

MIANXIANG DUIXIANG RUANJIAN GONGCHENG

面向对象软件工程

实践教程

王先国 主编

玉亚芳 罗中良 蔡妍 副主编

SHIJIAN JIAOCHENG

广东高等教育出版社

面向对象软件工程 实践教学

主 编 王先国

副主编 王亚芳 罗中良 蔡 妍

广东高等教育出版社

· 广州 ·

内 容 提 要

本书是一本面向对象技术的实践教程，结合大量软件开发实例，重点介绍了软件工程基本概念、面向对象技术、软件开发方法、软件开发过程、项目组织管理、系统建模，以及面向对象需求分析、面向对象设计、面向对象实现等。

本书重点突出面向对象技术的分析、设计和实现流程以及建模方法。书中所有的概念、技术、开发方法都通过实例来演示，内容精炼，表达简明，实例丰富，非常适合用作高等院校计算机专业及相关专业的教材，也可以作为培训机构相关专业的培训教材。

图书在版编目 (CIP) 数据

面向对象软件工程实践教程/王先国主编. —广州: 广东高等教育出版社, 2010. 8

ISBN 978 - 7 - 5361 - 3958 - 9

I. ①面… II. ①王… III. ①面向对象语言 - 软件工程 - 教材
IV. ①TP311.5

中国版本图书馆 CIP 数据核字 (2010) 第 153650 号

| | |
|------|---|
| 出版发行 | 广东高等教育出版社 |
| | 社址: 广州市天河区林和西横路 |
| | 邮编: 510500 营销电话: (020) 87557232 |
| | 网址: http://www.gdgjs.com.cn |
| 印 刷 | 佛山市浩文彩色印刷有限公司 |
| 开 本 | 787 毫米×1 092 毫米 1/16 |
| 印 张 | 18.75 |
| 字 数 | 480 千字 |
| 版 次 | 2010 年 8 月第 1 版 |
| 印 次 | 2010 年 8 月第 1 次印刷 |
| 印 数 | 1~3 000 册 |
| 定 价 | 30.00 元 |

(版权所有, 翻印必究)

面向对象软件工程或者面向对象分析与设计是计算机专业学生和软件工程专业学生一门非常重要的必修课程。尽管市面上介绍面向对象技术的图书不少，但是没有一本书的内容能全面、具体、正确地涵盖整个软件开发过程：从需求获取、系统建模、系统分析和系统实现。并且，从理论到实际项目的介绍也是脱节的。因此，学生对面向对象技术的理解常常是不完整的，甚至是错误的。

学生对面向对象技术的误解主要集中在以下几点：第一，没有真正理解 UML 表示法，不知道如何使用它们；第二，对统一开发过程（RUP）不理解，不知道在哪种情况下采用哪种模型来构造系统；第三，市面上没有一本面向对象的图书全面、系统、具体介绍整个软件生命过程中实际项目的开发步骤、开发技术、建模技术和 RUP 裁剪技术；第四，多数学生和软件开发者并不知道软件开发的三个要素，即开发过程、表示法和技术，更不知道如何系统地、合理地利用这三个要素来开发软件系统。

本书就是为学生和软件开发者提供一本涵盖整个软件开发过程的教材，它能够为大中型软件系统的开发提供开发步骤、技术提示和表示方法。

读者对象

本书是一本面向对象技术的软件工程教材，既适合软件工程初学者阅读，也适合系统分析员、设计者和系统测试者阅读。在写作上，本书以软件开发过程为主线，以系统建模为目标，运用实例系统地阐明了面向对象技术、软件开发过程、UML 建模方法。本书理论与应用配合紧密，知识表达通俗易懂，既可作为高等院校计算机专业及相关专业的教材，也可以作为培训机构相关专业的培训教材。

本书特色

市面上出现的面向对象技术的图书很多，但是，没有一本内容能全面、具体、正确地涵盖整个软件开发过程，并且，从理论到实际项目的介绍也是脱节的。本书与这些图书的不同之处在于强调内容的系统性、连贯性、逻辑性；强调软件过程、建模方法和面向对象技术的合理运用；强调理论与实践的结合性和可操作性。其具体特点如下。

1. 在内容组织上，强调知识的系统性、连贯性、逻辑性。对基本概念、技术、建模方法和软件开发过程的介绍由易到难逐层展开，便于读者理解。

2. 在实例写作上，强调软件工程三要素的运用。即，每个开发实例都贯穿了软件开发过程、建模方法和面向对象技术。整个开发流程是可以操作的，也是可以模拟的，学生能真正做到学以致用。

3. 知识表达风格上，知识表达方法采用从框架到细节，即首先对知识进行概要描述，

然后分解知识，简化知识，对知识进行详细描述，这样，将复杂过程、技术、方法简单化，抽象问题具体化。

本书组织

本书总共 15 章。下面对每一章的内容简要介绍：第一章介绍了软件工程的基本概念和定义；第二章介绍了面向对象技术的基本概念和特点；第三章介绍了传统软件开发方法和面向对象软件开发方法；第四章介绍了 RUP 软件开发过程；第五章介绍了软件开发过程的沟通、组织和管理；第六章介绍了结构化建模的技术和建模过程；第七章介绍了用例建模技术和过程；第八章介绍了动态建模技术和过程；第九章介绍了面向对象的需求定义和需求管理；第十章介绍了面向对象的分析内容、分析方法、分析过程和分析管理；第十一章介绍了系统概要设计的内容和目标；第十二章介绍了对各子系统进行详细设计的方法和过程；第十三章介绍了对象设计的内容、方法；第十四章介绍了把设计类映射为语言代码的方法和步骤，以及双向工程的内容；第十五章介绍了面向对象的测试概念和测试案例。

作者情况

本书作者过去十多年在大型软件公司从事计算机系统分析和设计工作，积累了丰富的系统分析和设计经验，近几年从事高校计算机教学工作，作者既有丰富的实践经验，又有丰富的教学经验，是主讲软件工程、UML 统一建模、设计模式、java 技术和 jsp 技术的一线教师。

本书第一章由王亚芳编写；第二至第四章由罗中良教授编写（惠州学院）；第五章由蔡妍编写；第六至十五章由王先国编写。全书由王先国统稿。

参与本教材编写的专家还有程汉湘教授（广东工业大学）、杨振野教授（广东技术师范学院）、罗兵教授（五邑大学）、罗先录教授（东软软件学院计算机系主任）、周娴玮博士。

致 谢

感谢肖应旺博士、冼广铭博士、刘刚博士、张承忠老师、杨桂芝老师、曲超博士为本书提出了许多建设性的意见和指导。

联系方式

书中实例虽然经过了多次复查，但难免会存在疏漏和错误，恳请读者批评指正。如有好的建议或在学习中遇到疑难问题，欢迎大家发电子邮件与本书作者联系（wangxg288@tom.com）。

编著者

2010 年 5 月

目 录

| | |
|-------------------------------|------|
| 第一章 软件工程概述 | (1) |
| 第一节 什么是软件工程 | (1) |
| 一、传统软件工程 | (1) |
| 二、面向对象的软件工程 | (2) |
| 三、其他软件工程 | (2) |
| 第二节 软件工程的三大要素 | (3) |
| 第三节 基本概念 | (4) |
| 一、参与者和角色 | (5) |
| 二、系统和模型 | (5) |
| 三、工作产品 | (7) |
| 四、活动、任务和资源 | (7) |
| 五、需求 | (8) |
| 六、分析与设计 | (9) |
| 本章小结 | (9) |
| 习 题 | (9) |
| 第二章 面向对象的技术 | (10) |
| 第一节 传统软件开发方法存在问题 | (10) |
| 第二节 面向对象的基本概念 | (11) |
| 第三节 面向对象的特点 | (12) |
| 第四节 模型 | (13) |
| 一、对象模型 | (13) |
| 二、功能模型 | (14) |
| 三、动态模型 | (15) |
| 第五节 面向对象的分析和设计 | (15) |
| 本章小结 | (17) |
| 习 题 | (17) |
| 第三章 软件开发方法 | (18) |
| 第一节 软件工程方法概述 | (18) |
| 一、大型软件开发中的困难 | (18) |
| 二、软件开发的规范化 | (18) |

| | |
|-----------------------------|------|
| 三、软件开发方法概述 | (19) |
| 四、方法学 | (21) |
| 第二节 软件开发过程中的阶段 | (22) |
| 一、需求 | (22) |
| 二、分析 | (24) |
| 三、设计 | (25) |
| 四、规范 | (25) |
| 五、实现 | (25) |
| 六、测试 | (25) |
| 七、部署 | (26) |
| 八、维护 | (26) |
| 第三节 传统软件开发方法学 | (26) |
| 一、瀑布方法学 | (26) |
| 二、螺旋式方法学 | (28) |
| 三、迭代式方法学 | (29) |
| 四、递增式方法学 | (29) |
| 五、合并方法学 | (30) |
| 第四节 面向对象的方法学 | (30) |
| 一、面向对象方法学的四个要素 | (30) |
| 二、面向对象方法概要 | (31) |
| 三、建模活动 | (31) |
| 四、问题求解活动 | (33) |
| 五、知识获取活动 | (33) |
| 本章小结 | (34) |
| 习 题 | (34) |
| 第四章 RUP (统一软件过程) | (35) |
| 第一节 当前流行的软件过程 | (35) |
| 第二节 RUP 概述 | (36) |
| 一、RUP 的核心 workflow | (37) |
| 二、RUP 的四个阶段 | (38) |
| 三、RUP 中的迭代模型 | (40) |
| 第三节 RUP 中的核心 workflow | (41) |
| 一、需求 workflow | (41) |
| 二、分析 workflow | (44) |
| 三、设计 workflow | (47) |
| 四、实现 workflow | (49) |
| 五、测试 workflow | (52) |
| 第四节 RUP 裁剪 | (56) |
| 第五节 RUP 的十大要素 | (56) |
| 本章小结 | (59) |

| | |
|-----------------------------|------|
| 习 题 | (59) |
| 第五章 项目组织和管理 | (60) |
| 第一节 沟通 | (60) |
| 一、计划内的沟通 | (61) |
| 二、计划外的沟通 | (64) |
| 第二节 软件配置管理 | (65) |
| 第三节 项目管理 | (65) |
| 一、项目综述 | (65) |
| 二、项目组织 | (66) |
| 第四节 质量管理 | (67) |
| 一、软件质量 | (67) |
| 二、软件质量保证 | (68) |
| 第五节 进度管理 | (69) |
| 一、软件开发项目的并行性 | (69) |
| 二、计划 | (69) |
| 三、进度安排 | (70) |
| 四、进度跟踪与控制 | (72) |
| 第六节 风险管理 | (72) |
| 一、软件风险 | (72) |
| 二、风险识别 | (73) |
| 三、风险预测 | (73) |
| 四、风险控制 | (74) |
| 第七节 人员组织与管理 | (75) |
| 一、项目参与者 | (75) |
| 二、人员分配 | (75) |
| 第八节 软件开发成本估算 | (77) |
| 一、软件成本估算过程 | (77) |
| 二、软件成本估算策略 | (78) |
| 三、常用的成本估算模式 | (78) |
| 本章小结 | (80) |
| 习 题 | (81) |
| 第六章 结构化建模与分析技术 | (82) |
| 第一节 结构化建模技术 | (82) |
| 一、对象分类 | (82) |
| 二、类和对象的表示 | (83) |
| 三、类间的关系 | (84) |
| 四、结构模型示例 | (88) |
| 第二节 结构分析技术 | (89) |
| 一、识别类和对象 | (89) |

| | |
|----------------------------|-------|
| 二、系统模型优化 | (89) |
| 三、结构分析中的启发式规则 | (90) |
| 第三节 领域建模与分析过程 | (90) |
| 一、领域模型开发流程 | (90) |
| 二、领域建模与分析示例 | (92) |
| 本章小结 | (100) |
| 习 题 | (100) |
| 第七章 用例建模与分析技术 | (101) |
| 第一节 用例建模技术 | (101) |
| 一、参与者和角色 | (101) |
| 二、用例 | (103) |
| 三、系统边界 | (103) |
| 四、用例模型示例 | (103) |
| 第二节 用例分析技术 | (105) |
| 一、用例关系 | (105) |
| 二、组织用例 | (107) |
| 三、用例描述 | (109) |
| 四、用例优先级 | (111) |
| 第三节 用例建模与分析过程 | (112) |
| 一、用例建模流程 | (112) |
| 二、用例建模与分析示例 | (114) |
| 本章小结 | (121) |
| 习 题 | (121) |
| 第八章 动态建模与分析技术 | (122) |
| 第一节 动态建模技术 | (122) |
| 一、场景建模技术 | (122) |
| 二、状态图建模技术 | (125) |
| 三、活动图建模技术 | (126) |
| 第二节 动态分析技术 | (127) |
| 一、建立系统级顺序图 | (127) |
| 二、建立子系统级顺序图 | (127) |
| 三、建立第三层顺序图 | (129) |
| 第三节 动态建模与分析过程 | (130) |
| 一、开发用例场景 | (131) |
| 二、开发系统级顺序图 | (132) |
| 三、开发子系统级顺序图 | (133) |
| 四、开发子系统级状态图 | (134) |
| 五、开发三层顺序图 | (135) |
| 六、为控制对象开发状态图 | (135) |

| | |
|--------------------|-------|
| 本章小结 | (136) |
| 习 题 | (136) |
| 第九章 面向对象的需求 | (137) |
| 第一节 相关概念 | (137) |
| 第二节 收集需求的方法 | (137) |
| 一、访谈 | (138) |
| 二、问卷调查 | (138) |
| 三、建立联合需求分析小组 | (138) |
| 四、获取手工操作流程 | (138) |
| 五、快速原型法 | (139) |
| 六、需求分类 | (139) |
| 七、需求验证 | (139) |
| 第三节 定义需求 | (140) |
| 一、定义词汇表 | (141) |
| 二、业务用例模型 | (141) |
| 三、系统用例模型 | (143) |
| 四、修改系统用例模型 | (144) |
| 第四节 管理需求 | (150) |
| 一、联合应用设计 | (150) |
| 二、需求追踪维护 | (151) |
| 三、需求文档化 | (152) |
| 本章小结 | (153) |
| 习 题 | (154) |
| 第十章 面向对象的分析 | (155) |
| 第一节 分析的主要任务 | (155) |
| 一、分析模型 | (155) |
| 二、分析内容 | (156) |
| 第二节 静态分析 | (157) |
| 一、在用例中寻找类 | (157) |
| 二、添加关系 | (158) |
| 三、修改对象模型 | (158) |
| 四、添加属性 | (159) |
| 五、选择属性还是类 | (160) |
| 六、添加关联类 | (161) |
| 第三节 动态分析 | (161) |
| 一、动态分析的任务 | (161) |
| 二、构思用例的实现图 | (162) |
| 三、给类添加操作 | (163) |
| 四、为对象构思状态机 | (163) |

| | | |
|------|-------------------|-------|
| 第四节 | 架构分析 | (164) |
| 一、 | 寻找分析包 | (164) |
| 二、 | 消除包间的循环依赖 | (165) |
| 第五节 | 管理分析 | (166) |
| 一、 | 将分析文档化 | (166) |
| 二、 | 开发角色 | (167) |
| 三、 | 沟通 | (168) |
| 四、 | 分析模型的迭代 | (168) |
| 本章小结 | | (169) |
| 习题 | | (169) |
| 第十一章 | 系统概要设计 | (170) |
| 第一节 | 系统设计概述 | (170) |
| 第二节 | 系统分解 | (171) |
| 一、 | 子系统组成 | (172) |
| 二、 | 子系统接口 | (173) |
| 三、 | 子系统质量特性 | (173) |
| 四、 | 系统分层 | (176) |
| 第三节 | 体系结构风格 | (179) |
| 一、 | 仓库体系结构 | (179) |
| 二、 | MVC 体系结构 | (180) |
| 三、 | 客户/服务器体系结构 | (180) |
| 四、 | 对等体系结构 | (181) |
| 五、 | 层结构体系 | (182) |
| 六、 | 管道和过滤器体系结构 | (183) |
| 第四节 | 系统设计实例 | (184) |
| 一、 | 分析需求模型 | (184) |
| 二、 | 确定设计目标 | (186) |
| 三、 | 系统分解为子系统 | (188) |
| 本章小结 | | (190) |
| 习题 | | (190) |
| 第十二章 | 系统详细设计 | (191) |
| 第一节 | 详细设计概述 | (191) |
| 第二节 | 对子系统迭代设计 | (192) |
| 一、 | 将子系统部署到硬件平台 | (193) |
| 二、 | 选择存储管理策略 | (194) |
| 三、 | 提供访问控制 | (196) |
| 四、 | 设计全局控制流 | (199) |
| 五、 | 标识边界条件 | (199) |
| 第三节 | 评审系统设计 | (202) |

| | |
|---------------------------|-------|
| 第四节 管理系统设计 | (203) |
| 一、系统设计文档化 | (203) |
| 二、系统设计人员 | (205) |
| 三、系统设计迭代 | (205) |
| 本章小结 | (206) |
| 习 题 | (206) |
| 第十三章 对象设计 | (207) |
| 第一节 对象设计概述 | (207) |
| 第二节 对象设计原则 | (208) |
| 一、单一职责原则 (SRP) | (208) |
| 二、开放封闭原则 (OCP) | (209) |
| 三、Liskov 替换原则 (LSP) | (212) |
| 四、依赖倒置原则 (DIP) | (216) |
| 五、接口隔离原则 (ISP) | (218) |
| 第三节 确定设计类 | (219) |
| 一、什么是设计类 | (219) |
| 二、设计类剖析 | (220) |
| 三、如何设计良好的设计类 | (220) |
| 第四节 设计中的继承 | (221) |
| 第五节 设计类关系 | (222) |
| 一、设计类关系 | (222) |
| 二、把分析关联精化成设计关系 | (224) |
| 三、将关联关系具体化 | (228) |
| 第六节 结构化类的组合 | (230) |
| 一、结构化类元 | (230) |
| 二、结构化类 | (230) |
| 第七节 接口规格说明 | (230) |
| 第八节 对象开发者 | (231) |
| 第九节 对象设计文档化 | (232) |
| 本章小结 | (234) |
| 习 题 | (235) |
| 第十四章 面向对象的实现 | (236) |
| 第一节 面向对象语言和编程风格 | (236) |
| 一、面向对象的语言 | (236) |
| 二、程序设计风格 | (237) |
| 第二节 从设计产品到代码 | (238) |
| 一、将类图映射为代码 | (238) |
| 二、将活动图映射为代码 | (246) |
| 三、将状态图映射为代码 | (248) |

| | |
|----------------------------|-------|
| 四、将顺序图映射为代码 | (249) |
| 第三节 Rose 双向工程 | (250) |
| 一、正向工程 | (250) |
| 二、逆向工程 | (252) |
| 三、实例应用 | (253) |
| 本章小结 | (258) |
| 习 题 | (258) |
| 第十五章 面向对象的测试 | (259) |
| 第一节 测试目标 | (259) |
| 第二节 测试准则 | (259) |
| 第三节 测试过程 | (260) |
| 第四节 白盒测试技术 | (261) |
| 一、逻辑覆盖 | (261) |
| 二、基本路径测试 | (264) |
| 三、条件测试 | (265) |
| 四、循环测试 | (265) |
| 第五节 黑盒测试技术 | (266) |
| 一、等价类划分 | (266) |
| 二、边界值分析 | (268) |
| 三、错误推测法 | (268) |
| 第六节 测试步骤 | (268) |
| 一、单元测试 | (269) |
| 二、集成测试 | (270) |
| 三、验收测试 | (271) |
| 四、系统测试 | (272) |
| 第七节 面向对象的软件测试 | (273) |
| 一、面向对象测试模型 | (273) |
| 二、面向对象分析测试 | (273) |
| 三、面向对象设计测试 | (273) |
| 四、面向对象编程测试 | (274) |
| 五、面向对象的单元测试 | (274) |
| 六、面向对象的集成测试 | (274) |
| 七、面向对象的系统测试 | (274) |
| 八、测试用例 | (275) |
| 本章小结 | (276) |
| 习 题 | (277) |
| 附录 UML 图总结 | (278) |

第一章

软件工程概述

“软件工程”一词是1968年北大西洋公约组织（NATO）在联邦德国召开的一次会议上首次提出的，这个会议专门讨论了软件危机问题。它反映了软件人员认识到软件危机的出现及谋求解决这一危机的努力，因此，这次会议被看作是软件发展史上一个重要的里程碑。

当时的软件开发者无法制定具体目标，无法预算实现目标所需的资源，也无法实现客户的愿望。他们通常承诺的是给用户摘月亮，而制造的却是登月车，最后交付的是一副方形车轮，这与目标相去甚远。

软件工程的重点既在软件，也在工程。一个工程师应该能够在一定时间和预算内，通过集成构件来构造高质量的软件产品。工程师通常面临的是问题定义不清晰，解决方案不全面，不得不依靠实验方法来评估解决方案。在飞机设计和桥梁建设等应用领域，工程师们已成功地迎接了类似的挑战，而软件工程师们却没那么幸运和成功。

第一节 什么是软件工程

软件工程是设计软件的工程，是指导计算机软件开发和维护的工程学科。采用工程的概念、原理、技术和方法来开发与维护软件，把经过时间考验而证明是正确的管理技术和当前的最好技术方法结合起来，这就是软件工程。

软件工程的目的是采用软件方法，开发出高效率、高质量的软件系统。

按照开发软件采用的不同技术，将软件工程分为：传统软件工程、面向对象的软件工程、模型驱动软件工程，等等。下面分别介绍其概念。

一、传统软件工程

传统软件工程采用结构化方法开发软件系统。

1. 结构化方法概要

结构化方法是指采用结构化分析方法（Structured Analysis, SA）对软件进行需求分析；然后采用结构化设计方法（Structured Programming, SP）进行概要设计和详细设计；最后采用结构化编程实现代码。

结构化方法把软件开发工作划分成结构化分析、结构化设计和结构化实现几个阶段，

每个阶段相对独立，而且是按照一定的顺序完成各阶段的任务。每个阶段的开始和结束都有严格的标准；每个阶段结束时都要进行严格的技术审查和管理复审。从技术和管理两方面对每个阶段的开发成果进行检查和评审，通过评审之后，开发流程才能进入下一个阶段；如果没有通过评审，则必须进行必要的返工，并且返工后还要再次经过审查。

结构化设计方法又可以细分为：面向数据流的设计方法、面向数据结构的设计方法。

2. 结构化方法的缺点

结构化方法的主要问题是构造的软件系统不够稳定。结构化方法以功能分解为基础，而功能又经常发生变化，必然导致软件构架不稳定；另外从数据流图到软件结构图之间的过渡有明显的断层，导致设计回溯需求有困难。但由于该方法非常简单、实用，并可有效地控制系统的复杂度，至今仍然有许多软件开发机构在使用结构化方法，或改进的结构化方法。

目前，结构化方法仍然是人们在开发软件时使用得十分广泛的软件工程方法学。这种方法学历史悠久，为广大软件工程师所熟悉，而且在开发某些类型的软件时也比较有效，因此，在相当长的一段时期内这种方法学还会有生命力。

结构化方法适合开发中小型软件系统。

二、面向对象的软件工程

面向对象的软件工程采用面向对象的方法开发软件。面向对象方法学用对象分解取代了结构化方法中的功能分解，该方法通过对象相互协作完成系统功能。对象之间的协作是通过消息来完成的。

面向对象方法学具有下述四个要点：

(1) 对象：对象把数据及在数据上的操作封装在一起。面向对象程序是由对象、接口、类组成的，复杂对象由比较简单的对象组合而成。

(2) 类：将系统中的所有对象划分成类，每个类都定义了一组属性和一组操作。类是对象的模板，对象是类的实例，对象是以类为模板创建出来的。

(3) 继承：系统中的所有类构成一个类层次树。按照父类（或称为基类）与子类（或称为派生类）的关系，把若干个相关类组成一个树结构的系统，在类层次结构中，下层子类自动拥有上层父类中定义的属性和操作，这种现象称为继承。

(4) 对象通过发送和接收消息实现交互：面向对象方法学的出发点和基本原则，是尽可能模拟人类的思维方式，使开发软件的方法与过程尽可能接近人类认识世界、解决问题的方法与过程，从而使得描述问题的问题空间与实现解法的解空间在结构上尽可能一致。

三、其他软件工程

1. 基于构件的开发方法

基于构件的开发（Component-Based Development, CBD）或基于构件的软件工程（Component-Based Software Engineering, CBSE）是一种软件开发新方法。这种开发方法主要是通过采用构件库中多个软件构件，通过组合手段高效率、高质量地构造自己的软件系统。

如果我们设计和实现的构件（Component）在某个领域中具有一定的通用性，可以在

第一章 软件工程概述

不同的计算机软件系统中复用，那么，将这些构件存储起来，就形成一个构件库。使用构件时，在构件库中进行查找，如果存在这种构件，就直接复制使用，否则就重建构件，并存储到构件库中，供以后使用。

因为，基于分布式的构件实现技术日趋成熟，基于构件的开发已经成为现今软件开发的热点技术，因此，基于构件的软件开发技术是软件工程发展方向之一。

近年来，各应用领域在构件设计与生产方面十分积极，构件的使用可以渗透到符合构件标准规范的所有系统中，不必耗费巨资从头开发自己的系统，使软件的开发者 and 使用者均可受益。因此，软件开发构件化是软件开发技术的一个发展趋势。

2. 模型驱动方法

模型驱动方法被认为是下一代软件开发方法的主流。

模型驱动的软件开发是以面向对象开发方法、UML 建模方法为基础发展起来的一种新型的软件开发方法。

模型驱动开发是 MDA 中的一部分。MDA 是模型驱动体系架构 (Model-Driven Architecture) 的简称。MDA 表示了一种模型驱动开发方法的概念框架。然而，尽管完整的 MDA 还没有成为现实，模型驱动开发方法已成为未来软件开发的可能。

第二节 软件工程的三大要素

随着软件开发方法的发展，人们逐步认识到编写程序仅是软件开发过程中的一个环节。典型的软件开发工作中，编写程序所需的工作量只占软件开发全部工作量的 10%~20%。软件开发工作应包括需求分析、软件设计、编写代码、测试代码、软件部署等几个阶段。对于传统软件工程来说，软件开发包括“结构化分析”、“结构化设计”、“结构化实现”和“结构化测试”等阶段。

对于大型软件系统的开发过程，必须确定软件开发的各个阶段，规定每一阶段的活动、产品、验收的步骤和完成准则。软件工程包含三个要素，即软件开发方法、软件开发工具和软件开发过程。

软件过程是指开发软件所需要完成的活动构成的框架。在每个活动期间，都要通过一些工具，采用一些方法、技术构造工作产品（如计划、文档、模型、代码、测试用例和手册等）。

软件开发期间，必须使用某些工具，采用某种方法（结构化方法或面向对象方法），按照某种软件过程开发软件系统。工具、方法和过程在开发软件时的关系见图 1-1 所示。



图 1-1 工具、方法和过程在开发软件时的关系

工具层为软件过程和方法提供了自动或半自动的支持；方法在技术上说明了需要如何去开发软件；软件过程提供一个活动框架，在这个框架下可以建立一个软件开发的综合计划。

1. 软件开发方法

软件开发方法指在技术上说明了需要如何去开发软件。如结构化方法、面向对象方法、构件方法、模型驱动方法等。

2. 软件开发工具

软件开发工具为软件工程方法提供了自动的或半自动的软件支撑环境。目前，已经推出了许多软件工具，这些软件工具集成起来，建立起称之为计算机辅助软件工程（CASE）的软件开发支撑系统。CASE 将各种软件工具、开发机器和一个存放开发过程信息的工程数据库组合起来形成一个软件工程环境。

3. 软件开发过程

软件开发过程指软件开发中的主要活动和顺序。过程定义了方法使用的顺序、要求交付的文档资料、为保证质量和协调变化所需要的管理及软件开发各个阶段完成的里程碑。

第三节 基本概念

在本节中，我们将介绍全书使用的主要术语和概念。一个项目由许多活动组成，每种活动则由许多任务构成。一个任务通过使用资源，创建工作产品。一个工作产品可能是一个系统、一个模型或者一个文档。资源包括参与者、时间和设备。这些概念可以用图 1-2 表示。每一个矩形表示一个概念。矩形间的线段表示概念间的不同关系。图 1-2 是用统一建模语言记号表达的。我们全书都用 UML 来表达软件模型和其他系统。

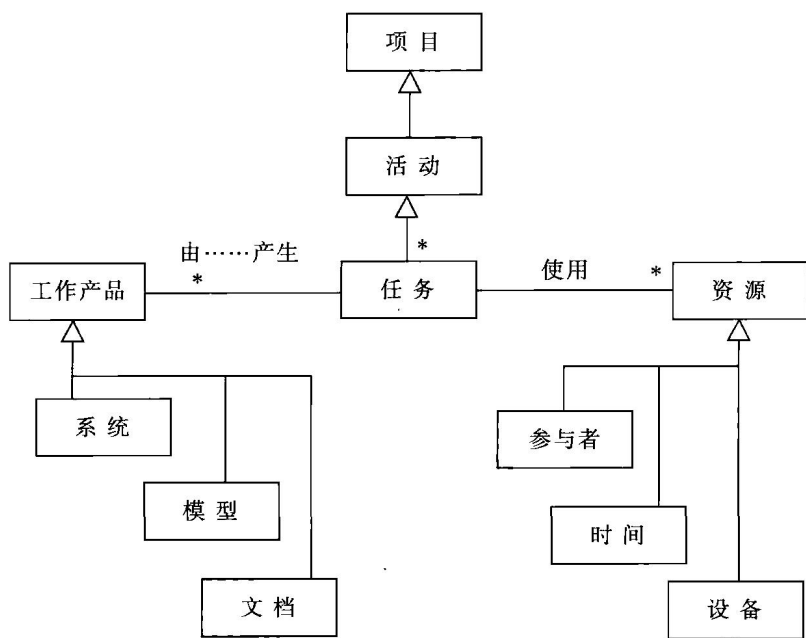


图 1-2 基本概念