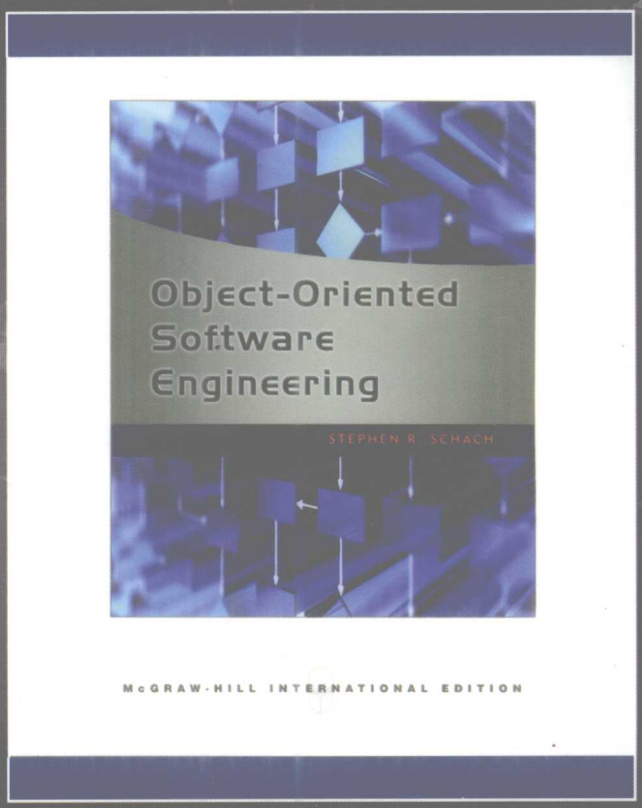
HZ Books 华章教育
Mc Graw Hill
计 算 机 科 学 丛 书

面向对象软件工程

(美) Stephen R. Schach 范德比尔特大学 著
黄林鹏 徐小辉 伍建焜 上海交通大学 译



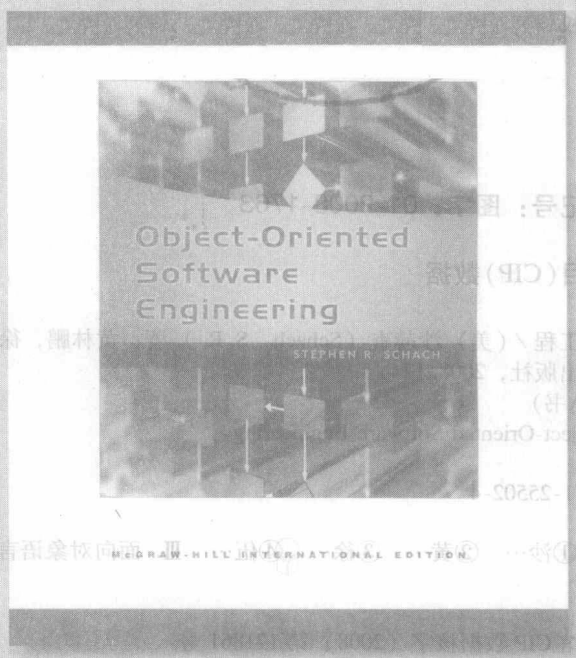
Object-Oriented Software Engineering

机械工业出版社
China Machine Press

计 算 机 科 学 丛 书

面向对象软件工程

(美) Stephen R. Schach 范德比尔特大学 著 黄林鹏 徐小辉 伍建焜 上海交通大学 译



Object-Oriented Software Engineering

机械工业出版社
China Machine Press

本书从面向对象范型出发对软件工程进行重新演绎，全面、系统、清晰地介绍了面向对象软件工程的基本概念、原理、方法和工具，通过实例说明面向对象软件开发的整个过程。

本书分为两个部分：第一部分介绍面向对象软件工程的基本理论；第二部分以工作流的形式介绍软件生命周期。

本书可以作为计算机相关专业高年级本科生和研究生的教材，也可以作为软件工程领域专业人士的参考书。

本书版权登记号：图字：01-2008-1763

图书在版编目(CIP)数据

面向对象软件工程／(美) 沙赫查 (Schach, S. R.) 著；黄林鹏，徐小辉，伍建焜译. —北京：机械工业出版社，2008.12

(计算机科学丛书)

书名原文：Object-Oriented Software Engineering

ISBN 978-7-111-25502-4

Ⅰ. 面… Ⅱ. ①沙… ②黄… ③徐… ④伍… Ⅲ. 面向对象语言－程序设计 Ⅳ. TP312

中国版本图书馆 CIP 数据核字 (2008) 第 171861 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：王春华

北京诚信伟业印刷有限公司印刷

2009 年 2 月第 1 版第 1 次印刷

184mm×260mm · 23 印张

书号：ISBN 978-7-111-25502-4

定价：48.00 元

凡购本书，如有倒页、脱页、缺页，由本社发行部调换

本社购书热线：(010) 68326294

出版者的话

文艺复兴以降，源远流长的科学精神和逐步形成的学术规范，使西方国家在自然科学的各个领域取得了垄断性的优势；也正是这样的传统，使美国在信息技术发展的六十多年间名家辈出、独领风骚。在商业化的进程中，美国的产业界与教育界越来越紧密地结合，计算机学科中的许多泰山北斗同时身处科研和教学的最前线，由此而产生的经典科学著作，不仅擘划了研究的范畴，还揭示了学术的源变，既遵循学术规范，又自有学者个性，其价值并不会因年月的流逝而减退。

近年，在全球信息化大潮的推动下，我国的计算机产业发展迅猛，对专业人才的需求日益迫切。这对计算机教育界和出版界都既是机遇，也是挑战；而专业教材的建设在教育战略上显得举足轻重。在我国信息技术发展时间较短的现状下，美国等发达国家在其计算机科学发展的几十年间积淀和发展的经典教材仍有许多值得借鉴之处。因此，引进一批国外优秀计算机教材将对我国计算机教育事业的发展起到积极的推动作用，也是与世界接轨、建设真正的世界一流大学的必由之路。

机械工业出版社华章分社较早意识到“出版要为教育服务”。自1998年开始，华章分社就将工作重点放在了遴选、移译国外优秀教材上。经过多年的不懈努力，我们与Pearson，McGraw-Hill，Elsevier，MIT，John Wiley & Sons，Cengage等世界著名出版公司建立了良好的合作关系，从他们现有的数百种教材中甄选出Andrew S. Tanenbaum，Bjarne Stroustrup，Brain W. Kernighan，Dennis Ritchie，Jim Gray，Afred V. Aho，John E. Hopcroft，Jeffrey D. Ullman，Abraham Silberschatz，William Stallings，Donald E. Knuth，John L. Hennessy，Larry L. Peterson等大师名家的一批经典作品，以“计算机科学丛书”为总称出版，供读者学习、研究及珍藏。大理石纹理的封面，也正体现了这套丛书的品位和格调。

“计算机科学丛书”的出版工作得到了国内外学者的鼎力襄助，国内的专家不仅提供了中肯的选题指导，还不辞劳苦地担任了翻译和审校的工作；而原书的作者也相当关注其作品在中国的传播，有的还专程为其书的中译本作序。迄今，“计算机科学丛书”已经出版了近两百个品种，这些书籍在读者中树立了良好的口碑，并被许多高校采用为正式教材和参考书籍。其影印版“经典原版书库”作为姊妹篇也被越来越多实施双语教学的学校所采用。

权威的作者、经典的教材、一流的译者、严格的审校、精细的编辑，这些因素使我们的图书有了质量的保证。随着计算机科学与技术专业学科建设的不断完善和教材改革的逐渐深化，教育界对国外计算机教材的需求和应用都将步入一个新的阶段，我们的目标是尽善尽美，而反馈的意见正是我们达到这一终极目标的重要帮助。华章分社欢迎老师和读者对我们的工作提出建议或给予指正，我们的联系方法如下：

华章网站：www.hzbook.com

电子邮件：hzjsj@hzbook.com

联系电话：（010）88379604

联系地址：北京市西城区百万庄南街1号

邮政编码：100037

华章教育

译者序

软件工程的目标是按时交付满足客户需要、未超出预算、无错误、能随需求变化易于修改的软件。

在计算机界，范型一词最早用于描述编程风格。编程范型可以看成是程序员对程序执行的看法，而一些语言是专门为某个特定的范型设计的，当然也有一些语言支持多种范型。由于编程语言和软件开发的密切关系，范型一词也被引申至软件工程领域。面向对象的软件工程（object-oriented software engineering）就是一门利用面向对象范型实现软件工程目标的学科。

本书的作者 Stephen R. Schach 博士编写了 14 本与软件工程相关的畅销书，他撰写的书籍深入浅出，被许多学校选为教材并翻译为多国文字。Stephen R. Schach 认为，当前传统范型的使用在很大程度上仅限于遗留系统的维护。学生所学的第一门面向对象程序语言是C++ 或 Java，他们毕业后一般也将工作于一个使用面向对象范型的公司，因此有必要从面向对象范型出发对软件工程进行重新演绎，本书正是这样一本教科书。它全面、系统、清晰地介绍了面向对象软件工程的基本概念、原理、方法和工具，并通过实例说明了面向对象软件开发的整个过程。

本书分为两个部分，第一部分介绍了面向对象软件工程的基本理论，第二部分以工作流的形式介绍了软件生命周期，并通过一个小规模和一个中等规模的软件开发实例进行阐述。按照作者建议的课程内容安排，本书既可作为一个学期也可作为两个学期的教科书使用。

本书的习题也很有特色，可分为 5 类：一是每章都包含的与知识点相关的练习；二是每一章都选择了一篇经典论文，要求学生阅读并对相关的问题展开讨论，此类题目对于研究性的学习或讨论班类的课程特别有助；三是针对需求、分析和设计工作流设计的面向对象的分析和设计的小项目，通过实践，学生可快速掌握相关的工具和技术；四是针对每章讨论的实例，要求学生按照需求变化对实例进行某种修改，修改一个现成的产品与从头开始开发一个产品相比，有时前者对于知识的掌握更加有效；五是 project，其是为 3 个人组成的团队设计的，目的是锻炼团队的协同软件开发能力。

本书的翻译工作主要由黄林鹏负责，参与本书翻译的还有徐小辉、王欣、陈俊清、任建焜、王德俊、孙俊、沈飞、徐成、黄冠、曾慧清和杜思奇等。其中第 2 章至第 4 章由徐小辉初译，第 5 章至第 7 章由王欣初译，第 8 章至第 11 章由陈俊清初译，第 12 章至第 15 章由伍建焜初译。陆朝俊副教授对本书的翻译提供了不少建设性的帮助。本书译稿由黄林鹏修改、整理和定稿，其对最终出现的问题负责，请将批评意见发至 lphuang@ sjtu. edu. cn，不胜感激。

黄林鹏

2008 年 11 月于上海交通大学

前 言

在1988年，我撰写了一本教科书名为《Software Engineering》。事实上，在那本书中唯一提到面向对象范型的只有一节，即面向对象设计。

直到1994年，面向对象范型开始得到软件业界的认同，因此，我撰写了《Classical and Object-Oriented Software Engineering》一书。6年后，面向对象范型变得比传统范型更加重要。为了反映这个变化，我在2000年撰写的《Object-Oriented and Classical Software Engineering》（软件工程：面向对象和传统的方法）[⊖]中更换了这两个主题的顺序。

今天，传统范型的使用仅限于对遗留系统的维护。学生所学的第一门面向对象的程序语言是C++或Java，而面向对象语言在后续的计算机科学和计算机工程课程中也常被使用。学生们期望，他们毕业后在一个使用面向对象范型的公司工作。面向对象范型已经完全挤压了传统范型的生存空间，这也是本书命名为《面向对象软件工程》的原因。

本书特点

- 统一过程仍是面向对象软件开发的首选，因此本书中，学生所学习的仍是统一过程的理论和实践。
- 第1章深入分析了面向对象范型的优势。
- 第2章引入了迭代-递增生命周期模型。此外，本章还讨论了敏捷过程。
- 第3章介绍了统一过程的各个阶段和工作流（活动），解释了为何需要二维生命周期模型。
- 第4章讨论了软件团队的多种组织方式，包括敏捷过程团队和开源软件开发团队。
- 第5章介绍了一些重要的CASE工具。
- 第6章强调了连续测试的重要性。
- 第7章关注的重点是对象。
- 第8章强调了设计模式。
- 第9章给出了软件项目管理计划的新IEEE标准。
- 第10章、第11章、第12章和第13章主要阐述统一过程中的工作流（活动）。
- 第13章清晰地阐述了实现和集成的区别。
- 第14章强调了交付后维护的重要性。
- 第15章提供了更多的关于UML的资料。这一章对于采用本书作为两学期的软件工程课程教材的教师尤其有用。在第二学期，除了开发基于团队的学期项目或者“封顶”项目外，通过学习本章，学生可以获得更多的UML知识。
- 本书包含两个使用统一过程的运行实例：MSG基金会实例和电梯问题。其Java和C++语言的开发源码可从www.mhhe.com/schach下载。
- 除了两个用于说明完整软件生命周期的运行实例外，还有7个较小的实例，分别用于突出说明特定的主题，如移动目标、逐步求精和交付后维护等。
- 本书强调文档、维护、重用、可移植性、测试和CASE工具。学生只有认识到这些面向对象软件工程的基础的重要性，才能更好地掌握最新的技术。

⊖ 该书的影印版和中文版，已由机械工业出版社出版，书号分别为ISBN 978-7-111-20822-8和ISBN 978-7-111-21722-0。

- 本书注重面向对象生命周期模型、面向对象分析、面向对象设计、面向对象范型的管理建议、面向对象软件的测试和维护，还包括了面向对象范型的度量。除此之外，许多地方或多或少地会提及对象，因为面向对象范型不仅与执行不同的工作流相关，而且已经渗透到我们对软件工程的思考之中。对象技术的使用遍及全书。
- 软件过程仍是本书的整体基础。为了控制这个过程，必须能够度量项目中所发生的一切。因此，有必要强调度量。对于过程改进，本书包括了能力成熟度模型（CMM）、ISO/IEC 15504（SPICE）和 ISO/IEC 12207。在第 4 章中包括了人员能力成熟度模型（P-CMM）的内容。
- 本书与语言无关。虽然书中少量代码是以C++ 或 Java 语言编写的，但我尽量消除与语言相关的细节，以确保代码例子对于C++ 或 Java 语言背景的读者同样清晰。例如，对于输出，我不使用C++ 语言的 cout，也不使用 Java 语言的 System. out. println，而是使用伪代码指令 print（唯一的例外是第二个研究实例，其完整实现分别以C++ 和 Java 语言给出）。
- 本书包括 600 余条参考文献条目，其中有当前的研究文章、一些内容仍新鲜且相关的经典文献以及书籍。毫无疑问，面向对象软件工程是一门快速发展的学科，学生需要了解最新的研究成果以及在哪里可以找到它们。同时，今天的前沿研究是在昨天的成果之上进行的，如果一篇文章的思想今天仍然实用，就没有理由将其排除在参考文献之外。
- 对于先修课程，假定读者熟悉一种高级面向对象程序设计语言（如C++ 或 Java 等），除此之外，希望读者学习过数据结构课程。

本书的结构

本书既可为传统的一个学期的软件工程课程使用，也同样适用于目前普遍流行的较为新颖的两个学期软件工程课程。在传统的一个学期（或者四分之一学年）的课程中，理论部分内容不得不一掠而过，这样，教师才有时间给学生讲授完成学期项目所需要的知识和技能。如此匆忙的目的是使学生能尽早开始项目并在学期结束前完成项目。为了满足一个学期的、基于项目的软件工程课程教学的需要，本书第二部分以一个工作流接着一个工作流的形式介绍了软件生命周期，而第一部分则提供了为理解第二部分所需要的理论基础。例如，第一部分介绍了CASE、度量和测试，而第二部分的每一章都包含一节讲授工作流的 CASE 工具、一节讲授工作流的度量和一节介绍工作流期间的测试。为了尽早讲授第二部分内容，第一部分的教学内容应保持简短。另外，第一部分最后两章（第 8 章和第 9 章）的内容的教学可以推迟，留待和第二部分的内容一起并行讲授。由此，可以尽快地开始学期项目。

现在来看一下两个学期的软件工程课程的教学情况。越来越多的计算机科学系和计算机工程系的教师认识到大部分毕业生受聘的职位是软件工程师，因此，许多大学都开设两个学期（或两个四分之一学年）的软件工程系列课程。第一学期的课程主要是理论性的（但通常也包括某种形式的小型项目），而第二学期的课程则是以基于团队的学期项目开发为主。学期项目通常是课程的最后一部分内容，因此，如果学期项目是第二学期的内容，教师的授课时间就不会显得捉襟见肘。

综上所述，使用本书作为一个学期（或四分之一学年）课程的教科书的教师，可首先讲授本书第 1 章至第 7 章的大部分内容，然后开始讲授第二部分（第 10 章至第 15 章）的内容。第 8 章和第 9 章的内容可以和第二部分的内容平行教授或在学期末（学生完成项目）时讲授。当本书作为两个学期使用的教材时，应当按顺序讲授各章节的内容。第一学期课程结束之后，学生就应该做好参与第二学期基于团队的学期项目的准备。

为了确保学生能够真正理解第二部分所介绍的一些关键软件工程技术，本书对每项技术都

介绍了两次。首先，通过电梯问题引入该技术并进行讲解。电梯问题大小适中，读者借此可以看到技术在一个完整问题上的应用，而且还包括了足够多的细节，可以凸显所教技术的优缺点。然后，给出 MSG 基金会案例中与该项技术相关的部分，借助案例的详细解决方案展开对应技术的第二次阐述。

习题类型

本书有 5 种类型的习题。第一，在第 10 章、第 11 章和第 12 章末尾都有连续的面向对象的分析和设计项目。因为只有通过实践才能掌握如何执行需求、分析和设计工作流。

第二，每一章的末尾都包含意在突出知识点的习题。这些习题是独立的，所有相关的技术信息都可以在本书中找到。

第三，有一个学期项目。这个项目是为 3 个人组成的团队设计的，3 是最小的不用通过标准电话交换意见的团队成员的数目。学期项目由 14 个独立的组件组成，每一组件都位于某一相关章节之后。例如，第 12 章中，学期项目的组件所涉及的就是软件设计。通过将一个大的项目分解为较小的、明确定义的若干部分，指导教师就能更密切地掌控学生的学习进度。学期项目的构造目的是使指导教师可以自由地将这 14 个组件应用于其他所选择的项目。

本书是为研究生和本科高年级学生编写的，因此，第 4 种类型的习题是基于软件工程领域的研究论文而设计的。每一章都选择一篇重要的论文。要求学生阅读文章并回答与其内容相关的问题。当然，教师也可自由选择其他的研究论文，每一章的“延伸阅读材料”中包含了多篇相关的论文。

第 5 种类型的习题和实例研究相关。这种类型的题目是应许多授课教师的要求加入的。许多教师感到学生通过修改一个现成的产品比从头开始开发一个产品所学的要多。许多业界的资深工程师也同意这个观点。由此，在每章给出实例研究之处都有需要学生对实例进行某种修改的问题。例如，其中一章就要求学生回答如下问题：若以不同的顺序执行面向对象分析的步骤，效果如何。为了使学生便于修改实例，在 www. mhhe. com/schach 上可获得实例的源代码。

该网站也给授课教师提供了 PowerPoint 形式的授课笔记以及包含学期项目在内的所有习题的详细解答。

UML 材料

本书大量使用统一建模语言（UML）。如果学生以前没有学过 UML，该方面的内容可以两种方式讲授。我倾向于在需要时才进行教授，即每个 UML 概念仅在需要时才介绍。下表给出了本书使用的 UML 结构所对应的章节。

结　构	介绍对应 UML 图的章节
类图、注解、继承（泛化）、聚合、关联、导航三角形	7.7 节
用例	10.4.3 节
用例图、用例描述	10.7 节
固定形式	11.4 节
状态图	11.9 节
交互图（顺序图、协作图）	11.18 节

另外一种讲授方式是整体教学。本书第 15 章介绍了 UML，包含上述概念以及以后学习所需的材料。第 15 章可以随时讲授，这章的内容不依赖于前面 14 章。第 15 章包含的主题见下表：

结　构	介绍对应 UML 图的小节
类图、聚合、多态、结合、泛化、关联	15.2 节
注解	15.3 节
用例图	15.4 节
固定形式	15.5 节
交互图	15.6 节
状态图	15.7 节
活动图	15.8 节
包	15.9 节
组件图	15.10 节
配置图	15.11 节

致谢

感谢本书的评审，他们是

Michael A. Aars（Baylor University）
Keith S. Decker（University of Delaware）
Xiaocong Fan（The Pennsylvania State University）
Adrian Fiech（Memorial University）
Sudipto Ghosh（Colorado State University）
David C. Rine（George Mason University）
Keng Siau（University of Nebraska-Lincoln）
Anita Kuchera（Rensselaer Polytechnic Institute）
Matthew R. McBride（Southern Methodist University）
Michael McCracken（Georgia Institute of Technology）
Rick Mercer（University of Arizona）
Richard J. Povinelli（Marquette University）
John Sturman（Rensselaer Polytechnic Institute）
Levent Yilmaz（Auburn University）

衷心地感谢对本书以及我先前书籍做出重要贡献的三位同仁，其一是 Kris Irwin，他提供了用 Java 和C++ 语言实现的学期项目的完整解答；其二是 Jeff Gray，他给出了 MSG 基金会案例的实现；其三是 Lauren Ryder，他和我一起完成了本书的教师指导手册和 PowerPoint 幻灯片。

接着要感谢的是出版商 McGraw-Hill。真诚地感谢资深出版编辑 Faye Schilling，她在出版中期主动承担了产品经理的工作，对于她在需要时能欣然应允修改日程安排表示感激。策划编辑 Lora Kalb，她自始至终都是本书出版的顶梁柱，非常高兴能再次和她一起工作。衷心感谢本书排版 Lucy Mullins、校对 Dorothy Wendell 和产品经理 Joyce Berendes。最后，感谢 Brenda Rolwes，在他的协调下，Studio Montage 公司的封面设计师 Jenny Hobein 将位于悉尼海港的桥梁变形为能在读者脑海产生深刻印象的封面图案。

感谢世界各地的教师，他们发来了关于我的其他书籍的意见。我期待着收到他们针对本书发来的反馈意见。我的 E-mail 地址是 srs@ vuse. vanderbilt. edu。

学生们对本书的出版也做出了巨大的贡献。再次感谢 Vanderbilt 大学的学生提出的来自课堂内外的挑战性问题和建设性意见。非常感谢来自世界各地的学生通过 E-mail 发给我的问题和建议。我真诚期望学生如对待我以前的书籍一样发来本书的反馈意见。

最后，感谢我的家人对我持之以恒的支持。和编写先前的书籍一样，我尽量确保家庭义务先于书籍写作。然而，当交稿临近时，有时这是不可能的。此时，他们总是能表示理解，为此我深深地感谢他们。

让我将这本我所撰写的第 14 本书籍和我的爱一起献给我的孙子 Jackson。

Stephen R. Schach

目 录

第二部分 软件生命周期工作流

第一部分
面向对象软件工程简介

本书前9章的作用有二：一是向读者介绍面向对象软件过程；二是作为本书后半部分内容的基础，后半部分描述的是面向对象软件开发的工作流（活动）。

软件过程是生产软件的方式，它开始于概念探究，结束于产品退役。在这期间，产品将经历一系列步骤，包括需求、分析（规格说明）、设计、实现、集成、交付后维护和最终退役。软件过程不仅包括开发和维护软件所用的工具和技术，还涉及参与的软件工程人员。

第1章指出软件生产技术必须是有成本效益的，并且能促进软件生产团队成员之间的相互沟通。从这章开始，全书自始至终强这个目标的重要性。

第2章详细讨论了各种不同的软件生命周期模型，包括进化树模型、瀑布模型、快速原型开发模型、同步－稳定模型、开源模型、敏捷过程模型、螺旋模型以及最重要的迭代－递增模型（该模型是许多面向对象软件工程的基础）。为了使读者能够对具体的项目选用合适的生命周期模型，这章对各种生命周期模型进行了比较和对比。

第3章重点介绍了统一过程这一目前最有前景的软件开发方法。详细论述了敏捷过程这一流行的软件开发方法并介绍了开源软件。在本章结尾论述了软件过程的改进。

目前，大型软件项目仅凭个人力量是很难在给定的时间内完成的，这种项目通常由一组软件工程人员合作开发。第4章主要论述了团队该如何组织才能使其成员能富有成效地合作。本章论述各种不同团队，包括民主型团队、主程序员团队、同步－稳定团队、开源团队和敏捷过程团队的组织方法。

软件工程师要求能使用大量不同类型的工具，包括分析型工具和应用型工具。第5章将介绍不同的软件工程工具，其中之一是逐步求精，这是一种把大问题分解成较小但容易处理的问题的技术。另一种工具是成本－效益分析，这是一种判断软件项目经济可行性的工具。接着要描述的是计算机辅助软件工程（CASE）工具，这是一种用于辅助软件工程师开发和维护软件产品的工具。最后，为了管理软件过程，必须度量软件过程的不同指标，以判断该软件是否偏离了正常轨道。这些测量（度量）工具对一个项目的成功是至关重要的。

第10～13章详细介绍了第5章后面的两个主题——CASE工具和度量。这几章对支持每种工作流的CASE工具进行了讨论，同时也给出了恰当的工作流管理所需的度量。

第6章讨论了与测试有关的一些基本概念。对软件生命周期的每种工作流所使用的具体软件测试技术将在第10～14章中介绍。

第7章详细解释了类和对象，证明了为什么面向对象范型比传统范型更成功。本书的其余部分都会用到这一章介绍的概念，特别是在第10章、第11章和第12章。

第7章的思想在第8章中得到了扩展，编写可移植到不同硬件平台的可重用软件是非常重要的。这一章的第一部分讲的是重用，包括多种重用实例的研究以及一些重用策略（如模式和框架）。可移植性是这一章第二部分的主题，这部分对可移植性策略进行了比较深入的介绍。本章反复介绍了对象在获取可重用性和可移植性时所起的作用。

第一部分的最后一章是第9章。在软件设计项目开始之前，一个基本的要求是做一份详细的整体计划。项目一旦开始，管理者必须密切监督进度，注意项目是否偏离计划，并在必要时采取正确的行动。同样，向客户提供关于项目的时间和经费开销的准确估算也是很重要的。这一章描述了不同的估算技术，包括功能点技术和COCOMO II技术；给出了软件项目管理计划的详细描述。由于主要的计划活动都发生在分析工作流结束之后，所以在第11章将使用本章给出的材料。

第 1 章　面向对象软件工程的范畴

学习目标

通过本章学习，读者应能：

- 了解面向对象软件工程的定义。
- 解释现在面向对象范型被广泛接受的原因。
- 论述软件工程各方面的含义。
- 描述现代维护观点。
- 论述持续计划、测试和编制文档的重要性。
- 认识遵守伦理规范的重要性。

这是一个众所周知的故事，有一个公司的主管一天收到了一份计算机生成的账单，账单的金额为 0.00 美元，他与朋友一起尽情地讥讽了“愚蠢的计算机”一番后将账单扔掉了，一个月以后，他收到了一份标记过期 30 天的类似账单，接着，第 3 张账单也来了。又一个月之后，第 4 张账单来了，同时附有一份通知，提示如果不及时付清这个 0.00 美元的账单将可能采取法律行动。

第 5 张账单，上面标记过期 120 天，没有任何提示，直白而粗鲁地威胁道，如果不立即付清账单，将采取所有必须的法律手段。这位主管担心自己公司的信用会受到这个疯狂机器的影响，于是找了一位软件工程师朋友，跟他讲了这件恼人的事情。软件工程师忍住笑，让主管邮寄去一张 0.00 美元的支票。这产生了期望的结果，几天后一张 0.00 美元的收据寄来了，主管小心翼翼地收好这张收据，以防将来计算机宣称那张 0.00 美元的账单他还没有支付。

这个故事有一个不太为人知晓的结局。几天后，银行经理召见了这位主管。银行经理拿着一张 0.00 美元的支票问他，“这是你的支票吗?”

这位主管回答：“是的”。

“那你能告诉我为什么要签署一张 0.00 美元的支票吗?”银行经理问道。

于是，整个故事被重新讲述了一遍。当主管讲完时，经理盯住他，温和地问道“你付 0.00 美元对我们计算机系统会造成什么后果，你想过吗?”

计算机专业人员虽然会觉得这个故事可笑，但是也会感到一些窘迫。毕竟，任何一个人所设计或完成的产品，在其原型阶段，都有可能出现类似寄送催讨 0.00 美元信件这种问题。目前，虽然在测试中总能发现此类错误，但是计算机专业人员的笑声会包含一种恐惧感，他们担心这种错误没有在产品交付给顾客前被检测出来。

1979 年 11 月 9 日检测到的一个软件错误，却绝对称不上幽默。这天，美国战略空军司令部收到全球军事指挥和控制系统（WWMCCS）计算机网络发出的报告，报告称前苏联向美国发射了导弹，这引起了警报混乱［Neumann，1980］。而实际发生的情况和 5 年后放映的电影《War Games》中的剧情一样，是误把模拟演习当成了真实发生的战事。虽然出于可以理解的原因，美国国防部未能详细说明把实验数据当成真实数据的确切过程，但有理由把这个问题归于软件错误。整个系统在设计时就无法区分模拟和真实情景；或者是用户接口没有包括必要的检查，以确保系统终端用户能分辨真伪。换句话说，如果这个问题确实是由软件引起的，软件错误可能会给文明社会带来不愉快和灾难性的后果（有关由软件错误所导致的灾难方面的信息，请参见备忘录 1.1）。

无论涉及的是账单还是防空项目，许多软件都会推迟其交付时间，原因可能是预算超出、