

全国普通高等院校
信息管理与信息系统专业规划教材

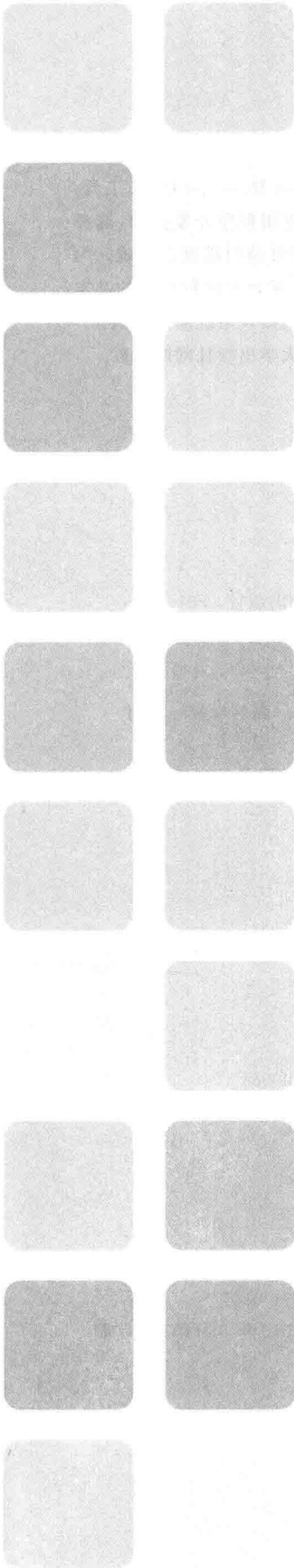
企业级典型Web 信息系统项目实战

李洪波 邹海林 主编

杨延村 李洪国 王惠敏 副主编

清华大学出版社





全国普通高等院校
信息管理与信息系统专业规划教材

企业级典型Web 信息系统项目实战

李洪波 邹海林 主编

杨延村 李洪国 王惠敏 副主编

清华大学出版社
北京

内 容 简 介

本书综合 PC Web 应用和移动应用,运用面向对象开发方法,依托 IBM Relation Rose 7.0 UML 工具,以移动扫描盘点系统为牵引,进行面向对象系统的建模,利用 ASP.NET Web 应用程序开发技术,选择 Microsoft Visual Studio 2010 开发工具,基于 SAP 8.5 系统平台,开发 B/S 结构的移动扫描盘点系统。为实现多点快速接入,利用 Web Service 技术,本书阐述了在 SAP 8.5 ERP 管理软件平台上进行开发的基本思路和具体实现,面向企业级应用,给出了移动扫描盘点系统的发布方法、部署方案以及系统测试知识。

本书具有配套的电子课件、源程序、案例数据库和自训题参考答案,可从清华大学出版社网站下载。

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

企业级典型 Web 信息系统项目实战/李洪波,邹海林主编. —北京:清华大学出版社,2015

全国普通高等院校信息管理与信息系统专业规划教材

ISBN 978-7-302-39543-0

I. ①企… II. ①李… ②邹… III. ①商业银行—计算机网络—项目开发—高等学校—教材
IV. ①F830.49

中国版本图书馆 CIP 数据核字(2015)第 041432 号

责任编辑:张 玥 薛 阳

封面设计:常雪影

责任校对:时翠兰

责任印制:何 芊

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:北京密云胶印厂

经 销:全国新华书店

开 本:185mm×260mm

印 张:24.25

字 数:571 千字

版 次:2015 年 8 月第 1 版

印 次:2015 年 8 月第 1 次印刷

印 数:1~2000

定 价:59.50 元

产品编号:055006-01

前言

当今新技术一日千里,互联网无处不在,移动应用无时不在。PC Web 应用和移动应用有机结合,是企业级信息系统实际项目开发的必然所向。信息系统分析和设计作为信息管理与信息系统专业、软件工程专业的必修课程。及时捕获时代技术新特征,与时俱进,调整课程内容和结构,是本书编写的出发点之一。此外,本书讲述企业必需的面向对象理论知识,训练社会亟须的互联网应用开发实用技术,达到培养理论与实践兼具、PC Web 应用和移动应用兼备、编程和测试互通、基础与工程融合的技术型、复合型开发人才的目标,为应用型人才培养铺路架桥。

本着即学即用的原则,本书先进行必需的理论或技术铺垫,然后辅以一个小实例,最后集成到移动扫描盘点系统项目中。全书按照分析、设计、编程、部署、测试的脉络进行组织。

全书共 4 篇。第一篇本着面向对象无处不在的观点,铺垫面向对象系统开发方法、分析和设计的基础内容,然后运用主流的 UML 建模工具 IBM Rational Rose,对移动扫描盘点系统进行面向对象系统的分析与设计。

第二篇对移动扫描盘点系统进行了面向对象的开发。基于 SQL Server 2008 R2,涵盖了 SAP 数据库的相关表的引用、专用数据库的设计与编程。然后,运用 Microsoft Visual Studio 2010 开发、发布与部署 Web Service。基于 Web Service,运用 ASP.NET 和开源 LigerUI 技术,运用 Web Form 架构,给出了 VS 2010 环境下 PC Web 应用程序的详细开发过程。同样,基于 Web Service 给出了 Pocket PC 2003 SE 智能移动设备开发环境的配置、应用程序的编写与发布的详细步骤和方法。

第三篇对系统部署涉及的软件环境、网络环境、版本环境、Web 环境配置进行了具体的论述。

第四篇对系统测试的白盒测试和黑盒测试进行了充分必要的讲述。

本书配套的电子资源有案例数据库、源程序、自训题解答和 PPT 课件。案例数据库对应的目录名为 Database,其中含有模拟 SAP 数据库、移动应用专用数据库和商业银行数据库。这些数据库均为 SQL Server 2008 R2 版本,通过附加操作完成。源程序对应的目录名为 SourceProgram,其中含有 MobileCountC、MobileTerminalApplication、PCWebMobileCount 和 SystemModeling 4 个文件夹,分别对应 Web Service、移动终端应用程序、PC Web 应用程序和 UML 建模文件。习题解答对应的文件夹为 PracticeAnswer,其中含有的子文件夹名称的首字母为数字,表示章号。

本书由李洪波布局谋篇并对所有模块进行开发、调试、部署与测试,全书由邹海林教授审阅。此外,杨延村完成部分文档的收集和部分课件制作,李洪国进行了专用数据库的开发,王惠敏进行了商业银行系统的需求分析和自训题设计。本书自训题解答的数据

库部分由李洪波完成,其余由信息管理与信息系统专业 2012 级的同学完成。自训题的移动应用程序由李笑天完成,Web Service 由齐米完成,MVC PC Web 应用程序由赵鹏程和齐米完成。

编者

2015 年 6 月于烟台



第一篇 面向对象系统的分析与设计

第 1 章 面向对象的开发方法	3
1.1 结构化方法和面向对象方法的比较	3
1.1.1 基本思想的比较	3
1.1.2 基本原则的比较	4
1.1.3 优点的比较	5
1.1.4 缺点的比较	5
1.2 面向对象分析与设计	7
1.2.1 面向对象分析方法的基本概念	8
1.2.2 面向对象系统分析与设计的阶段、步骤与优越性	9
1.2.3 面向对象软件开发和过程	12
1.3 面向对象系统分析与设计的建模语言 UML	35
1.3.1 UML 发展历史	35
1.3.2 UML 建模的主要内容	37
1.3.3 UML 的主要特点和应用领域	41
第 2 章 面向对象系统分析	43
2.1 基本概念	43
2.1.1 概述	43
2.1.2 相关概念	44
2.2 UML 分析工具	48
2.3 需求分析	52
2.3.1 需求分析的重要性	53
2.3.2 业务需求建模	54
2.3.3 需求建模	57
2.4 系统分析建模	65
2.4.1 系统分析及分析过程	65
2.4.2 静态分析	66
2.4.3 动态分析	69
2.5 系统分析报告	72
第 3 章 面向对象的系统设计	73
3.1 设计概述	73

3.1.1	设计任务	73
3.1.2	基本设计准则	74
3.2	系统构架设计	76
3.2.1	系统构架设计概述	76
3.2.2	子系统设计	80
3.2.3	包的设计	81
3.3	设计类的建立	82
3.3.1	初步设计类图的建立	82
3.3.2	交互图设计	84
3.3.3	设计类图的完善过程	87
3.4	面向对象的数据库设计	90
3.5	系统设计报告	94
第 4 章	移动扫描盘点系统的 UML 建模	96
4.1	需求概述	96
4.2	业务需求建模	97
4.2.1	用例分析	97
4.2.2	静态分析	99
4.2.3	动态分析	106
4.2.4	部署图	109
第 5 章	商业银行管理系统的 UML 建模自训	110
5.1	需求分析	110
5.2	自训题目	111

第二篇 移动扫描盘点系统开发

第 6 章	数据库应用开发	115
6.1	SAP 平台数据库表的修改与引用	115
6.2	移动扫描盘点系统专用数据库的开发	116
6.2.1	数据库模式	116
6.2.2	数据库编程	117
6.3	自训题	120
第 7 章	Web Service 开发	121
7.1	Web Service 概述	121
7.1.1	发展趋势	121
7.1.2	技术支持	122
7.1.3	软件支持	123
7.1.4	应用	124
7.2	VS 2010 下创建与引用 Web Service	127

7.2.1	.NET Framework 4.0 框架下创建与引用 Web Service	128
7.2.2	.NET Framework 2.0\3.0\3.5 下创建 Web Service	135
7.3	移动扫描盘点 Web Service 开发	137
7.3.1	新建 Service1 Web 服务	137
7.3.2	MobileCount. CS	139
7.3.3	Service1. asmx. cs	167
7.3.4	Web Service1 的发布	177
7.4	SAP Business One DI API 8.82 相关技术	187
7.4.1	相关对象	187
7.4.2	SAP Business One Service 启动	190
7.4.3	添加与查看自定义的 Password 字段	193
7.4.4	Web Service 中引用 SAP Business One DI Server API 组件	195
第 8 章	PC Web 应用开发	197
8.1	ASP.NET Web Form 基础	197
8.1.1	Web Form 简介	197
8.1.2	Web Form 网页	198
8.1.3	Web Form 控件	200
8.1.4	Web Form 事件	203
8.1.5	ASP.NET Web 窗体	204
8.1.6	ASP.NET——TextBox 控件	205
8.1.7	ASP.NET——Button 控件	206
8.1.8	ASP.NET 数据绑定	207
8.1.9	ASP.NET 2.0 母版页	224
8.1.10	ASP.NET 2.0 导航	225
8.2	LigerUI	227
8.2.1	布局组件 ligerLayout	227
8.2.2	面板组件 ligerAccordion	230
8.2.3	Tab 组件 ligerTab	232
8.2.4	树组件 ligerTree	236
8.2.5	ligerGrid 组件	246
8.2.6	综合示例	265
8.3	PC Web 应用开发	273
8.3.1	首页	273
8.3.2	主页	277
8.3.3	盘点任务页	281
8.3.4	盘点明细页	300
8.4	自训题	317

第 9 章 移动应用开发	318
9.1 配置 Pocket PC 2003 SE 设备仿真器	318
9.2 编写 Pocket PC 2003 SE 智能设备应用程序	320
9.3 仿真运行与调试 Pocket PC 2003 SE 智能设备应用程序	338
9.4 自训题	344

第三篇 移动扫描盘点系统部署

第 10 章 移动盘点扫描系统的部署	347
10.1 环境配置	347
10.1.1 安装 IIS 6.0	347
10.1.2 安装 .NET Framework	347
10.1.3 组网	348
10.1.4 安装 Microsoft ActiveSync 6.1	348
10.2 系统发布	349
10.2.1 Release 版与 Debug 版的区别	349
10.2.2 WebApplication 的发布	350
10.2.3 WebConfig 文件	351

第四篇 移动扫描盘点系统测试

第 11 章 系统测试	363
11.1 白盒测试	363
11.1.1 概述	363
11.1.2 实例	366
11.2 黑盒测试	369
11.2.1 测试方法	370
11.2.2 测试流程	373
11.2.3 工具选择	374
11.2.4 功能测试	375
11.3 自训题	377
参考文献	378

第一篇 面向对象系统的 分析与设计

面向对象技术将数据模型和处理模型二者合一,将属性和方法封装在一个对象中,将信息系统看成是一起工作来完成某项任务的相互作用的对象集合;通过定义系统中所有对象类型并显示对象之间是如何通过相互作用来完成分析任务。面向对象开发方法使用对象、类和继承等机制,而且对象之间仅能通过传递消息实现彼此通信。面向对象技术具有稳定性好、可重用性好、较易开发大型软件产品、可维护性好等优点。

结构化方法是将系统看成过程的集合,过程与数据实体之间交互,过程接收输入并产生输出。面向对象方法则不再把程序看成工作在数据上的一系列过程或函数的集合,而是把程序看作是相互协作而又彼此独立的对象的集合。

从概念方面看,结构化软件是功能的集合,通过模块以及模块和模块之间的分层调用关系实现;面向对象软件是事物对象的集合,通过对象以及对象和对象之间的通信联系实现。从构成方面看,结构化软件是过程和数据的集合,以过程为中心。面向对象软件是数据和相应操作的封装,以对象为中心。从运行控制方面看,结构化软件采用顺序处理方式,由过程驱动控制;面向对象软件采用交互式、并行处理方式,由消息驱动控制。从开发方面看,结构化方法的工作重点是设计;面向对象方法的工作重点是分析;但是,在结构化方法中,分析阶段和设计阶段采用了不相吻合的表达方式,需要把在分析阶段采用的具有网络特征的数据流图转换为设计阶段采用的具有分层特征的软件结构图,在面向对象方法中设计阶段的内容是分析阶段成果的细化,则不存在这一转换问题。从应用方面看,相对而言,结构化方法更加适合数据类型比较简单的数值计算和数据统计管理软件的开发;面向对象方法更加适合大型复杂的人机交互软件的开发。

本书依托两个信息系统实战项目,分别是“基于 SAP 的移动扫描盘点系统”和“商业银行管理信息系统”,前者作为讲述背景,后者仅作自训案例。这两个系统的软件编程将面向对象的观点一贯到底。为此,在第 1 章讲述面向对象的方法学,第 2 章讲解运用 UML 进行面向对象的分析,第 3 章涉及运用 UML 展开面向对象的系统设计。

第1章 面向对象的开发方法

结构化开发方法严格按照阶段划分的建模方式进行系统的开发工作,开发周期较长;原型法符合人们的认知习惯但不适合开发大型管理信息系统;而面向对象方法是尽可能模拟人类习惯的思维方式,使开发软件的方法和过程尽可能接近人类认识世界、解决问题的方法和过程。与前两种方法相比,面向对象方法是一种全新的开发方法,它以对象为出发点,以类为基础,以继承为手段,提出一种系统开发的新思路。

1.1 结构化方法和面向对象方法的比较

结构化方法是结构化分析、结构化设计和结构化编程的总称,是最早、最传统的软件开发方法,也是迄今为止信息系统中应用最广泛、最成熟的一种,它引入了工程思想和结构化思想,使大型软件的开发和编程都得到了极大的改善。

面向对象方法不仅是一些具体的软件开发技术与策略,而且是一整套关于如何看待软件系统与现实世界的关系,用什么观点来研究问题并进行问题求解,以及如何进行软件系统构造的软件方法学。人们普遍认为面向对象方法是一种运用对象、类、继承、封装、聚合、关联、消息和多态性等概念和原则来构造系统的软件开发方法。结构化方法和面向对象方法二者的比较如下所述。

1.1.1 基本思想的比较

结构化方法的基本思想主要体现在以下三个方面。

1. 自顶向下逐步求精

把程序看成是一个逐步演化的过程,把一个系统与程序看成是一层一层的。例如,户籍管理系统可以划分为户籍迁入迁出、人员迁入迁出、户籍注销、人员注销等几个子系统,而每个子系统又划分为接收用户信息、实施处理、提供返回信息等部分。

2. 模块化

模块化即将几个系统分成几个模块,每个模块实现特定的功能,最终整个系统由这些模块组成。模块与模块之间通过接口传递信息。模块最重要的特点就是独立性,模块之间还有上下层的关系,上层模块调用下层模块来实现一些功能。

3. 语句结构化

顺序结构、分支结构、循环结构,都是常用的语句结构。

而面向对象方法的基本思想主要有:客观世界中的事物都是对象,对象之间存在一定

的关系,并且复杂对象由简单对象构成;具有相同属性和操作的对象属于一个类,对象是类的一个实例;类之间可以有层次结构,类可以有子类。其中,子类继承父类的全部属性和操作,而且子类有自己的属性和操作;类具有封装性,把类内部的属性和一些操作隐藏起来,只有公共的操作对外是可见的,对象只可通过消息来请求其他对象的操作或自己的操作;强调充分运用人在日常逻辑思维中经常采用的思想方法与原则。例如,抽象、分类、继承、聚合、封装、关联等。

1.1.2 基本原则的比较

结构化方法遵循的基本原则有以下几条。

1. 抽象原则

抽象原则是一切系统科学方法都必须遵循的基本原则,它注重把握系统的本质内容,而忽略与系统当前目标无关的内容,它是一种基本的认知过程和思维方式。

2. 分解原则

分解原则是结构化方法中最基本的原则,它是一种先总体后局部的思想原则,在构造信息系统模型时,它采用自顶向下、分层解决的方法。

3. 模块化原则

模块化原则是结构化方法最基本的分解原则的具体应用,它主要出现在结构化设计阶段中,其目标是将系统分解成具有特定功能的若干模块,从而完成系统指定的各项功能。

面向对象方法遵循的基本原则有以下几条。

1. 抽象

抽象是处理现实世界复杂性的最基本方式,在 OO 方法中它强调一个对象和其他对象相区别的本质特性,对于一个给定域确定合理的抽象集是面向对象建模的关键问题之一。

2. 封装

封装是对抽象元素的划分过程抽象,由结构和行为组成。封装用来分离抽象的原始接口和它的执行。封装也称为信息隐藏,它将一个对象的外部特征和内部的执行细节分割开来,并将后者对其他对象隐藏起来。

3. 模块化

模块化是已经被分为一系列聚集的和耦合的模块的系统特性,对于一个给定的问题,确定正确的模块集几乎与确定正确的抽象集一样困难,通常每个模块应该足够简单以便能够被完整地理解。

4. 层次

抽象集通常形成一个层次,层次是对抽象的归类和排序,在复杂的现实世界中有两种非常重要的层次,一个是类型层次,另一个是结构性层次。确定抽象的层次是基于对象的继承,它有助于在对象的继承中发现抽象间的关系,搞清问题的所在,理解问题的本质。

1.1.3 优点的比较

结构化方法在程序设计时,先考虑问题大的方面,在确定了主要方向后,再由表及里深入到问题具体的细节,由易到难,逐层解决问题,使得整个程序设计过程由模糊到清晰,由概括到具体。结构化方法强调功能抽象和模块化,采取了分块处理问题的方法,可以把一个比较复杂的问题分解为若干个容易处理解决的部分,从而降低了问题处理的难度。由于结构化方法思路清晰,条理清楚,又有效地分解了繁复的问题,使得编写程序时清晰明了,简化了编程人员繁杂的工作,而在阅读时也能够一气呵成,给人以明朗的感觉。

面向对象方法与人类习惯的思维方法一致,使得使用者和维护人员都容易理解,在用户使用不会理解困难,在软件维护中可维护性也较高,并且易于测试和调试。它的稳定性好,软件局部修改不会引起整体的变化,而且局部修改容易实现。它的可重用性好,并且由于它是把大的问题分解成相互独立的小问题处理,降低了开发的技术难度,开发工作的管理也变得容易,开发大型软件变得容易,开发成本随之降低。面向对象方法直接由问题入手,来进行系统的建模,以降低使用成本,让用户在使用一个新的软件时,能够在最短的时间里了解使用方法,能正确、清楚地使用软件。

在编写程序方面,结构化方法有着比较强的优势,它思路清晰,条理严谨,步骤整洁,便于阅读理解;它是注重功能的开发方法。而面向对象方法则更容易理解,它操作简单,界面清楚,使得软件使用变得简单。

1.1.4 缺点的比较

结构化的系统分析和设计方法在实际的管理信息系统开发中已经被成功地应用多年,至今该方法仍占据着很重要的地位,一些大型信息系统的开发仍然采用这种方法。但结构化方法在得到问题本质性的描述之前,不断分解出的结论和需要处理的信息越来越多、越来越复杂,它要求系统分析员具有一种“俯视全局”的能力,能够透过问题的表象直接把握到问题本质。当软件项目较小、系统分析员的分析能力足够高的时候,结构化方法能快速地找到最简洁、高效率的逻辑模型。结构化方法对复杂问题的帮助有限,但它有助于使用面向分析方法的系统分析员确认系统最初的高阶模型。具体来说结构化开发方法存在以下缺陷。

1. 软件重用性差

重用性是指同一事物不经修改或稍加修改就可多次重复使用的性质。软件重用性是软件工程追求的目标之一。

2. 软件可维护性差

软件工程强调软件的可维护性,强调文档资料的重要性,规定最终的软件产品应该由完整、一致的配置成分组成。在软件开发过程中,始终强调软件的可读性、可修改性和可测试性是软件的重要的质量指标。实践证明,用传统方法开发出来的软件,维护时其费用和成本仍然很高,其原因是可修改性差,维护困难,导致可维护性差。

3. 开发出的软件不能满足用户需要

用传统的结构化方法开发大型软件系统涉及各种不同领域的知识,在开发需求模糊或需求动态变化的系统时,所开发出的软件系统往往不能真正满足用户的需要。

用结构化方法开发的软件,其稳定性、可修改性和重用性都不尽人意,这是因为结构化方法的本质是功能分解,从代表目标系统整体功能的单个处理着手,自顶向下不断把复杂的处理分解为子处理,这样一层一层地分解下去,直到只剩下若干个容易实现的子处理功能为止,然后用相应的工具来描述各个最低层的处理。因此,结构化方法是围绕实现处理功能的“过程”来构造系统的。然而,用户需求的变化大部分是针对功能的,因此,这种变化对于基于过程的设计来说是灾难性的。用这种方法设计出来的系统结构常常是不稳定的,用户需求的变化往往造成系统结构的较大变化,从而需要花费很大代价才能实现这种变化。

因此,结构化开发方法并未完全解决软件危机问题,因此,IT界一直在研究和探索新的信息系统开发方法,面向对象的开发技术应运而生。例如,可视化的编程语言 Visual C++、Java、.NET Framework 改变了传统的面向过程的编程方式,建立了面向对象开发环境,大大提高了系统的开发效率。根据面向对象程序设计思想,人们提出了面向对象的开发方法,它与结构化开发相比,是一个进步。

面向对象方法试图抽象出更公用的类,因此对系统分析员抽象事物和把握最初分析方向的要求很高,常常难以控制抽象对象的层次、粒度,甚至抽象出与问题本质面目全非的对象模型,掌握难度高于结构化分析方法。面向对象方法通过信息隐藏和封装等手段屏蔽了对象内部的执行细节,控制了错误的蔓延,但发生错误时,定位故障的代价大,尤其是继承的深度很大时。对于需求变化频繁的系统,得到一个高度可复用的面向对象软件系统设计是很困难的事情。

面向对象(Oriented Object, OO)是当前计算机界关心的重点,它是 20 世纪 90 年代软件开发方法的主流。面向对象的概念和应用已超越了程序设计和软件开发,扩展到很宽的范围,如数据库系统、交互式界面、应用结构、应用平台、分布式系统、网络管理结构、CAD 技术、人工智能等领域。

OMT(Object-Oriented Modeling Technique)是由 James Rumbaugh 等人提出的,是一种面向对象的软件开发方法,以对象建模为基础,从对象的角度对系统进行分析和设计。面向对象的软件开发方法基本思想是:采用了自底向上的归纳、自顶向下的分解的方法。第一步是从问题的陈述入手,构造系统模型。系统模型建立后的工作就是按服务(Service)来分解。第二步是将对问题的描述建立成三种模型,这三种模型分别是对象模型、动态模型和功能模型。对象模型是通过描述系统中的对象、对象间的关系和每个对象类的属性和操作

来表示系统的静态数据结构。动态模型描述了系统中与时间和变化有关的内容,它说明何时发生。功能模型描述系统的数据转换。三种模型都是必需的。但是,对象模型都是最重要、最基本、最核心的。每个对象类由数据结构(属性)和操作(行为)组成,有关的数据结构(包括输入输出数据结构)都成了软件开发的依据。

从编程技术和程序设计的角度来说,采用面向对象的分析方法是将问题分解为一系列实体——这些实体被称为对象(Object),然后围绕这些实体建立数据(对象)和针对该对象而定义的接口(Interface)来组织描述程序。正是由于这种方法是从对象的角度对系统进行分析和设计。对象就是对现实世界实体的正确抽象,而具有相同或相似性质的对象的抽象就是类。对象包含数据和操作,实现了数据和操作的结合,使数据和操作封装于对象的统一体中。每个类由属性和操作组成。对象彼此之间仅能通过发送消息互相联系。类中操作的实现过程叫作方法,一个方法有方法名、参数、方法体。

面向对象方法的编程技术重点是说明类,而不是定义函数。利用类(Class)来描述对象的特性,每个类都有描述对象属性的数据和操作数据的一组方法。面向对象程序设计主要是类的设计,程序代码主要是类的描述,而构成系统核心的对象在程序运行时才创建。对象之间的相互作用称为一个对象向另一个对象发送消息。

因此,结构化方法与面向对象方法不是对立的,没有谁先进谁过时之说,不论哪一种设计方法,正确清晰的需求界定都是开发一个成功的软件系统必不可少的前提条件,所以恰当地运用方法解决问题,在运用时关注运用方法的成本和价值才是根本性的问题。

1.2 面向对象分析与设计

面向对象程序设计是一种适用于设计、开发各类软件的范型。它是将软件看成一个由对象组成的社会,这些对象具有足够的智能,能接受从其他对象发来的信息,并以适当的行为予以响应;允许低层对象从高层对象继承属性和行为。通过这样的设计思想和方法,将所模拟的现实世界中的事物直接映射到软件系统的解空间。与传统的结构式程序设计相比,面向对象程序设计吸取了结构式程序设计的一切优点(自顶向下、逐步求精的设计原则)。而二者之间的最大差别表现如下。

1. 面向对象程序采用数据抽象和信息隐藏技术

这使组成类的数据和操作是不可分割的,避免了结构式程序由于数据和过程分离引起的弊病。面向对象程序是由类定义、对象(类实例)和对象之间的动态联系组成的。而结构式程序是由结构化的数据、过程的定义以及调用过程处理相应的数据组成的。

2. 用面向对象的感知方法模拟现实世界的问题概念

这种抽象过程分为知性思维和具体思维两个阶段。知性思维是从感性材料中分解对象,抽象出一般规定,形成了对对象的普遍认识。而具体思维是从知性思维得到的一般规定中揭示的事物的深刻本质和规律,其目的是把握具体对象的多样性的统一和不同规定的综合。

1.2.1 面向对象分析方法的基本概念

面向对象设计模式是建立在对“面向对象”纯熟、深入的理解的基础上的经验性认识。掌握面向对象设计模式的前提是首先掌握“面向对象”。这需要掌握面向对象的三大机制。从编程语言直观了解面向对象,各种面向对象编程语言相互有别,但都能看到它们对面向对象三大机制的支持,即封装、继承、多态。

1. 封装

封装是数据隐藏设计思想的实现,它必须具备三个要素:一个确定的边界,对象的所有私有数据、内部程序(成员函数)细节都被固定在这个边界内部;一个接口,这个接口描述对象之间的相互作用的规格说明,即消息;对象内部的实现代码受到封装的保护,其他对象不能直接修改该对象所拥有的数据和代码。

2. 继承

表达了一种对象类之间的相交关系,它使得子类对象可以继承父类对象的特征和功能,即子类能够复用父类的现有代码。

3. 多态性

多态性就是多种表现形式,具体来说,可以用“一个对外接口,多个内在实现方法”表示。举一个例子,计算机中的堆栈可以存储各种格式的数据,包括整型,浮点型或字符型。不管存储的是何种数据,堆栈的算法实现是一样的。针对不同的数据类型,编程人员不必手工选择,只需使用统一接口名,系统可自动选择。

多态性也分为静态多态性和动态多态性两种。静态多态性是指定义在一个类或一个函数中的同名函数,它们根据参数表(类型以及个数)区别语义,并通过“静态联编”实现。例如,在一个类中定义的不同参数的构造函数。动态多态性是指定义在一个类层次中不相同类内的重载函数,它们一般具有相同的函数,因此要根据指针指向的对象所在类来区别语义,它通过“动态联编”实现。

“多态性”一词最早用于生物学,指同一种族的生物体具有相同的特性。在面向对象理论中,多态性的定义是:同一操作作用于不同的类的实例,将产生不同的执行结果,即不相同类的对象收到相同的消息时,得到不同的结果。多态性包含编译时的多态性、运行时的多态性两大类。

动态多态是指发出同样的消息被不同类型的对象接收时,有可能导致完全不同的行为。即在用户不做任何干预的环境下,类的成员函数的行为能根据调用它的对象类型自动适应性地进行调整,而且调整是发生在程序运行时。

多态是面向对象程序设计的重要特征之一,是扩展性在“继承”之后的又一重大表现。

描述面向对象程序设计的基本概念有对象、消息、类、实例、公有消息、私有消息和消息序列。

1) 对象

对象在现实世界中,就是可以感觉到的实体。每个对象具有一个特定的名字以区别于