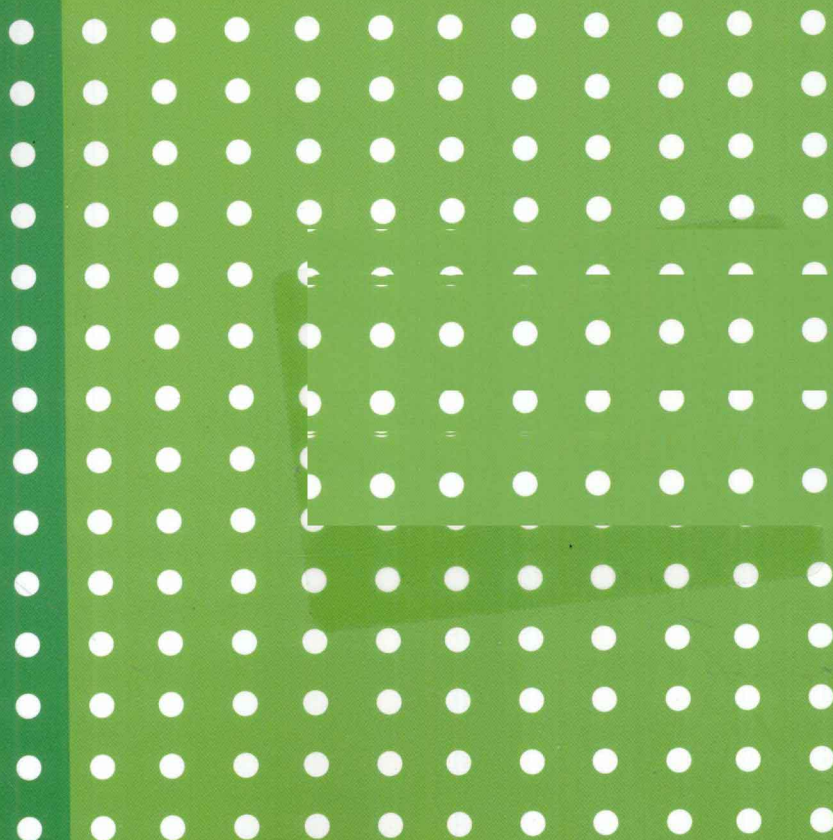


重点大学软件工程规划系列教材

设计模式实训教程

刘 伟 编著

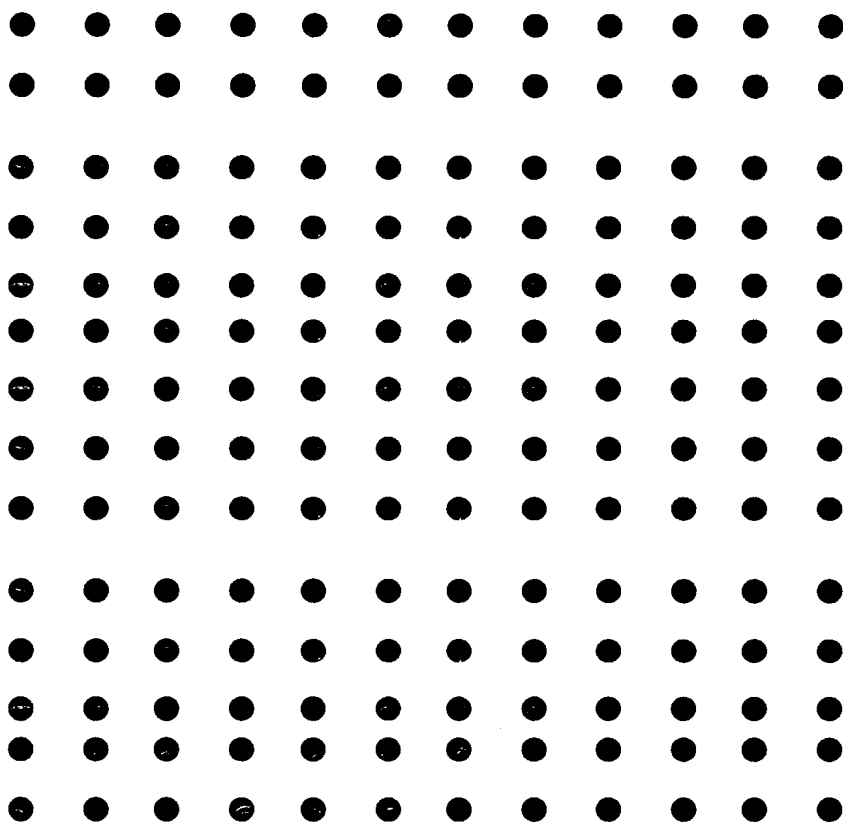


清华大学出版社

重点大学软件工程规划系列教材

设计模式实训教程

刘伟 编著



清华大学出版社
北京

内 容 简 介

本书通过大量项目实例让读者加深对 GoF 设计模式的理解,在学习模式的同时掌握如何在实际软件开发中运用模式,并通过大量练习来强化对设计模式的理解和掌握。

本书共分 6 章,内容包括 UML 类图实训,面向对象设计原则实训,创建型模式实训,结构型模式实训,行为型模式实训。从第 3 章到第 5 章,结合实例和大量实训练习学习如何在项目开发中使用设计模式;第 6 章对设计模式的相关知识进行补充,提供了 6 个模式联用解决方案,通过两个综合实例学习如何在应用开发中使用设计模式,同时提供了一些企业招聘过程中出现的面试和笔试试题并给出参考答案。

本书既可作为各类高等院校计算机和软件相关专业本专科生和研究生软件设计模式、软件体系结构等课程教学辅导用书和教师参考用书,也可作为全国计算机技术与软件专业技术资格(水平)考试辅导用书和软件架构师、软件工程师等开发人员的参考用书,还可以作为设计模式爱好者和自学者的习题集和实训教程以及就业之前的复习用书。

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

设计模式实训教程/刘伟编著. --北京:清华大学出版社,2012.1

(重点大学软件工程规划系列教材)

ISBN 978-7-302-27067-6

I. ①设… II. ①刘… III. ①面向对象语言—程序设计 IV. ①TP312

中国版本图书馆 CIP 数据核字(2011)第 205373 号

责任编辑:魏江江 顾 冰

责任校对:时翠兰

责任印制:杨 艳

出版发行:清华大学出版社

地 址:北京清华大学学研大厦 A 座

<http://www.tup.com.cn>

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 刷 者:北京富博印刷有限公司

装 订 者:北京市密云县京文制本装订厂

经 销:全国新华书店

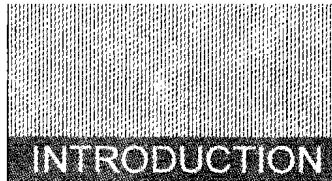
开 本:185×260 印 张:21 字 数:508 千字

版 次:2012 年 1 月第 1 版 印 次:2012 年 1 月第 1 次印刷

印 数:1~2500

定 价:34.00 元

产品编号:042974-01



出版说明

随着信息时代的来临,软件已被广泛应用到工业、农业、商业、金融、科教卫生、国防、航空等各个领域,成为国民经济和社会信息化的一个基础性、战略性产业。因此,与之相关联的软件工程专业也越来越受到社会的关注。

从国际范围来看,1996年,美国Rochester技术大学(RIT)率先设立软件工程专业,其后美国、加拿大、英国和澳大利亚的许多大学相继跟进。1998年,ACM和IEEE-CS两大计算机学会联合设立软件工程教育项目(SWEEP),研究软件工程课程设置。2001年,IEEE和ACM发布CC2001教程,将计算(computing)学科划分为计算机科学、计算机工程、软件工程、信息系统和信息技术五个二级学科。2003年6月,《计算机课程——软件工程》(CCSE)大纲第一稿发表,后正式更名为《软件工程2004教程》(SE2004)。

在我国,教育部十分重视软件工程专业的发展。2001年,教育部和原国家计委联合下文,成立了35所示范性软件学院(全部下设于重点大学);2005年5月,教育部和清华大学出版社联合立项支持的研究课题组发布《中国软件工程学科教程》;同年,教育部组织编写了《软件工程专业规范》;2006年3月,在教育部高等学校教学指导委员会成立大会上,宣布成立软件工程专业教学指导分委员会。截至2007年初,全国有139所高等院校设立了软件工程专业。显然,软件工程已经成为一门迅速兴起的独立学科。

从我国的国民经济和社会发展来看,软件人才的需求非常迫切。随着国家信息化步伐的加快和我国高等教育规模的扩大,软件人才的培养不仅在数量的增加上也在质量的提高上对目前的软件工程专业教育提出更为迫切的要求,社会需要软件工程专业的教学内容的更新周期越来越短,相应地,我国的软件工程专业教育在不断地发展和改革,而改革的目标和重点在于培养适应社会经济发展需要的、兼具研究能力和工程能力的高质量专业软件人才。

截至2007年,我国共有72个国家一级重点学科,绝大部分设置在教育部直属重点大学。重点大学的软件工程学科水平与科研氛围是培养一流软件人才的基础,而一流的学科专业教材的建设已成为目前重点大学学科建设的重要组成部分,一批具有学科方向特色优势的软件工程教材作为院校的重点建设项目成果得到肯定。清华大学出

版社一向秉承清华的“中西兼容、古今贯通的治学主张,自强不息、厚德载物的人文精神,严谨勤奋、求实创新的优良学风”。在教育部相关教学指导委员会专家的指导和建议下,在国内许多重点大学的院系领导的大力支持下,清华大学出版社规划并出版本系列教材,以满足软件工程学科专业课程教学的需要,配合全国重点大学的软件工程学科建设,旨在将这些专业教育的优势得以充分的发扬,强调知识、能力与素质的系统体现,通过这套教材达到“汇聚学科精英、引领学科建设、培育专业英才”的目的。

本系列教材是在软件工程专业学科课程体系建设基本成熟的基础上总结、完善而成,力求充分体现科学性、先进性、工程性。根据几年来软件工程学科的发展与专业教育水平的稳步提高,经过认真的市场调研并参考教育部立项课题组的研究报告《中国软件工程学科教程》,我们初步确定了系列教材的总体框架,原则是突出专业核心课程的教材,兼顾具有专业教学特点的相关基础课程教材,探索具有发展潜力的新的专业课程教材。

本系列教材在规划过程中体现了如下一些基本组织原则和特点。

一、体现软件工程学科的发展和专业教育的改革,适应社会对现代软件工程人才的培养需求,教材内容坚持基本理论的扎实和清晰,反映基本理论和原理的综合应用,在其基础上强调工程实践环节,并及时反映教学体系的调整 and 教学内容的更新。

二、反映教学需要,促进教学发展。教材规划以新的专业目录为依据。教材要适应多样化的教学需要,正确把握教学内容和课程体系的改革方向,在选择教材内容和编写体系时注意体现素质教育、创新能力与实践能力的培养,为学生知识、能力、素质协调发展创造条件。

三、实施精品战略,突出重点。规划教材建设仍然把重点放在专业核心(基础)课程的教材建设;特别注意选择并安排了一部分原来基础较好的优秀教材或讲义修订再版,逐步形成精品教材;提倡并鼓励编写体现工程型和应用型的专业教学内容和课程体系改革成果的教材。

四、支持一纲多本,合理配套。专业核心课和相关基础课的教材要配套,同一门课程可以有多种具有不同内容特点的教材。处理好教材统一性与多样化,基本教材与辅助教材、教学参考书,文字教材与软件教材的关系,实现教材系列资源的配套。

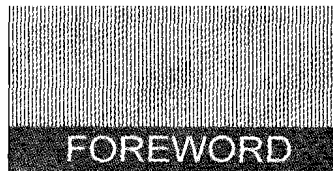
五、依靠专家,择优落实。在制订教材规划时依靠各课程专家在调查研究本课程教材建设现状的基础上提出规划选题。在落实主编人选时,要引入竞争机制,通过申报、评审确定主编。

六、严格把关,质量为重。实行主编责任制,参与编写人员在编写工作实施前经过认真研讨确定大纲和编写体例,以保证本系列教材在整体上的技术领先与科学、规范。书稿完成后认真实行审稿程序,确保出书质量。

繁荣教材出版事业、提高教材质量的关键是教师。建立一支高水平的、以老带新的教材编写队伍才能保证教材的编写质量,希望有志于教材的教师能够加入到我们的编写队伍中来。

“重点大学软件工程规划系列教材”丛书编委会

联系人: 付弘宇 fuhy@tup.tsinghua.edu.cn



前 言

随着面向对象技术的发展和广泛应用,设计模式已成为面向对象开发人员的必备技能之一。无论是面向对象的初学者还是具有一定开发经验的程序员,都可以通过对设计模式的学习和应用加深对面向对象思想的理解,开发出具有更好的可扩展性和复用性的软件。笔者在多年的面向对象教学和实践开发中也深刻体会到学习设计模式的意义,在教授 C++、Java、C# 等课程的同时,结合一些常用的设计模式可以让学生更好地理解面向对象的特性、抽象层的作用、合成复用的优点等原本很抽象、较难理解思想和概念。正如笔者经常和学生以及学员们说的:掌握设计模式后,就会发现面向对象设计是一门艺术,就会发现有些模式是如此巧妙,就一定受益于 GoF 所做的工作。

当前,在很多高校的软件工程专业的本科或研究生培养方案中都设置了面向对象分析与设计、软件设计模式、软件体系结构等课程,不少企业也开始注重对员工面向对象编程思想和设计模式等的培训,部分软件培训机构也将设计模式作为软件工程师培训课程的基本内容之一。在我国较为权威的全国计算机技术与软件专业技术资格(水平)考试的系统架构设计师(高级)、软件设计师(中级)等级别的考试中,关于设计模式的试题也占据一定的比例,近几年软件设计师考试题中固定有一道 15 分的设计模式大题,2009 年开考的系统架构设计师考试也有不少设计模式相关试题,笔者也有幸以湖南省第一、全国第四的成绩成为第一批国家认证系统架构设计师。

近年来,笔者一直承担中南大学软件学院 Java 实训、软件体系结构、设计模式等课程的教学任务,同时也作为长沙创智新程教育技术有限公司的软件工程专业课顾问承担设计模式、系统分析和设计等课程的教学和教研工作,也给一些软件企业提供 UML 和设计模式等课程的企业内训,并主持和参与一些软件项目的开发工作,从这些教学和开发工作中,笔者发现通过实例,尤其是结合软件项目实例是学习和掌握设计模式的最佳途径,而目前已出版的很多设计模式书籍大多通过一些生活实例来引入和学习设计模式,虽然可以让读者很轻松地学习,但很难做到深入理解和熟练运用,此外,部分教材缺少相应的练习来加深对所学模式的理解和掌握,有的书中虽有一些练习,但数量不多且针对性不强。因此,笔者一直想将这些年积累下来的一些实例整理成册,于是本书诞生了。

1. 本书特色

本书是国内第一本设计模式实训教程和习题集,通过大量项目实例让读者加深对 GoF 设计模式的理解,在学习模式的同时掌握如何在实际软件开发中运用模式,并通过大量练习来强化对设计模式的理解和掌握。笔者整理了这些年在设计模式教学和企业项目开发经验中积累的设计模式实训素材,同时参考了大量已有的设计模式书籍和网站,广泛收集各类设计模式实例和试题,包括历年全国计算机技术与软件专业技术资格(水平)考试试题、知名软件公司招聘面试和笔试题、国内外高校设计模式课程考试试题等,同时结合实际项目设计了大量练习题,包括选择题、模式代码填空题、综合分析题等多种题型,让读者在学习设计模式之余检验学习效果并结合实例来巩固所学知识。收集和整理的过程虽然很辛苦,也很耗时,但若能为我国软件事业的发展和面向对象技术的推广尽一份绵薄之力,所有付出都是值得的。

为了让设计模式的初学者也能够看懂本教程,在本书中的每一章前面都包含了“知识讲解”单元,让读者可以较快了解模式的基本知识,再结合后续实例进行深入学习。当然,如果配有一本专门的设计模式教材来学习则效果更好,本书也是笔者另外一本设计模式书籍《设计模式》的配套习题集,部分实例来自《设计模式》每一章后的思考和练习题。如果在学习完《设计模式》中的每一个模式之后,再通过本书的练习来进一步巩固,笔者相信会取得更好的学习效果。在本书中,针对每一个模式都提供了一个完整的实例,包括 UML 类图、源代码和实例分析,且每个模式都对应多道选择题、一道代码填空题和一道综合分析题,所有练习都提供了参考答案,部分综合分析题还提供了完整代码,本书所有类图均严格按照 UML 2.0 标准绘制,所有代码均在 JDK 1.6 环境下通过测试且运行无误。

2. 本书内容

本书包括 6 章:第 1 章介绍 UML 类图并通过实训让读者掌握如何阅读和绘制类图,学习使用类图来构造软件的静态模型;第 2 章介绍 7 个常用的面向对象设计原则,结合实例学习如何使用这些原则对系统进行重构;第 3 章介绍 6 个创建型设计模式,第 4 章介绍 7 个结构型设计模式,第 5 章介绍 11 个行为型设计模式,从第 3 章到第 5 章,结合实例和大量实训练习来进行学习;第 6 章对设计模式的相关知识进行补充,提供了 6 个模式联用解决方案,并通过两个综合实例来学习如何在应用开发中使用设计模式,同时提供了一些企业招聘过程中出现的面试和笔试试题并给出参考答案。本书所有源代码和类图都可通过清华大学出版社网站(<http://www.tup.tsinghua.edu.cn>)下载。

3. 目标读者

本书既可作为各类高等院校计算机和软件相关专业本专科生和研究生软件设计模式、软件体系结构等课程教学辅导用书、实验课教材和教师参考用书,也可作为全国计算机技术与软件专业技术资格(水平)考试辅导用书和软件架构师、软件工程师等开发人员的参考用书,还可以作为企业内训、设计模式爱好者和自学者的习题集和实训教程以及就业之前的复习用书。

4. 感谢

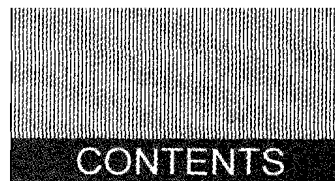
由于本书是一本实训教程和实战手册,涉及大量的实例及分析,这些实例的收集、整理和设计工作离不开众多同事和学生的帮助,在此对他们一并表示感谢。感谢中南大学软件

学院胡志刚教授在百忙之中抽出宝贵时间对本书进行细致的审校,感谢中南大学软件学院江林、杨柳和中南大学信息科学与工程学院郭克华、王斌等在本书写作过程中提出的宝贵意见和建议,感谢长沙创智新程教育技术有限公司任波、夏绪武、赵璟等参与部分内容的校稿工作,感谢唐磊、段玉冰、马元文、李辉、袁磊、谭华、雷松等学生发现本书中的文字纰漏和错误,感谢湖南科创信息技术股份有限公司彭宇波、刘剑峰等员工提供的一些企业应用实例,感谢我的爱人夏莉,她一直在我背后默默地付出,感谢我的父母和家人,他们给了我极大的支持。在编写过程中我参考和引用了国内外很多书籍和网站的相关内容,个别实例和练习的初始原型也来源于网络,由于涉及的网站和网页太多,没有一一列举,在此一并予以感谢。最后特别感谢清华大学出版社为本书出版所作出的努力。

由于时间仓促、学识有限,书中不足和疏漏之处难免,恳请广大读者将意见和建议通过清华大学出版社反馈给我,以便在后续版本中不断改进和完善。

刘 伟

2011 年 11 月 2 日于长沙岳麓山下



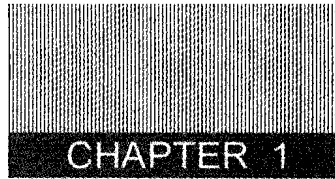
目 录

第 1 章 UML 类图实训	1
1.1 知识讲解	1
1.1.1 UML 概述	1
1.1.2 类与类的 UML 表示	2
1.1.3 类之间的关系	4
1.2 实训实例	8
1.2.1 类图实例之图书管理系统	8
1.2.2 类图实例之商场会员管理系统	10
1.3 实训练习	11
第 2 章 面向对象设计原则实训	18
2.1 知识讲解	18
2.1.1 面向对象设计原则概述	18
2.1.2 单一职责原则	18
2.1.3 开闭原则	19
2.1.4 里氏代换原则	19
2.1.5 依赖倒转原则	20
2.1.6 接口隔离原则	20
2.1.7 合成复用原则	21
2.1.8 迪米特法则	21
2.2 实训实例	22
2.2.1 单一职责原则实例分析	22
2.2.2 开闭原则实例分析	23
2.2.3 里氏代换原则实例分析	24
2.2.4 依赖倒转原则实例分析	25
2.2.5 接口隔离原则实例分析	26
2.2.6 合成复用原则实例分析	27
2.2.7 迪米特法则实例分析	28
2.3 实训练习	30

第 3 章 创建型模式实训	37
3.1 知识讲解	37
3.1.1 设计模式	37
3.1.2 创建型模式概述	39
3.1.3 简单工厂模式	40
3.1.4 工厂方法模式	41
3.1.5 抽象工厂模式	41
3.1.6 建造者模式	43
3.1.7 原型模式	43
3.1.8 单例模式	44
3.2 实训实例	45
3.2.1 简单工厂模式实例之图形工厂	45
3.2.2 工厂方法模式实例之日志记录器	48
3.2.3 抽象工厂模式实例之数据库操作工厂	50
3.2.4 建造者模式实例之游戏人物角色	53
3.2.5 原型模式实例之快速创建工作周报	58
3.2.6 单例模式实例之多文档窗口	60
3.3 实训练习	63
第 4 章 结构型模式实训	81
4.1 知识讲解	81
4.1.1 结构型模式概述	81
4.1.2 适配器模式	82
4.1.3 桥接模式	83
4.1.4 组合模式	84
4.1.5 装饰模式	86
4.1.6 外观模式	87
4.1.7 享元模式	88
4.1.8 代理模式	89
4.2 实训实例	90
4.2.1 适配器模式实例之算法适配	90
4.2.2 桥接模式实例之跨平台视频播放器	94
4.2.3 组合模式实例之杀毒软件	97
4.2.4 装饰模式实例之界面显示构件库	100
4.2.5 外观模式实例之文件加密	103
4.2.6 享元模式实例之围棋棋子	106
4.2.7 代理模式实例之日志记录代理	109
4.3 实训练习	111

第 5 章 行为型模式实训	132
5.1 知识讲解	132
5.1.1 行为型模式概述	132
5.1.2 职责链模式	133
5.1.3 命令模式	134
5.1.4 解释器模式	135
5.1.5 迭代器模式	136
5.1.6 中介者模式	137
5.1.7 备忘录模式	138
5.1.8 观察者模式	139
5.1.9 状态模式	140
5.1.10 策略模式	141
5.1.11 模板方法模式	142
5.1.12 访问者模式	143
5.2 实训实例	145
5.2.1 职责链模式实例之在线文档帮助系统	145
5.2.2 命令模式实例之公告板系统	147
5.2.3 解释器模式实例之机器人控制程序	152
5.2.4 迭代器模式实例之商品名称遍历	156
5.2.5 中介者模式实例之温度转换器	160
5.2.6 备忘录模式实例之游戏恢复点设置	166
5.2.7 观察者模式实例之股票变化	169
5.2.8 状态模式实例之银行账户	172
5.2.9 策略模式实例之电影票打折	177
5.2.10 模板方法模式实例之数据库操作	179
5.2.11 访问者模式实例之奖励审批	182
5.3 实训练习	186
第 6 章 模式联用与综合实例实训	219
6.1 设计模式补充知识	219
6.1.1 反射与配置文件	219
6.1.2 GRASP 模式	223
6.1.3 架构模式与 MVC	226
6.2 模式联用实训	227
6.2.1 适配器模式与桥接模式联用	228
6.2.2 组合模式与命令模式联用	228
6.2.3 外观模式与单例模式联用	229
6.2.4 原型模式与备忘录模式联用	229

6.2.5 观察者模式与组合模式联用.....	231
6.2.6 访问者模式、组合模式与迭代器模式联用	231
6.3 综合实例实训	232
6.3.1 多人联机射击游戏.....	232
6.3.2 数据库同步系统.....	239
6.4 实训练习	248
附录A 参考答案	252
A.1 第1章实训练习参考答案	252
A.2 第2章实训练习参考答案	255
A.3 第3章实训练习参考答案	260
A.4 第4章实训练习参考答案	266
A.5 第5章实训练习参考答案	276
A.6 第6章实训练习参考答案	301
参考文献.....	321



第 1 章

UML类图实训

UML(Unified Modeling Language,统一建模语言)是当前软件系统建模的标准语言,它融合了众多软件建模技术的优点,通过一系列标准的图形符号来描述系统。在 UML 2.0 的 13 种图形中,类图是使用最广泛的图形之一,它用于描述系统中所包含的类以及它们之间的相互关系,每一个设计模式的结构都可以使用类图来表示,它帮助人们简化对系统的理解,是系统分析和设计阶段的重要产物,也是系统编码和测试的重要模型依据。

1.1 知识讲解

UML 已经成为面向对象软件分析与设计的标准建模语言,其应用越来越广泛。在设计模式的学习和使用过程中也需要掌握一些 UML 相关技术,尤其是 UML 类图。通过类图,可以更好地理解每一个模式的结构并对每一个模式实例进行分析。

1.1.1 UML 概述

20 世纪 80 年代至 90 年代,面向对象分析和设计方法发展迅速,随着面向对象技术的广泛应用,其相关研究也十分活跃,涌现了大量的方法和技术,据不完全统计,最多的时候高达 50 多种,其中最具代表性的当属 Grady Booch 的 Booch 方法、Jim Rumbaugh 的 OMT(Object Modeling Technology,对象建模技术)和 Ivar Jacobson 的 OOSE(Object Oriented Software Engineering,面向对象软件工程)等,而 UML 正是在这三位大师的联手之下共同打造而成的,现在它已经成为面向对象软件分析与设计建模的标准。

UML 是通用可视化建模语言,不同于编程语言,它通过一些标准的图形符号和文字来对系统进行建模,用于对软件进行描述和可视化处理、构造和建立软件系统制品的文档。UML 适用于各种软件开发方法、软件生命周期的各个阶段、各种应用领域以及各种软件开发工具,是一种总结了以往建模技术的经验并吸收了当今最优秀成果的

标准建模方法。

UML 主要包括以下 4 个组成部分。

(1) 视图(View): UML 视图用于从不同的角度来表示待建模系统。视图是由许多图形组成的一个抽象集合。在建立一个系统模型时,只有通过定义多个视图,每个视图显示该系统的一个特定方面,才能构造出该系统的完整蓝图,视图也将建模语言链接到开发所选择的方法和过程。UML 视图包括用户视图、结构视图、行为视图、实现视图和环境视图。其中用户视图以用户的观点表示系统的目标,它是所有视图的核心,用于描述系统的需求;结构视图表示系统的静态行为,描述系统的静态元素(如包、类与对象),以及它们之间的关系;行为视图表示系统的动态行为,描述系统的组成元素(如对象)在系统运行时的交互关系;实现视图表示系统中逻辑元素的分布,描述系统中物理文件以及它们之间的关系;环境视图表示系统中物理元素的分布,描述系统中硬件设备以及它们之间的关系。

(2) 图(Diagram): UML 图是描述 UML 视图内容的图形。UML 1.0 提供了 9 种图, UML 2.0 提供了 13 种图,分别是用例图(Use Case Diagram)、类图(Class Diagram)、对象图(Object Diagram)、包图(Package Diagram)、组合结构图(Composite Structure Diagram)、状态图(State Diagram)、活动图(Activity Diagram)、顺序图(Sequence Diagram)、通信图(Communication Diagram)、定时图(Timing Diagram)、交互概览图(Interaction Overview Diagram)、组件图(Component Diagram)和部署图(Deployment Diagram),通过它们之间的相互组合可提供待建模系统的所有视图。其中,用例图对应用户视图;类图、对象图、包图和组合结构图对应结构视图;状态图、活动图、顺序图、通信图、定时图和交互概览图对应行为视图;组件图对应实现视图;部署图对应环境视图。

(3) 模型元素(Model Element): 模型元素是指 UML 图中所使用的一些概念,它们对应于普通的面向对象概念,如类、对象、消息以及这些概念之间的关系,如关联关系、依赖关系、泛化关系等。同一个模型元素可以在多个不同的 UML 图中使用,但是,无论在哪个图中,同一个模型元素都必须保持相同的意义并具有相同符号。

(4) 通用机制(General Mechanism): UML 提供的通用机制为模型元素提供额外的注释、信息和语义,这些通用机制也提供了扩展机制,允许用户对 UML 进行扩展,如定义新的建模元素、扩展原有元素的语义、添加新的特殊信息来扩展模型元素的规则说明等,以便适用于一个特定的方法、过程、组织或用户。

1.1.2 类与类的 UML 表示

类图是使用频率最高的 UML 图之一。在设计模式中,可以使用类图来描述一个模式的结构并对每一个模式实例进行分析和解释。

1. 类

类(Class)封装了信息 and 行为,是面向对象的重要组成部分,它是具有相同属性、操作、关系对象集合的总称。在系统中,每个类都应该具有一定的职责,职责是指类所担任的任务。一个类可以有多种职责,设计得好的类一般至少有一种职责,在定义类时,将类的职责分解为类的属性和操作,其中属性用于封装数据,操作用于封装行为。设计类是面向对象设计中最重要的重要组成部分,也是最复杂和最耗时的部分。

在软件系统运行时,类将被实例化成对象(Object),对象对应于某个具体的事物。类是

对一组具有相同属性、表现相同行为的对象的抽象,对象是类的实例(Instance)。

类图(Class Diagram)使用出现在系统中的不同类来描述系统的静态结构,类图用来描述不同的类和它们之间的关系。

在系统分析阶段,类通常可以分为三种,分别是实体类(Entity Class)、控制类(Control Class)和边界类(Boundary Class),下面对这三种类加以简要说明。

(1) 实体类:实体类对应系统需求中的每个实体,它们通常需要保存在永久存储体中,一般使用数据库表或文件来记录,实体类既包括存储和传递数据的类,还包括操作数据的类。实体类来源于需求说明中的名词,如学生、商品等。

(2) 控制类:控制类用于体现应用程序的执行逻辑,提供相应的业务操作,将控制类抽象出来可以降低界面和数据库之间的耦合度。控制类一般是由动宾结构的短语(动词+名词)转化来的名词,如增加商品对应有一个商品增加类,注册对应有一个用户注册类等。

(3) 边界类:边界类用于对外部用户与系统之间的交互对象进行抽象,主要包括界面类,如对话框、窗口、菜单等。

在面向对象分析和设计的初级阶段,通常首先识别出实体类,绘制初始类图,此时的类图也可称为领域模型,包括实体类以及它们之间的相互关系。

2. 类的 UML 表示

在 UML 中,类使用具有类名称、属性、操作分隔的长方形来表示,如定义一个类 Employee,它包含属性 name、age 和 email,以及操作 modifyInfo(),在 UML 类图中该类如图 1-1 所示。

在 UML 类图中,类一般由三部分组成。

(1) 第一部分是类名:每个类都必须有一个名字,类名是一个字符串。

(2) 第二部分是类的属性(Attributes):属性是指类的性质,即类的成员变量。类可以有任意多个属性,也可以没有属性。

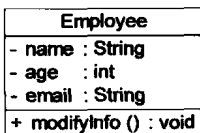


图 1-1 类的 UML 图示

UML 规定属性的表示方式为:

可见性 名称:类型 [= 默认值]

其中:

① “可见性”表示该属性对于类外的元素是否可见,包括公有(public)、私有(private)和受保护(protected)三种,在类图中分别用符号+、-和#表示。

② “名称”表示属性名,用一个字符串表示。

③ “类型”表示属性的数据类型,可以是基本数据类型,也可以是用户自定义类型。

④ “默认值”是一个可选项,即属性的初始值。

(3) 第三部分是类的操作(Operations):操作是类的任意一个实例对象都可以使用的行为,操作是类的成员方法。

UML 规定操作的表示方式为:

可见性 名称(参数列表):返回类型

其中:

- ① “可见性”的定义与属性的可见性定义相同。
- ② “名称”即操作名或方法名,用一个字符串表示。
- ③ “参数列表”表示操作的参数,其语法与属性的定义相似,参数个数是任意的,多个参数之间用逗号“,”隔开。
- ④ “返回类型”表示操作的返回值类型,依赖于具体的编程语言,可以是基本数据类型,也可以是用户自定义类型,还可以是空类型(void),如果是构造方法,则无返回类型。

1.1.3 类之间的关系

在软件系统中,类并不是孤立存在的,类与类之间存在各种关系,对于不同类型的关系,UML 提供了不同的表示方式。

1. 关联关系

关联关系(Association)是类与类之间最常用的一种关系,它是一种结构化关系,用于表示一类对象与另一类对象之间有联系,如汽车和轮胎、师傅和徒弟、班级和学生等。在 UML 类图中,用实线连接有关联关系的对象所对应的类,在使用 Java、C# 和 C++ 等编程语言实现关联关系时,通常将一个类的对象作为另一个类的属性。在使用类图表示关联关系时可以在关联线上标注角色名,一般使用一个表示两者之间关系的动词或者名词表示角色名(有时该名词为实例对象名),关系的两端代表两种不同的角色,因此在一个关联关系中 can 包含两个角色名,角色名不是必需的,可以根据需要增加,其目的是使类之间的关系更加明确。

如在一个登录界面类 LoginForm 中包含一个 JButton 类型的注册按钮 loginButton,它们之间可以表示为关联关系,代码实现时可以在 LoginForm 中定义一个名为 loginButton 的属性对象,其类型为 JButton,如图 1-2 所示。

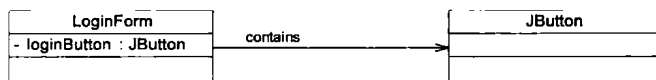


图 1-2 关联关系实例

在 UML 中,关联关系包含如下几种形式。

(1) 双向关联

默认情况下,关联是双向的。例如,顾客(Customer)购买商品(Product)并拥有商品,反之,卖出的商品总有某个顾客与之相关联。因此,Customer 类和 Product 类之间具有双向关联关系,如图 1-3 所示。

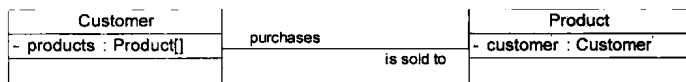


图 1-3 双向关联实例

(2) 单向关联

类的关联关系也可以是单向的,单向关联用带箭头的实线表示。例如,顾客(Customer)拥有地址(Address),则 Customer 类与 Address 类具有单向关联关系,如图 1-4 所示。

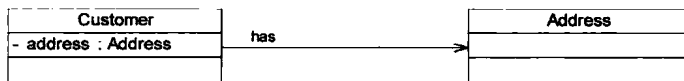


图 1-4 单向关联实例

(3) 自关联

在系统中可能会存在一些类的属性对象类型为该类本身,这种特殊的关联关系称为自关联。例如,一个节点类(Node)的成员又是 Node 类型的对象,如图 1-5 所示。

(4) 多重性关联

多重性关联关系又称为重数性关联关系(Multiplicity),表示两个关联对象在数量上的对应关系。在 UML 中,多重性关联可以直接在关联直线上用一个数字或一个数字范围表示。

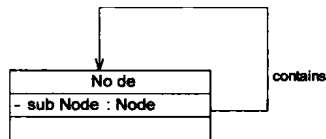


图 1-5 自关联实例

类的对象之间存在多种多重性关联关系,常见的多重性表示方式如表 1-1 所示。

表 1-1 多重性表示方式列表

表示方式	多重性说明
1..1	表示另一个类的一个对象只与一个该类对象有关系
0..*	表示另一个类的一个对象与零个或多个该类对象有关系
1..*	表示另一个类的一个对象与一个或多个该类对象有关系
0..1	表示另一个类的一个对象没有或只与一个该类对象有关系
m..n	表示另一个类的一个对象与最少 m 、最多 n 个该类对象有关系 ($m \leq n$)

例如,一个表单(Form)可以拥有零个或多个按钮(Button),但是一个按钮只能属于一个表单,因此,一个 Form 类的对象可以与零个或多个 Button 类的对象相关联,但一个 Button 类的对象只能与一个 Form 类的对象关联,如图 1-6 所示。

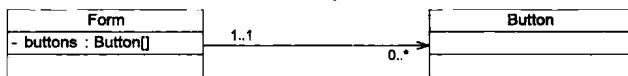


图 1-6 多重性关联实例

(5) 聚合关系

聚合关系(Aggregation)表示一个整体与部分的关系。通常在定义一个整体类后,再去分析这个整体类的组成结构,从而找出一些成员类,该整体类和成员类之间就形成了聚合关系。如一台计算机包含显示器、主机、键盘、鼠标等组成部分,就可以使用聚合关系来描述整体与部分之间的关系。在聚合关系中,成员类是整体类的一部分,即成员对象是整体对象的一部分,但是成员对象可以脱离整体对象独立存在。在 UML 中,聚合关系用带空心菱形的直线表示。例如,汽车发动机(Engine)是汽车(Car)的组成部分,但是汽车发动机可以独立存在,因此,汽车和发动机是聚合关系,如图 1-7 所示。

(6) 组合关系

组合关系(Composition)也表示类之间整体和部分的关系,但是组合关系中部分和整体