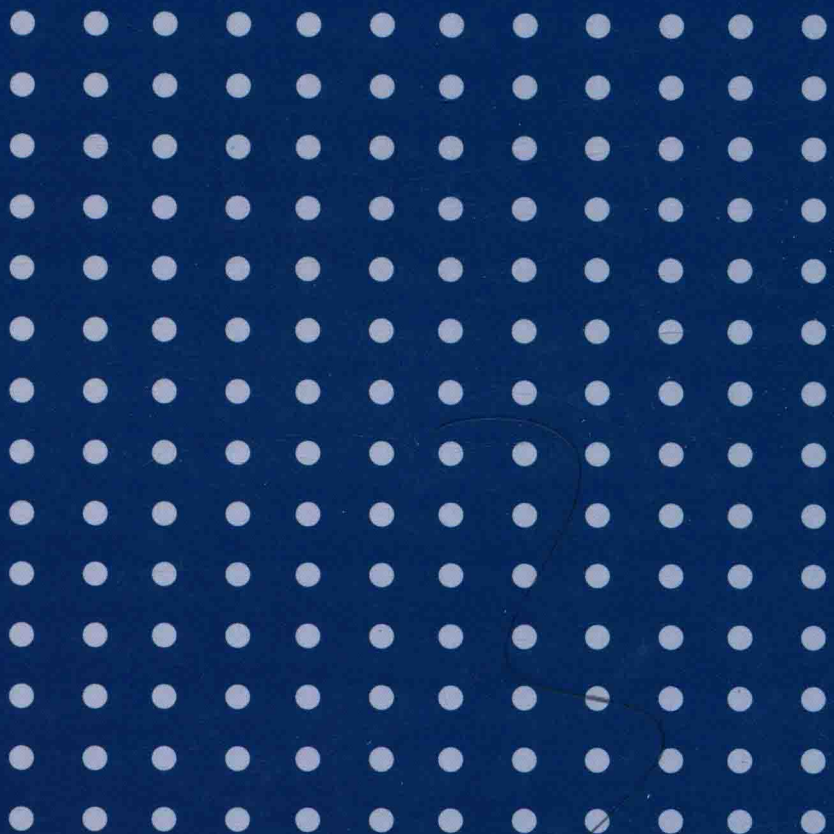
重点大学计算机专业系列教材

实用软件设计模式教程

（第2版）

徐宏喆　董丽丽　侯迪　编著



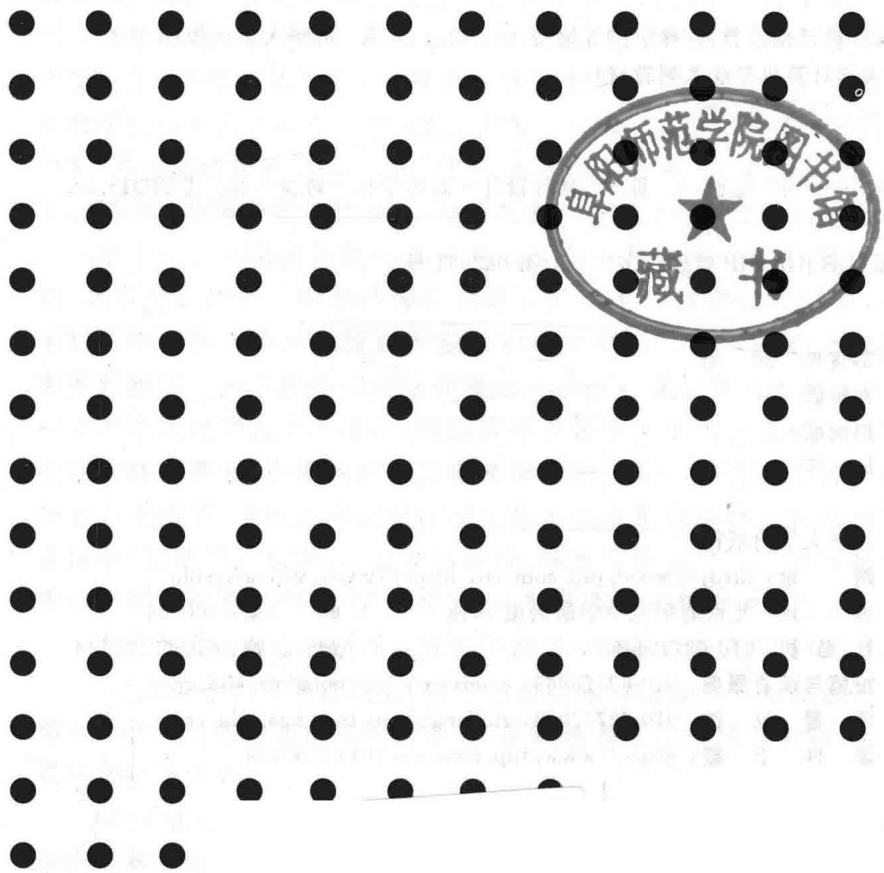
清华大学出版社

业系列教材

实用软件设计模式教程

（第2版）

徐宏喆　董丽丽　侯迪　编著



清华大学出版社
北京

内 容 简 介

设计模式是面向对象编程的热门话题之一，也是近年来国内外广泛使用和研究的热点。

本书是一本介绍软件设计模式内容及原理的教材，作者以设计模式的概念、原则、分类及构成为出发点，详细分析了24种设计模式。在介绍每种模式时，以一个软件设计开发中的实际问题为引子，探讨一般实现方法的缺陷，进而介绍新模式的结构，再以一个实际的例子展现模式的编程方法，最后对使用模式的效果进行分析，通过应用实例展示设计模式在应用系统开发实践中的应用。同时，本书紧跟业界技术发展，对最新的软件架构建模技术进行了分析和介绍。

本书是为有一定编程基础的读者编写的，内容全面，概念清晰，例题丰富，循序渐进，易于学习，是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软件开发工作有关人员的参考书。

图书在版编目(CIP)数据

实用软件设计模式教程/徐宏喆等编著. —2版. —北京：清华大学出版社，2017
(重点大学计算机专业系列教材)
ISBN 978-7-302-43597-6

Ⅰ. ①实… Ⅱ. ①徐… Ⅲ. ①软件设计－高等学校－教材 Ⅳ. ①TP311.5

中国版本图书馆CIP数据核字(2016)第082051号

责任编辑：郑寅堃　薛　阳
封面设计：常雪影
责任校对：焦丽丽
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：三河市君旺印务有限公司
装 订 者：三河市新茂装订有限公司
经　　销：全国新华书店
开　　本：185mm×260mm　　**印　　张**：21.5　　**字　　数**：522千字
版　　次：2009年7月第1版　2017年2月第2版　　**印　　次**：2017年2月第1次印刷
印　　数：1～1500
定　　价：45.00元

产品编号：063913-01

INTRODUCTION

出版说明

随着国家信息化步伐的加快和高等教育规模的扩大，社会对计算机专业人才的需求不仅体现在数量的增加上，而且体现在质量要求的提高上，培养具有研究和实践能力的高层次的计算机专业人才已成为许多重点大学计算机专业教育的主要目标。目前，我国共有16个国家重点学科、20个博士点一级学科、28个博士点二级学科集中在教育部部属重点大学，这些高校在计算机教学和科研方面具有一定优势，并且大多以国际著名大学计算机教育为参照系，具有系统完善的教学课程体系、教学实验体系、教学质量保证体系和人才培养评估体系等综合体系，形成了培养一流人才的教学和科研环境。

重点大学计算机学科的教学与科研氛围是培养一流计算机人才的基础，其中专业教材的使用和建设则是这种氛围的重要组成部分，一批具有学科方向特色优势的计算机专业教材作为各重点大学的重点建设项目成果得到肯定。为了展示和发扬各重点大学在计算机专业教育上的优势，特别是专业教材建设上的优势，同时配合各重点大学的计算机学科建设和专业课程教学需要，在教育部相关教学指导委员会专家的建议和各重点大学的大力支持下，清华大学出版社规划并出版本系列教材。本系列教材的建设旨在“汇聚学科精英、引领学科建设、培育专业英才”，同时以教材示范各重点大学的优秀教学理念、教学方法、教学手段和教学内容等。

本系列教材在规划过程中体现了如下一些基本组织原则和特点。

(1) 面向学科发展的前沿，适应当前社会对计算机专业高级人才的培养需求。教材内容以基本理论为基础，反映基本理论和原理的综合应用，重视实践和应用环节。

(2) 反映教学需要，促进教学发展。教材要能适应多样化的教学需要，正确把握教学内容和课程体系的改革方向。在选择教材内容和编写体系时注意体现素质教育、创新能力与实践能力的培养，为学生知识、能力、素质协调发展创造条件。

(3) 实施精品战略，突出重点，保证质量。规划教材建设的重点依然是专业基础课和专业主干课；特别注意选择并安排了一部分原来基础比较好的优秀教材或讲义修订再版，逐步形成精品教材；提倡并鼓励编写体现重点大学

计算机专业教学内容和课程体系改革成果的教材。

(4) 主张一纲多本,合理配套。专业基础课和专业主干课教材要配套,同一门课程可以有多本具有不同内容特点的教材。处理好教材统一性与多样化的关系;基本教材与辅助教材以及教学参考书的关系;文字教材与软件教材的关系,实现教材系列资源配套。

(5) 依靠专家,择优落实。在制订教材规划时要依靠各课程专家在调查研究本课程教材建设现状的基础上提出规划选题。在落实主编人选时,要引入竞争机制,通过申报、评审确定主编。书稿完成后要认真实行审稿程序,确保出书质量。

繁荣教材出版事业,提高教材质量的关键是教师。建立一支高水平的以老带新的教材编写队伍才能保证教材的编写质量,希望有志于教材建设的教师能够加入到我们的编写队伍中来。

教材编委会

FOREWORD

前言

面向对象程序设计已经成为软件设计开发领域的主流，而学习使用设计模式非常有助于软件开发人员开发出易维护、易扩展、易复用的代码。而且，目前越来越多的大学和培训机构也把面向对象技术作为主要教学内容。

本书从面向对象的基本概念入手，介绍面向对象程序设计的主要原理和方法，重点探讨了在程序设计中怎样使用著名的 24 个设计模式。本书编者在十余年的项目开发实践中积累了丰富的开发经验，在近年来的项目开发中，也有意识地大量使用设计模式来提高系统的可复用性。在对各类设计模式的使用中常常沉醉于设计模式精妙的构思和优雅的结构，于是产生了编写一本用实例来透彻讲解设计模式使用的参考书的想法，鉴于高等院校对设计模式相关教材的迫切需要，因此决定将本书以教材的形式撰写。

作者于 2009 年编写了《实用软件设计模式教程》，由清华大学出版社出版。该书出版后，受到了广大读者的欢迎，认为该书概念清晰，叙述详尽，深入浅出，通俗易懂。根据发展的需要，作者于 2016 年编写了《实用软件设计模式教程(第 2 版)》，本书第 2 版相较第 1 版，对最新的软件架构技术进行了补充阐述，紧跟当前技术发展，同时改用业界使用较为广泛的 C# 程序设计语言作为设计模式的描述语言，为读者的工作和学习提供有益的帮助。

本书严格执行面向对象设计标准并使用实例讲解每个设计模式，使读者易于理解、便于使用，最后一章还用实际项目开发实践中的实例作为例子，介绍各种设计模式在实际项目中综合应用的方法。

本书的章节安排如下。

第 1 章为面向对象基础，详细分析面向对象方法，从面向对象方法的产生、面向对象方法的概念引出了面向对象方法的优势，又结合一个具体的应用系统实例，细致分析了面向对象分析、面向对象设计、面向对象编程实现、面向对象的测试以及面向对象软件设计原则的主要步骤和方法。

第 2 章为 C# 面向对象编程基础，介绍 C# 语言的相关概念和技术，为后续的设计模式学习打下基础。

第 3 章为设计模式，按创建型、结构型、行为型分类，详细分析 23 种设计模式，在介绍每个模式时，以一个软件设计开发中的实际问题为引子，探讨一

般实现方法的缺陷,进而介绍新模式的结构,再以一个实际的例子展现模式的编程方法,最后对使用模式的效果进行分析。

第4章为综合实例,该实例集中使用了多种设计模式,展示设计模式在应用系统开发实践中的应用。

第5章为软件架构与架构建模技术,介绍软件架构的定义和发展史,分析了几种常用的软件架构模式,并简要介绍软件架构建模技术。

第6章为面向服务的软件架构——SOA,简要介绍SOA、SOA的框架及其应用实例。

第7章为云计算环境下的软件架构,主要介绍在云计算环境下软件架构的技术和内容。

本书的编写过程中,林春喜、杨刚、吴夏、武晓周、姚智海、王婵、陈鹏、闫志浩等人完成了大量校正、录入工作,在此对他们的工作表示感谢。

在此谨对所有曾经支持和帮助过我们的同志和朋友表示真挚的谢意。

由于我们水平有限,再加上时间紧迫,书中难免有疏漏和不妥之处,盼望专家和广大读者不吝指正。

编　者

2016年12月

CONTENTS

目录

CHAPTER 1

第1章 面向对象基础

本章阐述面向对象的思想和基本概念，并通过编程语言的发展历史来说明面向对象技术的优势所在；同时，介绍了面向对象的开发过程，包括面向对象分析、面向对象设计、面向对象的编程实现和面向对象的测试。本章主要为进一步学习设计模式打下良好的基础。书中所有的示例代码均为面向对象语言C#所编写，第2章将对C#语言面向对象编程基础进行简要介绍。

1.1 面向对象方法

所谓面向对象，就是以对象观点来分析现实世界中的问题：从普通人认识世界的观点出发，把事物归类、综合，提取其共性并加以描述。在面向对象的系统中，世界被看成是独立对象的集合，对象之间通过消息相互通信。面向对象方法是一种运用对象、类、继承、封装、聚合、多态性和消息传送等概念来构造系统的软件开发方法，其基本思想是从现实世界中客观存在的事物（即对象）出发来构造系统，并在系统中尽可能运用人类的自然思维方式。

1.1.1 面向对象方法的特点

在现实世界中，大到日月星辰，小至沙粒微尘，均可视为单独的个体对象，它们具有自己独有的运动规律和内部状态，其相互作用组成了多姿多彩的世界。

计算机软件可以认为是现实世界中相互联系的对象所组成的系统在计算机中的模拟实现。在程序的世界，一个用户、一条消息、一个过程或是某种算法，均可将其看作为一种对象；而每一种对象，其创建和销毁，内在属性和表现行为，以及它与外界之间的关系，无不具有某种哲学的意味。

人类在长期的认识和改造世界的过程中逐渐形成了一种面向对象的世界观——用分类的方式来认识世界，通过对象及对象间的相互关系来理解世界。面向对象方法是面向对象的世界观在软件开发方法中的直接运用。与传统的面向过程方法相比，面向对象方法的核心思想是从现实世界中客观存

在的事物(即对象)出发来构造软件系统,并在系统构造中尽可能运用人类的自然思维方式。它强调直接以问题域(现实世界)中的事物为中心来思考问题、认识问题,并根据这些事物的本质特点,把它们抽象地表示为系统中的对象,作为系统的基本构成单位,这可以使系统直接地映射问题域,保持问题域中事物及其相互关系的本来面貌。

具体地讲,面向对象方法有如下一些主要特点。

- 从问题域中存在的客观事物来抽象对象,并以此作为软件系统的基本构成单位。
- 事物的静态特征由对象的属性来表示;事物的动态特征由对象的方法来表示。
- 对象的属性和方法结合为一体,成为一个独立的实体,对外屏蔽其内部细节,也即封装。
- 对事物分类,将具有相同属性和方法的对象归为一类。类是这些对象的抽象描述,每个对象是它所属类的一个实例。
- 通过较多或较少地忽略事物之间的差异,来实现不同程度上的抽象,以得到较一般的类和较特殊的类,特殊类继承一般类的属性和方法。面向对象方法支持这种继承关系的描述和实现。
- 复杂的对象可以把简单对象作为其构造成分,即聚合。
- 对象之间通过消息进行通信。
- 用关联来表达对象之间的静态关系。

1.1.2 面向对象方法的基本概念

1. 对象的概念

在不同领域中,对于对象有不同的解释。一般认为,对象就是一种事物、一个实体。在面向对象的领域中,可以从两个角度来理解对象,一是从现实世界角度,二是从系统构成角度。在现实世界中,客观存在的任何事物都可以被看作是对象。这样的对象可以是有形的,例如一辆汽车;也可以是无形的,例如一项计划或一个抽象的概念。无论从哪个方面看,对象都是一个独立的单位,它具有自己的静态特征和动态特征。对于所要建立的特定系统模型来说,现实世界中的有些对象是有待于抽象的实物。

在面向对象的系统中,对象是用来描述客观事物的一个实体,是构成系统的一个基本单位。一个对象由一组属性和对这组属性进行操纵的一组操作构成,属性是用来描述对象静态特征的一个数据项,操作是用来描述对象动态特征的一个动作序列,对象、对象的属性和操作都有自己的名字。对象实现了信息隐藏,对象与外部是通过操作接口联系的,方法的具体实现方式对外部是不可见的。封装的目的就是阻止非法的访问,操作接口提供了这个对象的功能。对象是通过消息与另一个对象进行通信的,每当一个操作被调用,就有一条消息被发送到这个对象上,消息带来了将被执行的这个操作的详细内容。一般情况下,消息传递的语法随系统的不同而不同,其组成部分包括目标对象、所请求的方法和参数等。

2. 类的概念

类是具有相同属性和操作的一组对象的集合,为属于该类的全部对象提供统一、抽象的描述,其内部包括属性和操作两个主要部分。类的作用是用来创建对象,对象是类的一个实例。例如,在一个学生信息管理系统中,类“学生”具有姓名、性别、学号和年龄等属性,以及“注册”和“上报”等操作。一个具体的学生就是“学生”这个类的一个实例。

类是创建对象的样板，它包含着所创建对象的内部状态的描述和方法的定义。类完整地描述了外部接口和内部算法以及数据结构的形式。类是对象的抽象及描述，是具有共同行为的若干对象的统一描述体，类中要包含生成对象的具体方法。一个类的所有对象都要有相同的数据结构，并且共享相同的实现操作的代码，而各个对象有着各自不同的状态，即私有的存储。因此，类是所有对象的共同的行为和不同状态的集合体。

3. 继承的概念

类提供了一组说明对象结构的机制，再借助继承这一重要机制扩充其定义。继承提供了创建新类的一种快捷方法，可以通过对已有类进行修改或扩充来满足新类的要求。新类可以共享已有类的行为，也可以修改或额外添加行为。可以说，继承的本质特征是行为共享。将被继承的类称为父类或基类，而将新产生的类称为子类或派生类。

在继承关系中，子类自动地拥有或隐含地复制其父类的全部属性与操作，这种机制也称作一般类对特殊类的泛化。继承具有“是一种(is a kind of)”的含义，如图1.1所示，学生是一种学校人员，教师也是一种学校人员，二者作为特殊类继承了一般类“学校人员”的所有属性和操作。

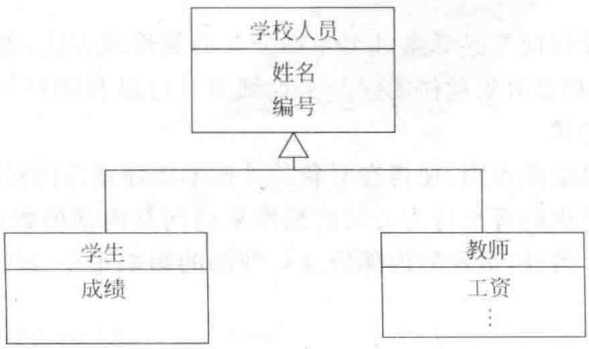


图1.1 继承的示例

面向对象方法以其独特的特点，有效地支持了软件复用：面向对象方法中的对象(类)是复用构件的一种雏形，类和类继承是强有力的复用机制，继承和重载是复用构件的修改技术，继承和消息传递是复用构件合成新构件的构件合成机制。面向对象方法对可复用性的支持具体体现在以下方面。

对象(类)——可复用构件的雏形。在面向对象技术中，对象是组成系统的基本成分和独立的活性单元，是对实体的数据和功能的统一抽象。由于对象的一系列特有的性质，使其具有了可复用构件的雏形：

① 对象是数据和功能的统一体，集内部状态属性和外部行为属性于一身，具有极高的独立性；

② 对象之间是一种松耦合关系，通过接口发送或接收信息来建立对象之间的请求和被请求关系，对象之间的联系简单、明了，便于应用和组装；

③ 对象(解空间)是对实体(问题空间)的模拟，缩短了问题空间中的问题与解空间中的软件实体之间的距离。在问题空间中具有共性的实体，其共性能够在与之对应的软件模块之间反映出来，从而得到复用。

类与类继承——复用机制。类是对具有相似性质的对象的描述,从而使得这些对象可以共享这些描述,类继承提供了复用不同类的描述的可能性,进一步扩大了共享的范围和灵活性。

继承和修改——修改技术。在面向对象方法中,当要设计一个新的对象(类)时,如果已有一个相似的对象(类)存在,则不必从头开始来设计这个对象(类),而可以作为已有类的子类,继承父类的所有特征,只需描述与已有对象(类)不同的部分。继承和重载使对象(类)的修改可以不通过直接对其进行增删和修改的方法来进行。

继承及消息传递——构件合成技术。若类 subclass 是类 class 的子类,则类 subclass 是类 class 与它自身定义的"合成"。若在类 A 的定义中有诸如 B 和 b 的声明(b 是 B 的一个实例),则类 A 的定义就可以通过 b 来利用类 B 的功能,这就是消息传递合成技术。这两种合成方法相互配合使用并相辅相成,是有效的构件合成技术。

4. 封装和隐藏的概念

封装是面向对象技术的主要特征之一。其把过程和数据包围起来,使对数据的访问只能通过已定义的接口。封装保证了模块具有较好的独立性,使得程序维护起来较为容易且实现了信息隐藏。

信息隐藏是指对象仅向外界暴露(Expose)公共的属性或方法,外界不能直接访问对象内部的私有信息,也不知道对象操作的内部实现细节。可以利用封装或使用接口与实现相分离的方法实现信息隐藏。

通过封装以及信息隐藏原则,使得在对象的外部不能随意访问对象的内部数据和操作,而只允许通过由对象提供的可见行为公共的操作来访问其内部的数据,这就避免了外部错误对其的"交叉感染"。另外,对象的内部修改对外部的影响很小,减少了由修改引起的"波动效应"。

1.1.3 面向对象语言的产生

软件开发人员对问题域的认识是人类的一种思维活动。人类的思维活动总是借助自己所熟悉的某种自然语言进行的,而软件系统的开发则必须用一种计算机所能够阅读和理解的语言来完成。

人类的自然语言和计算机所能够理解的编程语言之间存在着巨大的差异:一方面,自然语言所产生的对问题域的认识远远不能被计算机理解和执行;另一方面,计算机能够理解和执行的编程语言又很不符合人类的思维方式。这种人机之间语言交流的矛盾称为"语言的鸿沟"。

而计算机编程语言的不断发展使得这种"语言的鸿沟"逐渐变窄。图 1.2 表示编程语言由低级到高级的发展过程。程序设计语言的发展是一个不断演化的过程,其根本的推动力就是抽象机制更高的要求,以及对程序设计思想更好的支持。具体地说,就是把机器能够理解的语言提升到也能够很好地模仿人类思考问题的形式。程序设计语言的演化从最开始的机器语言到汇编语言到各种结构化高级语言,最后到支持面向对象技术的面向对象语言,反映的就是一条抽象机制不断提高的演化道路。机器语言和汇编语言几乎没有抽象,对于机器而言是最合适的描述。它们可以操作到机器里面的位单位,并且任何操作都是针对机器的。这就要求人们在使用机器或者汇编语言编写程序时,必须按照机器的方式去思考问题。

因为没有抽象机制，所以程序员很容易陷入复杂的事物之中。随着 C、Pascal、Fortran 等结构化高级语言的诞生，使程序员可以离开机器层次，在更抽象的层次上表达意图。由此诞生的三种重要控制结构以及一些基本数据类型，都能够很好地让程序员以接近问题本质的方式去思考和描述问题。随着程序规模的不断扩大，在 20 世纪 60 年代末期出现了软件危机，在当时的程序设计范型中都无法克服错误随着代码的扩大而级数般地扩大的问题，以致到了无法控制的地步，这个时候就出现了一种新的思考程序设计方式和程序设计范型——面向对象程序设计，由此也诞生了一批支持此技术的程序设计语言，例如 Eiffel、C++ 和 Java。这些语言都以新的观点去看待问题，即问题是由各种不同属性的对象以及对象之间的消息传递构成的。面向对象语言由此必须支持新的程序设计技术，例如数据隐藏、数据抽象、用户定义类型、继承和多态等。

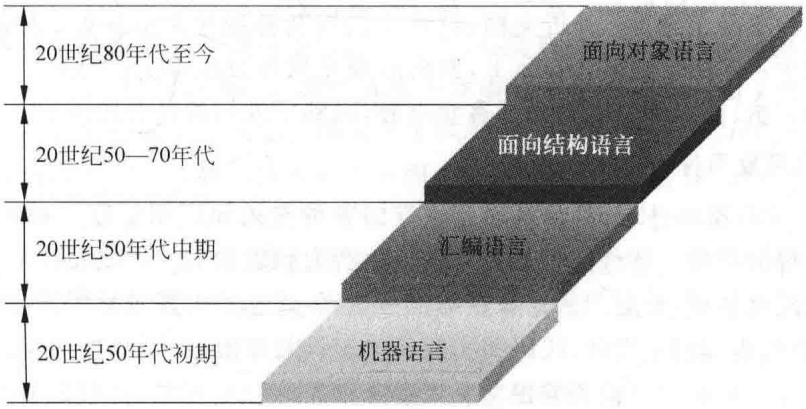


图 1.2　程序设计语言的发展

1.1.4　面向对象的优势

面向对象方法与传统的程序设计方法相比，其优势主要体现在如下几个方面。

1. 优良的可维护性

需求是不断变化的，因为客户业务关系、竞争形势、技术发展和规章制度等因素都在不断地在发生变化，这就要求系统对变化要有弹性。面向过程方法从功能的角度进行建模，所有的软件都用功能作为其主要构造块。人们把大量精力都集中在控制和大算法的分解上，最终的系统设计错综复杂，对一处修改往往会引起连锁反应。这种建模的缺点是模型脆弱，难以适应不可避免的错误修改和需求变动，以至于使系统维护困难，而过程和数据的分离是造成这种状况的根本原因。另外，分析模型与设计模型的不匹配也是结构化方法的一个重大缺点，这样的缺点也使得复用难以实现。

在面向对象方法中，把数据和处理数据的过程作为一个整体，即对象。面向对象方法以众多的对象及其交互模式为中心，把易变的数据结构和部分功能封装在对象内并加以隐藏，提高了对象的内聚性，并减少了与其他对象的耦合，保证了对该对象的修改不会影响其他的对象；其他对象只有通过消息才能直接或间接地影响对象的行为，这样就避免了由于随意访问对象的属性而造成的对象状态的不确定性。按照面向对象的封装和信息隐藏原则，对象内部的属性和部分操作仅供对象自己使用，对其修改不会影响其他对象。对对象的接口

(此处指广义的接口,即对象中供其他对象访问的那些操作)的更改则会影响其他的对象,若在设计模型时遵循一定的原则,这样影响能被局限在一定的范围之内。此外,由于将操作与实现的细节进行了分离,若接口中的操作仅在实现上发生了变化,则不会影响其他对象。

2. 生产效率的提高

按照现今的软件质量观点,不是仅仅要求软件能够在事后通过测试排除错误,而是要着眼于软件开发过程的每个环节,从分析、设计阶段就开始质量保证活动。高质量不仅是指系统没有错误,更是指系统要达到好用、易用、可移植和易维护,并让用户由衷地感到满意。相对传统的软件开发方法,采用面向对象方法进行软件开发是更容易做到这些的。

有数据表明,使用面向对象技术在开发阶段能提高效率20%,在维护阶段提高得就更多。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。实际上,面向对象是软件方法学的返璞归真。软件开发从过分专业化的方法、规则和技巧回到了客观世界,回到了人们的日常思维。

3. 优良的可复用性

软件复用(SoftWare Reuse)是将现有软件的各种有关知识用于建立新的软件,以缩减软件开发和维护的花费。软件复用是提高软件生产力和质量的一种重要技术。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代码,后来扩大到包括领域知识、开发经验、设计决定、体系结构、需求、设计、代码和文档等一切有关方面。

面向对象方法从面向对象的编程发展到面向对象的分析与设计,使这种方法支持软件复用的固有特征能够在软件生命周期各个阶段都发挥作用,从而对软件复用的支持达到了较高的级别。与其他软件工程方法相比,面向对象方法的一个重要优点是,可以在整个软件生命周期达到概念、原则、术语及表示法的高度一致。这一优点使面向对象方法不但能在各个级别支持软件复用,而且能对各个级别的复用形成统一的、高效的支持,达到良好的全局效果。做到这一点的必要条件是,从面向对象软件开发的前期阶段——面向对象分析,就把支持软件复用作为一个重点问题来考虑。

1.2 面向对象分析

自软件工程学问世以来,已出现过多种分析方法,其中最有影响力的是功能分解法、数据流法、信息建模法和20世纪80年代后期兴起的面向对象方法。面向对象分析正是在借鉴了以往的许多分析方法的基础上建立起来的。

1.2.1 概论

面向对象的分析,就是运用面向对象方法进行系统分析,它是软件生命周期的一个阶段,具有一般分析方法所共有的内容、目标及策略。与其他几种分析方法相比,面向对象分析的优势在于,它对问题域的观察、分析和认识是很直接的。对问题域的描述也是很直接的。它所采用的概念与问题域中的事物保持了最大程度的一致,问题域中有哪些值得考虑的事物,面向对象分析模型中就有哪些对象,而且对象、对象的属性和操作的命名都强调与