

可下载教学资料

<http://www.tup.tsinghua.edu.cn>



高等学校教材
计算机科学与技术

实用软件设计模式教程

习题解答与实验指导

徐宏喆 王建新 朝鲁蒙 等 编著

清华大学出版社



高等学校教材

计算机科学与技术

实用软件设计模式教程 习题解答与实验指导

清华大学出版社
北京

内 容 简 介

本书是《实用软件设计模式教程》(徐宏喆等编著,清华大学出版社出版)的配套辅导和实验指导,主要为教材各章的课后习题提供参考答案,并为学生上机实验提供指导。本书旨在通过难易适度的习题,加深学生对面向对象设计方法的理解,培养学生良好的编程风格,并使学生能够熟练地应用 VS2008 集成开发环境,掌握常用的开发和调试方法及工具。

本书分为两大部分,第一部分为与《实用软件设计模式教程》教材相配套的知识点总结和习题解答;第二部分为实验指导,一共包括 7 个实验,指导学生熟悉 VS2008 开发环境,并通过几个典型的关于设计模式的实验指导,加深学生对设计模式的理解和认识。

本书是为有一定编程基础的读者编写的,可以作为大学计算机专业本科生、研究生学习设计模式的基础教材,也可以作为从事软件研究和开发相关人员的参考书。

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

实用软件设计模式教程习题解答与实验指导 / 徐宏喆等编著. —北京:清华大学出版社, 2010.11

(高等学校教材·计算机科学与技术)

ISBN 978-7-302-23356-5

I. ①实… II. ①徐… III. ①软件设计—高等学校—教学参考资料 IV. ①TP311.5

中国版本图书馆 CIP 数据核字(2010)第 152208 号

责任编辑:郑寅堃 李玮琪

责任校对:白 蕾

责任印制:李红英

出版发行:清华大学出版社

地 址:北京清华大学学研大厦 A 座

<http://www.tup.com.cn>

邮 编:100084

社 总 机:010-62770175

邮 购:010-62786544

投稿与读者服务:010-62795954, jsjic@tup.tsinghua.edu.cn

质 量 反 馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

印 装 者:北京国马印刷厂

经 销:全国新华书店

开 本:185×260 印 张:9 字 数:215 千字

版 次:2010 年 11 月第 1 版 印 次:2010 年 11 月第 1 次印刷

印 数:1~3000

定 价:16.00 元

产品编号:038341-01

设计模式 (Design Pattern) 是一套被反复使用、多数人知晓的、经过分类编目的代码设计经验的总结。使用设计模式是为了可重用代码、让代码更容易被他人理解、保证代码的可靠性。本书是《实用软件设计模式》(徐宏喆等编著,清华大学出版社出版)的配套用书,包括习题解答和实验指导两大部分。

第一部分,习题解答。该部分与教材的章节安排完全对应,详细总结了各章节的重点难点,这些重点难点的总结,可以帮助读者加深对教程基本知识点的理解和认识。此外,该部分还精选了一些习题并给出了参考答案,这些习题难度适中,内容紧扣教材,以考查学生基础知识为主。本书对每一道编程题目都进行了注释,力求使程序结构清晰易懂。通过本书的学习,读者能够较全面地掌握面向对象设计模式的相关知识,并能较快地提高自己的编程水平。

第二部分,实验指导。该部分包括 7 个实验,每个实验 4 个学时。

实验 1 熟悉 Visual Studio 2008 开发环境。本实验介绍了 VS2008 集成开发环境,旨在为初次接触 VS2008 的读者提供一些必要的指导。

实验 2 桥接模式,介绍了桥接模式在绘图程序中的作用。

实验 3 装饰者模式,介绍了装饰者模式在网站新闻模块中的应用。

实验 4 外观模式,介绍了外观模式在网络流量计费系统中的应用。

实验 5 观察者模式,模拟 RSS 订阅和获取新闻的整个过程。

实验 6 职责链模式,使用职责链模式创建了一个简单的、通用的浏览器程序。

实验 7 命令模式,介绍了命令模式在模拟复合文档的处理程序的创建过程中的应用。

以上每个实验都分为实验目的、实验任务、实验内容和步骤以及实验小结 4 个单元,以帮助读者深刻体会设计模式的用处和好处,加深读者对设计模式的理解和感悟。读者如果能按照实验的要求,在编程实践中仔细学习体会这 6 个设计模式,认真感悟面向对象设计的思想和原则,必定会受益匪浅。书中所有程序都经过上机调试,可以确保编译通过。

本书可以作为计算机专业人员的参考用书，特别适合于计算机本科和研究生的学生、参加国家高等教育自学考试的考生，以及参加计算机等级考试的考生使用。

由于面向对象设计模式博大精深，加之时间仓促，作者水平有限，书中难免存在不足之处，恳请广大读者批评指正。

编 者

2010年8月

第一部分 习题解答	1
第1章 绪论	2
第2章 面向对象方法概述	10
第3章 UML 建模语言	16
第4章 软件设计模式基础	21
第5章 创建型设计模式	25
第6章 结构型设计模式	34
第7章 行为型设计模式	46
第8章 软件架构与建模技术	63
第9章 软件架构集成技术	70
第二部分 实验指导	79
实验1 熟悉 Visual Studio 2008 开发环境	80
实验2 桥接模式	92
实验3 装饰者模式	98
实验4 外观模式	104
实验5 观察者模式	111
实验6 职责链模式	117
实验7 命令模式	129

第一章

計 量

第一章 計量

第一部分 习题解答

第1章

绪 论

1.1 本章要点

软件危机是指计算机软件在它的开发和维护过程中所遇到的一系列严重问题。主要包括两方面问题：如何开发软件，怎样满足人们对软件日益增长的需要；如何维护数量不断膨胀的已有软件。

软件危机的主要表现有如下几个方面。

- (1) 对软件开发成本和进度的估计常常很不准确。
- (2) 用户对“已完成的”软件系统不满意的现象经常发生。
- (3) 软件产品的质量常常靠不住。
- (4) 软件常常是不可维护的。
- (5) 软件通常没有适当的文档资料。
- (6) 软件成本在计算机系统总成本中所占比例逐年上升。

软件危机的原因主要有两方面：一方面与软件本身的特点有关；另一方面来自于软件开发人员的一些弱点。软件生产过程中的这种知识密集和人力密集的特点是软件危机的根源所在。从软件开发危机的种种表现和软件开发作为逻辑产品的特殊性可以发现软件开发危机的原因如下：用户需求不明确、缺乏正确的理论指导、软件开发的规模越来越大和软件开发的复杂度越来越高。

软件工程学和程序设计方法学就是在解决软件危机的过程中逐渐形成的两门学科。软件工程学从软件开发的应用角度出发，主要研究软件生产的客观规律性，建立与系统化软件生产有关的概念、原则、方法、技术和工具，指导和支持软件系统的生产活动，以期达到降低软件生产成本、改进软件产品质量、提高软件生产率水平的目标。程序设计方法学主要运用数学知识，研究算法与逻辑方法，从微观的角度研究一些较小的程序模块的正确性。

最早提出的软件方法是结构化程序设计方法，其核心就是模块化。结构化方法主张使用顺序、选择、循环三种基本结构来嵌套联结成具有复杂层次的“结构化程序”，其设计思想是“自顶而下，逐步求精”。

结构化系统分析与设计 (SSA&D) 的基本思想是：用系统的思想，系统工程的方法，按用户至上的原则，结构化、模块化、自顶向下对系统进行分析与设计。软件分析和设计的基本阶段可以划分为需求分析、概要设计、详细设计三个阶段。

结构化分析 (Structured Analysis, SA) 是面向数据流的需求分析方法。其基本思想是“分解”和“抽象”。数据流图 (Data Flow Diagram, DFD) 是结构化分析的最基本工具，它包括如下基本元素：数据流、数据存储、数据处理的和数据字典。

结构化设计 (Structured Design, SD) 方法包括如下两种：面向数据流的设计和面向数据结构的设计。

面向对象方法是最常用的软件方法，其最基础的概念就是类和对象。基本特征有抽象、继承、封装、多态性。面向对象开发技术因为拥有与人类的思维习惯类似、具有良好的稳定性、可重用性好等技术优势，目前已经成为最流行的软件开发技术。

将设计模式引入软件设计和开发过程的目的在于充分利用已有的软件开发经验。软件设计模式是一套被反复使用、多数人知晓的、经过分类编目的代码设计经验的总结。使用设计模式是为了可重用代码，让代码更容易被他人理解，保证代码的可靠性。

设计模式实现了软件代码级别的复用，但随着软件系统越来越复杂和庞大，代码级别的软件复用已经远远不能满足大型软件开发的需求。而对整个系统的设计和描述变得越来越重要，由此便产生了软件架构的概念。软件架构描述的对象是直接构成系统的抽象组件。各个组件之间的连接则明确和相对细致地描述组件之间的通信。在实现阶段，这些抽象组件被细化为实际的组件，比如具体某个类或者对象。

从提出软件开发计划的那一刻开始直到软件在实际应用中完全报废为止，可以认为是一个完整的软件生存周期。软件生存周期的提出是为了更好地管理、维护和升级软件。一种典型的阶段划分方法为问题定义、可行性分析和项目开发计划、需求分析、概要设计、详细设计、编码与测试、运行与维护。

问题定义是将一个软件构想酝酿形成一个目标明确的主题，是软件开发的起始阶段。问题定义阶段的基本任务是要确定“软件要解决的问题是什么”。

可行性分析和项目开发计划，这两个活动通常被整合在一起进行，要做的事情就是回答“需要做什么？要如何去做？可不可能完成？”

需求分析是指在建立一个新的或改变一个现存软件系统时描写新系统的目的、范围、定义和功能时所要做的一切工作。需求分析是软件工程中的一个最关键的过程。

概要设计描述软件的整体体系结构。概要设计的基本任务如下：设计软件系统结构、设计数据结构和数据库、编写概要设计文档、评审。

详细设计就是把在概要设计里所划分出来的模块要实现的功能，用相应的设计工具详细地描述出实现步骤来，也就是写出代码的算法。在详细设计里所有的表述无论是语言或是图表，都应做到有精确的唯一解释。

编码阶段是将软件设计的结果转换成计算机可运行的程序代码。

软件维护是软件生存周期中非常重要的一个阶段，但是它的重要性往往被人们忽视。国外很多软件开发组织都把 60% 以上的人力用于维护已运行的软件，它的重要性不能被忽略。

最传统的软件开发模型是瀑布模型，随着软件工程技术不断发展，在软件开发实践

中，出现了许多新的或改进的软件开发过程模型，较常见的有瀑布模型、原型模型、喷泉模型、螺旋模型等。

瀑布模型的核心思想是按工序将问题化简，将功能的实现与设计分开，便于分工协作，即采用结构化的分析与设计方法将逻辑实现和物理实现分开。瀑布模型的示意图如图 1.1.1 所示。

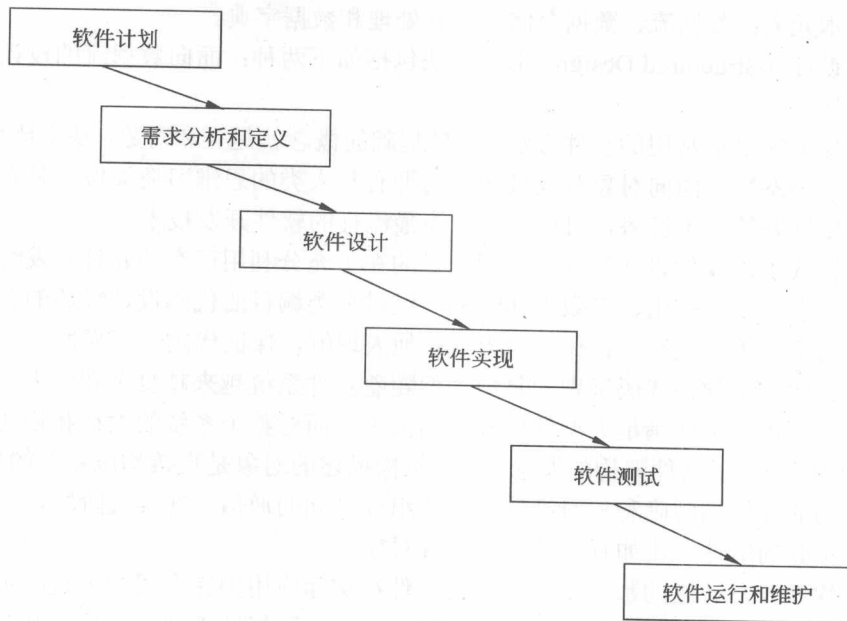


图 1.1.1 瀑布模型示意图

原型模型也叫样品模型，其思想是：先借用已有系统作为原型模型，通过“样品”不断改进，得到的最后产品就是用户所需要的。其示意图如图 1.1.2 所示。

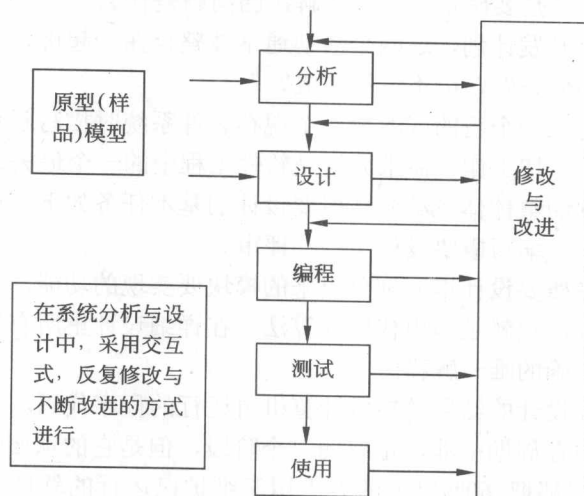


图 1.1.2 原型模型示意图

喷泉模型主要用于面向对象的软件开发项目，软件的某个部分通常被重复多次，相关对象在每次迭代中随之加入渐进的软件成分，其示意图如图 1.1.3 所示。

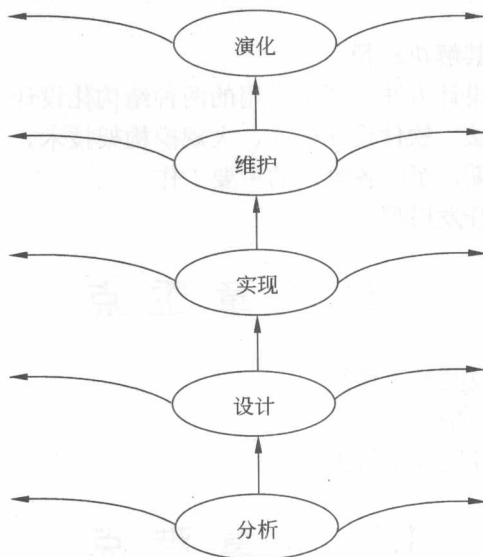


图 1.1.3 喷泉模型示意图

螺旋模型是将瀑布模型和快速原型模型结合起来，强调了其他模型所忽略的风险分析，特别适合于大型复杂系统，其示意图如图 1.1.4 所示。

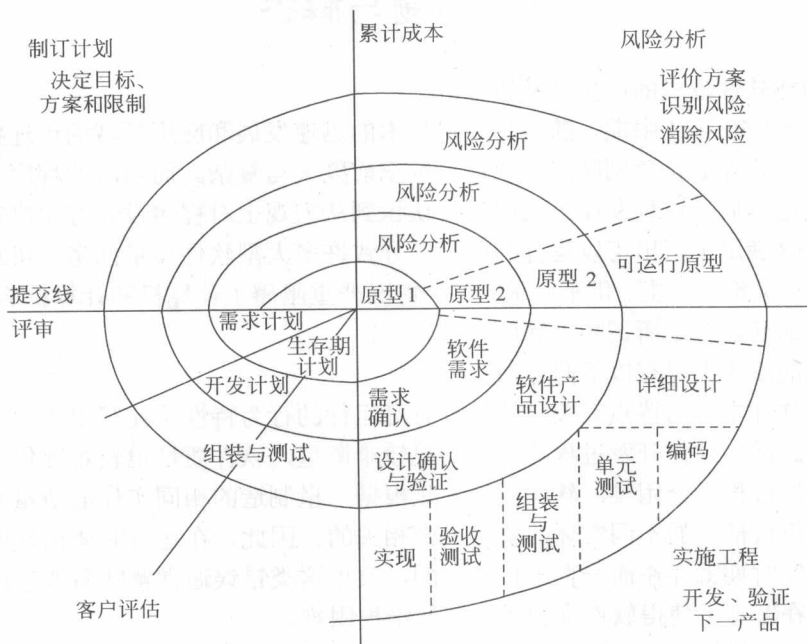


图 1.1.4 螺旋模型示意图

1.2 本章目标

- (1) 理解软件危机及其解决途径。
- (2) 理解结构化程序设计方法,了解常用的两种结构化设计方法。
- (3) 了解面向对象方法、软件设计模式、大规模构架技术。
- (4) 理解软件生存周期,了解各阶段的主要工作。
- (5) 了解常用的软件开发模型。

1.3 本章重点

- (1) 软件危机的产生及其解决途径。
- (2) 软件设计模式的由来。
- (3) 几种常见的软件开发模型的优缺点。

1.4 本章难点

- (1) 两种常见的结构化设计方法。
- (2) 软件生存周期中各个阶段的主要工作。

1.5 习题与解答

- (1) 请简述软件危机的产生及其原因。

答:20世纪60年代中期,随着计算机技术的迅速发展和应用领域的迅速拓宽,软件需求迅速增长,软件数量急剧膨胀,软件系统空前庞大与复杂。而当时的程序设计与软件开发技术却远远落后于这种发展。人们没有认识到从宏观上对程序设计方法进行研究的重要性,许多人只满足于写出可以运行的程序,导致许多大型软件质量低劣、可靠性不高、可维护性差,却又价格昂贵、供不应求。这种情况严重阻碍了计算机和计算机应用的发展,这一系列严重的问题就是所谓的“软件危机”。

软件危机的原因主要有以下两方面。

一方面与软件本身的特点有关。软件具有可运行的行为特性,在写出程序代码并在计算机上试运行之前,软件开发过程的进展情况较难衡量,软件质量也较难评价,因此管理和控制软件开发过程十分困难;软件质量不是根据大量制造的相同实体的质量来度量的,而是与每一个组成部分的不同实体的质量紧密相关的。因此,在运行时所出现的软件错误几乎都是在开发时期就存在而一直未被发现的,改正这类错误通常意味着改正或修改原来的设计,这就在客观上使得软件维护远比硬件维护困难。

另一方面,危机原因来自于软件开发人员的如下弱点。

其一,软件产品是人的思维结果,因此软件生产水平最终在相当程度上取决于软件人员的教育、训练和经验的积累。

其二,对于大型软件往往需要许多人合作开发,甚至要求软件开发人员深入应用领域的问题研究,这样就需要在用户与软件人员之间以及软件开发人员之间相互通信。在此过程中难免发生理解的差异,从而导致后续错误的设计或实现,而要消除这些误解和错误往往需要付出巨大的代价。

其三,由于计算机技术和应用发展迅速,知识更新周期加快,软件开发人员经常处在变化之中,不仅需要适应硬件更新的变化,而且还要涉及日益扩大的应用领域问题研究;软件开发人员所进行的每一项软件开发几乎都必须调整自身的知识结构以适应新的问题求解的需要,而这种调整是人所固有的学习行为,难以用工具来代替。

(2) 程序设计方法学和软件工程学有何区别和联系?

答:程序设计方法学与软件工程学研究的途径和侧重点有所差别,主要表现在以下几个方面。

研究方法和途径不同。软件工程学应用的是工程方法,而程序设计方法学运用的是数学方法;软件工程学注重工程方法与工具研究,程序设计方法学则注重算法与逻辑方法的研究。

研究对象侧重点不同。软件工程的对象所指的软件,一般指“大型程序”,是一个系统;而程序设计方法学的研究对象则侧重于一些较小的具体程序模块。

软件工程学注重“宏观可用性”,程序设计方法学注重“微观正确性”。例如软件工程学研究软件的“可靠性”的方法是“软件测试”,程序设计方法学研究的内容则是程序的“正确性证明”。

随着软件技术的迅速发展,软件工程学和程序设计方法学的研究内容也在不断发展,研究的内容和方法相互渗透。事实上,人们已经很少也没有必要区分什么是软件工程学的范畴,什么是程序设计方法学的范畴了。这两条研究途径的界限逐渐地模糊化和一体化了。

一方面,程序设计方法学研究已发生了较大的变化,逐渐从“纯粹的程序”正确性证明等较老的课题转向“软件”的结构化、正确性、可靠性及软件设计方法方面的研究。另一方面,软件工程从一开始就是以程序设计方法学为基础的一门工程学科,而且在不断吸收程序设计方法学和计算机科学理论新成果和新技术。从某种意义上可以说,软件工程学实际上就是“应用设计方法学”。

(3) 简述结构化程序分析与设计的基本思想。

答:结构化程序分析与设计的基本思想是:用系统的思想,系统工程的方法,按用户至上的原则,结构化、模块化、自顶向下对统进行分析与设计。具体来说,就是先将整个系统开发过程划分出若干个相对独立的阶段,如系统规划、系统分析、系统设计、系统实施等。在前三个阶段坚持自顶向下地对系统进行结构化划分。在系统调查或理顺管理业务时,应从最顶层的管理业务入手,逐步深入到最底层。在系统分析阶段,提出新系统方案和系统设计时,先考虑系统整体的优化,然后再考虑局部的优化问题。在系统实施阶段,则应坚持自底向上的逐步实施。

(4) 什么是面向对象方法?其基本特征是什么?与结构化方法相比有什么优点和不足?

答：面向对象方法是一种把面向对象的思想应用于软件开发过程中，指导开发活动的系统方法。其基本特征是抽象、继承、封装、多态性。与结构化方法相比，它的优点如下。

与人类的思维习惯类似。面向对象技术对问题空间进行自然分割，以更接近人类思维的方式建立问题域模型，以便对客观实体进行结构模拟和行为模拟，从而使设计出的软件尽可能直接地描述现实世界。

具有良好的稳定性。运用面向对象技术，软件开发时间短，效率高，所开发的程序更稳定。由于面向对象编程的可重用性，可以在应用程序中大量采用成熟的类库，从而缩短了开发时间。

可重用性好。应用程序更易于维护、更新和升级。继承和封装使得应用程序的修改带来的影响更加局部化。

其缺点是不易于大系统的开发。

(5) 什么是设计模式？为什么需要设计模式？

答：软件设计模式（Design Pattern）是一套被反复使用、多数人知晓的、经过分类编目的代码设计经验的总结。使用设计模式是为了可重用代码、让代码更容易被他人理解、保证代码的可靠性。

(6) 软件生存周期一般包括哪几个阶段？每个阶段都完成哪些工作？

答：软件生存周期的阶段划分有多种方法。一种典型的阶段划分为：问题定义、可行性分析和项目开发计划、需求分析、概要设计、详细设计、编码与测试、运行与维护。

各阶段完成的工作如下。

问题定义：确定要为用户建立什么样的软件系统，软件叫什么样的名称。通过问题定义阶段的工作，应提交出关于问题性质、工程目标和规模等书面报告。

可行性分析和项目开发计划：本阶段要做的事情就是回答“需要做什么？要如何去？做？可不可能完成？”

需求分析：问题识别，分析与综合，制订规格说明，评审。

概要设计：设计软件系统结构，数据结构及数据库设计，编写概要设计文档，评审。

详细设计：为每个模块所要完成的功能进行具体而精确的描述，要根据功能描述再转化成精确的、结构化的软件过程描述。

编码与测试：此阶段是将软件设计的结果转换成计算机可运行的程序代码。在软件设计完成后要经过严密的测试。

运行与维护：后续维护性工作。

(7) 瀑布模型与原型模型有何区别？它们各自应用于什么场合？

答：瀑布模型将软件生命周期划分为制订计划、需求分析、软件设计、程序编写、软件测试和运行维护 6 个基本活动，并且规定了它们自上而下、相互衔接的固定次序，如同瀑布流水，逐级下落。原型模型也叫样品模型，其主要思想是：先借用已有系统作为原型模型，通过“样品”不断改进，使得最后的产品就是用户所需要的。

瀑布模型的优点：方法规范，每个阶段质量可保证，每个阶段归定的文档使错误得到及早发现和处理，容易维护。缺点：靠文档驱动，用户不能全面地认识动态的软件产品；

过于理想化,可能出现设计上的错误。适用范围:完全定义好了需求,而且没有时间压力的系统。

原型模型优点:开发后期不会因为发现了规格说明文档的错误而进行较大的返工;在设计和编码阶段发生错误的可能性比较小。缺点:开发者常常需要实现上的折衷,以使原型能够尽快工作;用户看到的并不是软件的工作版本。适用范围:用户定义了一组一般性目标,但不能标识出详细的输入、处理及输出需求;开发者可能不能确定算法的有效性、操作系统的适应性或人机交互的形式。

第2章

面向对象方法概述

2.1 本章要点

面向对象方法最基本的概念就是对象和类。对象是对客观世界中事物^①的主观认识，对象是一个独立单位，它具有自己的静态特征^②和动态特征^③。在面向对象的系统中，把具有相同静态特征和动态特征的对象抽象为类，类被看作是对象的模型。面向对象方法有 3 个显著特性。

(1) 封装性：对象中的属性和操作都被隐藏起来，只把接口暴露给外界。

(2) 继承性：对象间的一种层次关系，某个对象继承了另一个对象就具有了那个对象所有的属性和方法。

(3) 多态性：同一个操作作用在不同对象的时候表现出的不同行为。

面向对象方法从现实世界客观存在的事物特征出发，并按照符合人类自然思维方式的方法来构造系统。面向对象方法具有以下优势。

(1) 有利于系统的维护。它使用了封装的方法把易变的数据结构和处理数据的过程作为一个整体封装在对象内并加以隐藏，保证了对某个对象的修改不会影响其他对象，避免了面向过程方法中对数据的随意所带来的数据安全问题；使用了继承的方法可以轻易地对模型（类）进行进一步扩展，以适应需求变化；使用了接口的方式将维护所带来的影响局限在一定的范围之内；使用消息传递的方式使得对程序执行流程的修改更加灵活。这些具体方法的综合运用使得系统更加易于修改和完善。

(2) 有利于开发效率的提高。这主要体现在如下几方面：分析与设计的投入在编程、测试时会得到回报；面向对象方法使系统模型更易于理解；分析文档、设计文档以及源程序对应良好；功能变化引起的全局性修改较少；有利于复用。

(3) 有利于产品复用的扩展。早期的软件复用主要是代码级复用，被复用的知识专指程序，而在面向对象方法及其延伸中扩大到包括领域知识、开发经验、设计决定、体系结

① 事物可以有形的，如诸葛亮、123 班成绩单；也可以是无形的，如草船借箭方案、升序排序算法。

② 静态特征：如属性和字段。

③ 动态特征：如可操作的方法。

构、需求、设计、代码和文档等一切有关方面。

面向对象方法的开发过程包含了面向对象分析、面向对象设计以及编程实现三个阶段。

(1) 面向对象分析就是运用面向对象方法进行系统分析,提取和整理用户需求,并建立问题域精确模型的过程,它是软件生命周期的一个阶段。它强调用对象的概念对问题域中的事物进行完整的描述,刻画事物的数据特征和行为特征,同时也要如实地反映问题域中事物之间的各种关系。

在面向对象的分析中,系统分析员应该深入理解用户需求,抽象出系统的本质属性,提取系统需求规格说明,并用模型准确地表示出来。通常,面向对象分析模型包括对象模型、动态模型和功能模型,其中对象模型是最基本、最重要、最核心的。

面向对象分析有两个任务,一是形式地说明所面对的应用问题,最终成为软件系统基本构成的对象,还有系统所必须遵从的、由应用环境所决定的规则和约束。二是明确地规定构成系统的对象如何协同合作完成指定的功能。为了建立一个分析模型,也应该遵循以下基本原则。

- ① 建立信息域模型。
- ② 描述系统的所有功能。
- ③ 表示系统的全部行为。
- ④ 划分数据、功能、行为模型的层次结构,更好地表示模型的细节。
- ⑤ 采用逐步求精技术,早期模型表示问题的本质,后期模型提供实现细节。

(2) 面向对象设计则是把分析阶段得到的需求转变成符合成本和质量要求的、抽象的系统实现方案的过程。在进行面向对象设计时应当遵循以下设计准则。

- ① 模块化和抽象化原则。
- ② 信息隐藏和封装原则。
- ③ 对象的高内聚和弱耦合原则。
- ④ 可扩充性和可重用性原则。

系统设计应当首要考虑问题域部分的设计、人机交互部分的设计、任务管理部分的设计以及数据管理部分的设计。

(3) 面向对象编程实现主要包括两项工作:把面向对象设计结果翻译成某种程序设计语言写成的面向对象程序;测试并调试面向对象程序。

2.2 本章目标

- (1) 掌握面向对象的基本概念和特点。
- (2) 理解面向对象方法的优点。
- (3) 掌握面向对象方法的开发过程。

2.3 本章重点

- (1) 面向对象方法的基本概念和特点。
- (2) 采用面向对象方法的重要意义。