

高 等 学 校 计 算 机 课 程 规 划 教 材

UML面向对象需求分析与建模教程

冯洪海 主编

清华大学出版社

高等学校计算机课程规划教材

UML面向对象需求分析与建模教程

冯洪海 主编



清华大学出版社
北京

内 容 简 介

全书共分 17 章,主要讲解面向对象分析与设计及面向过程分析与设计的区别、UML 的基本组成与功能、RUP 统一软件过程、Rational Rose 的安装与使用方法、用例图建模以及活动图、状态图、类图、包图、时序图、协作图、部署图与组件图等的建模。另外还介绍了架构模式,通用职责分配模式、设计原则、设计模式。最后两章是两个简单案例——图书馆管理系统和在线考试系统。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

UML 面向对象需求分析与建模教程/冯洪海主编. —北京:清华大学出版社,2012.12

高等学校计算机课程规划教材

ISBN 978-7-302-30899-7

I. ①U… II. ①冯… III. ①面向对象语言—程序设计—高等学校—教材 IV. ①TP312

中国版本图书馆 CIP 数据核字(2012)第 291667 号

责任编辑:汪汉友

封面设计:傅瑞学

责任校对:李建庄

责任印制:李红英

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 装 者:清华大学印刷厂

经 销:全国新华书店

开 本:185mm×260mm 印 张:16

字 数:401千字

版 次:2012年12月第1版

印 次:2012年12月第1次印刷

印 数:1~3000

定 价:29.00元

产品编号:049799-01

出版说明

信息时代早已显现其诱人魅力,当前几乎每个人随身都携有多个媒体、信息和通信设备,享受其带来的快乐和便捷。

我国高等教育早已进入大众化教育时代,而且计算机技术发展很快,知识更新速度也在快速增长,社会对计算机专业学生的专业能力要求也在不断翻新。这就使得我国目前的计算机教育面临严峻挑战。我们必须更新教育观念——弱化知识培养目的,强化对学生兴趣的培养,加强培养学生理论学习、快速学习的能力,强调培养学生的实践能力、动手能力、研究能力和创新能力。

教育观念的更新,必然导致教材的更新。一流的计算机人才需要一流的名师指导,而一流的名师需要精品教材的辅助,而精品教材也将有助于催生更多一流名师。名师们在长期的一线教学改革实践中,总结出了一整套面向学生的独特的教法、经验、教学内容等。本套丛书的目的就是推广他们的经验,并促使广大教育工作者进一步更新教育观念。

在教育部相关教学指导委员会专家的帮助和指导下,在各大学计算机院系领导的协助下,清华大学出版社规划并出版了本系列教材,以满足计算机课程群建设和课程教学的需要,并将各重点大学的优势专业学科的教育优势充分发挥出来。

本系列教材行文注重趣味性,立足课程改革和教材创新,广纳全国高校计算机专业一线优秀名师参与,从中精选出佳作予以出版。

本系列教材具有以下特点。

1. 有的放矢

针对计算机专业学生并站在计算机课程群建设、技术市场需求、创新人才培养的高度,规划相关课程群内各门课程的教学关系,以达到教学内容互相衔接、补充、相互贯穿和相互促进的目的。各门课程功能定位明确,并去掉课程中相互重复的部分,使学生既能够掌握这些课程的实质部分,又能节约一些课时,为开设社会需求的新技术课程准备条件。

2. 内容趣味性强

按照教学需求组织教学材料,注重教学内容的趣味性,在培养学习观念、学习兴趣的同时,注重创新教育,加强“创新思维”、“创新能力”的培养、训练;强调实践,案例选题注重实际和兴趣度,大部分课程各模块的内容分为基本、加深和拓宽内容3个层次。

3. 名师精品多

广罗名师参与,对于名师精品,予以重点扶持,教辅、教参、教案、PPT、实验大纲和实验指导等配套齐全,资源丰富。同一门课程,不同名师分出多个版本,方便选用。

4. 一线教师亲力

专家咨询指导,一线教师亲力;内容组织以教学需求为线索;注重理论知识学习,注重学

习能力培养,强调案例分析,注重工程技术能力锻炼。

经济要发展,国力要增强,教育必须先行。教育要靠教师和教材,因此建立一支高水平的教材编写队伍是社会发展的需要,特希望有志于教材建设的教师能够加入到本团队。通过本系列教材的辐射,培养一批热心为读者奉献的编写教师团队。

清华大学出版社

前 言

由于计算机领域的有关技术都是人工技术,因此都有为什么设计该项技术的理由(why),以及如何应用该项技术的方法(how),当然还有该项技术的内容(what)。本系列教程就是从这3个方面(3W)去详细阐述。由于一般的教程并不涉及 why 和 how 这两个方面,所以读者并不知道该项技术的由来以及如何应用,而这3个方面结合也就成为该系列教程的特色之一。

1. 为什么写本书

随着专业分工的细化,面向对象分析和设计已经各自成为独立的工种,是事关软件工程专业成败、更具全局性质、更吸引高端人才的工作。目前,我国外包项目众多,其中大部分的分析 and 设计工作由外方完成,而大量费力费时的开发工作由国人来完成。因此,我们国家的软件产业要避免给外国人打工,必须培养更多的熟悉分析和设计人才。而现实是,高校的软件工程和分析设计相关的课程严重与实际脱节。本书吸收大量工作在一线的软件工程师的实践经验,着眼实际工作需要,给读者提供实用的面向对象分析与设计的教程。

2. 本书的特点

- (1) 区别于其他书籍,本书包含大量如何使用 UML 分析设计的实际经验,即 how。
- (2) 区别于其他书籍,本书包含大量为什么设计该项技术的理由,即 why。
- (3) 区别于其他书籍,本书包含大量概念或技术之间关系的阐述。
- (4) 条理清楚。
- (5) 案例较丰富。

3. 主要内容

全书共分 17 章,第 1 章主要讲解面向对象分析与设计与面向过程分析与设计的区别;第 2 章概要讲解 UML 的基本组成与功能;第 3 章讲解 RUP 统一软件过程的概念、9 个核心过程、特点、最佳实践等;第 4 章讲解 Rational Rose 的安装与使用方法;第 5 章讲解用例、用例图、与用例建模,将用例区分为业务用例、概念用例、系统用例;第 6 章讲解活动图;第 7 章讲解状态图;第 8 章讲解类图与包图;第 9 章讲解时序图;第 10 章讲解协作图;第 11 章讲解部署图与组件图;第 12 章讲解架构模式;第 13 章讲解通用职责分配模式;第 14 章讲解设计原则;第 15 章讲解设计模式;第 16 章是简单案例——图书馆管理系统;第 17 章是案例——在线考试系统。

4. 读者对象

本书适合高等学校计算机及相关专业“需求分析与建模”、“面向对象分析与设计”有关课程的教材或参考书;适合从事面向对象分析设计的工作人员学习和参考。

本书由冯洪海主编,另外,参加本书编写的还有王豪爽、杜佳恒、马睿、杨园园、于静洋、姚莉、杨一平、陈小潘、曹志伟、秦勉、刘宇、周星、黄俊辉、陈永、王赵翔、李丽。感谢清华大学

出版社的支持。

书中引用了《Thinking in UML》的一些观点,有兴趣的读者可以参考该书作展开阅读。另外,书中一些资料借鉴于网络,在此感谢那些在网上提供资料的作者。

书中有不少是作者一家之言,对于不妥之处,欢迎批评指正。

本书受到河南大学教材建设基金的资助。

作者

2012年10月

目 录

第 1 章 面向对象分析与设计	1
1.1 认识论	1
1.2 工程方法与软件工程方法	1
1.3 分析与设计的区别	2
1.4 面向对象的设计过程	3
1.5 面向对象分析与面向过程分析的区别	4
第 2 章 UML 简介	6
第 3 章 RUP 统一软件过程	13
3.1 软件过程.....	13
3.2 统一过程.....	13
3.3 RUP 主要特点	14
3.4 RUP 最佳实践	15
3.5 RUP 过程框架模型	15
3.6 开发过程的静态结构.....	16
3.7 9 个核心过程工作流	18
3.8 RUP 的角色	21
3.9 RUP 与软件工程	21
第 4 章 UML 工具 Rational Rose	23
4.1 建模概论.....	23
4.2 Rational Rose 2003 简介	23
4.2.1 Rational Rose 概述	23
4.2.2 Rational Rose 的安装	24
4.2.3 Rational Rose 的使用	29
4.3 Rose 的双向工程	36
4.3.1 双向工程概述	36
4.3.2 正向工程	36
4.3.3 逆向工程	38
第 5 章 用例图	40
5.1 衍生型.....	40
5.2 使用者.....	40

5.3	用例	46
5.3.1	业务用例	47
5.3.2	业务用例实现	48
5.3.3	概念用例	48
5.3.4	系统用例	48
5.3.5	系统用例实现	49
5.3.6	业务用例模型与系统用例模型之间的关系	49
5.3.7	用例的特征	51
5.3.8	用例的优势	52
5.3.9	用例的局限性	52
5.3.10	用例方法与 SRS 方法比较	52
5.3.11	设计用例的标准	53
5.3.12	什么才是有效用例	53
5.3.13	发现用例	53
5.3.14	使用用例的误区	56
5.3.15	用例的适用性	60
5.4	边界	60
5.5	用例之间的关系	61
5.5.1	关联关系	61
5.5.2	包含关系	61
5.5.3	扩展关系	63
5.5.4	用例的纵向方面的关系——泛化	64
5.5.5	关系之间的比较	66
5.6	用例图	68
5.6.1	业务用例视图	69
5.6.2	业务用例实现视图	70
5.6.3	概念用例视图	70
5.6.4	系统用例视图	70
5.6.5	系统用例实现视图	71
5.7	用例描述	71
第 6 章	类图与包图	75
6.1	类图	75
6.2	类图中的类	76
6.2.1	类的表示	76
6.2.2	类的种类	77
6.2.3	类之间的关系	79
6.3	类的衍生型	80
6.3.1	边界类	80

6.3.2	实体类	81
6.3.3	控制类	82
6.4	类图的抽象层次	83
6.4.1	概念层	83
6.4.2	说明层	84
6.4.3	实现层	84
6.5	如何构造类图	85
6.5.1	建立类图的步骤	85
6.5.2	识别、寻找类的方法	85
6.5.3	类图建模技术	87
6.6	包图	87
6.6.1	包	87
6.6.2	分包的原则	88
6.6.3	包的表示法	89
6.6.4	包图	90
6.6.5	UML 工具：从代码逆向工程产生包图	91
6.6.6	使用层进行设计	91
6.6.7	包与子系统、层、组件等关系	91
第 7 章	状态图	93
7.1	举例与图示	93
7.2	状态图的元素	93
7.2.1	状态	93
7.2.2	状态图元素	94
7.3	状态图	97
7.4	实例	101
第 8 章	活动图	103
8.1	举例与图示	103
8.2	状态图元素	103
8.2.1	动作状态	104
8.2.2	活动状态	104
8.2.3	动作流	104
8.2.4	分支与合并	105
8.2.5	分叉与汇合	105
8.2.6	泳道	106
8.2.7	对象流	106
8.2.8	分解	107
8.3	活动图	108

8.4	实例	110
第9章	时序图	112
9.1	交互模型	112
9.2	举例与图示	112
9.3	组成元素	113
9.3.1	对象	114
9.3.2	消息	114
9.3.3	生命线	118
9.3.4	激活(控制焦点)	118
9.3.5	组合片段	119
9.3.6	总结	121
9.4	时序图	123
9.5	实例	129
第10章	协作图	131
10.1	举例与图示	131
10.2	协作图元素	131
10.2.1	消息	131
10.2.2	链	132
10.2.3	对象	132
10.3	协作图	132
10.4	实例	138
10.5	几种建模图形的表达焦点和语言特点	141
10.6	状态图、时序图、合作图和活动图之间的比较	142
第11章	组件图与部署图	143
11.1	组件图	143
11.1.1	举例与图示	143
11.1.2	组件图元素	143
11.1.3	组件图	144
11.2	部署图(配置图)	145
11.2.1	举例与图示	145
11.2.2	组成元素	145
11.2.3	配置图建模技术	147
第12章	架构模式	148
12.1	模式	148
12.2	架构	149

12.3	软件框架	152
12.4	分层模式	153
12.5	管道——过滤器	159
12.6	黑板模式	162
12.7	MVC 模式	163
12.8	客户/服务器	165
12.9	端到端	167
第 13 章 通用职责分配模式		170
13.1	通用职责分配模式概述	170
13.2	信息专家	171
13.3	创建者	171
13.4	高内聚	172
13.5	低耦合	173
13.6	控制者	174
13.7	多态	175
13.8	纯虚构	176
13.9	间接性	177
13.10	变化预防	178
第 14 章 设计原则		180
14.1	面向对象设计原则简介	180
14.2	对象单一职责原则	180
14.3	里氏替换原则	182
14.4	依赖倒置原则	184
14.5	接口隔离原则	187
14.6	迪米特法则	188
14.7	开闭原则	190
第 15 章 设计模式		193
15.1	综述	193
15.2	单例模式	193
15.3	抽象工厂模式	195
15.4	建造者模式	196
15.5	工厂方法模式	197
15.6	原型模式	198
15.7	适配器模式	198
15.8	桥接模式	199
15.9	组合模式	200

15.10	装饰模式	201
15.11	外观模式	202
15.12	享元模式	203
15.13	代理模式	204
15.14	责任链模式	205
15.15	命令模式	206
15.16	解释器模式	207
15.17	迭代器模式	207
15.18	中介者模式	208
15.19	备忘录模式	209
15.20	观察者模式	210
15.21	状态模式	211
15.22	策略模式	212
15.23	模板方法	213
15.24	访问者模式	213
 第 16 章 图书馆管理系统		215
16.1	用例模型——系统需求的获取	215
16.2	分析模型——开发者的视野	217
16.3	系统设计——实现方案	221
 第 17 章 在线考试系统		226
17.1	用例模型图示	226
17.1.1	网上考试管理系统功能概述	226
17.1.2	网上考试管理系统中各种角色的划分	226
17.1.3	考试模块用例图	226
17.1.4	账户管理用例图	227
17.2	分析模型图示	228
17.2.1	管理账户域模型	228
17.2.2	修改账户信息域模型	228
17.2.3	用户登录域模型	229
17.2.4	注册账户域模型	229
17.3	组件图	230
17.3.1	在线考试管理表示层组件图	230
17.3.2	在线考试管理系统业务层组件图	230
17.3.3	在线考试管理系统数据访问层组件图	231
17.3.4	在线考试管理系统组件图	231
17.4	包图	232
17.4.1	采用 Struts+Spring+Hibernate 框架技术的经典系统架构图	232

17.4.2	紧密耦合架构示例图示·····	232
17.4.3	松耦合架构示例图示·····	234
17.5	类图·····	235
17.5.1	在线考试管理系统表示层类图·····	235
17.5.2	在线考试管理系统控制层类图·····	235
17.5.3	在线考试管理系统业务层类图·····	235
17.5.4	在线考试管理系统业务实体类图·····	235
17.5.5	在线考试管理系统数据访问层类图·····	237
17.6	时序图·····	237
17.6.1	在线考试管理系统用户登录时序图·····	237
17.6.2	在线考试管理系统查询题库信息时序图·····	238
17.7	协作图·····	239
17.7.1	在线考试管理系统用户登录协作图·····	239
17.7.2	在线考试管理系统查询题库信息协作图·····	239
17.8	活动图·····	240
17.8.1	在线考试管理系统学生申请考试活动图·····	240
17.8.2	在线考试管理系统教师新建题库活动图·····	241
17.9	状态图·····	241
17.9.1	在线考试管理系统学生状态图·····	241
17.9.2	在线考试管理系统教师状态图·····	242

第 1 章 面向对象分析与设计

1.1 认识论

一直以来,人们认为面向过程和面向对象的区别源自方法论的区别,其实二者的区别首先源自认识论的区别,并且认识论决定方法论。认识论讨论的是世界的本质是由对象组成的还是由过程组成的。面向对象的认识论认为世界是由对象组成的,面向过程的认识论认为世界是由过程组成的。对于如何制造汽车,人们显然按照面向对象方法论去做,即制造出不同的零部件,然后组装在一起,但是对于如何修理汽车,可能很多人会将其认为是方法和步骤的组合或序列,因为此时方法或步骤是重点,而对象是非重点。还有,解决某类问题的方法、舞蹈的编排、绘画的过程、游泳的步骤、维修方法、算法序列等,都是过程步骤是重点,而对象淹没其中。那么,如果由此认为部分世界是由过程组成的,便形成了面向过程的认识论观点。

马克思主义哲学思想认为,世界是物质的,物质是运动的,而没有说世界是由运动组成的。显然,运动(行为、变化)是物质的一个属性(特征、表现)。关于世界、对象、属性、行为的认识观可以总结如下:

- (1) 世界由对象组成,而不是由过程(或方法或行为)组成。
- (2) 对象是独立存在的,行为是对象发出的,不是独立存在的。
- (3) 行为随着对象的属性而变化,如有腿的动物有走的行为,没有腿的动物没有走的行为。
- (4) 对象是客观存在的实体,是具体的;行为是对象随时间的变化过程,也是具体的,是千变万化的。没有脱离对象而存在的具体过程,因此,抛开对象去模拟现实世界的过程无法面对千变万化的对象的行为。

1.2 工程方法与软件工程方法

1. 工程方法

现实中的工程都是一个个实体,如楼房、汽车等,容易理解应该将这些实体分解成一个个标准的部件,然后根据不同要求进行组装、替换等。但是像行为过程、方法序列这些是否也应视作是对象的组合呢?即将其视作行为的组合,还是视作对象(其拥有行为)的组合?很显然,两种都行,但哪一种更合理呢?

2. 软件工程方法

软件工程是一个关于软件开发各阶段的定义、任务、作用,建立在理论上的一门工程学科。它对解决软件危机,指导人们利用科学、有效的方法来开发软件,提高及保证软件开发的效率和质量起到了一定的作用。

经典的软件工程思想将软件开发分成以下 5 个阶段：需求分析(Requirements Capture)阶段、系统分析与设计(System Analysis and Design)阶段、系统实现(Implementation)阶段、测试(Testing)阶段和维护(Maintenance)阶段。

如前所述,除了实体工程外,还有方法步骤之类的系统,如组装、维修、操作、计算等,它们都是行为过程,而且是多个对象参与、互相协作、有因有果的过程。

显然,软件系统是一个计算方法的执行过程。因此,早期人们以为软件系统就是数据结构+算法,认为应该将软件系统视为方法的组合,因此,出现了面向过程即面向函数的编程方法,即整个软件过程由一系列函数互相调用组成。这种方法将数据与行为隔离开,扩展性差。

而面向对象的思维将这类过程型的工程视为对象运动变化的结果,同样由对象组成,但对对象都有运动和变化。当我们将视角针对对象时,发现不但对象可以替换,方法也可以替换。

比如,将大象装进冰箱需要 3 步:

- (1) 打开冰箱门。
- (2) 大象进入冰箱。
- (3) 关上冰箱门。

如果采用面向过程(函数)思维,那么这 3 步就是 3 个行为“打开”、“进入”和“关上”的组合。

如果是采用面向对象思维,便是两个对象的行为的组合,即冰箱、大象的行为的组合。即“冰箱打开”、“大象进入”、“冰箱关闭”。可以替换冰箱为其他的容器,也可以替换大象以装进其他的物体,同样可以替换不同的冰箱打开和关闭的方式,如上下推拉式、左右推拉式、左右扇开式等;也可以替换不同的进入方式,如走进、爬进等。显然面向对象的思维更能适应外界的变化。

面向过程编程思想将行为(软件中的函数)从行为的主体(软件中的对象)中剥离出来,然后将这些行为连接成一个整体。而现实中的行为变化多样,连接成一体的行为很难适应变化。而现实是行为并不能脱离主体而存在,主体的行为根据主体的属性不同而不同,而且,主体组合成了现实世界,人们可以根据需要变更主体以适应变化。

认识论上面向对象和面向过程的差别,导致了方法论上软件分析设计方法的革命。分析设计从过程分析变成了对象获取,从数据结构变成了对象结构。

1.3 分析与设计的区别

分析模型和设计模型最早在 Jacobson 的 OOSE 中就提出来了,不是在 RUP 中首先提出的。

这两个模型的本质差别在于,分析模型是对问题域的描述,是独立于实现技术的。设计模型是使用具体的实现技术实现分析模型,是依赖于实现技术的。

面向对象分析包括哪些活动?应该建立哪些类型的模型?

面向对象分析的主要活动包括理解用例模型、识别分析类、定义交互行为、建立分析类图以及评审分析模型等。

面向对象分析应该建立功能模型、分析对象模型和动态模型 3 种类型,其中功能模型由用例和场景表示,分析对象模型由类图和对象图表示,动态模型由状态图和时序图表示。具体过程概括如下:理解用例模型,识别分析类,定义交互行为,建立分析类图,评审分析模型,如图 1-1 所示。

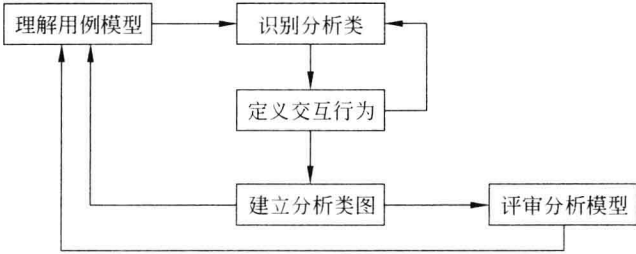


图 1-1 面向对象分析过程

1.4 面向对象的设计过程

面向对象的设计过程涉及几个本源问题。

(1) 什么样的设计结果才会满足需求?

比如设计哪些类或方法,是依靠经验还是理论确定? 经验不足的人在面对复杂的需求时,一般尝试着设计出几个类来,发现解决不了问题时,再重新设计……许多项目就这样在试错中不断变更、拼凑。

(2) 如何证明设计结果满足了需求?

比如如何证明设计出的一些类已经满足了需求,而不是等到项目后期通过不断测试来证明? 如果后期测试后不能满足需求,往往要投入大量的返工成本。

很多设计师不知道如何去验证自己的设计能否满足实际的需求,而是寄希望于通过设计模式,设计柔性的系统,来应付将来需求的变化。

(3) 设计结果如何让开发人员理解? 如何与开发人员交流?

许多程序员并不能理解设计师的成果及系统分析员编写的需求说明书之间的关系,不知道二者是如何映射的,设计出的类为什么是这些而不是那些。

之所以有这些困惑,是因为现实世界和软件世界之间存在着一道鸿沟,通过将现实世界抽象可以得到软件系统。抽象是面向对象的精髓,同时也是面向对象的困难所在。要想跨越这道鸿沟,需要通过以下方法:

- (1) 把现实世界映射到对象世界的方法。
- (2) 用对象世界描述现实世界的方法。
- (3) 验证对象世界行为是否正确反映了现实世界的方法。

面向对象的分析、设计方法正好架起了跨越这道鸿沟的桥梁。UML 可以用来解决这些问题。