高等学校软件工程专业系列教材

实用性和系统性

项目案例

课件和源码资源

UML 2 面向对象分析与设计（第2版）

谭火彬 编著

微课版

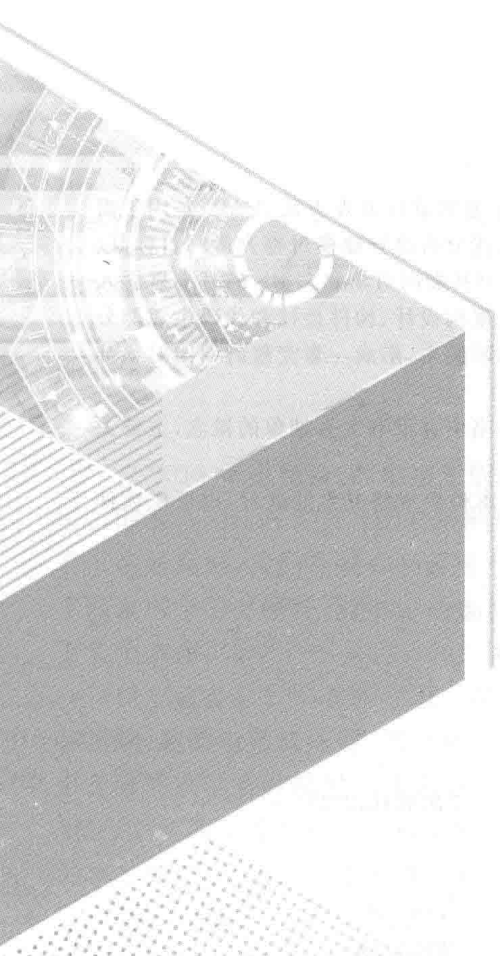
300分钟 教学视频

全程语音讲解

非外借

清華大學出版社

高等学校软件工程专业系列教材



UML 2 面向对象分析与设计（第2版）

◎ 谭火彬 编著

清華大學出版社
北京

内 容 简 介

分析和设计是软件开发中至关重要的一环，面向对象的方法是主流的软件开发方法，UML 是用于面向对象分析设计的标准化建模语言。本书围绕这 3 个方面展开，以论述分析设计建模过程为最终目标，以面向对象方法作为建模的理论基础，以 UML 作为建模支撑语言。全书从面向对象和 UML 的基本概念入手，循序渐进地讲解业务建模、需求建模、需求分析、设计原则和模式、架构设计、构件设计和代码生成等分析设计中的各个知识点，并通过多个贯穿全书的案例将各个知识点串联起来，形成一套完整的面向对象分析设计方法论。

本书是作者多年从事软件工程教学和软件项目开发实践的总结，书中并没有太多抽象的概念，主要关注实际软件开发中所需要的知识和实践技能，力求做到通俗易懂。

本书既可作为高等院校软件工程专业及计算机相关专业高年级本科生或研究生的教材，也可供软件开发人员阅读和参考。

图书在版编目(CIP)数据

UML 2 面向对象分析与设计/谭火彬编著. —2 版. —北京：清华大学出版社，2019
(高等学校软件工程专业系列教材)
ISBN 978-7-302-50698-0

Ⅰ. ①U…　Ⅱ. ①谭…　Ⅲ. ①面向对象语言－程序设计　Ⅳ. ①TP312.8

中国版本图书馆 CIP 数据核字(2018)第 163071 号

策划编辑：魏江江
责任编辑：王冰飞
封面设计：刘　键
责任校对：胡伟民
责任印制：丛怀宇

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,010-62795954
印 刷 者：清华大学印刷厂
装 订 者：三河市溧源装订厂
经　　销：全国新华书店
开　　本：185mm×260mm　　**印　　张**：23.25　　**字　　数**：[illegible]
版　　次：2013 年 5 月第 1 版　2019 年 1 月第 2 版　　**印　　次**：2019 年 1 月第 1 次印刷
印　　数：16001～17500
定　　价：59.50 元

产品编号：071387-01

FOREWORD

第2版前言

自2013年出版后，本书第1版被多所高校选用，得到了大家的认可。随着UML语言自身的发展和广泛应用，以及在这期间部分使用者反馈的意见，我们发现有必要对书中的内容进行修订。本次修订除了修改书中的差错外，对书中的部分内容也进行了调整，主要包括以下几个方面。

(1) 增加了200道练习题。在每一章的最后新增了数量不等的练习题，题型包括选择题、简答题和应用题。这些习题主要来自编者多年来的教学实践积累。其中，选择题主要涉及书中的重要概念和典型应用，可用于课堂教学过程中的随堂测试；简答题涵盖了每个章节的核心知识点，可用于学生课后复习；而应用题则为案例实践和部分课外调研，可用于学生课外综合实践。读者可联系出版社或编者本人获取习题答案。

(2) 更新部分内容。结合UML 2.5和建模领域的新发展与应用，更新了书中的部分内容，包括第2章UML组成结构、第3章UML活动图等内容。

(3) 更新有关建模工具的使用。本书第1版中建模实践的内容主要是围绕Rational Rose工具开展的。新版在保留原有的内容基础上，增加了目前流行的Enterprise Architect 12工具使用的介绍，对书中涉及的案例模型同时提供了这两个工具的版本。

(4) 增加了多媒体教学资料。除了第1版提供的教学PPT和课程案例模型外，新版还配套录制了一些教学视频，主要讲解各类建模操作实践，欢迎读者观看。

当然，书中还难免存在一些不足或者疏漏之处，欢迎各位读者批评指正；如有疑问或问题，请随时与编者联系，电子邮箱：thbin@buaa.edu.cn，编者在博客园上的个人博客主页会发布图书相关信息，网址为http://www.cnblogs.com/thbin/。

编者

2018年9月

FOREWORD

第1版前言

毋庸置疑，面向对象的方法已成为现代软件开发中最主流的方法，即便是SOA、云计算等概念也均是建立在面向对象方法基础之上的进一步抽象。与此同时，自1997年UML正式诞生，到2011年发布的UML 2.4.1，经历了多个版本的发展和完善，UML已成为建模语言的国际标准(ISO 19501和ISO 19505)，基于UML的面向对象分析设计方法也日益成熟。然而，UML只是提供了一种标准的表示法，在分析设计过程中，什么时候以什么方式使用什么UML模型等具体的建模实践并没有在UML中定义，而这才是广大软件开发人员所要掌握的实践技能，也是本书所关注的内容。

本书目标

本书系统地介绍了利用UML 2进行面向对象分析与设计的过程，主要目标包括以下3个方面。

- OO(面向对象)：建立对象的思维方式，对面向对象思想和理论有深入的理解。
- UML(统一建模语言)：能够熟练地使用UML表达面向对象的设计思想。
- Model(建模)：运用面向对象的一般原则和模式进行应用系统的分析和设计建模。

组织结构

本书总体结构可以分为三大部分。

第一部分为基础概念，包括第1章和第2章。其中，第1章为上升到面向对象，通过案例引出面向对象的方法，并重点介绍了对象技术中的几个核心概念；第2章为可视化建模技术基础，全面介绍了有关UML 2的组织结构和内容。这些基础概念将在后续的分析设计中被广泛使用。

第二部分为面向对象的分析，包括第3～5章。其中，第3章为业务建模，原始业务是需求分析的出发点，本章简要地介绍了业务建模的基本概念和方法，并提供了一些实践指南；第4章为用例建模，系

统地介绍了利用UML用例模型进行需求定义的过程和实践；第5章为用例分析，介绍了如何围绕第4章所建立的用例模型进行面向对象分析的方法和实践。

第三部分为面向对象的设计，包括第6～10章。其中，第6～7章为设计基础，分别介绍了有关面向对象设计的基本原则和模式，这些原则和模式将有效地指导后续的设计过程；第8章为架构设计，介绍了如何在系统的全局范围内，基于分析活动的成果定义设计元素、设计机制等内容，从而构造系统的组织结构；第9章为构件设计，介绍了如何在系统的局部设计各个细节，包括用例设计、子系统设计、类设计和数据库设计等方面的内容；第10章为从模型到代码，简单地介绍了设计模型和代码之间的映射，为后续编码做准备。

在案例设计方面，本书设计了两个贯穿全书的案例：旅店预订系统和旅游业务申请系统。这两个案例各有侧重，通过它们，读者不仅可以掌握UML建模的基本方法，还可以全面了解在整个系统开发过程中从分析模型到设计模型不断演化的过程。此外，在各个章节中，针对一些特定的知识点，设计了各种小的案例进行阐述，这些案例包括第1章开篇的素数问题、第2章的图书馆管理系统、第3章的饭店系统、第6章的咖啡机系统、第7章的可复用按钮等。

有关UML内容，本书从两个层面进行介绍：首先在第2章中对UML基本概念、组织结构和各种模型进行了系统、初步的介绍；然后在后续的分析设计实践中针对一些重点UML模型的使用进行详细、深入的论述，使读者在掌握UML基本概念后，能够在需要的地方进行应用。有关各章节中涉及的UML模型和核心概念如下表所示。

章　节	章节名称	章节性质	UML模型	核心概念
第1章	上升到面向对象	基础		面向对象技术、类、对象
第2章	可视化建模技术	基础	*（全部）	UML组织结构
第3章	业务建模	业务	活动图	业务参与者、业务用例、活动
第4章	用例建模	需求	用例图	用例、参与者、用例关系
第5章	用例分析	分析	顺序图、类图	用例实现、分析类（边界类、控制类和实体类）
第6章	面向对象的设计原则	设计基础	通信图	设计原则
第7章	面向对象的设计模式	设计基础		模式、设计模式、GRASP
第8章	架构设计	设计	包图、部署图	构架、设计元素、设计机制、进程、线程
第9章	构件设计	设计	类图、状态机图、构件图	接口、子系统、组合结构操作、方法、状态、关系
第10章	从模型到代码	实现		正向工程、逆向工程

有关UML工具的选择，也是读者所关心的问题。市面上有很多商业的或开源的UML工具，这些工具各有特点，但核心建模功能都相差不大。工具本身只是一种实现手段，选择哪款UML工具，并不影响对本书概念的理解和实践。本书中的UML模型绘制主要采用IBM Rational Rose 2003和Sparx Systems Enterprise Architect 7.5工具，都不算最新的工具，够用即可。选择Rose是因为编者从最早学习UML开始就一直使用该工具，虽然有点老，但能满足大部分建模需求；不过由于Rose 2003不支持最新的UML 2，因此针对UML 2中的新概念选择了Enterprise Architect（没有选择IBM Rational的后续版本RSA是因为

这个工具集过于庞大，更倾向于一个集成开发平台，不适合作为一个普通的 UML 工具进行介绍；此外，其默认的图形样式颜色较淡，不适合放在书中展示）。当然，这两个工具也无法覆盖到所有的 UML 概念，因此书中有些模型是选择其他的 UML 工具或一些绘图工具完成的。

致谢

本书是编者在北京航空航天大学软件学院近 10 年的研究生面向对象分析与设计课程教学基础上编写而成的，非常感谢软件学院的领导、教师和学生提供这样的交流平台，书中大部分内容和案例都是通过此课程的教学不断形成和完善的。在此，还要特别感谢在课程教学和教材编写过程中提供帮助的姚淑珍教授、林广艳副教授、杨文龙教授、孟岩老师等学院教师和企业导师。此外，编者还曾参加过 IBM 组织的面向对象分析与设计课程培训、UML China 公开课等，也曾为一些软件企业进行过有关方面的培训，这些来自企业的经历也充实了课程和教材内容，在此一并表示感谢。最后，还需要感谢我的家人，正是因为她们无私的奉献才使得我有更多的精力投入到课程教学和教材编写中。

编 者

2012 年 10 月

CONTENTS

目　录

CHAPTER 1

第 1 章

上升到面向对象

从早期的手工开发阶段到软件工程的出现，从传统的结构化开发方法到面向对象的方法，软件开发方法正逐渐扮演着更加重要的角色。而面向对象的方法也已取代了传统的软件开发方法，成为软件开发方法的主流。对象、类、封装、继承和多态等概念也已被广泛接受。

本章从一个简单案例入手，分析面向对象方法的特点，并对对象技术中的各类关键概念进行详细介绍，从而帮助读者建立面向对象的思维方式，为后续的分析和设计打下理论基础。

本章目标

本章是基础章，通过对本章的学习，读者能够快速地掌握面向对象领域的核心概念，了解面向对象技术、系统分析与设计及它们与 UML 之间的关系，并建立面向对象的思维方式。

主要内容

(1) 从结构化到面向对象：理解传统结构化方法与面向对象方法之间的思维差异，掌握它们在具体应用中的区别和联系。

(2) 面向对象技术：掌握面向对象技术的定义，了解面向对象技术的发展历史，对面向对象技术的优势要有一定的认识。

(3) 对象和类：掌握并理解对象和类的定义及它们之间的关系。

(4) 面向对象技术相关原则：掌握抽象、封装、分解、分层、复用等面向对象的基本原则，掌握并理解泛化和多态机制的作用。

(5) 上升到面向对象：了解面向对象、建模和 UML 之间的关系，并对面向对象的建模要有一定的认识。

1.1 从素数问题看面向对象

视频讲解

随着 C++、Java、C# 等面向对象编程语言的日益普及，面向对象技术已经得到了广泛的应用。从面向对象的编程语言到面向对象软件工程方法也日益被人们重视起来。面向对象的更进一步应用(如面向构件、面向服务、面向模式等)也逐步发展起

来,而这些新方法都是建立在面向对象的思维方式上的。由此可见,深入理解面向对象的思维方式不仅可以帮助我们理解当前面临的应用模式,还是我们进一步学习和发展的必经之路。

很多人都经历过传统的结构化思维方式,让我们先通过一个经典的数据结构中的算法问题来探讨如何走出传统的思维模式。利用面向对象的思维方式来解决问题,进而理解到底什么是面向对象思维方式,为后续的对象建模热身。

1.1.1 问题的提出

这里,我们所面临的是一个求素数的问题。素数(也叫质数)是指除了1和它本身外,不能被其他正整数整除的数。按照习惯规定,1不算素数,最小的素数为2,其余的有3、5、7、11、13、17、19等。

根据上面的定义可以推导出判断素数的算法:对于数n,判断n能否被i($i=2,3,4,5,\ldots,n-1$)整除,如果全部不能被整除,则n是素数,只要有一个能除尽,则n不是素数。事实上,为了压缩循环次数,可将判断范围从2 ~ $n-1$改为2 ~ sqrt(n)。

筛选法是一个经典的求素数的算法,它的作用并不是判断某个数是否为素数,而是求小于数n的所有素数。下面通过一个简单的例子来说明筛选法的求解过程。

我们需要求解50以内的所有素数$i(2<i<n)$,为此需要进行如下的筛选(见图1-1)。

筛掉2的倍数:	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	...
筛掉3的倍数:	2	3	5	7	9	11	13	15	17	19	21	23	25	27	29	31	33	...		
筛掉5的倍数:	2	3	5	7	11	13	17	19	23	25	29	31	35	37	41	43	...			
筛掉7的倍数:	2	3	5	7	11	13	17	19	23	29	31	37	41	43	47	49				
留下素数序列:	2	3	5	7	11	13	17	19	23	29	31	37	41	43	47					

图1-1 筛选法求素数示意图

为了利用程序实现此算法,需要进行算法设计。考虑分别采用结构化方法和面向对象的方法实现此算法,并将设计方案以合适的方式记录下来。下面将通过对两种不同方法的比较,讲解结构化方法和面向对象方法在本质上的区别。

1.1.2 传统的结构化解决方案

按照传统的结构化方法,算法的执行过程如下所示。

(1) 以当前最小的数(即2)作为因子,将后面所有可以被2整除的数去掉(因为它们肯定不是素数,参见图1-1的第1行,去掉后面的4、6、8等,剩余结果见第2行)。

(2) 取剩余序列中第二小的数(即3)作为因子,将后面所有可以被3整除的数去掉(参见图1-1中的第2行,去掉后面的9、15、21等,剩余结果见第3行)。

(3) 如此继续,直到所取得最小数大于sqrt(n)(图1-1中第4行为最后一次筛选,此时的因子为7,因为下一个因子即为11,大于sqrt(50))。

(4) 剩余的序列即为n以内的所有素数。

为了更清楚地描述该算法,可以采用流程图来阐述算法流程,该算法的流程图如图1-2所示。

需要说明的是,上述的流程图和前面描述的算法有所出入。在具体实现时,考虑到算法的执行效率,并没有将当前因子的倍数直接删除(因为如果采用数组存储当前数字序列,则

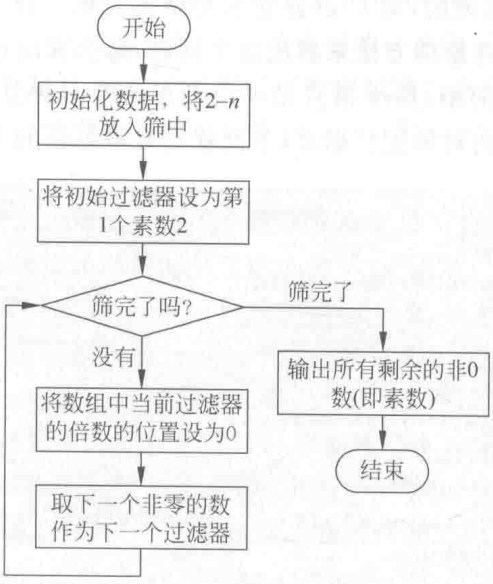


图 1-2 筛选法求素数流程图

删除过程的算法复杂度都为 O(n)),而是将相应的位置设为 0,表明该位置已经没有数据了。

设计出这样一个算法后,后面的实现过程就是"水到渠成"的事了,我们可以选择一种合适的编程语言来实现。下面列出了该算法的 C 语言实现[①]。

```
int main(){
    int * sieve, n;
    int iCounter = 2, iMax, i;
    printf("Please input max number:");
    scanf(" % d", &n);
    sieve = (int * )malloc((n - 1) * sizeof(int))
    for(i = 0;i < n - 1;i++) { sieve[i] = i + 2; }
    iMax = sqrt(n);
    while (iCounter < = iMax) {
        for (i = 2 * iCounter - 2; i < n - 1; i += iCounter)
            sieve[i] = 0;
        iCounter++;  }
    for(i = 0; i < n - 1; i++)
        if sieve[i]! = 0 printf(" % d ",sieve[i]);
    return 0;
}
```

1.1.3 面向对象的解决方案

在上面的问题中,可以很容易地从算法描述中构造目标程序。很自然,这也似乎很符合人们的思维习惯。那么这种方法是面向对象的方法吗?也许读者都会说不是。因为很明

① 需要说明的是,本书提供的代码主要是便于读者理解设计方案,多数代码只是一个片段,并不完整,而且很多示例性的代码语法也较不严格。另外,这些代码大多数采用了 C++或 Java 语法表示。