高等职业教育教学用书

软件工程

主 编 薛志明

副主编 钱 杰 周 剑 龚 雪 张 博

本书配有电子教学参考资料包

電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

高等职业教育教学用书

软件工程

薛志明 主 编

钱 杰 周 剑 龚 雪 张 博 副主编

電子工業出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

软件工程是计算机技术专业的主干课程。本书是根据普通高职院校计算机技术专业软件工程课程需要而编写的教材。本书共 7 章。其中，第 1 章介绍了面向对象设计技术，第 2 章介绍了软件工程及其相关概念，第 3 章介绍了软件开发过程中的需求分析，第 4 章介绍了软件开发过程中的概要设计，第 5 章介绍了软件开发过程中的详细设计，第 6 章介绍了软件开发过程中的系统实施，第 7 章介绍了与软件开发有关的软件复用、XML 技术及分布对象技术。

本书可作为高等职业学校计算机类专业的教材或教学参考书，也可作为软件开发人员的参考书。

图书在版编目（CIP）数据

软件工程 / 薛志明主编. —北京：电子工业出版社，2013.9
高等职业教育教学用书
ISBN 978-7-121-21466-0

Ⅰ. ①软… Ⅱ. ①薛… Ⅲ. ①软件工程－高等职业教育－教材 Ⅳ. ①TP311.5

中国版本图书馆 CIP 数据核字（2013）第 215103 号

策划编辑：施玉新
责任编辑：韩玉宏
印　　刷：北京京师印务有限公司
装　　订：北京京师印务有限公司
出版发行：电子工业出版社
　　北京市海淀区万寿路 173 信箱　邮编　100036
开　　本：787×1 092　1/16　印张：9.75　字数：249.6 千字
印　　次：2013 年 9 月第 1 次印刷
印　　数：3 000 册　　定价：19.50 元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：（010）88254888。

质量投诉请发邮件至 zlts@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线：（010）88258888。

前　言

软件工程是计算机学科中一个非常有价值且具有广阔发展空间的研究领域。很多年来，随着计算机硬件技术的迅速发展，人们开发优质软件的能力远远落后于社会各个领域对计算机软件的需求，即在软件开发过程中仍经受着软件危机的困扰。为克服软件危机，自20世纪60年代以来，人们在这一领域做了大量的研究与实践工作，积累了大量的软件开发技术和方法，进而逐渐形成了系统的软件项目开发与管理理论。于是，一门新兴的学科——软件工程诞生了。软件工程所研究的范围十分广泛，主要包括软件项目开发和软件维护的有关理论、技术、方法、标准、计算机辅助工具和环境及软件项目管理等诸多方面。软件工程领域的研究成果为缓解软件危机发挥了关键性的作用。

软件工程是计算机技术专业的主干课程。本书是根据普通高职院校计算机技术专业软件工程课程需要而编写的教材。本书共7章。其中，第1章介绍了面向对象设计技术，第2章介绍了软件工程及其相关概念，第3章介绍了软件开发过程中的需求分析，第4章介绍了软件开发过程中的概要设计，第5章介绍了软件开发过程中的详细设计，第6章介绍了软件开发过程中的系统实施，第7章介绍了与软件开发有关的软件复用、XML技术及分布对象技术。

本书具有以下特点。

(1) 结构合理，系统介绍了软件工程的基本原理、概念、方法和开发要求。

(2) 在选材上注重实效性，增加了典型的开发实例，做到理论与实践相结合，达到学以致用的目的。

(3) 书中对概念的讲解较为清楚，易于理解。各章后配有相关的习题与教材内容配合紧密。

本书可作为高职院校软件工程课程教材或教学参考书，也可作为软件开发人员的参考书。

本书由薛志明主编，由钱杰、周剑、龚雪、张博任副主编。其中，第1章由薛志明编写，第2章由龚雪、张博编写，第3章由蒋政编写，第4章由钱杰编写，第5章由何元编写，第6章由王松编写，第7章由周剑编写。

本书由无锡广播电视大学的胡云老师主审。在本书的编写过程中，得到了无锡高等师范学校肖敏副教授的大力支持。宜兴张渚职业高级中学的王斌生老师参与了大纲的讨论和制订，并提出了很多宝贵的意见和建议。在此，编者向他们表示衷心的感谢。

由于时间仓促及编者水平有限，书中难免存在不妥或疏漏之处，恳请广大读者批评指正。

编　者

2013年8月

目　录

第1章 面向对象设计技术

本章学习要点

掌握面向对象的基本概念及对象的基本特性。
掌握有关概念：抽象、封装、继承、分类、聚合、关联、消息等。
了解面向对象方法的基本特征。
了解面向对象的软件工程方法。
掌握对象间通信的方式。
了解面向对象的分析和设计方法及其过程。

开发一个系统并认识这个系统是一个渐进的过程，是在继承了以往有关知识的基础上，多次迭代往复并逐步求精深化而成的。在这种认识的深化过程中，既包括从一般到特殊的演绎，也包括从特殊到一般的归纳。到目前为止，用于分析、设计和实现系统的过程和方法大部分是瀑布型的，即后一步实现前一步所提出的需求，或者是进一步发展前一步所得出的结果。

因此，越接近系统设计或实现的后期阶段，对系统设计或实现的结果修改就越困难。同时也只有在系统设计或实现的后期阶段才能发现前期所形成的一些差错。而且当这个系统规模越大、问题越复杂时，认识这个系统的过程和设计或实现这个系统的过程不一致所引起的困扰也就越大。

为了解决上述问题，就应使分析、设计和实现的方法尽可能地接近认识系统的方法。换言之，就是应使上述问题的问题空间和解决问题的方法空间在结构上尽可能地一致，也就是使分析、设计和实现系统的方法学原理与认识客观世界的过程尽可能地一致。这就是面向对象方法学的出发点和所追求的基本原则。

1.1 面向对象的基本概念

1.1.1 面向对象的基本元素

在面向对象方法中，最重要的两个概念就是类和对象。首先，类和对象能反映出系统中各事物之间的联系；其次，对象的各种状态和特性构成了面向对象的各种特征。

1. 对象

从广义上讲，对象（object）既可以是具体有形的，也可以是抽象无形的。面向对象的程序就是由相互作用的对象组成的，对象是由数据和处理这些数据的操作组成的。

通常将对象定义为：对象是问题域中某些事物的一个抽象，它反映该事物在系统中需要保存的信息和发挥的作用；它是一组属性和有权对这些属性进行操作的一组服务的封装体。

对象也可以由相对比较简单的对象以某种方式组成，对整个世界来说，就是以一些原始的对象为基础，经过层层组合而成。对象的框图表示法如图 1-1 所示。

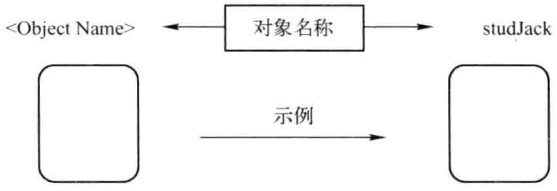

图 1-1 对象的框图表示法

2. 类及类层次结构

（1）类

类（class）是具有相同属性和服务的一组对象的集合，它为属于该类的全部对象提供了统一的抽象描述，其内部包括属性和服务两个部分。

为了创建一个对象，系统必须提供一个类的定义，这样才能在程序中用指令创建对象。类是创建对象的模型或模板，对象被称为“类的实例”（instance）。如图 1-2 所示，studZhang 和 studLi 这两个对象就是 stud 类的实例。只要一个类已被定义，用户即可根据程序的需要创建这个类的多个对象。

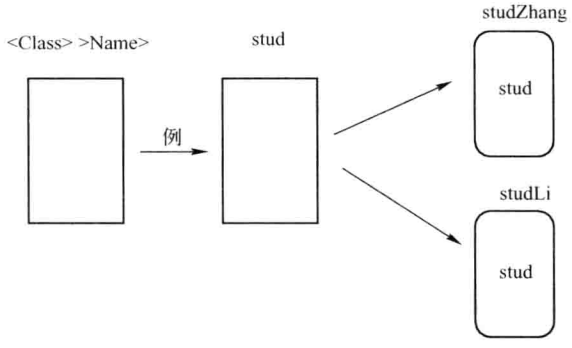

图 1-2 类及其对象

（2）类层次结构

一个类的上层可以有父类（超类），下层可以有子类，这几个类就形成了一种层次结构。这种层次结构的一个重要特点是继承性。一个类继承其父类（超类）的全部特性，同时继承具有传递性。

类可分为一般类和特殊类，其描述如下：如果类 A 具有类 B 的全部属性和全部服务，而且具有自己特有的某些属性或服务，则 A 类叫做 B 类的特殊类，B 类叫做 A 类的一般类。

3．消息和方法

（1）消息

在面向对象方法中，把面向对象发出的服务请求称做消息（message）。也就是说，如果要求某个类或对象执行一项任务，则必须要向其发送一个消息。

消息是用来请求对象处理或回答某些信息的要求。程序的执行就是通过在对象间传递消息来完成的。

（2）方法

方法（method）就是对象上的各种操作。处理消息的类或对象需要通过相应的编程，不能随便将消息发送给一个类或对象，而只能将消息发送给能够理解消息的类或对象。处理所收到消息的类或对象必须有对应的方法，即类或对象为完成一项任务而执行的指令序列。

对象间数据的传递是通过“消息的变量”来实现的，发送给类或对象的消息名必须与其方法名相同。在系统内部，当面向对象的程序执行时，一般要做 3 件事：首先，根据需要创建对象；其次，当程序处理信息或响应来自用户的输入时，要求从一个对象传递消息到另一个对象；最后，若不再需要该对象，则应删除它并回收它所占用的存储单元。

（3）举例

在面向对象的程序设计中，现要求查询某一学生的课程成绩。如图 1-3 所示，其程序设计如下：首先，以 stud 类创建一个实例（studZhang 对象）；其次，向 studZhang 对象发送 search 消息，查询其课程 1（kc1）的成绩；最后，删除创建的对象（studZhang 对象），回收其所占用的存储单元。

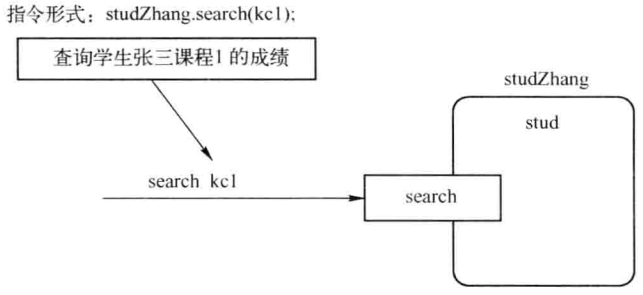

图 1-3　发送 search 消息给 studZhang

1.1.2　对象的基本特性

1．继承性

在面向对象方法中，继承是子类自动地共享其父类中定义的属性和服务的机制。继承性使得相似的对象可以共享程序代码和数据结构，从而大大减少了程序的冗余信息。在程序

执行期间，对象某一性质的查找是从该对象所在的层次开始的，沿类的结构逐层向上进行，所找到的第 1 个匹配的性质作为该对象的性质，如图 1-4 所示。

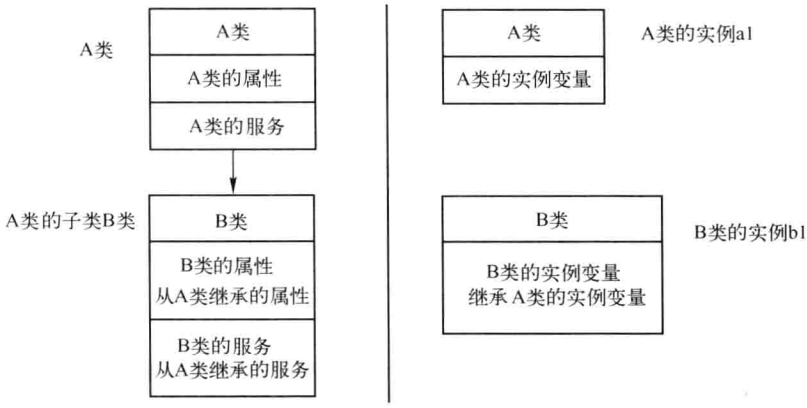

图 1-4 类的继承

继承是面向对象方法中一个十分重要的概念。继承性使得软件的维护和修改工作量减少了许多。当需要扩充软件原有功能时，先从修改的类派生出一个子类（派生类），子类继承父类所有属性和服务，根据需要在派生类中添加必要的代码。当需要完全改变父类中某个操作时，可以在派生类中定义一个同名的方法。如果要新增加一个功能，则可以在派生类中定义并实现一个全新的方法。因此，利用类和对象的继承性可大大提高软件开发效率。

继承具有传递性。在面向对象的设计中，要特别注意将问题的一般性与特殊性分别进行设计。应该定义基类解决一般性问题，在派生类中解决特殊性问题，这样便于软件的复用和修改。

多继承：如果一个类是多个一般类的特殊类，则它从多个一般类中继承属性和服务。在多继承过程中，可能会出现一个命名冲突问题，即一个特殊类继承了多个相同的属性名或服务名时，系统无法判定它的语义到底是哪一个类中的属性或服务。解决此问题的方法是：其一，禁止多继承结构中各个一般类的属性和服务取相同的名字；其二，提供更名机制，使程序可以在特殊类中更换从各个一般类继承来的属性或服务的名字。

2. 封装性

封装属于一种信息隐蔽技术，是面向对象方法的一个重要原则。它有两个含义：其一是将对象的全部属性和服务结合在一起，形成一个不可分割的独立单位（即对象）；其二是信息隐蔽，即尽可能隐蔽对象的内部细节，对外形成一个边界，只保留有限的对外接口使之与外部发生关系。

3. 结构和连接

在系统开发过程中，可能会涉及很多事物，为了使系统能够有效地映射问题域，开发者需要认识并描述对象之间的 4 种关系，即对象之间的分类关系、对象之间的组成关系、对象之间的静态关系、对象之间的动态关系。在面向对象方法中，利用一般/特殊结构、整体/部分结构、实例连接和消息连接，描述对象之间的 4 种关系。

4．多态性

对象的多态性是指一般类中定义的属性或服务被特殊类继承后，可以具有不同的数据类型或表现出不同的行为。在面向对象方法中，多态性是指在类的不同层次上可以使用相同的服务名。当对象接收到发送给它的消息时，对象根据所属类，动态地选择在该类中定义的实现算法。

例如，有一服务名为“绘图”，消息中给出的服务名均为“绘图”，而椭圆、多边形、矩形等类的对象在接收到这个消息时却各自执行不同的绘图操作。

多态性属于一种比较高级的功能，而支持多态性的语言应具备下列功能。

① 重载：在特殊类中对继承来的属性或服务进行重新定义。

② 动态绑定：运行时根据对象接收到的消息动态地确定要连接哪一段服务代码。

③ 类属：服务参数的类型可以参数化。

1.1.3　面向对象方法

面向对象方法涉及领域广泛，没有人能明确而清晰地界定它的范围、严格而准确地对它进行定义。在 20 世纪 80 年代以前，面向对象方法被人们认为是一种新兴的程序设计方法，因为其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。在 20 世纪 80 年代以后，面向对象方法已涉及计算机软件领域的几乎所有分支，远远超出了程序设计语言和编程技术范畴。与此同时，面向对象方法也深入到计算机软件以外的其他领域，因而说它是一种软件开发方法已不够全面。

目前，面向对象方法最主要的应用范畴仍是软件开发，在软件生命周期的各个阶段（包括分析、设计、编程、测试与维护）及所涉及的各个领域（如界面设计、数据库、软件复用、CASE 工具等），都已形成相应的理论与技术体系。

1．面向对象方法概述

面向对象技术是一整套关于如何看待软件系统与现实世界的关系、以什么样的观点研究问题进行求解及如何进行系统构造的软件方法学。而面向对象方法是一种运用对象、类、继承、封装、聚合、消息传送、多态性等概念构造系统的软件开发方法。面向对象方法的基本思想是从现实世界中客观存在的事物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。

在使用面向对象方法开发的软件中涉及的业务范围统称为该软件的问题域。面向对象方法所强调的是：直接以问题域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征，把它们抽象地表示为系统中的对象，并把这些对象作为系统中的基本构成单位，使系统直接映射问题域，保持问题域中事物间相互关系的本来面貌。

开发软件的每一种方法都有自己的理论依据和思想体系，也建立自身独有的概念、符号、规则、策略等规范。结构化方法采用许多符合人类思维习惯的解决问题的方法，如自顶向下、逐步求精；面向对象方法则更强调运用人类在日常思维中经常采用的思想方法与原则，如抽象、分类、继承、聚合、封装等，这就使得软件开发者能更有效地思考问题，并以其他人也能看得懂的方式把自己的认识表达出来。

综上所述，面向对象方法是一种基于对象、类、实例和继承等概念的软件开发方法，即以对象为基本单位，实现消息传递、类、继承、多态和动态绑定来开发的问题域模型。

2．面向对象方法的特点

在用面向对象方法开发的系统中，主要以类的形式进行描述并通过对类的实例化创建对象，对象是系统的基本构成单位。每一对象均对应于问题域中的某个事物，它的属性与服务描述了事物的静态特征和动态特征。对象间的继承关系、聚合关系、消息和关联如实地反映了问题域中事物之间实际存在的各种关系。

面向对象方法主要有以下特点。

① 对象作为问题域中事物的抽象，是系统的基本构成单位。

② 事物的静态特征：对象的属性。

③ 事物的动态特征：对象的服务（操作）。

④ 封装：将对象的属性和服务结合作为一个独立的实体，对外屏蔽其内部细节。

⑤ 实例：对象是类的实例。

⑥ 聚合：指将若干个简单对象构成一个复杂对象，即复杂对象可以用简单对象作为其构成部分。

⑦ 动态联系：对象之间通过消息进行通信（单向或双向）。

3．举例

要对数据库进行操作，最基本的是需要建立必要的连接，然后才能完成一系列的相关操作。例如，如果在 ADO.NET 中不存在一个 COMMAND 类，那么就不能对数据库做任何事情，包括对数据库进行插入、更新操作；因此，对数据库的所有操作均从 COMMAND 类开始。

（1）用户创建 COMMAND 类的对象并设置连接属性

```
splConnection conn=new sqlConnection( );    /创建 conn 对象
…
sqlCommand cmd=new sqCommand( );            /创建 cmd 对象
cmd.connection=conn;                        /设置 connection 的属性值
cmd.CommandText="SELECT * FROM   Student";  /设置 CommandText 的属性值
```

（2）执行对象的一个操作

```
conn.open( );                               /open( ) 就是一个方法
```

1.1.4 面向对象的软件工程

随着计算机语言的发展，它与自然语言之间的差距或鸿沟越来越小。而软件开发主要是对问题域的认识和描述：从认识方面看，软件工程学在分析阶段提供了一些对问题域的分析和认识方法；从描述方面看，它在分析和设计阶段提供了一些从问题域逐步过渡到编程语言的描述手段。

在传统的软件工程方法中，从需求分析到设计之间存在一定的差距，即没有真正填平

语言之间的鸿沟，如图 1-5 所示。而在面向对象的软件工程方法中，从面向对象的分析到面向对象的设计，再到面向对象的编程和面向对象的测试都是紧密衔接的，没有了语言之间的鸿沟，如图 1-6 所示。

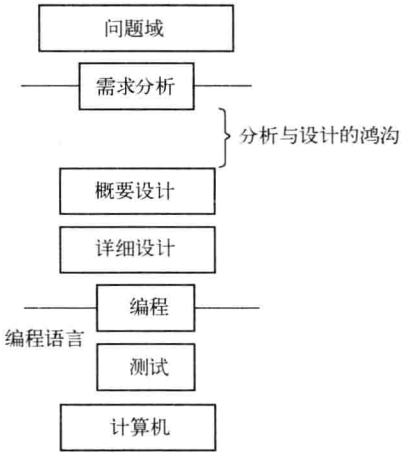

图 1-5　传统的软件工程方法

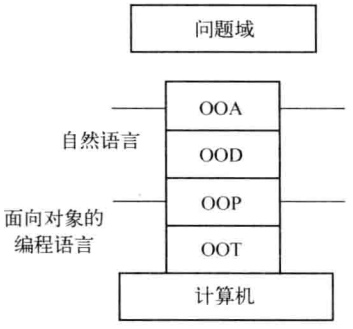

图 1-6　面向对象的软件工程方法

1. 传统的软件工程方法

传统的软件工程方法是指在面向对象方法出现之前的各种软件工程方法，如结构化软件工程方法。

（1）需求分析

需求分析具有两个方面的意义：其一，在认识事物方面，它具有一整套分析、认识问题域的方法、原则和策略，使开发人员（系统分析员）对问题域的理解比不遵守软件工程方法更为全面、深刻和有效；其二，在描述事物方面，它具有一套表示体系和文档规范，比用自然语言来表达更为准确，也更接近后期的开发阶段。

局限性：在全局范围内以功能、数据或数据流为中心来进行分析；例如，功能分解法把整个问题域看做一些功能和子功能，数据流法将其看做一些数据流来加工分析；这种分析不是直接映射问题域，而是经过了不同程度的转化和重新组合，存在问题理解上的偏差。

（2）概要设计和详细设计

概要设计是以需求分析的结果作为出发点来构造出一个具体的系统设计方案，主要包括系统的模块结构划分、模块间的数据传送及调用关系。

详细设计是在概要设计的基础上考虑每个模块的内部分析和算法，完成每个模块的程序流程图。

经过概要设计和详细设计，开发人员对问题域的认识和描述越来越接近系统的具体实现即编程。

局限性：在传统的软件工程方法中，设计文档很难与分析文档对应，原因是两者的表示体系不一致。例如，结构化分析的结果——数据流图（DFD）和结构化设计的结果——模块结构图（MSD）是两种不同的表示体系。

（3）编程和测试

编程要完成的任务是选择一编程语言完成代码的编写，最终产生一个能够被机器理解和执行的系统。测试是发现和排除程序中的错误，最终产生一个正确的系统。

（4）软件维护

通常软件维护有两种情况：其一，在系统使用过程中针对出现的问题进行维护或修改；其二，因系统需要完善或变化而进行补充和完善性维护。

局限性：对于第 1 种情况，需要从程序逆向追溯到发生错误的开发阶段，而程序不能映射问题域及各个阶段的文档不能对应，每一步追溯都存在许多理解障碍；对于第 2 种情况，应对需求到程序的每一过程增加理解，特别是对开发文档进行理解。

2．面向对象的软件工程方法

面向对象的软件工程方法是面向对象方法在软件工程领域的全面运用。它包括面向对象的分析（OOA）、面向对象的设计（OOD）、面向对象的编程（OOP）、面向对象的测试（OOT）和面向对象的软件维护等主要内容。

（1）面向对象的分析

OOA 强调直接针对问题域中客观存在的各个事物建立 OOA 模型中的对象。用对象的属性和服务分别描述事物的静态特征和动态特征。对对象及其服务的命名都强调与客观事物一致。OOA 模型也保留了问题域中事物之间的关系，即把具有相同属性和相同服务的对象归结为一类，用一般/特殊结构（又称为分类结构）描述对象之间的关系（即继承关系）；用整体/部分结构（又称为组装结构）描述对象之间的组成关系；用实例连接和消息连接描述对象之间的静态关系和动态关系。静态关系是指一个对象与另一个对象的属性关系，动态关系是指一个对象的服务与另一个对象的服务有关。

（2）面向对象的设计

OOA 针对问题域运用面向对象方法，建立一个反映问题域的 OOA 模型，不考虑与系统实现有关的因素（如编程语言、图形用户界面、数据库等），从而使 OOA 模型独立于具体的实现。

而 OOD 则是针对系统的一个具体的实现运用面向对象方法。它包括两个方面的工作：一是将 OOA 模型直接转化到 OOD 模型，作为 OOD 模型的一个部分；二是针对具体实现中的人机界面、数据访问、任务管理等因素补充一些与实现有关的部分，这部分采用与 OOA 相同的表示法和模型结构。因此，OOA 与 OOD 间不存在传统方法中分析和设计的鸿沟，降低了从 OOA 过渡到 OOD 的难度和出错率，并减少了工作量。

（3）面向对象的编程

面向对象的编程（OOP）又称为面向对象的实现（OOI）。OOP 工作就是用其一种面向对象的编程语言把 OOD 模型中的每个成分书写出来。

在编制程序时，程序员只需要用具体的数据结构来定义对象的属性，用具体的语句来实现服务流程图所表示的算法即可。因为对象类及其内部构成（属性和服务）与外部关系（结构和静态、动态关系）都已明确地在 OOA 和 OOD 中建立好。

（4）面向对象的测试

OOT 是指以对象概念为中心的软件测试，即使用面向对象技术和面向对象概念与原则来组织测试。这样可以更准确可靠，并能提高测试效率。一个原因是 OOT 以对象的类作为基本测试单位，查错范围主要是类定义之内的属性和服务，以及有限的对外接口所涉及的部

分；另一个原因是对象的类具有继承性，对父类的测试完成后，子类的测试重点只是那些重新定义的属性和服务。

（5）面向对象的软件维护

面向对象的软件工程方法为改进软件维护提供了有效的途径：其一，程序和问题域一致，各阶段表示一致，大大降低了理解的难度；其二，由于对象的服务被封装在对象内部，而对象的封装性使一个对象的修改对其他影响很小。

1.2　面向对象的分析

分析过程实际上是提取系统需求的过程，主要包括理解、表达和验证。在面向对象的分析过程中，主要由对象模型、动态模型和功能模型组成。

提取系统需求的过程是系统分析员和用户及领域专家多次反复交流和多次修改的过程，即对软件需求规格说明的正确性、完整性和有效性需要进行多次验证及反复迭代。

面向对象分析的关键是识别出问题域内的对象，并分析它们相互间的关系，最终建立起问题域的简洁、精确、可理解的正确模型。在建立模型的过程中，对象模型是最基本、最重要、最核心的模型。

1.2.1　面向对象分析的基本过程和原则

1．分析问题的层次

面向对象建模得到的模型包含对象的 3 个要素，即静态结构（对象模型）、交互次序（动态模型）和数据变换（功能模型）。

对大型复杂系统而言，对象模型由 5 个层次组成，即主题层、类及对象层、结构层、属性层和服务层，如图 1-7 所示。

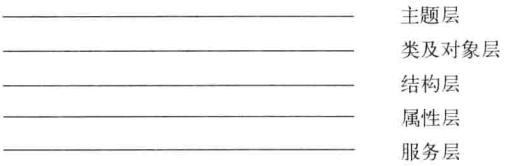

图 1-7　复杂系统对象模型的 5 个层次

在以上 5 个层次中，一层比一层更详细地描述对象模型。同时通过分层可以将整个系统中所有的复杂对象分解成几个不同的概念，便于用户阅读和理解。

在面向对象的分析过程中，以上 5 个层次分别对应建立对象模型的 5 个主要活动，即确定主题、找出对象和类、识别结构、定义属性、定义服务。通常在完整定义每个类中的服务之前，必须先建立起动态模型和功能模型，并通过对这两种模型的研究，更准确更合理地确定每个类应该提供哪些服务。

2．面向对象分析的主要概念及主要原则

面向对象分析的主要概念有：对象、类、服务、结构（可分为一般/特殊结构和整体/部分结构）、连接（包括实例连接和消息连接）、主题（指导用户理解大型而复杂的对象模型）。

面向对象分析的主要原则是：尽可能全面运用抽象、封装、继承、分类、聚合、关联、消息通信、粒度控制、原型开发等原则形成高质量、高效率的分析。