞生后就迅速在全世界流行，成为了程序设计的主流技术，且长盛不衰。

近20年来，国内许多高校纷纷将面向对象技术列入教学计划，特别是信息类专业，如计算机科学与技术、通信工程等专业甚至将其作为本科生或专科生的必修课。在欧美等发达国家的工科专业该课程也已十分普及。随着面向对象技术，尤其是以C++为面向对象程序设计语言的开发技术的发展，旧的教材已经越来越不能适应当前课堂教学和自学的需求。为此，我们结合自己多年的教学经验和对面向对象技术(C++)在新形势下的要求，编写了这本教材，以适应时代发展的需要。

本教材是在参考了国内许多高校面向对象技术(C++)课程的教学大纲的基础上编写的，为了适应工科信息类专业本、专科生的教学需要，本书从面向对象程序设计的概述讲起，介绍了面向对象的基本概念，C++的基本概念和编程特点，重点阐述类与对象，继承与派生以及多态性等内容。同时，为了进一步提高学生的实践和动手能力，我们还介绍了模板与异常处理、C++的输入与输出流以及ODBC与数据库编程等，使得学生能够很快熟悉C++的程序设计开发与调试，并能够解决一些实际问题，从而深化学生对面向对象编程的理解，培养学生的学习兴趣。

本书内容概要如下。

第1章是面向对象程序设计概述，其中包括面对对象的基本概念和基本特征及其发展简史等。第2章是C++概述，介绍C++语言的特点、构成、编译连接、C++数据类型、函数、指针、引用以及其他若干C++的重要特性。从第3～5章是本书的重点内容，其中第3章详细介绍类和对象，包括C++类、构造函数和析构函数、自引用this指针、对象数组与对象指针、静态成员与友元、友元类和友元方法等。第4章介绍继承与派生，其中包括继承与派生的概念、派生类的声明、定义基类和派生类、构造函数访问顺序、子类隐藏父类的成员函数、多重继承以及虚基类等。第5章是多态性，首先介绍什么是多态，然后详细介绍函数重载、运算符重载、虚函数与抽象类、类型转换等问题。从第6章开始，着重培养的是学生的实践能力，

第6章是模板与异常处理,其中包括模板的概念、函数模板和模板函数、类模板和模板类、类模板的派生以及异常处理的实现等。第7章是C++的输入输出,在这一章里主要介绍流概念、iostream流类库、预定义的I/O流、输入输出的格式控制、文件流、文件输入和输出流、用户自定义的I/O流,包括重载插入运算符和提取运算符等。第8章介绍ODBC与数据库编程,其内容包括ODBC与数据库概述、MFC ODBC连接数据库、ODBC的构成、MFC ODBC类以及MFC DAO连接数据库。第9章是最后一章,是课程设计实例分析,其中包括开发背景、需求分析、总体设计、详细设计、系统测试和软件使用说明书等完整的课程设计过程。

本书内容新颖,重点突出,层次清晰,语言通俗易懂,内容覆盖面广。现在流行的面向对象C++类的教材大都牵涉的概念比较多、语法比较复杂、内容十分广泛,使不少刚刚开始接触面向对象编程的本科生感到学习难度较大,难以入门。针对国外教材基础起点高、涉及内容广、难度大等特点,以及国内教材C语言知识点编写繁杂、不注重MFC等软件工程应用开发等特点,结合普通本科工科院校学生的实际情况,结合我们多年的教学经验和讲义积累,本教材定位准确、取舍合理、是本、专科生易于学习的教材,并且以通俗易懂的语言讲解许多复杂的概念,减少初学者学习C++的困难。本书各章均提供了丰富的实例和练习,同时也提供相应的多媒体课件(PPT格式)。本书可作为高等院校应用型本科(含部分专科、高职类)各相关专业(如计算机、电子信息、通信、物联网等)的程序设计教材,也适合编程开发人员培训、广大计算机技术爱好者自学使用。

本书的所有实例全部经过上机调试。根据我们的教学体会,本书的教学可以安排为48～60学时。

尽管我们在写作过程中投入了大量的时间和精力,但由于水平有限,错误和不足之处仍在所难免,敬请读者批评指正(任何建议都可以发至邮箱gdnfz@163.com)。我们会在适当时间对本书进行修订和补充,并公布在兰州理工大学精品课程网站。

本书第1、第2、第9章由朱红蕾老师编写,第3、第4、第8章由年福忠老师编写、第5～7章由庞淑侠老师编写。全书由年福忠老师统稿。本书的最终出版得到了许多老师和同学的帮助。兰州理工大学规划教材基金对本书的编写进行了资助。清华大学出版社的广大员工为本书的编写和出版付出了辛勤劳动。另外,研究生刘伟龙同学协助做了许多工作。在此,一并表示感谢。

本书的配套课件等教学资料可以从清华大学出版社网站www.tup.com.cn下载,关于下载与使用的任何问题,请联系邮箱fuhy@tup.tsinghua.edu.cn。

编　者

2014年12月

目　录

第1章 面向对象程序设计概述

面向对象程序设计(Object Oriented Programming,OOP)方法是一种全新的设计和构造软件的方法。这种程序设计模式更符合人类的思维方式,能更直接地反映和描述客观世界。面向对象程序设计可以增加程序的可重用性和可扩充性,并且极大地降低了软件维护的开销。现在,面向对象程序设计方法已经被越来越多的软件开发人员所接受,成为当今主流的程序设计方法。本章首先介绍面向对象技术的基本概念、基本特征和面向对象与面向过程程序设计的区别,然后介绍面向对象程序设计发展简史以及当今主流的面向对象程序设计语言,最后介绍 Visual C++ 6.0 开发环境。

1.1 什么是面向对象程序设计

面向对象程序设计方法以对象为基础,利用特定的软件工具直接完成从对象客体的描述到软件结构之间的转换,这是面向对象设计方法最主要的特点和成就。面向对象程序设计方法的应用解决了传统面向过程开发方法中客观世界描述工具与软件结构不一致的问题,缩短了软件开发周期,解决了从分析设计到软件模块结构之间多次转换映射的繁杂过程,是一种很有发展前途的程序设计方法。

1.1.1 程序设计范型

程序设计范型(Paradigm)是指设计程序的规范、模型和风格,它是一类程序设计语言的基础。一种程序设计范型体现了一类语言的主要特征,这些特征能用以支持应用领域所希望的设计风格。不同的程序设计范型有不同的程序设计技术和方法学。

面向过程程序设计范型是流行很广泛的程序设计范型,这种范型的主要特征是,程序由过程定义和过程调用组成(简单地说,过程就是程序执行某项操作的一段代码,函数是最常用的过程)。从这个意义出发,基于面向过程的程序可以用以下的公式来表述:

程序=过程+调用

基于面向过程程序设计范型的语言称为面向过程性语言,如 Ada、C、Fortran、Pascal 等都是典型的面向过程性语言。除面向过程程序设计范型外,还有许多其他程序设计范型。如函数式程序设计范型也是较为流行的程序设计范型。它的主要特征是,程序被看做"描述输入与输出之间关系"的数学函数(典型语言是 LISP)。此外,还有模块程序设计范型(典型语言是 Modula)、逻辑式程序设计范型(典型的语言是 PROLOG)、进程式程序设计范型、类型系统程序设计范型、事件程序设计范型、数据流程序设计范型等。

面向对象程序设计是一种新的程序设计范型。这种范型的主要特征是:

程序＝对象＋消息

面向对象程序的基本元素是对象。面向对象程序的主要结构特点是：第一，程序一般由类的定义和类的使用两部分组成；第二，程序中的一切操作都是通过向对象发送消息来实现的，对象接收到消息后，启动有关方法完成相应的操作。一个程序中涉及的类，可以由程序设计者自己定义，也可以使用现成的类(包括类库中为用户提供的类和他人已构建好的类)。尽量使用现成的类，是面向对象程序设计范型所倡导的程序设计风格。

需要说明的是，某一种程序设计语言不一定与一种程序设计范型相对应。实际上具有两种或多种范型特征的程序设计语言，称为混合型语言。例如 C++就不是纯粹的面向对象程序设计范型的语言，而是具有面向过程程序设计范型和面向对象程序设计范型的混合型程序设计语言。

1.1.2 面向对象程序设计的基本概念

面向对象程序设计方法为程序员提供了表示问题空间中各种事物元素的工具，并且这种表示方法是通用的。问题空间中的事物和它们在解空间中的表示被称为“对象”。这些对象通过外部接口访问其他对象。同时，可以向程序中添加新的对象类型并根据问题空间中的术语对程序进行调整。这样，阅读描述问题解决方案的代码就是在阅读表达该问题的文字。较之面向过程的设计方法，这是一种更灵活、更强大的语言抽象。因此，面向对象程序设计允许程序员用问题空间中的术语来描述问题。对于面向对象程序设计而言，每个对象就像是一个个小的自治单元，有状态和可执行的运算，如同现实世界中的对象一样，有特性和行为。可见，对象是组成一个系统的基本逻辑单元，是一个有组织形式的含有信息的实体。

在现实世界中，“类”是对一组具有相同属性和行为的对象的抽象。而在面向对象程序设计中，“类”就是具有相同数据和相同操作(函数)的一组对象的集合，也就是说，类是对具有相同数据结构和相同操作的一组对象的描述。因此，类是创建对象的样板。面向对象程序设计总是先声明类，再由类生成其对象。这种方式可以避免重复编码。因为类只需编码一次，就可以创建本类的所有对象。

对象由属性和行为两部分组成。对象只有在具有属性和行为的情况下才有意义。属性是用来描述对象静态特征的数据项，行为是用来描述对象动态特征的操作。对象是包含客观事物特征的抽象实体，是属性和行为的封装体。在面向对象程序设计领域，可以用“对象＝数据＋作用于这些数据上的操作”这一公式来表达。

类是具有相同属性和行为的一组对象的集合，它为属于该类的全部对象提供了统一的抽象描述，其内部包括属性和行为两个主要部分。类是对象集合的再抽象。

类与对象的关系如同一个模具与用这个模具铸造出来的铸件之间的关系。类给出了属于该类的全部对象的抽象定义，而对象则是符合这种定义的一个实体。因此，类是创建对象的“模板”，按照这个模板所建立的一个个类的实际例子，就是具体的对象。所以，对象又被称为类的实例。

在面向对象程序设计中，类的确定与划分非常重要，是软件开发中关键的一步，划分的结果直接影响到软件系统的质量。如果划分得当，既有利于程序进行扩充，又可以提高代码的可重用性。因此，在解决实际问题时，需要正确地进行分“类”。理解一个类究竟表示哪一

组对象，如何把实际问题中的实物汇聚成“类”，而不是一组数据，这是面向对象程序设计中的一个难点。

1.1.3 面向对象程序设计的基本特征

面向对象程序设计方法模拟人类惯常的认识世界的方式，代表了计算机程序设计的新颖的思维方法。这种方法的提出是对软件开发方法的一场革命，是解决软件开发面临困难的最有希望、最有前途的方法之一。本节介绍面向对象程序设计的4个基本特征，即抽象、封装、继承和多态。

1. 抽象

抽象是人类认识世界的最基本方法之一。抽象就是将有关事物的共性归纳、集中的过程。抽象是对复杂世界的简单表示，例如，把所有具有大学学籍的人归为一类，称为“大学生”，这就是抽象。因此，抽象需要忽略事物中与当前目标无关的非本质特征，而强调与当前目标有关的本质特征，从而找出事物的共性，并把具有共性的事物划为一类，得到一个抽象的概念。抽象实现了客观世界向计算机世界的转化。将客观事物抽象成对象及类是比较难的过程，但却是面向对象程序设计非常关键的一步。

2. 封装

在现实世界中，封装就是把某个事物包围起来，使外界不知道该事物的具体内容。在面向对象程序设计中，封装就是把对象的属性和行为结合成一个独立的单位，并尽可能隐藏对象的内部细节。对象好像是一个黑盒子，表示对象属性的数据和实现各种操作的代码都被封装在黑盒子里。这些数据和代码从外部看不到，也不能从外部直接访问和修改。

数据封装是面向对象程序设计的一个基本特征。封装有两个含义：一是把对象的全部属性和行为结合在一起，形成一个不可分割的独立单位。对象的属性值(除了公有的属性值)只能由这个对象的行为来读取和修改；二是尽可能隐藏对象的内部细节，对外形成一道屏障，与外部的联系只能通过外部接口实现。

数据封装无论是对于使用者或实现者都是十分有利的。封装机制将对象的使用者与实现者分开。从使用者的角度来看，不必知道对象行为实现的细节，只需了解类定义的接口部分，即可操作对象，而不必关心对象的内部实现细节。这样，使用者在开发过程中就可以集中精力解决应用中出现的问题，从而使问题处理得到简化。封装的结果实际上隐藏了复杂性，并提供了代码重用性，从而降低了软件开发的难度。从实现者的角度来看，数据封装有利于编码、测试及修改。封装的结果使对象以外的部分不能随意存取对象的内部属性，从而有效地避免了外部错误对它的影响，大大减小了查错和排错的难度。另一方面，当对对象内部进行修改时，由于它只通过少量的外部接口对外提供服务，因此同样减小了内部修改对外部的影响，从而极大地提高了程序的可靠性和稳定性。

3. 继承

在客观世界中，存在着一般和特殊的关系，特殊具有一般的属性，同时又具有自己的新特性。如果只考虑事物的一般性，而不考虑事物的特殊性，就不能反映出客观世界中事物之间的层次关系，不能完整地、正确地对客观世界进行抽象描述。例如，动物、哺乳动物、灵长类哺乳动物之间可以构成一个由一般到特殊的层级关系。因此，运用抽象的原则，提取事物的共性，就可以得到适合一组事物的一个类。如果在这个类的基础上，再考虑抽象过程中被

舍弃的一部分对象的特征,则可以形成一个新的类。这个类具有前一个类的全部特征,同时其自身又具备更丰富的特性,成为前一个类的子集,并与前一个类形成一种层次结构,这就是继承。

继承是一种连接类与类的层次模型。继承意味着"自动地拥有",即特殊类中不必重新定义已在一般类中定义过的属性和行为,而是自动地拥有其一般类的属性与行为。继承允许和鼓励类的重用,提供了一种明确表述共性的方法。一个特殊类既有自己新定义的属性和行为,又有继承下来的属性和行为。尽管继承下来的属性和行为是隐式的,但无论是在概念上还是在实际效果上,都是这个类的属性和行为。当这个特殊类又被它更下层的特殊类继承时,它继承来的和自己定义的属性和行为又将被下一层的特殊类继承下去。

由此可见,继承是对客观世界事物传承关系的一种直观反映。通过类的继承,能够实现对某一特定类型问题的深入抽象描述。这一特性完全符合人类认识问题的客观规律。

4. 多态

客观世界的多态体现在不同的对象收到相同的消息时可以产生多种不同的行为方式。具体来说,多态是指类中同一函数名对应多个具有相似功能的不同函数,并可以使用相同的调用方式来调用这些具有不同功能的同名函数。用一个计算机操作的例子可以很好地说明多态的概念。例如,通过鼠标操作发送"双击"消息给计算机上的不同对象,则这些对象的响应可能各不相同。"文件夹"对象收到"双击"消息后,会打开该文件夹;而"计算器"对象收到"双击"消息后,则会打开计算器程序。

C++语言支持两种多态性,即编译时的多态性和运行时的多态性。编译时的多态性通过函数重载(包括运算符重载)来实现,而运行时的多态性则通过虚函数来实现。

多态是面向对象程序设计的又一个基本特征。多态性减轻了程序设计人员的记忆负担,使程序的设计、修改更加灵活,程序的使用者只需要记住有限的几个接口就可以完成各种所需要的操作。同时,在程序中可以用简单的操作完成同一类体系中不同对象的操作。因此,多态特性增强了软件的灵活性和可重用性,为软件的开发和维护提供了极大的便利。

1.2 为什么要使用面向对象程序设计

随着现代科技的飞速发展,人们越来越多地依赖于计算机来满足各个领域的需求。程序设计面临的任务也越来越艰巨和复杂。人们对软件功能的要求越来越多,对软件性能的要求也越来越高。在这种情况下,面向对象程序设计具有多方面的吸引力。对管理人员,它实现了更快和更廉价的开发与维护过程。对分析与设计人员,建模处理变得更加简单,能生成清晰、易于维护的设计方案。对程序员,对象模型显得如此浅显和易于使用。因此,面向对象程序设计已经成为软件开发领域的主流设计方法。

1.2.1 传统程序设计方法的局限性

传统的程序设计是面向过程的结构化程序设计。面对日益快速增长的大规模、高性能、高复杂度软件设计的需求,传统的程序设计方法已远远不能满足需要。传统程序设计方法的局限性至少体现在以下几个方面。

首先,在面向过程的程序设计中,问题被看作一系列需要完成的任务。各种函数(在此