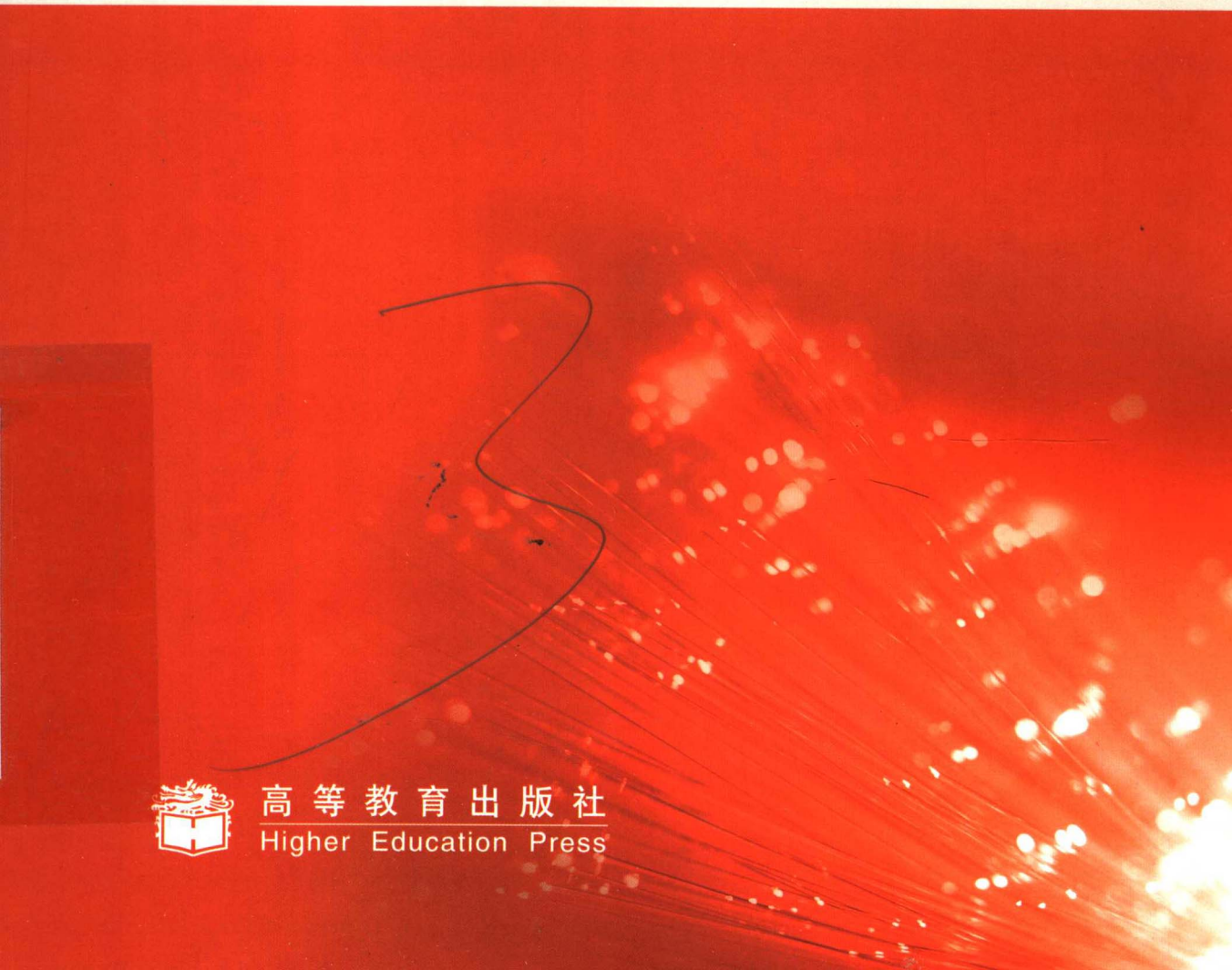
普通高等教育“十一五”国家级规划教材

高等学校计算机科学与技术系列教材

面向对象方法

石峰　宋红



高等教育出版社
Higher Education Press

普通高等教育“十一五”国家级规划教材

高等学校计算机科学与技术系列教材

面向对象方法

石峰　宋红

高等教育出版社

内容提要

本书根据新的计算机专业规范编写而成，是普通高等教育“十一五”国家级规划教材。本书内容丰富，覆盖面向对象方法学的基本原理、面向对象分析与设计方法、面向对象的程序设计方法以及面向对象方法在多个软件领域的最新应用等多项内容。本书从C++如何提供相应的语法以支持面向对象的角度，讲解C++语言的主要语法和程序设计方法，使得学生能够更加深入地理解面向对象基本原理在实际工作中的具体应用和体现，帮助读者建立面向对象的思维方式。

本书适合作为高等学校计算机专业、软件工程专业本科及硕士研究生相关课程的教材，也可作为从事IT方面工作的工程技术人员的参考读物。

图书在版编目（CIP）数据

面向对象方法／石峰，宋红．—北京：高等教育出版社，
2008.4

ISBN 978－7－04－023223－3

Ⅰ．面… Ⅱ．①石…②宋… 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(2008)第015177号

策划编辑	刘 艳	**责任编辑**	焦建虹	**封面设计**	于文燕	**责任绘图**	尹 莉
版式设计	王艳红	**责任校对**	俞声佳	**责任印制**	尤 静		

出版发行	高等教育出版社	**购书热线**	010－58581118
社 址	北京市西城区德外大街4号	**免费咨询**	800－810－0598
邮政编码	100011	**网 址**	http://www.hep.edu.cn
总 机	010－58581000		http://www.hep.com.cn
		网上订购	http://www.landraco.com
经 销	蓝色畅想图书发行有限公司		http://www.landraco.com.cn
印 刷	北京东光印刷厂	**畅想教育**	http://www.widedu.com
开 本	787×1092 1/16	**版 次**	2008年4月第1版
印 张	14.5	**印 次**	2008年4月第1次印刷
字 数	320 000	**定 价**	18.50元

物料号 23223－00

高等学校计算机科学与技术系列教材编审委员会

序

计算机和通信技术的迅猛发展，不仅形成了融合度最高、潜力最大、增长最快的信息产业，而且成为推动全球经济快速增长和全面变革的关键因素。进入21世纪，我国的信息产业虽然已取得了长足的发展，但与发达国家相比，还有不小的差距。国家信息化的发展和信息产业国际竞争能力的提高，迫切需要高素质、创新型的计算机专业人才。

高素质计算机专业人才的培养离不开高质量的计算机教育。我们的专业虽然机会多，处于非常有利的条件，但是我们同样面临着一件事，就是从规模发展向质量提高的转变。怎么提高质量？专业素质的教育和应用素质的训练非常重要。尤其是我国高等教育进入大众化发展阶段，社会对计算机专业人才呈现出了多样化的需求。而与此同时，计算机学科的发展已极大地突破了原有的学科体系框架，形成了在“计算机科学与技术”之下向多个专业方向发展的新格局。在这种背景下，教育部高等学校计算机科学与技术教学指导委员会编制了《高等学校计算机科学与技术专业发展战略研究报告暨专业规范(试行)》(以下简称“专业规范”)。专业规范按照“培养规格分类”的指导思想，提出了三种类型、四个方向，即科学型(计算机科学方向)，工程型(计算机工程方向、软件工程方向)，应用型(信息技术方向)的计算机专业发展建议，体现了社会对不同人才类型的需求，对于指导我国计算机教学改革与建设，规范计算机教学工作，促进计算机教学质量的提高都具有重要的意义。

高水平的教材是一流教育质量的重要保证。为了配合专业规范的试行，便于广大高校教师按照新的专业规范组织实施教学，高等教育出版社在大力支持专业规范研究与起草工作的同时，还邀请规范起草小组的有关专家成立“高等学校计算机科学与技术系列教材编审委员会”，组织规划了结合计算机专业规范、面向全国高等学校计算机专业本科生的“高等学校计算机科学与技术系列教材”。令人高兴的是，一批有创新、改革精神，且有丰富教学经验的高等学校教师投身到新体系计算机专业教材的编写中来，他们用自己创造性的思维、辛勤的汗水诠释专业规范的思想，把新的课程体系和教学内容生动地传达给师生，并进行着有意义的教学实践。

“高等学校计算机科学与技术系列教材”以专业规范和CC2001－CC2005有关教程为依据，以强化基础、突出实践、注重创新为原则，体现了学科课程体系和教学内容改革的新成果。此外，这一系列教材还配有丰富的教学辅助资源，并与现代教育技术手段相结合，充分发挥网络平台的作用，使教材更有利于广大教师和学生使用。目前，这一系列教材有不少选题已列入普通高等教育“十一五”国家级规划教材，希望这些教材的出版能够对新形势下我国高等学校计

算机专业课程改革与建设起到积极的推动作用，使我国高校的计算机专业教学质量再上一个台阶。

中国科学院院士

2006—2010年教育部高等学校计算机科学与技术教学指导委员会主任

二〇〇七年十一月

前　言

每当回忆起学习传统软件工程和程序设计的艰辛和曲折，人们总会感到不寒而栗。当听到面向对象程序设计可以按照人们的认知规律和习惯去进行软件开发时，人们不觉为之一振，然而真的学习面向对象语言和软件工程时却会发现，它远不像人们想像得那么容易。

为什么会这样？尽管人们的绝大部分认知过程是与人类认知规律相符合的，然而由于人们对此运用得过于娴熟，几乎变成了条件反射，以至于人类的认知规律有哪些，具体认知时使用了哪些思维手段对人们而言已经熟视无睹，所以突然让人们改变多年来按照逻辑思维的方式思考和认知问题，转而按照自己与生俱来的本能进行软件开发，难免有些无从下手，对于那些从未总结或分析过人类认知过程和规律的学习者更是如此。所以，在学习面向对象语言和面向对象软件工程这种以人类认知规律为基础的软件技术和思维方式之前，十分有必要先了解一下人类的认知过程和规律，细细体会这种认知规律性在具体软件开发过程中的应用，才能起到事半功倍的效果。

面向对象技术与方法这门课程的宗旨是，通过对人类认知规律的总结以及在具体软件开发过程中的应用，使学习者能够做到“知其然且知其所以然”，体会面向对象的妙用。

本书根据教育部计算机科学与技术教学指导委员会编制的计算机专业规范相应课程要求，通过对人类认知规律的总结和简介，逐步引入面向对象的思想和基本概念，在简单介绍对象结构与功能的分析和描述方法的基础上，介绍面向对象模型的C++语言实现方式，以便学习者可以将学习面向对象语言时积累的经验与面向对象方法学的基本原理相对比，从而更深入地理解面向对象的本质。面向对象分析与设计是面向对象技术的精华，然而在本书中作者更希望它能够作为学习者了解、体会和掌握面向对象思想的工具。

全书共分9章。第1～4章由石峰编写，主要介绍人类认知规律及其与面向对象的关系。后5章由宋红编写：第5章介绍面向对象的软件开发过程，并对每个开发阶段的工作做了介绍；第6～8章通过两个具体的系统实例——自动取款机系统和图书馆信息管理系统，介绍了面向对象的软件分析和设计过程，包括软件需求分析的用例建模，面向对象的系统分析以及面向对象的设计；第9章给出一个比较完整的面向对象软件开发过程实例——R公司网上家电销售系统。

对于本书来说，学习的重点应该放在对面向对象思想的掌握上，而不是具体的技术细节。本书例题中的视图采用UML描述，所给出的程序是在Windows XP环境中使用Visual Studio .Net 2005所带的Visual C++编写的。

本书是在编者多年教学的基础上编写的，适合作为计算机及其相关专业高年级本科或硕士研究生教学用书，也可用做其他理工科专业的计算机教学参考书。对于有志于深入学习面向对

象技术与方法的广大计算机爱好者，本书亦有一定的参考价值。书中所用到的部分素材来源于教学过程中的积累和其他中外文教材、资料，由于无法在此一一列举，现谨对这些教材和资料的作者表示衷心的感谢。在本书编写过程中，东南大学徐宝文教授给予了大量的关心和指导，在此表示最衷心的感谢！

由于作者水平有限，加之时间仓促，本书虽然已经几次修改但疏漏之处在所难免，欢迎各位专家和广大读者不吝赐教。

编　者

2007 年 10 月于北京理工大学

目　录

第1章 引 论

软件的核心是程序，软件开发的核心工作是程序设计，而程序设计的实质是把人对客观事物的认知结果用程序设计语言描述出来，因此认知方法和描述成为软件开发两个最重要的组成部分，可以说软件技术的发展基本上都是围绕着如何认知和如何描述这两个话题进行的，包括怎样认知客观事物、怎样去抽象客观事物、如何提高描述的效率、如何提高描述的可读性和易理解性等方方面面的具体内容，软件技术的每一次飞跃莫不与此有关。

本章的主要内容是对软件技术发展历史过程的简单回顾，期望读者通过软件技术发展不同阶段之间的联系理解面向对象方法出现的原因，通过了解传统软件工程方法的缺点和不足，更好地理解和掌握现代软件工程的理论基础——面向对象方法。

1.1 软件开发过程

软件产品是对客观事物认知结果的描述，是一种思维结果，同时包含了物和人的因素，导致软件的生产过程和软件产品本身具有明显的不确定性。

因此，尽管软件生产的工程化是软件规模化生产的必由之路，但软件产品的特殊性决定了它不可能像其他有形产品那样按照物理及化学原理进行工程化生产。

软件生产的工程化经历了软件工程出现前的“软件艺术”阶段、传统软件工程出现后强调标准化和规范化的“软件技术”阶段以及现代软件工程的“面向对象方法”和基于构件的程序开发阶段。从一开始强调充分挖掘程序设计者个人思维潜能，到努力克服程序设计者个人因素对开发工作的影响，进而保证软件生产过程和软件产品的标准化，发展到目前的按照人类认知规律进行软件开发，从而将开发者对客观事物的认知与开发过程有机地统一在一起。

随着软件工程的研究和发展，软件开发已经从最初的单一程序编写演化为包括可行性论证、需求分析、建模、编码、测试、维护等在内的软件过程。

1.1.1 软件工程史前期

从1946年世界上第一台电子计算机诞生到1957年高级语言FORTRAN出现的10年间，计算机的应用基本属于专业人士的专利，这个时期的软件几乎都是规模较小的单一程序，解决的问题也比较简单。人们使用令人感到晦涩难懂的机器码、助记符语言、汇编语言编写各种程序，由于存储器资源和运算速度所限，即使同样的功能，为了节省在今天看来微乎其微的存储空间和

几个机器周期的操作时间，人们也会针对应用场合的特殊性重新实现，人们陶醉于在程序中使用各种小技巧，追求的是个性化的程序设计风格，各类程序五花八门，因此这个阶段的程序设计往往被称为“程序设计艺术”，像艺术一样，个性发展，百花齐放，推陈出新，没有统一风格和规范，程序员的工作基本上是“单干”，缺乏合作。

后期为了便于提高程序可读性和重用已有程序，出现了模块、子程序、程序包等概念，尤其是FORTRAN出现后又有了函数和函数库的概念，程序的编写也出现了类似手工作坊式的初级合作。FORTRAN还使得越来越多的非计算机专业人员涉足程序设计，于是就出现了一个很现实的问题——如何教会非专业人员编写程序。这不是一个简单的问题，就像人们学会了单词、学会了造句、学会了语法和修辞，但未必就能写出一篇文章一样，掌握了FORTRAN的语法未必能够编写出程序，于是就出现了所谓的“程序设计方法学”，出现了所谓的面向过程程序设计方法。

这段时期的程序基本都集中在科学计算领域，程序的核心基本都是已经由专业领域科学家或数学家推导出的各种数学公式，因此程序设计过程主要体现在各个已有公式的具体实现上。

1.1.2 传统软件工程期

20世纪60年代中期，随着计算机技术的发展，程序从最原始的单纯代码演化为由程序和设计说明组成的程序系统，进而摇身一变成为“软件”(程序＋文档＋数据)。以软件开发为谋生手段的从业人员越来越多，于是出现了很多软件公司，程序规模和应用领域的扩大导致软件开发的分工合作，以往由专业领域科学家和数学家从事的“公式推导”也逐步由领域专家和软件开发人员共同承担，软件测试和维护工作也逐步形成规模，成为软件开发过程的独立阶段，软件开发逐步演化为一种工程技术。

相对软件功能和性能，人们越来越强调软件的可读性、易维护性、易操作性，人们开始从工程的角度研究软件生产过程，将软件的生产划分为多个阶段，对各阶段的特点、任务、作用以及技术规范等进行了定义，进而于1968年提出了软件工程的概念，期待以工程化的管理手段和开发技术促进软件的发展，提高软件的质量和生产效率。程序设计过程也演变成软件工程。

对于一个软件而言，都会经历从问题提出、开发、使用、维护、修订，直至最终终止使用而被另一个软件取代的过程，就像是一个生命体从孕育、出生、成长到最后消亡，软件的这个状态变化的过程称为生命周期或生存期(Life Cycle)。

显然，软件生命周期的演化具有阶段性，依据一定的原则，可以把软件生命周期划分为若干不同阶段，相邻的阶段既相互区别又相互联系，每个阶段都以其前一阶段的工作成果作为本阶段工作的基础。

依据不同的原则可以得到软件生命周期的不同划分，软件工程国家标准《计算机软件开发规范》(GB 8566—88)中将软件生命周期划分为8个阶段：可行性研究与计划、需求分析、概要设计、详细设计、实现(包括单元测试)、组装测试(集成测试)、确认测试、使用和维护。这里仅对概要设计、详细设计、实现3个阶段加以简介。

(1) **概要设计**：概要设计的输出为概要设计说明书，该阶段的主要研究焦点是软件系统的总

体结构、各模块功能及其相互关系。

概要设计说明书描述了软件系统的基本处理流程、组织结构、功能分配、模块划分、接口设计、运行设计、数据结构设计和出错处理设计等方面的规定，为详细设计奠定基础。其中对软件层次结构的设计，常见方法有 SD(Structured Design，结构化设计)、SC(Structure Chart，程序结构图)、Jackson 方法等。

(2) **详细设计**：详细设计的输入是概要设计的输出，输出是详细设计说明书。该阶段的主要任务是对概要设计中每个模块的功能进行分析，建立每一个待实现功能的抽象数学模型，将实际问题转化为数学问题，然后选择或制定解决相应数学问题的算法，并将该算法以适当的方式(如流程图)描述出来，通过具体的描述形式直观、明确地反映程序设计思想，以待进入编码阶段。

详细设计说明书描述了软件系统各层次中每一个程序(模块、函数等)的设计细节，包括功能描述、模块性能、实现算法及其逻辑流程、接口定义、存储分配、测试计划、待解决问题等各方面的信息。

(3) **实现**：常称为编码，该阶段的任务是在所选用的软硬件环境中，根据详细设计说明书中每个程序的规格说明，将所描述算法的逻辑流程编码为具体程序，并在编码过程中对每一个程序模块进行必要的测试。

Pascal、C 语言等是传统软件工程时期编码常采用的程序设计语言，用它编写的多为结构化的程序。

传统软件工程(如上述 3 个阶段)的核心思想是规范程序设计者的行为，尽量去除程序设计者个人因素对程序设计结果的影响，希望借此得到一个所有设计者都能读懂的、十分规范的程序设计。因此，传统软件工程对各个阶段进行了规范化定义，将客观事物的行为抽象为数学模型和算法，编码就是描述算法或数学公式，最终程序将复杂的客观事物抽象成 3 种简单控制结构的嵌套应用，这几乎是一个十分客观的抽象，因为它的结构太简单和规范了。

1.1.3 现代软件工程

在软件工程出现之前对于程序设计没有具体的规范，完全由设计者根据个人的喜好随意发挥，因此程序的可读性很差，不利于合作开发大型程序，因此出现了结构化程序设计，为编程者提供了程序设计的统一风格和规范，而软件工程的出现则进一步规范了软件生产过程，这一切都是为了得到客观的抛去了程序设计者个人因素影响的程序设计结果，最终目的就是方便所有合作者理解。

一味强调去除人的因素的影响所造成的结果就是，程序设计越接近完成，程序设计描述的内容越抽象，与人的距离越远，直至程序编写完成后几乎完全与客观事物脱离了直接的关系，当程序规模较大时，尽管微观上的结构十分规范，但整体上已经很难读懂了。

考虑上述情况，现代软件工程的发展逐步将人的因素引入软件开发过程中，但这种人的因素不是个人因素，而是将人类认知的普遍规律引入软件开发中，认为软件的开发过程就是开发者对软件所对应的客观事物的认知过程，软件就是对客观事物的描述，程序就是使用程序设计语言对

客观事物的描述或客观事物在计算机中的模型实现。

显然，如果在软件开发过程中按照人类认知的一般规律去认知事物，认知结果的描述（软件或程序）也符合人类认知规律，那么认知结果就会很容易为大多数人所理解。因此诞生了源于人类认知一般规律、考虑软件开发特点的面向对象方法学，在其指导下出现了多种面向对象程序设计方法。与传统软件工程中概要设计、详细设计和实现 3 个阶段对应的典型的面向对象程序设计过程是面向对象分析（OOA）、面向对象设计（OOD）和面向对象程序设计（OOI）。

实际上现代软件工程逐步被融于具体的计算机辅助软件工程工具中，此类工具提供对实施软件开发和维护中的各阶段、方法、技术的全面支撑，包括计划、文档、建模、编码、测试等具体环境和辅助工具。例如，RUP（Rational Unified Process）、OOSP（Object-Oriented Software Process）、DSDM（Dynamic System Development Method）等。

1.2 程序的组织结构

程序是人与计算机进行交互的手段和工具，是由多条语句按照特定的语义和一定的风格汇聚在一起形成的一段“话”或一篇“文章”。

一部长篇小说总是被分解成多个章节，每一节分解成多个段，每个段又分解成多个层次，每个层次分解为多个句子，然后再用不同词汇进行书写。小说章节的划分既可以按照事件发展的顺序进行，也可能按照事件的不同组成部分进行，抑或按照引发事件的原因和事件产生的结果进行，甚至按照事件发展所涉及的地理位置进行。

这种出现在人们视野中的错落有致的段落和章节划分就是小说的物理结构，这样的结构便于人们对长篇小说内容的理解，而这种结构的划分是根据不同部分（章节、段落）之间的关系进行的（时间顺序、组成部分、原因结果、地理位置等）。

与小说类似，书写程序会因为所使用程序设计语言的特点及编程者的经验而有不同的风格，程序内部通常会有许多具有特定功能的模块，这些模块被人们用某种方式组织在一起，形成程序的物理层次结构。不同模块的划分及形成物理层次结构的策略直接影响程序的可读性、易理解性和可维护性，所产生的程序风格有所不同，形成了不同的软件设计模式。

在软件技术发展历史上，依据不同的模块划分规则和组织策略，曾出现过 3 种比较典型的程序结构和设计风格，即面向过程程序、结构化程序、面向对象程序。

1.2.1 面向过程程序的结构

面向过程的程序设计是追踪事物发展的过程展开的，对于一个复杂的系统，往往首先对系统的任务进行划分，将大规模的任务分解为相对独立、规模较小的子任务的集合，子任务继续分解，直到每个任务成为规模足够小的程序模块，如图 1-1(a)所示。

面向过程程序的基本模块一般通过函数实现，每个函数内部代码的编写是按照算法解题过程进行的，即代码的物理位置受控于解题过程——代码执行顺序，由于不同问题具有不同的算

法，具体过程变化多端，语句的排列很难有统一规则和风格，而且此类程序中到处充斥着 goto 语句，将各条语句紧密地联系在一起形成一个很难进一步分解的整体，这时谈论其内部结构已经意义不大，此类程序如图 1－1(b)所示。

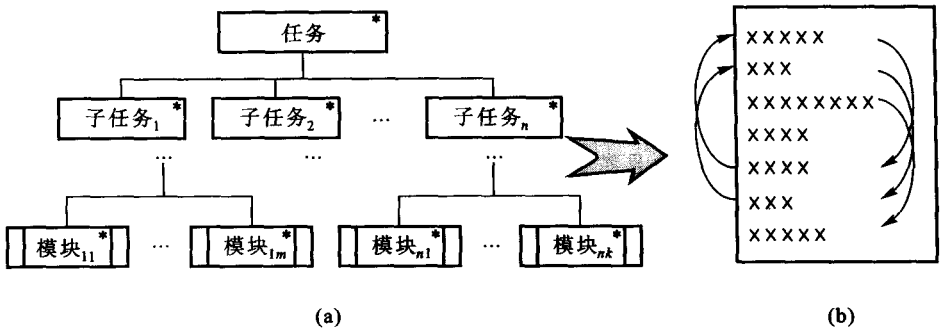


图 1－1　面向过程程序的结构示意图

1.2.2　结构化程序的结构

结构化程序的出现规范了程序风格，在函数内部形成规范化的层次结构，即函数由顺序语句、循环语句、分支语句 3 类语句根据其相应的控制结构，形成直观、嵌套的层次化物理结构，其风格如图 1－2 所示。

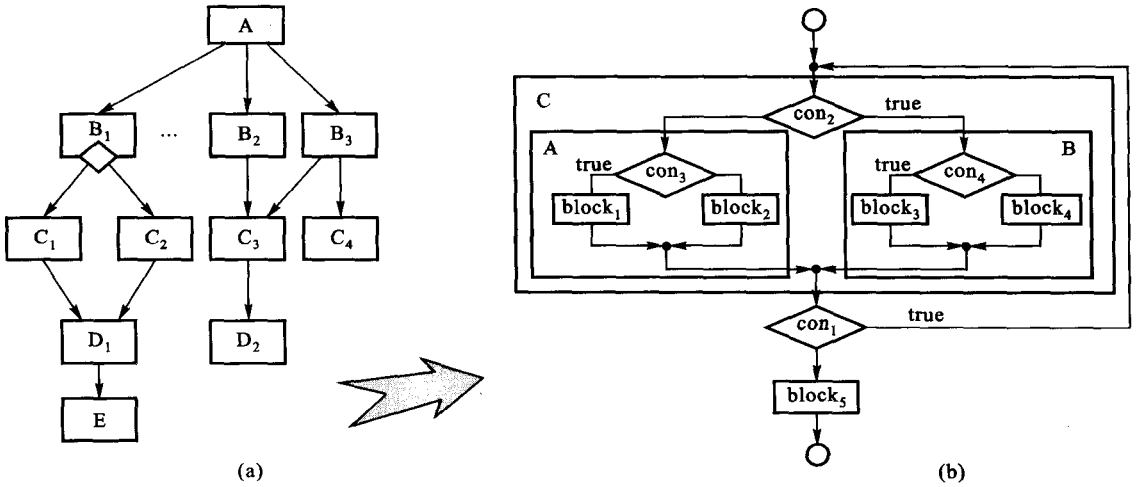


图 1－2　结构化程序的结构示意图

结构化的精髓在于使用语句运行顺序的控制结构，将程序划分为不同组成模块，进而根据控制结构的规律形成程序的物理结构，当程序规模不是很大、模块功能不是很复杂时，结构化程序具有十分清晰的结构，可读性很强。

1.2.3 面向对象程序的结构

结构化程序的风格统一，结构简单，可读性好，但是当程序规模较大或模块复杂时，由于嵌套层次过多，导致程序结构（由 3 种简单结构构成）十分混乱，尽管规范，但可读性却降低了。出现这种情况的原因之一是用于描述结构的手段太单一，将复杂的客观事物均抽象成只包含 3 种简单结构的程序，而不是根据不同层次的特点采用不同的手段，自然给人以结构层次过多、过乱的感觉。

结构化程序中语句的组织也是按照算法进行的，而算法是客观事物行为多次抽象的结果，由于抽象程度太高，而且都是客观事物的行为，不含行为主体的信息，所以很难与客观事物之间建立起对应关系，因此程序的可读性随程序规模变大而明显下降。

面向对象程序克服了上述程序结构的缺点，其风格如图 1－3 所示。其中的模块是与客观事物组成部分（实体）相对应的数据及其操作（函数）的封装体（称为类），是其所封装行为（函数）的主体，主体之间不同种类的相互关系与客观事物组成部分之间的关系相对应。由于客观事物往往是人们能够观察到或直接感受到的，因此尽管程序的规模可能很大，但只要相应的客观事物是容易被认知的，程序的可读性也就必然好。

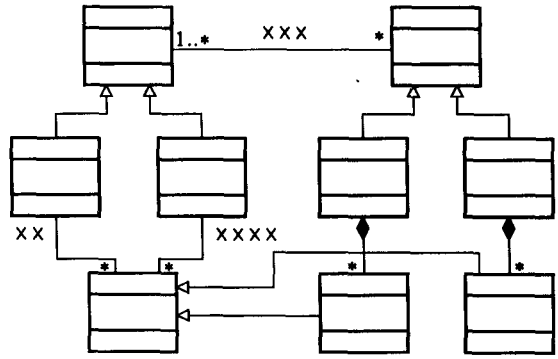


图 1－3　面向对象的程序结构示意图

图 1－3 中的程序结构在不同层次上的描述手段不同：首先，在与客观事物组成部分相对应的实体之间采用了多种关系（泛化关系、聚合、关联等）；其次，在实体内部采用了任务分解的方式，将实体的行为分解为多个函数；最后，在函数内部通常采用结构化程序的结构。显然，不同的描述手段适合不同的层次，多种结构描述手段汇聚在一处比起结构化程序中单调的基本结构更加丰富多彩，使得程序的层次结构变得更为清晰。

1.3 设计模式

1.3.1 面向过程程序设计

最早出现的程序设计方法学和软件工程可以追溯到 20 世纪 60 年代末。正像人们掌握了字、词不一定会写文章一样，当第一个高级语言 FORTRAN 出现后，对于非计算机专业人员，不但要学习具体的 FORTRAN 语法，而且要学会如何使用这些语法去编写程序，这就是程序设计方法学，后来大规模软件的开发改变了程序设计初期手工作坊式的独家单干，合作开发导致了软