

<<面向对象的分析与设计>>

图书基本信息

书名：<<面向对象的分析与设计>>

13位ISBN编号：9787302301202

10位ISBN编号：7302301204

出版时间：2013-1

出版时间：邵维忠、杨芙清 清华大学出版社 (2013-01出版)

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

<<面向对象的分析与设计>>

内容概要

《面向对象的分析与设计》是一本全面论述面向对象分析与设计方法的专著。

全书分为3部分。

第一部分为基础篇，论述面向对象方法的基本思想和主要概念，对不同的分析与设计方法进行综述，对统一建模语言（UML）和《面向对象的分析与设计》提出的方法做概要介绍。

第二部分为分析篇，详细介绍面向对象分析的全过程，围绕面向对象建模中的需求模型（用况图）、基本模型（类图）、辅助模型（包图、顺序图、活动图以及其他各种UML模型图）和模型规约，对面向对象的分析进行深入讨论，给出详细的过程指导和工程策略，并对相关的UML内容做较详细的介绍。

第三部分为设计篇，介绍如何在面向对象分析模型基础上，针对具体的实现条件进行面向对象的系统设计，包括问题域部分、人机交互部分、控制驱动部分和数据接口部分4个主要部分的设计，最后介绍如何进行构件化与系统部署。

<<面向对象的分析与设计>>

作者简介

邵维忠，北京大学信息科学技术学院教授、博士生导师。

1970年毕业于北京大学数学力学系，1979—1983年在计算机科学技术系任教并攻读硕士学位。

早期主要从事操作系统和软件工程领域的研究。

1987—1989年在新加坡国立大学参加科技合作。

回国后在杨芙清院士主持的国家“八五”、“九五”重点科技攻关课题中担任主要技术负责人。

自1991年起注重于面向对象建模方法的研究，开设了研究生课程和本科生课程，组织研制了一系列面向对象建模工具。

撰写和翻译了多部关于面向对象方法的学术著作。

在软件工程环境、面向对象方法、建模语言、软件复用、构件技术和中间件技术等领域承担了多项国家高技术研究发展计划（863）项目、国家重大基础研究（973）项目和国家自然科学基金项目。

曾获国家科技进步二等奖及国家部委级奖励多项。

杨芙清，中国科学院院士，计算机软件科学家、教育家。

北京大学信息科学技术学院教授、博士生导师。

现任北京大学信息与工程科学学部主任、软件工程国家工程研究中心首席科学家、软件与微电子学院理事长。

1958年北京大學数学力学系研究生毕业。

1957—1959年在前苏联学习程序设计和计算机软件。

1962—1964年任莫斯科杜勃纳联合核子物理研究所计算中心中国专家。

20世纪70年代主持研制了我国第一台百万次集成电路计算机多道运行操作系统和第一个全部用高级语言书写的操作系统。

1983—1999年担任北京大学计算机系系主任期间，将该系建成国内一流和国际知名的计算机科学技术研究和人才培养基地。

在国内率先倡导开展软件工程研究，创办了国内第一个软件工程学科，主持了历经多个五年计划的国家重点科技攻关项目——青鸟工程和国家863计划若干重点课题的研究。

创建了软件工程国家工程研究中心，提出“人才培养与产业建设互动”的理念，创建了以新机制、新模式办学的示范性软件学院。

曾多次获得各种荣誉称号和奖励，如全国科学大会奖、国家科技进步二等奖、电子工业部科技进步特等奖、国家级教学成果奖一等奖等十六项国家及部委级的奖励，以及全国“三八”红旗手、全国高等学校先进科技工作者、“光华科技基金”一等奖、何梁何利科学与技术进步奖、中国计算机学会终身成就奖等。

发表论文150余篇、著作10余部，培养了百余名硕士、博士和博士后。

<<面向对象的分析与设计>>

书籍目录

第一部分基础篇 第1章 面向对象方法概论 1.1 什么是面向对象 1.2 从认识论看面向对象方法的形成 1.2.1 软件开发——对事物的认识和描述 1.2.2 语言的鸿沟 1.2.3 编程语言的发展使鸿沟变窄 1.2.4 软件工程学的作用 1.2.5 面向对象是软件方法学的返璞归真 1.3 面向对象方法的基本概念 1.3.1 对象 1.3.2 类 1.3.3 封装 1.3.4 继承 1.3.5 聚合 1.3.6 关联 1.3.7 消息 1.3.8 多态性 1.3.9 其他 1.3.10 术语对照 1.4 面向对象方法的历史及现状 1.4.1 历史 1.4.2 发展到软件生存周期全过程 1.4.3 渗透到计算机软件的各个领域 第2章 不同的分析与设计方法 2.1 功能分解法 2.2 结构化方法 2.3 信息建模法 2.4 面向对象方法 2.4.1 什么是OOA 2.4.2 什么是OOD 2.4.3 面向对象方法的主要优点 2.4.4 几种典型的面向对象方法 2.4.4.1 Booch方法 2.4.4.2 Coad/Yourdon方法 2.4.4.3 Jacobson方法 2.4.4.4 Rumbaugh方法 第3章 统一建模语言UML简介 3.1 UML的背景与发展历史 3.2 UML1概况 3.2.1 UML1规范的主要组成部分 3.2.2 语言体系结构和定义方式 3.2.3 UML1的各种图和扩展机制 3.3 UML2概况 3.3.1 从UML1到UML2 3.3.2 UML2的4个规范 3.3.2.1 UML基础结构 3.3.2.2 UML上层结构 3.3.2.3 对象约束语言 3.3.2.4 UML图交换 3.3.3 UML2的各种图 第4章 本书的OOA&D方法概貌 4.1 引言 4.2 主要概念 4.2.1 主要模型元素 4.2.2 面向对象建模的主要原则 4.3 模型及其规约 4.3.1 基本模型——类图 4.3.2 需求模型——用况图 4.3.3 辅助模型——其他各种图 4.3.4 模型规约 4.3.5 OOA模型框架 4.3.6 OOD模型框架 4.4 建模过程 4.4.1 OOA过程 4.4.2 OOD过程 4.5 OOA与OOD的关系 4.5.1 一致的概念与表示法 4.5.2 不同的目标、内容和抽象层次 4.5.3 在软件生存周期中的位置 4.5.4 两种不同的分工观点 4.5.5 从模型驱动的体系结构看OOA与OOD的关系 第二部分分析篇 第5章 建立需求模型——用况图 5.1 需求分析和系统分析 5.2 基于用况的需求分析基本思路 5.3 系统边界与参与者 5.3.1 系统边界 5.3.2 参与者 5.4 用况 5.4.1 什么是用况 5.4.2 用况的内容与格式 5.4.3 如何定义用况 5.5 用况图 5.5.1 模型元素及表示法 5.5.2 几个值得商榷的问题 5.6 开发过程与建议 第6章 发现对象、定义对象类 6.1 对象和类的概念及其运用 6.1.1 概念 6.1.2 类的语义 6.1.3 在类的抽象层次建模 6.1.4 如何运用对象和类的概念 6.2 表示法 6.3 发现对象 6.3.1 研究问题域 6.3.2 正确地运用抽象原则 6.3.3 策略与启发 6.3.4 审查与筛选 6.4 对象分类 6.4.1 将对象抽象为类 6.4.2 审查与调整 6.4.3 类的命名 第7章 定义对象的属性和操作 7.1 属性和操作 7.2 表示法 7.3 定义属性 7.3.1 策略与启发 7.3.2 审查与筛选 7.3.3 推迟到OOD考虑的问题 7.3.4 属性的命名和定位 7.4 定义操作 7.4.1 行为分类 7.4.2 策略与启发 7.4.3 审查与筛选 7.4.4 认识对象的主动行为 7.4.5 描述操作流程 7.4.6 操作的命名和定位 7.5 接口的概念及其用途 第8章 建立对象间的关系 8.1 一般—特殊结构 8.1.1 相关概念 8.1.2 表示法 8.1.3 如何发现一般—特殊结构 8.1.4 审查与筛选 9.1.5 一般—特殊结构的简化 8.1.6 多继承及多态性问题 8.2 整体—部分结构 8.2.1 相关概念 8.2.2 表示法 8.2.3 如何发现整体—部分结构 8.2.4 审查与筛选 8.2.5 整体—部分结构的高级应用技巧 8.2.5.1 简化对象的定义 8.2.5.2 支持软件复用 8.2.5.3 表示数量不定的组成部分 8.2.5.4 表示动态变化的对象特征 8.2.6 两种结构之间的变通 8.3 关联 第三部分 设计篇 索引 参考文献

<<面向对象的分析与设计>>

章节摘录

版权页：插图：以相同的方式与系统进行交互的参与者可能有多个。

例如在一个超级市场中有多个收款员。

在识别收款员这种参与者时，是按照人类通常采用的抽象思维方式，把所有的收款员作为一个概念来考虑的，而不是单个地考虑每一个具体的收款员。

在用况图中，一个表示参与者的图形符号代表了同一类型的全部参与者。

在需求分析中要找出每一类参与者，但是对参与者的描述并不需要太多信息，通常只要给出它们的名称。

如果其名称不足以准确地表达这类参与者是一种什么人或者物，可以在用况图的规约中附加一点简明扼要的文字说明。

用况图中的参与者和类图中的类都是对现实世界中某些事物的抽象。

那么，二者之间有什么区别？

根本的区别是：类是系统的构成元素，位于系统边界以内，在编程时需要用程序来实现；参与者不是系统的构成元素，只是位于系统边界以外与系统交互的外界事物，不对应任何程序代码。

对现实世界中的某一事物，如何判断它应该成为系统中的对象还是作为系统以外的参与者？

可通过以下条件来判断：如果某一事物需要由系统来保存、管理和使用它的某些信息，或者由系统来实现它的某些行为，则它应该被抽象为系统中的对象，用类图中的类来表示；如果某一事物将与系统直接进行交互，以使用系统的某些功能，则它应该被看成系统的参与者，在用况图中表示；如果某一事物同时具备以上两个条件，则它既要被抽象为系统中的对象，又要作为系统的参与者，在类图和用况图中都要加以表示；如果某一事物不具备上述任何条件，则它与系统没有任何关系，应该被忽略。

发现参与者的基本思路是：分析用户所要求的每一项功能是由哪些人员、设备或外系统来使用的，或者说每一项功能的执行需要哪些人员、设备或外系统与系统进行信息交互。

这些与系统交互的人员、设备或外系统就是系统的参与者。

以下分别讨论3类参与者的发现策略。

1.人员 首先从接受系统服务的人员中发现参与者。

找出哪些人员是系统的直接使用者。

注意，这里强调的是“直接使用”。

间接地使用系统的人不能看成参与者。

例如，公司的经理如果只是阅读秘书从系统获得的报表，而自己并不直接操纵系统，那么，参与者应该是秘书而不是经理。

<<面向对象的分析与设计>>

编辑推荐

《面向对象的分析与设计》可供计算机软件领域的研究人员和工程技术人员，计算机软件专业的教师、研究生和高年级本科生阅读参考。

<<面向对象的分析与设计>>

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>