



21世纪高职高专规划教材·计算机类

基于UML的 管理信息系统实训

主编 姚 策

 北京理工大学出版社
BEIJING INSTITUTE OF TECHNOLOGY PRESS

基于 UML 的

地理信息系统的开发

王 强 王 强



中国图书馆学会图书馆分会 2006 年学术年会论文集

21 世纪高职高专规划教材·计算机类

基于 UML 的管理 信息系统实训

主 编 姚 策

副主编 刘学敏

 北京理工大学出版社

BEIJING INSTITUTE OF TECHNOLOGY PRESS

内 容 简 介

本书是“管理信息系统”课程的实践环节指导用书,介绍了UML语言及其在面向对象软件系统分析与设计中的应用。目的是通过课程设计的综合训练来培养学生的实际分析问题能力及动手能力,帮助学生系统地掌握本门课程的主要内容,更好地完成教学任务。

本书既注重知识的系统性,同时又注重管理信息系统的实践性和选材的先进性,着重突出实用性和可操作性,叙述深入浅出,可作为高职高专计算机与软件技术专业或计算机相关专业的实践指导教材或参考书,也可供软件开发人员、项目管理人员和软件项目客户阅读。

版权专有 侵权必究

图书在版编目(CIP)数据

基于UML的管理信息系统实训/姚策主编. —北京:北京理工大学出版社, 2007. 4

21世纪高职高专规划教材·计算机类

ISBN 978-7-5640-0936-6

I. 基… II. 姚… III. ①面向对象语言, UML-程序设计-高等学校:技术学校-教材②管理信息系统-高等学校:技术学校-教材
IV. TP312

中国版本图书馆CIP数据核字(2007)第018887号

出版发行/北京理工大学出版社

社 址/北京市海淀区中关村南大街5号

邮 编/100081

电 话/(010) 68914775(办公室) 68944990(批销中心) 68911084(读者服务部)

网 址/http://www.bitpress.com.cn

经 销/全国各地新华书店

印 刷/北京圣瑞伦印刷厂

开 本/787毫米×1092毫米 1/16

印 张/14.75

字 数/335千字

版 次/2007年4月第1版 2007年4月第1次印刷

印 数/1~4000册

定 价/24.00元

责任校对/陈玉梅

责任印制/李绍英

图书出现印装质量问题,本社负责调换

前 言

进入 21 世纪,信息技术的应用几乎已经遍及各行各业。软件作为信息技术的前沿,其所扮演的角色尤为重要。同时,随着管理信息系统的研究越来越深入和广泛,新的学科、新的技术、新的方法和工具不断涌现。20 世纪末,面向对象的技术被广泛应用,人们发现仅仅在代码阶段使用面向对象的技术是不够的,需要在分析和设计阶段就开始使用面向对象技术,于是面向对象分析和面向对象设计技术迅速发展起来,已逐渐取代了传统的方法,成为我国当前计算机管理信息系统学科中的主流方法。

以往使用的面向对象的分析与设计方法各有特色,也各有不足之处,而且术语不统一,缺乏共同的标准,常给软件开发人员带来困惑。著名的软件工程学家 Grady Booch, Ivar Jacobson, Jim Rumbaugh 先后聚集于 Rational 公司,携手合作、共同努力,综合了他们各自原创的面向对象的分析与设计方法,加以扩充改进并汲取其他同类方法的优点,提出了统一模型语言 UML。1997 年 UML 被美国标准化组织 OMG 接受,并在当年发布了标准版本 UML1.1。经过不断的使用、修改、补充、完善,UML 日趋成熟,已成为国际上领先的软件开发的有效方法和工具,得到了广泛的推广应用。本书特色之一就是主要集中讨论如何使用 UML 进行面向对象的系统分析与设计,重点讲述面向对象的系统分析和设计、系统编程及测试。本书具有以下特点:一是不依赖于具体的管理信息系统教科书,重点放在管理信息系统的实践应用上;二是结合实际应用的要求,使课程设计既覆盖知识点,又接近工程设计需要,激发学生的学习兴趣,提高学生独立分析问题及解决问题的能力,并养成良好的建模习惯;三是本书注重实用性和可操作性,叙述深入浅出,以学籍管理等系统为例,逐步介绍了使用 UML 和 Rose 进行设计的全过程,最终完成一个完整的解决方案;四是本书可以对计算机或相关专业学生设计撰写毕业论文起到指导作用。

全书共分为 11 章,主要内容为:第 1 章是课程设计的目的与要求,主要介绍了在“管理信息系统课程设计”中学生要达到的标准;第 2 章主要介绍面向对象的概念以及 UML 中 9 种图的基础知识;第 3 章介绍了使用建模工具 Rose 建立 UML 图的基本操作;第 4 章~第 7 章主要介绍面向对象的软件工程的分析、设计方法;第 8 章介绍了系统编程;第 9 章主要介绍软件测试的基本概念及常用的测试方法;第 10 章及第 11 章通过两个实例详细介绍了面向对象的软件分析与设计的具体流程。

本书由姚策主编,刘学敏副主编。刘世堂、王立、王春媚参与了本书的编写工作,并对本书的插图、资料收集及校对做了大量的工作,在此致以诚挚的感谢。

本书内容新颖、实用,讲述通俗易懂,读者阅读本书后,便可以仿照本书步骤进行实际操作。但是由于水平有限,时间紧张,书中疏漏在所难免,望各位读者批评指正。本书在编写当中参考了许多国内外文献,但由于篇幅有限,有一些未能列入,敬请谅解,在此致以诚挚的谢意!

编 者

2006 年 12 月

目 录

1	课程设计的目的与要求	1
1.1	课程设计的目的与要求	1
1.2	课程设计实施步骤	1
1.2.1	选题	1
1.2.2	拟出具体的设计方案	2
1.2.3	设计与调试	2
1.2.4	撰写课程设计总结报告	2
1.3	上交材料	2
1.4	课程设计报告的撰写规范	2
1.5	软件工程课程设计课题验收	3
1.6	指导教师的主要工作	3
1.7	成绩评定	3
2	面向对象的概念与 UML 介绍	4
2.1	面向对象的概念	4
2.1.1	对象的概念	4
2.1.2	面向对象的特征	4
2.1.3	面向对象的要素	4
2.1.4	面向对象的软件分析与设计方法	5
2.1.5	面向对象方法具有的优越性	5
2.2	UML 简介	5
2.2.1	标准建模语言 UML 的内容	5
2.2.2	UML 建模过程	7
2.2.3	UML 模型视图	8
2.3	用例图	8
2.3.1	用例图的元素	8
2.3.2	用例间的关系	9
2.4	静态图	10
2.4.1	类图	10
2.4.2	对象图	12
2.5	交互图	13
2.5.1	顺序图	13
2.5.2	协作图	14
2.6	行为图	15

2.6.1	状态图	15
2.6.2	活动图	16
2.7	实现图	17
2.7.1	组件图	17
2.7.2	配置图	18
	习题	18
3	软件开发工具 Rational Rose	20
3.1	Rational Rose 的主要功能	20
3.2	Rational Rose 的使用	21
3.2.1	使用 Rose 创建模型	21
3.2.2	系统主菜单窗口	23
3.2.3	在 Rose 中创建 UML 图的基本操作	26
	习题	46
4	系统可行性分析	47
4.1	可行性分析的概念	47
4.2	可行性研究报告的编写目的及任务	47
4.2.1	可行性研究报告的编写目的	47
4.2.2	可行性研究报告的编写任务	48
4.3	学籍管理系统可行性分析实例	48
4.3.1	引言	48
4.3.2	可行性研究的前提	49
4.3.3	对现有系统的分析	51
4.3.4	所建议系统技术可行性分析	51
4.3.5	所建议系统经济可行性分析（投资及收益分析）	53
4.3.6	社会因素可行性分析	54
4.3.7	结论意见	54
	习题	54
5	系统的需求分析	55
5.1	系统需求的分析过程	56
5.1.1	获取用户的需求	56
5.1.2	分析用户的需求	57
5.1.3	编写软件需求说明书	58
5.1.4	用户与开发人员共同验证需求	58
5.1.5	需求分析过程应注意的问题	59
5.2	系统功能和角色的识别	59
5.2.1	系统功能的识别	59
5.2.2	角色的识别	59
5.3	系统需求的 UML 静态建模	60
5.3.1	需求分析的过程	60

5.3.2	需求分析模型的组成	60
5.3.3	系统需求描述	61
5.3.4	需求分析	62
5.3.5	建立对象模型	67
5.4	系统需求的 UML 动态建模	72
5.4.1	对对象之间的交互关系进行建模: 时序图	72
5.4.2	对对象之间的交互关系进行建模: 协作图	74
5.4.3	针对单个对象重要行为和状态的建模: 状态图	75
5.4.4	检查分析模型	77
	习题	78
6	系统的总体设计	79
6.1	系统设计概述	79
6.2	软件体系架构	80
6.2.1	仓库体系结构	80
6.2.2	模型/视图/控制器体系结构	81
6.2.3	客户/服务器体系结构	82
6.2.4	B/S 结构	85
6.2.5	对等体系结构	86
6.2.6	管道和过滤器体系结构	86
6.3	子系统设计和访问控制设计	87
6.3.1	定义系统设计目标	89
6.3.2	定义子系统和功能模块	89
6.3.3	确定系统设计目标	93
6.3.4	确定子系统	94
6.3.5	数据管理设计	95
6.3.6	定义访问控制	95
6.3.7	设计全局控制流	96
6.3.8	确定系统的功能范围	98
6.3.9	系统配置设计与映射到软/硬件平台	98
6.4	总体设计报告	101
6.4.1	系统设计报告内容	101
6.4.2	系统设计报告的不断优化	102
	习题	102
7	系统的详细设计	103
7.1	详细设计概述	103
7.1.1	详细设计的任务	103
7.1.2	详细设计的原则	104
7.2	对象和类设计	104
7.2.1	对象和类设计	104

7.2.2	对象设计内容	104
7.2.3	对象设计过程	106
7.2.4	对象设计文档的编制方法和内容	110
7.3	用户界面设计	111
7.3.1	图形界面类型	111
7.3.2	用户界面设计过程	114
7.3.3	界面设计的技巧	115
7.4	数据库设计	115
7.4.1	数据模型	116
7.4.2	对象数据库模型	117
7.4.3	对象关系数据库模型	117
7.4.4	关系数据库模型	119
7.4.5	UML 映射 RDB	121
	习题	122
8	系统编程	123
8.1	程序设计的基本概念	123
8.1.1	程序设计	123
8.1.2	程序设计语言的分类	123
8.1.3	程序设计语言的选择	124
8.1.4	程序设计的步骤	124
8.2	编码风格	124
8.3	面向对象的编程	125
8.4	学籍管理系统编程实例	125
8.4.1	编程语言的选择	125
8.4.2	编程步骤	126
8.4.3	系统编程实例	126
8.4.4	系统的改进	152
	习题	153
9	系统测试	154
9.1	软件测试概述	154
9.2	自动化测试	156
9.2.1	自动化测试概念	156
9.2.2	自动化测试的工具介绍 (1) —— Rational Robot	157
9.2.3	自动化测试的工具介绍 (2) —— LoadRunner	158
9.3	学籍管理系统测试实例	162
9.3.1	系统开发背景	162
9.3.2	测试过程	163
9.3.3	测试计划	164
9.3.4	测试用例	165

9.3.5 缺陷报告	166
9.3.6 测试结果总结分析	167
习题	169
10 综合实例：图书管理系统的分析与设计	170
10.1 系统需求描述	170
10.1.1 系统的总体功能	170
10.1.2 系统的功能细化	170
10.2 系统的用例建模	171
10.2.1 识别参与者	171
10.2.2 识别用例	172
10.2.3 系统用例图	172
10.2.4 为用例编写文档	172
10.3 活动建模	174
10.3.1 创建用例“维护借阅者信息”活动图	175
10.3.2 创建用例“维护书刊信息”活动图	175
10.3.3 创建用例“维护物理书刊信息”活动图	176
10.4 静态结构建模	176
10.4.1 定义系统实体类	176
10.4.2 定义用户界面类	177
10.4.3 确定类之间的关系并建立类图	177
10.5 动态行为建模（交互建模）	179
10.5.1 建立顺序图	179
10.5.2 建立状态图	185
10.6 包设计	185
10.6.1 系统分解	185
10.6.2 系统包图	186
10.7 协作的设计	186
10.7.1 “登录”的协作图	186
10.7.2 “还书”的协作图	186
10.7.3 “借书”的协作图	187
10.8 建立物理模型	188
10.8.1 建立系统组件图	188
10.8.2 建立配置图	188
10.9 用户界面设计	189
10.10 数据库设计	189
11 综合实例：网上教学管理系统的分析与设计	190
11.1 系统需求描述	190
11.1.1 系统的总体功能	190
11.1.2 系统的功能细化	191

11.2 系统的用例建模.....	191
11.2.1 识别参与者.....	191
11.2.2 识别用例.....	192
11.2.3 系统用例图.....	192
11.2.4 为用例编写文档.....	193
11.3 活动建模.....	194
11.4 静态结构建模.....	196
11.4.1 定义实体类.....	196
11.4.2 定义类属性.....	196
11.4.3 确定类之间的关系.....	196
11.4.4 建立类图.....	197
11.5 动态行为建模（交互建模）.....	198
11.5.1 建立顺序图.....	198
11.5.2 建立状态图.....	200
11.6 包设计.....	202
11.7 协作设计.....	202
11.8 建立物理模型.....	203
11.8.1 建立系统组件图.....	204
11.8.2 建立配置图.....	204
11.9 用户界面设计.....	205
11.10 数据库设计.....	205
11.11 系统实施情况、测试和升级.....	206
附录 软件规格说明书文档模板.....	208
参考文献.....	225

1 课程设计的目的与要求

1.1 课程设计的目的与要求

开设“管理信息系统课程设计”的主要目的是提高学生综合应用所学专业知识的能力，训练和提高软件开发技能。要求每个学生在规定时间内完成一个规模适当的应用软件系统；在老师的指导下以软件设计为中心，独立地完成从需求分析、软件设计、编码到软件测试运行的软件开发全过程。通过软件开发的实践实现以下基本目标。

1. 深化已学的知识，完成从理论到实践的转化

通过软件开发的实践，进一步加深对软件工程方法和技术的了解，将软件工程的理论知识运用于开发的实践，并在实践中逐步掌握软件工具的使用。

2. 提高分析和解决实际问题的能力

课程设计是软件工程实践的一次模拟训练，通过软件开发的实践，积累经验，提高分析和解决问题的能力。

3. 培养“开拓创新”能力

大力提倡和鼓励在开发过程中使用新方法、新技术。激发学生实践的积极性和创造性，开拓思路设计新算法、新创意，培养创造性的工程设计能力。

1.2 课程设计实施步骤

1.2.1 选题

1. 选题原则

课程设计的课题要尽量结合实践教学，反映新技术，以取得更好的工程设计实践的训练效果。同时由于受到时间、开发环境及条件等的限制，命题应从实际出发，课题的规模大小及难易程度要适中。课题也要具有一定复杂度，通过激发学生参加开发实践的积极性和创造性努力完成实践任务，才能达到综合应用所学知识的目的。选题分两种方式，一是指导教师

命题，二是学生自己选题，学生选题应由指导教师批准后方可进行。

2. 选题要求

(1) 选题的内容要具有先进性、综合性、实践性，应适合实践教学并启发创新，选题内容不应太简单，难度要适中。

(2) 可以结合企事业单位应用的实际情况进行选题。

(3) 题目应该实现相对完整的功能。

1.2.2 拟出具体的设计方案

学生应在指导教师的指导下着手进行课程设计总体方案的总结与论证。学生根据自己所接受的设计题目设计出具体实施方案，经指导教师批准后方可实施。

1.2.3 设计与调试

学生在指导教师的指导下应完成所确定题目的设计及调试工作，并取得预期的成果。

1.2.4 撰写课程设计总结报告

课程设计总结报告是对课程设计工作的整理和总结，主要包括课程设计的总体设计方案、算法设计、程序测试与调试等部分，最后写出课程设计的总结报告。

1.3 上交材料

学生应按规定撰写课程设计总结报告。课程设计实践教学环节结束后，由指导教师批阅，并将学生完成的总结报告按要求送交院系存档。

1.4 课程设计报告的撰写规范

撰写课程设计的总结报告是为了在设计、安装、调试完成后，训练学生编写归纳技术文档、撰写科学技术总结报告的能力，也训练学生养成严谨的作风和科学的态度。通过撰写课程设计总结报告，不仅能对设计、安装、调试及技术参考等内容进行全面总结，而且还能够把实践内容提升到理论高度。总结报告按如下顺序用 A4 纸进行撰写打印并装订成册。

- (1) 统一的封面（含课程设计课题名称、专业、班级、姓名、学号、指导教师等）。
- (2) 设计任务和技术要求（由指导教师在选题时提供给学生）。
- (3) 内容摘要。
- (4) 目录。

- (5) 课程设计总结报告正文。
- (6) 按统一格式列出主要参考文献。
- (7) 指导教师成绩评定表。

1.5 软件工程课程设计课题验收

验收步骤如下：

- (1) 学生选择测试用例演示所完成软件的所有功能。
- (2) 教师对系统及模块功能进行抽查。

1.6 指导教师的主要工作

由担任本课程的教师或其他老师担任指导教师，指导教师具体负责课程设计的任务布置、实践指导和成绩评定。

首先，根据学生的实际情况确定题目，并使学生充分理解课程设计要完成题目的目的。指导教师应给出题目比较完整的介绍，使学生对要进行的设计内容有明晰的认识和理解。指导教师在公布课程设计课题时应包括以下内容：课题名称、设计任务、主要方法、主要参考文献等内容。

其次，严格按照软件工程要求的开发设计步骤进行，因为本课程设计的主要目标之一就是要学生养成良好习惯，即严格按照软件工程实施标准的、规范良好的开发设计方法和作风，提倡标准化的程序设计以降低设计成本并提高设计效率，因此本课程设计的完成步骤应该在指导教师的严格把握下进行。

指导教师在学生停课期间应及时了解学生的实践进度和出勤情况，为学生解决疑难问题。

1.7 成绩评定

成绩按照优、良、中、及格和不及格 5 级评定，具体要求如下：

- (1) 文档编写完整规范，编写过程符合软件工程开发设计流程等要求。
- (2) 软件功能达到设计要求，实现正确、程序稳定、界面美观大方、编程符合规范、程序具有易读性等。
- (3) 学习态度、独立设计能力的要求。
- (4) 课程设计报告。

2 面向对象的概念与 UML 介绍

本章实训

实训目的:

本章讲解的是实训前的预备知识。

实训内容:

- (1) 掌握 UML 的内容: 5 类 (9 种) 图以及图元素的使用。
- (2) 掌握 UML 建模过程。

2.1 面向对象的概念

在软件开发的各个方面几乎都应用了面向对象 (Object Oriented, OO) 的思想, 如面向对象的分析 (Object Oriented Analysis, OOA)、面向对象的设计 (Object Oriented Design, OOD) 以及人们经常说的面向对象的编程实现 (Object Oriented Programming, OOP) 等。

2.1.1 对象的概念

对象 (Object) 可以是任何具体事物。例如, 现实生活中的计算机、打印机、因特网中的站点、超级链接、图像等都可以看作对象。每个对象都有其属性和事件。对象属性标识了对象的性质、特征; 事件也叫操作, 是由对象识别和响应的某些操作, 是一些特定的预定义活动, 可由用户或系统启动。

2.1.2 面向对象的特征

- (1) 唯一性。
- (2) 分类性。
- (3) 继承性。
- (4) 多态性 (多形性)。

2.1.3 面向对象的要素

- (1) 抽象。

- (2) 封装性（信息隐藏）。
- (3) 共享性。

2.1.4 面向对象的软件分析与设计方法

目前，面向对象开发方法的研究已日趋成熟，国际上已有不少面向对象产品出现。这里简单介绍 Booch 方法、OOSE 方法和 OMT 方法。

1. Booch 方法

Booch 方法最先描述了面向对象的软件开发方法的基础问题。提出面向对象的软件分解更接近人对客观事务的理解，而功能分解只通过问题空间的转换来获得。

2. OOSE 方法

OOSE 方法是一个使用用例驱动的方法，它建立的所有模型都是以用例模型为基础的。OOSE 方法的主要特点是能够较好地描述系统的需求。

3. OMT 方法

OMT 方法是一种新兴的面向对象的开发方法，开发工作的基础是对真实世界的对象建模，然后围绕这些对象使用分析模型来进行独立于语言的设计，面向对象的建模和设计促进了对需求的理解，有利于开发出更清晰、更容易维护的软件系统。该方法为大多数应用领域的软件开发提供了一种实际的、高效的保证，努力寻求一种问题求解的实际方法。

2.1.5 面向对象方法具有的优越性

- (1) 更加接近现实世界。
- (2) 软件系统结构更加稳定。
- (3) 软件的可重用性更好。
- (4) 更加便于软件的维护与扩充。

2.2 UML 简介

UML（Unified Modeling Language，统一建模语言）是一种可视化的建模语言，它能够让系统构造者用标准的、易于理解的方式建立起能够表达他们设计思想的系统蓝图，并且提供一种机制，以便于不同的人之间可以有效地共享和交流设计成果。

2.2.1 标准建模语言 UML 的内容

首先，UML 把 Booch，OMT 和 OOSE 三种方法中的基本概念融合在一起，而且这些基

本概念与其他面向对象技术中的基本概念基本一致，所以 UML 成为开发者都愿意使用的一种统一建模语言；其次，UML 不单单是以上方法的简单融合，而是在这些方法的基础上广征大家意见，集众家之长，经多次修改而成，UML 扩展了现有方法的应用范围；第三，UML 是标准的建模语言，而不是标准的开发过程。尽管 UML 的应用必然以系统的开发过程为背景，但不同的组织和不同的应用领域需要采取不同的开发过程。

作为一种建模语言，UML 的定义包括 UML 语义和 UML 表示法两个部分。

(1) UML 语义：描述基于 UML 的精确元模型定义。元模型为 UML 的所有元素在语法和语义上提供了简单、一致、通用的定义性说明，使开发者能在语义上取得一致，消除了因人而异的最佳表达方法所造成的影响。此外 UML 还支持对元模型的扩展定义。

(2) UML 表示法：定义 UML 符号的表示法，为开发者或开发工具使用这些图形符号和文本语法及为系统建模提供了标准。这些图形符号和文字所表达的是应用级的模型，在语义上它是 UML 元模型的实例。

标准建模语言 UML 的重要内容可以由 5 类图（共 9 种图形）来定义，表 2-1 对 UML 中的 9 种图作了基本的描述：

表 2-1 UML 中的 9 种图

类型	图名	描述
用例图	用例图	从用户角度描述系统的功能，并指出各功能的操作者
静态图	类图	用于定义系统的类，包括描述类之间的关系（如关联关系、组合关系等）以及类的内部结构，即类的属性和操作。因此类图是描述系统中类的静态结构，即它描述的是一种静态关系，它的生命周期存在于系统的始终
	对象图	对象图是类图的实例，使用与类图基本一致的标识。不同点是对象图显示类的多个对象实例，而不是实际的类。由于对象存在生命周期，因此对象图只能在系统某一时间段内存在
行为图	状态图	描述一类对象可能出现的状态及事件发生时状态的转移条件。一般状态图是对类图的补充
	活动图	描述为满足用例要求所要进行的活动以及活动间的约束关系。使用活动图可以很方便地表示并行活动
交互图	顺序图	描述对象之间的动态合作关系，强调对象之间消息发送的时间顺序，也显示对象之间的交互过程
	协作图	与顺序图是等价的，但着重描述的是对象间的协作关系
实现图	组件图	描述代码部件的物理结构及各部件之间的依赖关系。一个部件可能是资源代码部件、二进制部件或可执行部件。包含关于逻辑类或实现类的一些信息，有助于分析和理解部件之间的相互影响程度
	配置图	定义系统中软硬件的物理体系结构。可以描述实际的计算机、设备（用节点表示）及它们之间的关系，也可描述部件之间的依赖性以及连接的类型。在节点内部放置可执行部件及对象，以描述节点与可执行软件单元的对应关系

从应用的角度看，采用面向对象技术进行系统设计时，第一步是描述需求；第二步是按