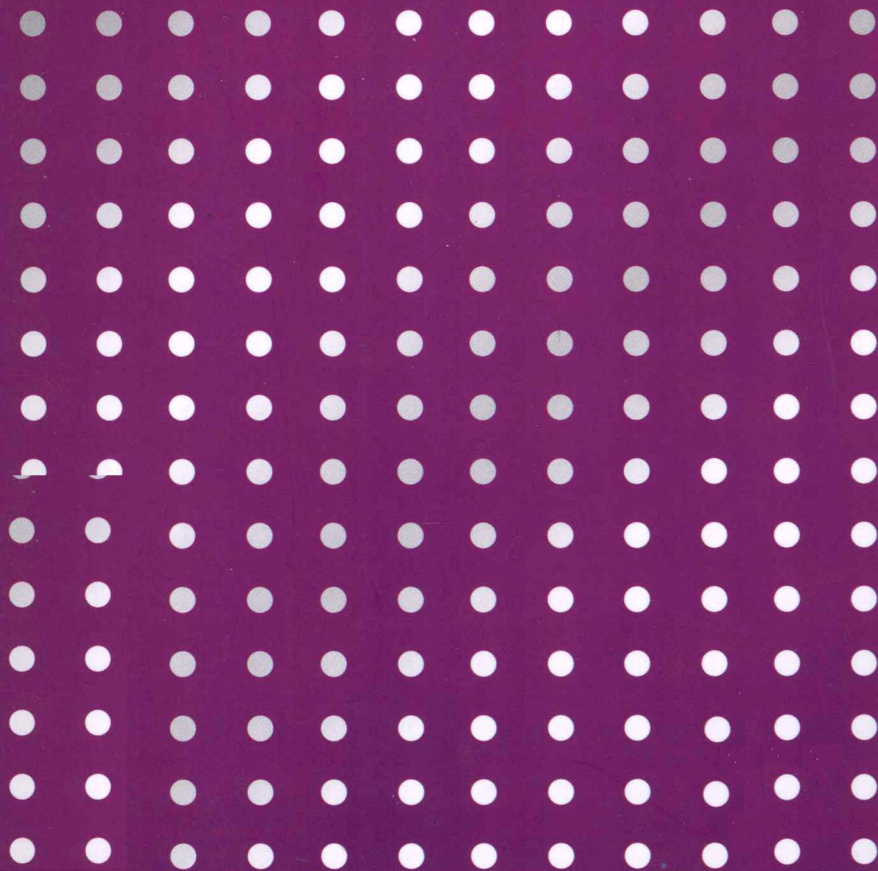


高等院校信息技术规划教材

面向对象程序设计教程

任宏萍 编著



清华大学出版社

高等院校信息技术规划教材

面向对象程序设计教程

任宏萍 编著



清华大学出版社
北京

内 容 简 介

本书从面向对象软件开发的角度出发,以 Java 语言为辅助工具,重点讲述面向对象程序设计的思想、方法、技术以及面向对象的编程原则,使读者能较全面地了解面向对象软件开发的方法和实现过程,更好地理解和掌握面向对象的程序设计。

全书分为 9 章,从面向对象软件开发概述讲起,包括面向对象的基本概念、统一建模语言 UML 部分内容、程序设计模式等,以帮助读者打下良好的面向对象程序设计基础;其次讲述了 Java 语言编程的基础知识;然后以三层程序设计模式为主线,分别讲解问题域类、图形用户界面类、数据访问类的设计与实现以及它们之间如何交互共同完成系统功能,其中包括设计、实现面向对象的重要概念:封装、继承、多态和类之间的各种关系,数据库访问以及异常处理等;最后讲解基于 Web 的应用开发。

本书的附录介绍 Java 应用开发环境的建立、MyEclipse(含 Tomcat)的安装和使用说明以及常用的面向对象程序设计的术语和词汇的解释。

本书可作为高等院校计算机相关专业及软件工程专业“面向对象程序设计”课程的教材,也可作为高校教师、软件开发人员和计算机科技人员的学习参考书。

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

面向对象程序设计教程/任宏萍编著. —北京:清华大学出版社,2012.11

(高等院校信息技术规划教材)

ISBN 978-7-302-30048-9

I. ①面… II. ①任… III. ①面向对象语言—程序设计—教材 IV. ①TP312

中国版本图书馆 CIP 数据核字(2012)第 211991 号

责任编辑:焦虹 顾冰

封面设计:常雪影

责任校对:梁毅

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

印刷者:清华大学印刷厂

装订者:三河市李旗庄少明印装厂

经 销:全国新华书店

开 本:185mm×260mm 印 张:20.75 字 数:490千字

版 次:2012年11月第1版 印 次:2012年11月第1次印刷

印 数:1~3000

定 价:33.00元

产品编号:043558-01

前言

foreword

随着计算机技术、信息技术行业的不断发展以及软件的广泛应用,面向对象的程序设计已经成为软件开发的主流程序设计方法,面向对象的软件开发也显现出前所未有的优势。

面向对象程序设计是一种程序设计范例,同时也是一种程序开发的方法。它将对象作为软件的基本单元,将数据和方法封装在对象中,以提高软件的重用性、灵活性和扩展性。面向对象程序设计问世以来,市面上出现了较多的相关书籍,但它们大多以讲述面向对象程序设计语言细节为主。学生学习后可能掌握了这个面向对象程序设计的语言,会使用这个语言的语句编写程序,但对面向对象的概念、思想、方法的应用还是不够清楚,面对实际问题写出来的软件可维护性差。因此不能说使用了面向对象的程序设计语言(如Java)编程,就实现了面向对象的软件开发。

笔者根据多年对面向对象软件开发的研究与实践以及教学经验和体会编写成此书,希望使读者对面向对象有一个较系统的认知,明确面向对象程序设计的地位,程序设计的依据是什么,什么是好的程序设计模式,面向对象程序设计的宗旨是什么,逐步建立和掌握使用面向对象认知世界的思想、方法来指导程序的实现,以提高认识问题和解决问题的能力。

全书共分9章,第1章面向对象软件开发概述,主要讲述面向对象的基本概念和特征,面向对象软件开发的基础知识和基本过程,以及程序设计模式。第2章Java基础知识,主要讲述Java语言的基础知识和基本编程技术。第3章Java的类及使用,主要讲述Java提供的常用类和包以及如何在程序中使用它们。第4~8章主要以三层程序设计模式(表示层、业务逻辑层、数据访问层)为主线,由浅入深地介绍各层中类的设计和实现,然后介绍它们之间如何交互共同完成应用系统的功能。其中贯穿如何实现面向对象的重要概念:封装、继承和多态。例如,第4章自定义类及使用,主要讲解如何设计一个问题域类(封装),如何定义使用问题域类,以及如何处理程序可能出现的异常情况,提高程序的健壮性。第5章继承与多态,主



要讲解类的继承关系如何设计,继承和多态的实现机制以及它们的作用,介绍抽象类和接口的设计与实现。第6章类之间的关系及实现,主要讲解如何建立类之间的关系以及如何编程实现这些关系。第7章图形用户界面类,主要讲述图形用户界面类的设计和实现,包括界面布局、事件处理机制以及如何编写处理事件的方法、用户界面类与问题域类如何交互等。第8章数据持久化和数据访问的实现,主要讲述利用数据库实现数据持久,讲解数据访问类的设计和实现,以及数据访问类与问题域类如何交互等。第9章Web应用系统的开发,主要讲述Web技术的基础知识,Web应用程序设计模式MVC,通过案例讲述Web应用系统的设计与实现。

本书实例丰富,解题思路清晰,步骤明确,解释详细,浅显易懂,便于读者理解和掌握面向对象的程序设计的方法和技术。

在本书出版之际,感谢华中科技大学软件学院领导和同事们的积极支持,叶倩参与了本书附录的编写,在此一并表示感谢。

由于时间关系,书中难免有误和不足之处,敬请广大读者批评指正。

编者

2012年6月

目录

Contents

第 1 章 面向对象软件开发概述	1
1.1 面向对象及软件开发	1
1.1.1 什么是面向对象	1
1.1.2 什么是面向对象程序设计	2
1.1.3 面向对象的软件开发	2
1.1.4 面向对象方法的优越性	4
1.2 面向对象的基本概念	5
1.2.1 对象	5
1.2.2 消息及消息发送	6
1.2.3 类与实例	6
1.2.4 类的特性	7
1.3 UML 简介	9
1.3.1 用例图	10
1.3.2 类图	11
1.3.3 序列图	14
1.3.4 活动图	16
1.3.5 UML 建模举例	19
1.4 程序设计模式及风格	23
1.4.1 三层程序设计模式	23
1.4.2 分层结构的优势和缺点	25
1.4.3 程序设计风格	26
1.5 本章小结	27
练习题	28
第 2 章 Java 基础知识	30
2.1 认识 Java	30
2.1.1 Java 的历史和特点	30



2.1.2	Java 开发环境和开发过程	32
2.2	标识符、关键字和分隔符	34
2.2.1	标识符	34
2.2.2	关键字	35
2.2.3	分隔符	35
2.3	变量和常量	36
2.3.1	变量	37
2.3.2	常量	38
2.4	数据类型及转换	38
2.4.1	基本类型	39
2.4.2	引用类型	39
2.4.3	数据类型的转换	40
2.5	运算符与表达式	40
2.5.1	算术运算符和表达式	40
2.5.2	逻辑运算符和表达式	41
2.6	控制流语句	42
2.6.1	条件语句	43
2.6.2	选择语句	45
2.6.3	循环语句	46
2.6.4	跳转控制语句	49
2.7	数组	53
2.7.1	一维数组	53
2.7.2	多维数组	55
2.8	本章小结	58
	练习题	58
第 3 章	Java 的类及使用	61
3.1	Java 的程序包	61
3.1.1	包的概念	61
3.1.2	创建和编译一个包	63
3.1.3	包的使用	64
3.2	字符串类 String	65
3.2.1	String 类的常用方法及使用	65
3.2.2	字符串与其他数据类型的转换	67
3.2.3	创建 String 数组	68
3.3	动态数组类 ArrayList	68
3.3.1	ArrayList 类的常用方法	69
3.3.2	ArrayList 类的使用	69

3.4	日期类 Date、Calendar 与 DateFormat	70
3.4.1	创建日期对象和日期的格式化	71
3.4.2	Calendar 类的应用	72
3.5	其他几个常用的类	74
3.5.1	包装类 Wrapper	74
3.5.2	数值计算类 Math	74
3.5.3	扫描器类 Scanner	75
3.6	什么是良好的编程习惯	77
3.7	本章小结	77
	练习题	77
第 4 章	自定义类(问题域类)	79
4.1	类的详细设计	79
4.2	类的定义	80
4.2.1	类定义的结构	80
4.2.2	声明类的属性变量	81
4.2.3	编写类的方法成员	81
4.3	类的使用	86
4.3.1	创建类的实例	86
4.3.2	调用类的方法成员	87
4.3.3	体会面向对象程序设计方法	94
4.3.4	优化自定义的类	95
4.4	静态变量和静态方法	96
4.4.1	定义静态变量和静态方法	96
4.4.2	静态变量和静态方法的应用	96
4.5	方法的重载	99
4.5.1	什么是方法的重载	99
4.5.2	重载方法的条件和使用	100
4.6	异常及异常处理	101
4.6.1	异常的分类	101
4.6.2	异常的捕获与处理	102
4.6.3	异常处理的一般原则	107
4.6.4	常见的 Java 异常类	108
4.7	本章小结	109
	练习题	109
第 5 章	继承与多态	111
5.1	类的继承	111

5.1.1	继承的案例	112
5.1.2	继承的实现	113
5.1.3	可访问修饰符	122
5.1.4	继承的应用举例——自定义异常类	123
5.2	抽象类与抽象方法	125
5.2.1	什么是抽象类和抽象方法	126
5.2.2	抽象类的应用	126
5.3	多态性	133
5.3.1	多态的概念	133
5.3.2	方法的重写及功用	133
5.3.3	实现多态的步骤	135
5.3.4	使用多态的好处	136
5.4	接口	139
5.4.1	接口的定义与实现	139
5.4.2	接口的应用	142
5.4.3	接口与继承的不同作用	148
5.4.4	接口与抽象类的比较	149
5.5	本章小结	150
	练习题	150
第 6 章	类之间的关系及实现	153
6.1	关联关系及实现	153
6.1.1	关联关系的概念及实例	153
6.1.2	实现 1 对 1 的关联关系	155
6.1.3	实现 1 对多的关联关系	158
6.2	聚合关系及实现	162
6.2.1	聚合关系的定义	163
6.2.2	聚合关系的实现	163
6.2.3	组合关系	166
6.3	依赖关系及实现	166
6.3.1	依赖关系的定义	167
6.3.2	依赖关系的实现	167
6.3.3	关联和依赖的区别	168
6.4	本章小结	168
	练习题	169
第 7 章	图形用户界面	170
7.1	Java 的 GUI 类及应用	170

7.1.1	组件和容器类	170
7.1.2	布局管理器类	174
7.2	用户界面事件的处理	178
7.2.1	用户界面事件	178
7.2.2	事件处理方法	179
7.3	自定义 GUI 类	180
7.3.1	定义 GUI 类	180
7.3.2	GUI 类的简单应用	180
7.4	用户界面类与问题域类的交互	186
7.4.1	实现交互的步骤	186
7.4.2	用户界面与业务逻辑分离的好处	189
7.5	用户界面设计的原则	190
7.6	本章小结	191
	练习题	191
第 8 章	数据持久化和数据访问的实现	193
8.1	数据持久化	193
8.2	文件及访问	194
8.2.1	文件的数据结构	194
8.2.2	Java I/O 包	195
8.2.3	创建一个文件	197
8.2.4	顺序文件的读和写	198
8.2.5	随机文件的读和写	201
8.3	数据库及 SQL	204
8.3.1	Access 数据库管理系统	204
8.3.2	建立数据库连接	208
8.3.3	数据库访问语言 SQL	212
8.3.4	Java SQL 程序包	214
8.4	数据访问的实现	217
8.4.1	数据访问类的设计	217
8.4.2	数据访问类的实现	219
8.4.3	问题域类与数据访问类的交互	226
8.5	较复杂的数据库访问的实现	229
8.5.1	访问 1 对 1 关系数据表	230
8.5.2	访问 1 对多关系数据表	241
8.6	本章小结	247
	练习题	247



第 9 章 Web 应用系统的开发	249
9.1 Web 基本知识	249
9.1.1 WWW 工作原理	250
9.1.2 URL	250
9.1.3 HTTP	251
9.1.4 HTML	252
9.1.5 Web 浏览器和 Web 服务器	254
9.2 Web 应用系统结构	255
9.2.1 C/S 结构	255
9.2.2 B/S 结构	255
9.3 Java Servlet	257
9.3.1 Servlet 的功能及生命周期	257
9.3.2 Java Servlet 包	259
9.3.3 自定义 Servlet	260
9.3.4 Servlet 运行环境	262
9.3.5 调用 Servlet 程序	263
9.4 JSP	265
9.4.1 JSP 页面结构	265
9.4.2 JSP 页面元素	266
9.4.3 JSP 与 Bean	269
9.4.4 JSP 的工作过程	270
9.5 Web 应用系统的设计模式与架构	271
9.5.1 Web 应用系统的设计模式	271
9.5.2 MVC 设计模式	272
9.5.3 Web 应用系统的架构	274
9.6 Web 应用系统开发实例	275
9.6.1 基于 MVC 的 Web 应用的实现步骤	275
9.6.2 基于 MVC 的 Web 应用开发举例	278
9.6.3 Web 应用系统软件在 TOMCAT 中的部署	286
9.7 本章小结	288
练习题	288
附录 A Java Application 开发环境的建立	289
A.1 下载和安装 MyEclipse	289
A.2 使用 MyEclipse 编写 Java 程序	291
A.2.1 创建 Java Project(项目)	291

A. 2. 2 创建自定义的类	291
A. 2. 3 编译一个类	293
A. 2. 4 运行一个类	293
A. 3 导入 Java Class	294
A. 4 导出 Java 项目	295
A. 5 调试(Debug)Java 程序	295
附录 B Java Web 应用开发环境的建立	301
B. 1 建立 Web 项目	301
B. 2 创建、编辑、编译 Web 应用文件	302
B. 3 调试运行 JSP 文件	303
B. 4 项目的发布	304
附录 C 常用术语或词汇表	305
参考文献	318

面向对象软件开发概述

随着计算机技术的迅猛发展,人们对计算机的依赖程度越来越高,期望利用计算机解决各类问题的欲望也越来越强烈,从而导致软件开发所面临的问题越来越复杂,这就需要软件开发人员掌握良好的软件开发方法,以便指导软件开发的全过程,提高软件产品的开发效率,确实保证软件产品的质量。面向对象的软件开发方法就是这样一种好的开发方法。

本章要点

- 面向对象软件开发的特征及内容;
- 面向对象的基本概念;
- 软件建模语言 UML 简介;
- 什么是好的程序设计模式和风格。

1.1 面向对象及软件开发

面向对象(Object-Oriented,OO)是一种新的软件开发和程序设计技术,它认为客观事物都是由对象(object)组成的,对象是在原事物基础上抽象的结果。任何复杂的事物都可以通过对象的某种组合结构构成。面向对象技术的发展,是朝着更加贴近人们世界的方向发展。

1.1.1 什么是面向对象

面向对象是一种思维方式,是观察和分析问题的方法。通常表现为我们是将问题按照过程方式来解决呢,还是将问题抽象为一个对象或多个对象来解决它。很多情况下,我们会不知不觉地按照过程方式来解决它,例如,先做什么,后做什么,而不是考虑将要解决的问题抽象为对象去解决它。

面向对象最重要的改进就是把世间万物都描述为对象(object),如一个学生、一本书、一台计算机等都是对象,复杂对象可以由简单对象组成,自然界包括软件系统是由一组彼此相关并能相互通信的对象所组成。例如,一个学生成绩管理系统是由学生、课程、教师和分数等对象组成,这些对象是对现实世界问题域的抽象表述。对象与对象之间通

过消息进行通信。各个对象各司其职,相互协作来完成目标任务。

面向对象也是一种用计算机语言模拟客观世界的技术,它所追求的目标是将客观世界的问题求解尽可能地简单化。通过面向对象的理念使计算机软件系统能与客观世界中的系统一一对应,使得软件开发更直观一些。

总之,面向对象不仅是一些具体的软件开发技术与策略,还是一整套关于如何看待软件系统与客观世界的关系,以及如何进行软件系统构造的方法学。

1.1.2 什么是面向对象程序设计

面向对象程序设计(Object Oriented Programming, OOP)是一种新兴的程序设计方法,或者是一种新的程序设计规范(paradigm),或者是一种计算机编程架构。其精髓在于程序的组织与构造。面向对象程序将对象作为程序的基本结构单元,程序由对象组合而成。作为动作的主体,对象将数据及对数据的操作封装在一起成为一个相对独立的实体,以简单的接口对外提供服务。

面向对象程序设计的基本思想是使用对象、类、继承、封装、消息等基本概念来进行程序设计,通过使用这些概念使面向对象的思想得到了具体的体现。

面向对象程序设计优于传统的结构化程序设计,其优越性表现在,它有希望解决软件工程的两个主要的问题——软件复杂性控制和软件生产率的提高,达到软件工程的三个主要目标:重用性、灵活性和扩展性。

1.1.3 面向对象的软件开发

面向对象的软件开发就是使用计算机语言将人们关心的现实世界所涉及的业务范围(即软件的问题域)映射到计算机世界的过程,即把一个复杂的问题分解成多个能够完成独立功能的对象(类),然后把这些对象组合起来去解决这个复杂的问题。面向对象的软件开发的本质是识别和组织问题领域中的对象模型,采用对象来描述问题域中的实体,再用程序代码模拟这些对象,使程序设计过程更自然、更直观。是使用对象、类、继承、封装、消息等基本概念来进行程序设计。

面向对象的软件开发的中心工作包括系统的需求分析、设计、实现及测试等。

(1) 需求分析(analysis)的任务是,首先应明确用户的需求,包括功能性的需求和非功能性的需求。功能性需求描述的是用户想要系统实现的功能;非功能性需求描述的是如何使这个系统能在实际环境中运行。其次将这些需求以标准化模型的形式规范地表述出来,即将用户和开发人员头脑中形成的需求以准确的文字、图、表等形式表述出来,形成双方都认可的文件。通过构造模型(问题域模型和应用模型)更加深入地理解需求。分析的目标是确定系统应该完成哪些功能,即做什么,而不是确定如何完成这些功能。分析阶段的工作是由用户和开发人员共同协作完成的。

(2) 设计(design)可分为系统架构设计和类的设计。系统设计是为解决应用问题而设计的一个高层策略——架构,制定政策以指导后续的分类设计;类设计是对分析模型进行扩展,将模型进一步细化,并考虑技术细节和限制条件,设计的目的是制定一个可行的

解决方案,即怎么做,以便能很容易地转变成编程代码。

设计阶段的重心是从应用概念转向计算机概念,开发者要选择合适的算法来实现系统主要功能。

(3) 实现(implementation)的任务是要选择一种合适的面向对象的编程语言,用面向对象程序设计的方法和技术进行编程,把软件设计模型转换成计算机可以接受的程序,本书将重点讲述这一部分。

(4) 在测试(test)阶段,测试人员利用开发人员和用户提供的测试样例分别检验编码,完成各个模块的测试(单元测试)和整个软件系统的功能测试(系统测试)。

面向对象的软件开发过程有多种形式,迭代化增量的开发过程是形式之一。这种开发过程是开发人员通过选择一组需求或变更请求来进行分析、设计、实现和单元测试,然后将其集成并发布的过程。开发人员将继续依照这种方式进行工作,再选择并实现一小部分的需求,并根据用户提出的变更请求做相应改变,直至完成分配给他们的所有工作任务。迭代特性如图 1-1 所示。

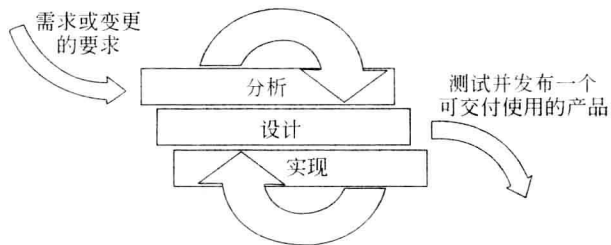


图 1-1 从开发人员角度进行的迭代

“增量”可以说是产品的一个发布。第 1 个增量实现系统的基本需求,客户对每一个增量的使用和评估都作为下一个增量发布的新特征和功能,这个过程在每一个增量发布后不断重复,直到产生最终的完善产品。增量模型强调每一个增量均发布一个可操作的产品。每一个增量可能有一个或几个迭代。这种基于迭代的反馈信息,可以准确地评估进度和调整计划,如果出现问题可及早发现并解决,使失败的风险最小化。

对于大型软件工程项目,可采用统一过程(unified process)模型来开发,它提供了在开发组织中分派任务和责任的纪律化方法。它的目标是在可预见的日程和预算的前提下,确保满足最终用户需求的高质量产品。统一过程模型如图 1-2 所示。

统一过程模型可用二维坐标来描述。横轴通过时间来组织,是过程展开的生命周期特征,体现开发过程的动态结构。在时间上被顺序分解为 4 个阶段,分别为初始阶段(inception)、细化阶段(elaboration)、构造阶段(construction)和交付阶段(transition)。纵轴以工作流程来组织,体现开发过程的静态结构,模型中有 9 个核心 workflow,即 6 个核心过程 workflow(core process workflows): 业务建模、需求、分析与设计、实施、测试和部署;3 个核心支持 workflow(core supporting workflows): 配置与变更管理、项目管理和开发环境。

统一过程模型的每个阶段可以进一步分解为迭代过程。迭代过程是导致可执行产品版本(内部和外部的)完整开发循环,是最终产品的一个子集,从一个迭代过程到另一

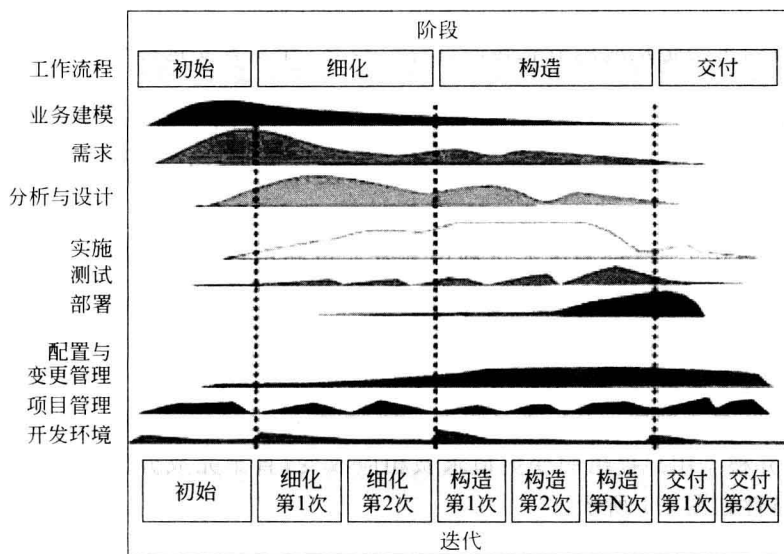


图 1-2 统一过程模型示意图

个迭代过程递增式增长形成最终的系统。迭代过程具有以下优点。

(1) 降低了在一个增量上的开发风险。如果开发人员重复某个迭代,那么损失只是这一个开发有误的迭代的花费。通过在开发早期确定风险,降低了产品无法按照既定进度进入市场的风险。

(2) 用户的需求并不能从一开始就作出完全的界定,通常是在后续阶段中不断细化的。因此,迭代过程这种模式更能适应需求的变化。

(3) 项目小组可以在开发中学习,积累经验。

(4) 高度的重用性,可获得较佳的总体质量。

软件开发过程是一个复杂的过程,很难对其进行机械的划分,我们能够做的就是把握平衡——成本和质量的平衡。

1.1.4 面向对象方法的优越性

面向对象开发方法的优越是相对传统的结构方法而言的。

传统的结构方法思考的着眼点在解决问题的流程上,是面向过程的模式。在面向过程的软件开发中,人们把焦点放在系统的功能、操作上,关心事物发展的过程,较少地涉及产生动作的主体,因此造成与人类对事物认识之间的差距。

面向对象方法思考的着眼点在于解决问题所需用到的对象及对象之间的互动上。它把一个系统定义为对象的集合,每个对象有特定的行为功能,是负有责任的角色,并维护它自己的状态(数据)。通过对象的互动(相互通信,消息传递)来完成系统的功能。

在软件发展初期,传统的结构方法没有很大的问题,但是当软件规模越来越大,变化的速度越来越快的时候,这两种观念就有了冲突。例如,在订单处理系统中,订单这个对

象是现实社会的一个普通的商业名词,它是相对稳定的,所不同的只是处理规则有所不同。在传统的语言编程中,“订单”这个名词并不是关心的重点,关心的重点放在订单的处理过程上。偏偏这个处理过程是不稳定的。一旦处理过程改变,软件也须修改,而且可能因涉及面较大,修改量也随之增加。而面向对象采用现实世界系统的思考方式,侧重于建立订单这个类,并构造订单类的体系,然后建立处理规则。所以,它与现实世界的变化规律基本一致,改变起来也就比较容易。

从本质上说,面向对象是确定动作的主体(对象)在先,执行动作在后。面向对象的分析和设计,就是先确定系统中的实体(问题域对象),再确定在这些对象上可能实施的操作。因此,面向对象模式又可称为主体-动作模式。

面向对象的开发之所以强调对象本身“是”什么,而不是它“怎样被使用”,其本质原因是:对象的使用依赖于应用程序的细节,而应用程序的细节在开发过程中经常会发生变化。随着需求的发展,对象提供的特征会比使用它的方式更加稳定。因此,构建在对象基础上的软件系统也会比较稳定。

总的来说,面向对象的开发方法有以下优点:

- (1) 把软件系统看成各种对象的集合,更接近人类的自然思维方式。
- (2) 软件需求的变动往往是功能的变动,而功能的执行者——对象一般不会有大的变化,这使得按对象设计出来的系统结构比较稳定。
- (3) 对象把数据(属性值)和行为(方法)一起封装起来,这使得方法和与之相关的数据不再分离,提高了每个对象的相对独立性,从而提高了软件的可维护性。
- (4) 可重用性。从一开始对象的产生就是为了重复利用,完成的对象将在今后的程序开发中被部分或全部地重复利用。
- (5) 可靠性。由于面向对象的程序代码包含了来源于成熟可靠的类库,因而新开发程序的新增代码明显减少,这是程序可靠性提高的一个重要原因。

1.2 面向对象的基本概念

面向对象的基本概念主要包括对象、类、消息、封装、继承、多态和消息传递等。

1.2.1 对象

对象是人们要进行研究的任何实体或事物,从最简单的整数到复杂的飞机等均可视为对象。每一个对象都具有标识符和相应的属性和行为。如汽车,它的属性有品牌、型号、颜色等;其行为有启动、加速、减速、刹车等。对象是构成软件系统的一个基本单位,它由一组属性和方法组成。属性(attribute)反映了对象的信息特征,如特点、值、状态等。而方法(method)则是用来定义改变属性状态的操作以及对对象行为的描述,即对象能做的工作。一个软件系统通常有:

- 用户界面对象,如菜单、按钮、文本框,它们的属性和方法如表 1-1 所示;
- 问题域对象,如在一个订单处理系统中的对象客户、订单和产品,它们的属性和方法如表 1-2 所示。