清华电脑学堂

- ✓ 总结了作者多年UML应用经验和教学心得
- ✓ 系统讲解了UML技术的要点和难点
- ✓ 实例众多、图例丰富、实用性强
- ✓ 提供丰富的课堂练习和课后习题

UML 建模 设计与分析

标准教程（2013-2015版）

王菁 赵元庆 等编著

清华大学出版社

清華
电脑学堂

UML 建模 设计与分析

标准教程（2013-2015版）

■ 王菁 赵元庆 等编著

清华大学出版社
北 京

内 容 简 介

本书全面介绍了使用 UML 进行软件设计、分析与开发的知识。全书共包含 18 章，内容涉及面向对象的分析方法和设计方法，面向对象分析的三层设计，现实开发模型中所存在的问题，用例图、类图、对象图和包图，活动图，通信图、时间图、状态机图、组件图和部署图，UML 的核心语义、UML 的体系结构以及面向对象约束语言等，最后两章通过具体的案例详细介绍如何使用 UML 中的模型图对系统建模。本书内容全面、实例丰富，适合作为高校相关专业和社会培训教材，也可以作为软件设计人员和开发人员的参考资料。

图书在版编目（CIP）数据

UML 建模、设计与分析标准教程（2013—2015 版）/王菁，赵元庆等编著. —北京：清华大学出版社，2013.7
（清华电脑学堂）
ISBN 978-7-302-31872-9

Ⅰ. ①U…　Ⅱ. ①王…　②赵…　Ⅲ. ①面向对象语言-程序设计-教材　Ⅳ. ①TP312

中国版本图书馆 CIP 数据核字（2013）第 070996 号

责任编辑：冯志强
封面设计：李晓春
责任校对：胡伟民
责任印制：王静怡

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
印 装 者：清华大学印刷厂
经　　销：全国新华书店
开　　本：185mm×260mm　　**印　　张**：22.5　　**字　　数**：565 千字
版　　次：2013 年 7 月第 1 版　　**印　　次**：2013 年 7 月第1 次印刷
印　　数：1～4000
定　　价：39.80 元

产品编号：050574-01

前　言

20 世纪 90 年代，人们推出了许多不同的面向对象设计和分析方法。这些不同的面向对象的方法具有不同的建模符号体系，这些不同的符号体系极大地妨碍了软件的设计人员、开发人员和用户之间的交流。因此，有必要在分析、比较不同的建模语言以及总结面向对象技术应用实践的基础上，建立一个标准的、统一的建模语言。UML 就是这样的建模语言，UML 在 1997 年 11 月 17 日被对象管理组织 OMG 采纳为基于面向对象技术的标准建模语言。统一建模语言 UML 不仅统一了面向对象方法中的符号表示，而且在其基础上进一步发展，并最终被统一为被人们所接受的标准。

UML 相当适合于以体系结构为中心的、用例驱动的、迭代式和渐增式的软件开发过程，其应用领域颇为广泛，除了可用于具有实时性要求的软件系统建模以及处理复杂数据的信息系统建模外，还可用于描述非软件领域的系统。

UML 适用于系统开发过程中从需求分析到完成测试的各个阶段：在需求分析阶段，可以用用户模型视图来捕获用户需求；在分析和设计阶段，可以用静态结构和行为模型视图来描述系统的静态结构和动态行为；在实现阶段，可以将 UML 模型自动转换为用面向对象程序设计语言实现代码。

本书主要内容

本书以渐进的顺序来介绍 UML，从需求分析开始，然后再构建和部署系统。

第 1 章介绍 UML 入门的基础知识。本章首先介绍面向对象的分析方法和设计方法，介绍现实软件开发模式所面临的问题，然后介绍面向对象分析的工具和方法——UML，最后简单介绍统一过程 RUP 的知识。

第 2 章介绍什么是用例图，主要包含用例图的构成、用例间的关系、用例描述以及如何使用用例图建模等内容。

第 3 章介绍 UML 中类图的基本概念，重点介绍了类图的概念、表示方法、接口以及类图中常用的几种关系，如泛化关系、关联关系、依赖关系、实现关系以及聚合关系和组合关系等。

第 4 章介绍对象图和包图的相关内容，包括对象图的概念、表示方法和类图的区别以及包图的概念、表示方法、包之间的关系与类图的区别等。

第 5 章介绍活动图的概念、组成元素和控制结点，还介绍了活动图与状态图之间的不同点。

第 6 章介绍顺序图的作用、定义、构成、使用以及创建方法等内容。

第 7 章介绍系统交互的动态视图——通信图，包括通信图的含义、构成、消息对象、消息迭代以及顺序图和通信图的比较等。

第 8 章介绍与时间图有关的内容，包括时间图的构成、时间约束和时间图的替换表示法等。

第 9 章介绍 UML 中属于行为图之一的状态机图，重点介绍状态机图的构成、标记符、转移、组合状态以及如何建模等内容。

第 10 章介绍交互结构图与交互概况图的内容。

第 11 章介绍组件图和部署图的概念、构成、组件间和部署间的关系以及如何建模等。

第 12 章介绍 RUP 的二维空间、核心工作流程以及十大开发要素等。

第 13 章介绍如何将 UML 模型映射到关系型数据库，其内容主要涉及模型结构的映射和模型功能的映射两部分。

第 14 章介绍 UML 的核心语义以及 UML 的体系结构。

第 15 章介绍对象约束语言的概念，对象约束语言的结构、语法、表达式和数据类型等内容，最后介绍集合和约束的使用。

第 16 章以面向对象的代表语言——C++为例介绍 UML 模型转换为实现的原理和方法，包括实现类，泛化的实现、类之间各种关系的实现以及接口等。

第 17 章介绍如何使用 UML 进行建模绘制不同的图，如用例图、类图、顺序图和组件图等。

第 18 章介绍如何使用 UML 绘制网上购物系统的相关模型图，通过本章的介绍，使读者更全面、更快速地了解 UML 中各种模型图的功能和建模步骤。

本书特色

本书是一本完整介绍 UML 在软件设计和开发过程中应用的教程，在编写过程中我们精心设计了丰富的实例，以帮助读者顺利学习本书内容。

- **理论紧密结合实践** 全书提供了 3 个完整的分析案例，通过示例分析、设计过程讲解 UML 的应用知识。
- **图文并茂** UML 理论知识比较抽象，本书绘制了大量 UML 图，帮助读者直观理解抽象内容。
- **网站互动** 我们在网站上提供了本书案例和扩展内容的资料链接，便于读者继续学习相关知识；授课教师也可以下载本书教学课件和其他教学资源。
- **思考与练习** 简答题测试读者对各章内容的掌握程度；分析题理论结合实际，引导读者深入掌握 UML 理论知识。

读者对象

本书在多家院校成熟教案以及自编教材的基础上整合编写，全面介绍使用 UML 进行软件设计、分析与开发的知识，适合作为普通高校计算机专业教材，也可以作为软件设计人员和开发人员的参考资料。

本书作者均从事软件分析、开发和教学工作，拥有丰富的 UML 开发案例。参与本

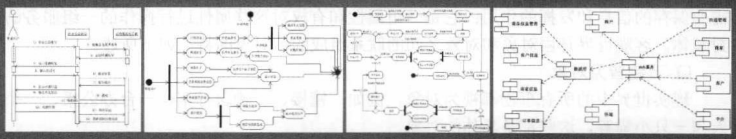
书编写的除了封面署名人员外，还有王敏、马海军、祁凯、孙江玮、田成军、刘俊杰、赵俊昌、王泽波、张银鹤、刘治国、何方、李海庆、王树兴、朱俊成、康显丽、崔群法、孙岩、倪宝童、王立新、王咏梅、辛爱军、牛小平、贾栓稳、郭磊、杨宁宁、郭晓俊、方宁、王黎、安征、亢凤林、李海峰等人。由于时间仓促、水平有限，疏漏之处在所难免，欢迎读者朋友登录清华大学出版社的网站 www.tup.com.cn 与我们联系，帮助我们改进提高。

编　者

目　录

第1章

UML入门



从20世纪80年代末到90年代中出现了一大批面向对象的分析与设计方法。各种形式的方法相互之间各不相同，它们之间的差异为面向对象程序开发方法的发展和应用带来了不便。在这种情况下，UML应运而生。UML是软件和系统开发的标准建模语言，它主要以图形的方式对系统进行分析、设计。UML解决了多种面向对象分析、开发与设计时的差异，是一种专用于系统建模的语言，而且还为开发人员与客户之间，以及开发人员之间的沟通与理解架起了“桥梁”。

作为本书的第一章将从UML的诞生背景开始介绍，使读者了解UML出现的必要性，以及对UML有一个全面、整体的认识。

本章学习要点：

- 理解面向对象中对象的概念
- 了解面向对象开发
- 熟悉面向对象的主要特性
- 了解面向对象的三层和三种模型
- 了解UML出现的前提
- 熟悉UML的4层体系结构
- 了解常用UML建模工具
- 熟悉UML中的视图、图、事物、关系和通用机制
- 了解什么是RUP及与UML的关系

1.1 认识面向对象

为了解决开发大型软件系统的复杂性和可维护性，在过去的几十年中出现了许多开发方法，像瀑布开发方法、螺旋式开发方法、迭代式开发方法等。

面向对象是一种新兴程序设计和开发方法，其基本思想是使用对象、类、封装、继承、关联、消息等基本概念来对系统进行分析与设计。

1.1.1 面向对象简介

面向对象（Object Oriented，OO）是计算机界关心的重点，也是20世纪90年代软件开发方法的主流。

面向对象的核心是对象，它是系统中用来描述客观事物的一个实体，它是构成系统的一个基本单位。一个对象由一组属性和对这组属性进行操作的一组服务组成。从更抽象的角度来说，对象是问题域或实现域中某些事物的一个抽象，它反映该事物在系统中需要保存的信息和发挥的作用；它是一组属性和有权对这些属性进行操作的一组服务的封装体。客观世界是由对象和对象之间的联系组成的。对象的特点如下所述。

❑ 万物皆为对象

现实世界中的所有事物都视为对象，例如一幢楼、一辆自行车、一台办公桌、一个人和一只小猫等，这些都是具体的对象。

❑ 对象都是唯一的

在现实世界中，每个对象都是与众不同的，就像是“世界上没有两个完全相同的人”一样的道理。

❑ 对象具有属性和行为

例如一位超市的顾客具有姓名、年龄、体重等属性，同时也具有购物、结账等行为。

❑ 对象具有状态

状态是指某一时刻对象的各个属性的取值。因为对象的属性并不是一直不变的，例如一位顾客的姓名和年龄等属性。

❑ 对象都属于某个类别

每个对象都是某个类别的实例。例如收银员布兰妮和顾客朱丽叶是两个不同的实例，一个是收银员类的实例，一个是顾客类的实例。同一个类的所有实例都具有相同的属性，只不过属性的取值不一定相同，例如，布兰妮和朱丽叶都有姓名、年龄、体重等属性，但是这些属性的值不一定相同。

面向对象可以分为面向对象的分析（Object Oriented Analysis，OOA）、面向对象的设计（Object Oriented Design，OOD）和面向对象的编程（Object Oriented Programming，OOP），并且涉及到数据库系统、交互式界面、应用平台、分布式系统和网络管理结构等领域。

1．面向对象的分析

OOA就是应用面向对象方法进行系统分析。OOA是面向对象方法从编程领域向分

析领域发展的产物。从根本上讲，面向对象是一种方法论，不仅仅是一种编程技巧和编程风格，而且是一套可用于软件开发全过程的软件工程方法，OOA 是其中的第一个环节。OOA 的基本任务是运用面向对象方法，从问题域中获取需要的类和对象，以及它们之间的各种关系。

2．面向对象的设计

OOD 指面向对象设计，在软件设计生命周期中发生于 OOA 后期或者之后。在面向对象的软件工程中，OOD 是软件开发过程中的一个大阶段，其目标是建立可靠的、可实现的系统模型；其过程是完善 OOA 的成果，细化分析。其与 OOA 的关系为：OOA 表达了“做什么”，而 OOD 则表达了“怎么做”，即分析只解决系统“做什么”，不涉及“怎么做”，而设计解决“怎么做”的问题。

3．面向对象的编程

OOP 就是使用某种面向对象的语言，实现系统中的类和对象，并使得系统能够正常运行。在理想的 OO 开发过程中，OOP 只是简单地使用编程语言实现了 OOA 和 OOD 分析和设计模型。

1.1.2 面向对象开发简介

在过去很长的时间内，使用面向对象开发的大多数人都专注于编程语言，而且文档的重点也都停留在实现上，而不是分析和设计。最初在解决传统开发语言中那些不灵活的问题时，面向对象编程语言显示出强大的威力。但是对于软件工程来说，这种专注可以说是某种意义上的倒退——因为它过度注重实现机制，而不是它们所支持的底层思维过程。

面向对象开发方法的原则是鼓励软件开发者在软件生命周期内应用其概念来工作和思考。只有较好地识别、组织和理解了应用领域的内在概念，才能有效表达出数据结构和函数的细节。

面向对象开发只有到了最后几个阶段才不是独立于编程语言的概念过程。所以可以将面向对象开发看作是一种思维方式，而不是一种编程技术。它的最大好处在于帮助规划人员、开发者和客户清晰地表达抽象的概念，并将这些概念互相传达。它可以充当规约、分析、文档、接口以及编程的一种媒介。

为了加深读者对面向对象开发的理解，下面将它与传统的软件开发作比较。面向对象的开发方法把完整的信息系统看成对象的集合，用这些对象来完成所需要的任务。对象能根据情况执行一定的行为，并且每个对象都有自己的数据。而传统开发方法则把系统看成一些与数据交互的过程，这些数据与过程隔离保存在不同文件中，当程序运行时，就创建或修改数据文件。图 1-1 显示了面向对象开发与传统软件开发之间的区别。

过程通过接收输入的数据，然后对它进行处理，随后保存数据或输出数据。面向对象则是通过接收消息来更新它的内部数据。这些差别虽然看起来简单，但对于整个系统的分析、设计和实现来说却非常重要。

在传统的结构化分析和设计中，开发人员也使用图形模型，如数据流图（DFD）用来表示输入、输出和处理，还要建立实体关系图（ERD）以表示有关存储数据的详细资料。它的设计模型主要由结构图等构成。

而在面向对象开发中，因为需要描述不同的对象，所以面向对象开发中所建立的模型不同于传统的模型。例如，面向对象开发不仅需要用数据和方法来描述建模，还需要用模型来描述对象之间的交互。面向对象开发中使用 UML 来构造模型。

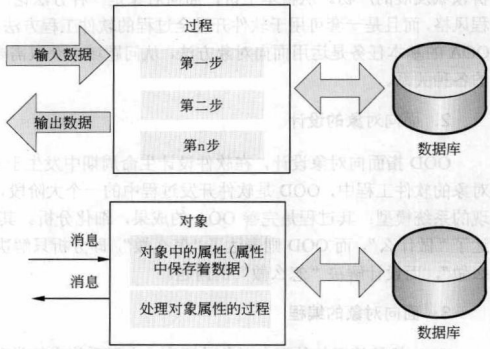


图 1-1 传统方法与面向对象方法的比较

1.1.3 面向对象的主要特性

为了使读者进一步理解面向对象的概念，本节将逐一介绍面向对象的核心特性。

1. 抽象

“物以类聚，人以群分”是指把众多的事物进行归纳和分类，也是人们在认识客观世界时经常采用的思维方法。而在这里分类所依据的原则是抽象。

抽象（Abstract）就是忽略事物中与当前目标无关的非本质特征，更充分地注意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。

例如，在设计一个学生管理系统的过程中以学生李华为例时，就只关心他的学号、班级、成绩等，而忽略他的身高、体重等信息。因此，抽象性是对事物的抽象概括和描述，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，也是面向对象方法的第一步。例如，将学生抽象成对象及类的过程如图 1-2 所示。

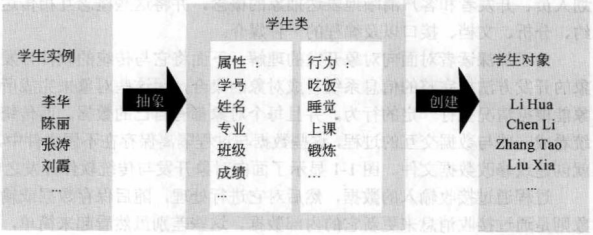


图 1-2 抽象过程示意图

2. 封装

封装（Encapsulation）是指把对象的属性和行为结合成一个独立的单位，并尽可能隐蔽对象的内部细节。例如图 1-2 中的学生类就实现了封装。

通常来说封装有两个含义：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；二是尽可能隐蔽对象的内部细节，对外形成一道屏障，与外部的联系只能通过外部接口实现。

封装的信息隐蔽作用反映了事物的相对独立性，可以只关心它对外所提供的接口，即能做什么，而不注意其内部细节，即怎么提供这些服务。例如，对于一台冰箱，我们不需要知道它具体的实现细节，怎样使用电能控制温度的冷藏与保鲜。我们只要知道，怎么打开冰箱，怎么调整温度，怎样存储食品即可。

3. 继承

客观事物既有共性，也有特性。如果只考虑事物的共性，而不考虑事物的特性，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。运用抽象的原则就是舍弃对象的特性，提取其共性，从而得到适合一个对象集的类。

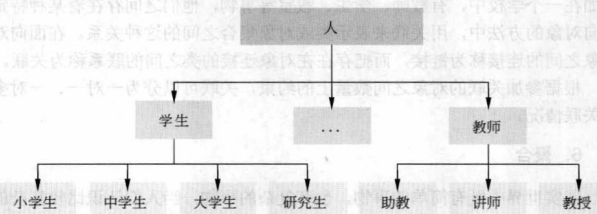


图 1-3 类的继承示意图

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性，则可形成一个新的类。这个新类具有前一个类的全部特征，是前一个类的子集，形成一种层次结构，即继承结构，如图 1-3 所示。

4. 多态

面向对象设计借鉴了客观世界的多态性，体现在不同的对象收到相同的消息时产生多种不同的行为方式。

例如，在一般类“几何图形”中定义了一个行为“绘图”，但并不确定执行时到底画一个什么图形。特殊类“椭圆”和“多边形”都继承了几何图形类的绘图行为，但其功能却不同，一个是要画出一个椭圆，另一个是要画出一个多边形。这样一个绘图的消息发出后，椭圆、多边形等类的对象接收到这个消息后各自执行不同的绘图函数。如图 1-4 所示，这就是多态性的表现。

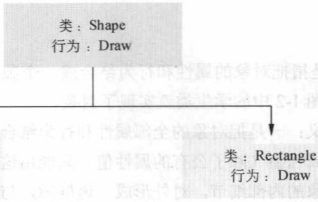


图 1-4　多态性示意图

提　示

继承性和多态性的结合，可以生成一系列虽类似但独一无二的对象。由于继承性，这些对象共享许多相似的特征；由于多态性，针对相同的消息，不同对象可以有独特的表现方式，实现特殊化的设计。

5. 关联

在现实世界中事物不是孤立的、互相无关的，而是彼此之间存在着各种各样的联系。例如在一个学校中，有教师、学生、教室等事物，他们之间存在着某种特定的联系。在面向对象的方法中，用关联来表示类或对象集合之间的这种关系。在面向对象中，常把对象之间的连接称为链接，而把存在在对象连接的类之间的联系称为关联。

根据参加关联的对象之间数量上的约束，关联可以分为一对一、一对多、多对多 3 种关联情况。

6. 聚合

现实世界中既有简单的事物，也有复杂的事物。当人们认识比较复杂的事物时，常用的思维方法为：把复杂的事物分解成若干个比较简单的事物。在面向对象的技术中像这样将一个复杂的对象分解为几个简单对象的方法称为聚合。

聚合是面向对象方法的基本概念之一。它指定了系统的构造原则，即一个复杂的对象可以分解为多个简单对象。同时它也表示为对象之间的关系：一个对象可以是另一个对象的组成部分。同时该对象也可以由其他对象构成。

7. 消息

消息是指对象之间在交互中所传递的通信信息。当系统中的其他对象需要请求该对象执行某个操作时，就向其发送消息，该对象接收消息并完成指定的操作，然后把操作结果返回到请求服务的对象。

一个消息一般应该含有如下信息：接收消息的对象、请求该对象提供的服务、输入信息和响应信息。