

---

高等学校信息管理与信息系统专业系列教材

# 信息系统分析与设计

甘仞初 主编

甘仞初 颜志军 杜晖 龙虹 编



高等教育出版社

## 内容提要

本书作为普通高等学校信息管理与信息系统本科专业核心课程之一——“信息系统分析与设计”的教材,阐述信息系统建设的理论、方法与技术,主要包括信息系统的概念、结构和类型、系统建设的基本思路与方法、系统规划与业务流程改革、系统开发的结构化方法与面向对象方法、基于 Web 的信息系统开发以及信息系统管理的基本知识。第二~六章均附有实例介绍,且各实例均来自作者多年的开发项目。全书讲解清晰、语言表达准确。

本书既可作为高等院校信息管理与信息系统专业教材,也可供从事信息化工作的技术与管理人员使用。

本书配有相关的电子教案,需要的教师可到高等教育出版社网站上下载:

wp.du. 或 wp.

## 图书在版编目(CIP)数据

信息系统分析与设计/甘仞初主编. —北京:高等教育出版社, 2003.6

ISBN 7-04-012312-6

. 信... . 甘... . 信息系统-系统分析-高等学校-教材 信息系统-系统设计-高等学校-教材 . G 202

中国版本图书馆 CIP 数据核字 (2003) 第 038047 号

---

出版发行	高等教育出版社	购书热线	010-64054588
社 址	北京市西城区德外大街 4 号	免费咨询	800-810-0598
邮政编码	100011	网 址	http://wp.du.
总 机	010-82028899		http://wp.
经 销	新华书店北京发行所		
印 刷			
开 本	787×960 1/16	版 次	年 月第 版
印 张	25.5	印 次	年 月第 次印刷
字 数	470 000	定 价	29.20 元

---

本书如有缺页、倒页、脱页等质量问题,请到所购图书销售部门联系调换。

版权所有 侵权必究

责任编辑 耿 芳  
封面设计 于文燕  
版式设计 陆瑞红  
责任校对 康晓燕  
责任印制

## 郑重声明

高等教育出版社依法对本书享有专有出版权。任何未经许可的复制、销售行为均违反《中华人民共和国著作权法》。行为人将承担相应的民事责任和行政责任，构成犯罪的，将被依法追究刑事责任。社会各界人士如发现上述侵权行为，希望及时举报，本社将奖励举报有功人员。

现公布举报电话及通讯地址：

电 话：(010) 84043279 13801081108

总 机：(010) 82028899

E-mail：dd@hep.com.cn

地 址：北京市西城区德外大街4号

邮 编：100011

# 前 言

本书是根据我国普通高等学校信息管理与信息系统本科专业核心课程之一“信息系统分析与设计”的教学大纲和教学要求编写的。这门课程的教学目的是使学生在已学的其他专业基础课和专业课的基础上,通过课堂教学与实践性教学环节,掌握信息系统建设的基本方法与技术,为毕业设计和今后的工作打下专业的方法与技术基础。

信息系统是正在发展中的领域。社会经济和信息技术迅速发展,国际国内信息化持续向前推进,使得信息化建设的人才十分紧缺,信息系统的理论与方法不断遇到新的挑战。作为高校本科生的教材,本书力求做到既反映这个领域的新的的发展,又能使学生掌握信息系统建设的基本方法与技能,培养学生分析解决本专业实际问题的能力。

全书共分七章。第一章是信息系统建设导论,主要讨论信息系统的发展历程、信息系统的结构与分类和概要论述系统建设的基本思路与主要方法,以作为后续各章的预备知识和概念基础。第二章论述信息系统规划与业务流程改革,信息系统作为促进组织变革、制度创新与知识创新的战略手段,在系统规划与业务流程改革中得到具体的体现。第三、四章讨论结构化系统分析、设计与系统实施的方法。尽管一些新的方法有了很大发展,结构化方法作为信息系统建设中比较成熟的方法与技术,仍然在信息系统建设中得到广泛应用,是本专业学生应该掌握的基本方法与技能。第五章讨论面向对象方法。近 20 年来,在信息系统领域,面向对象方法有了较大的发展,统一建模语言(UML)和统一过程(RUP)的建立和日趋完善使得面向对象方法在应用上有了实际的工业标准。第六章阐述基于 Web 的信息系统的开发方法与技术。这是新发展起来并逐渐成为主流模式的一类信息系统。本章着重介绍基于 Web 的信息系统的体系结构、系统开发的基本技术和工具。第七章讨论信息系统开发的项目管理与运行维护的基本知识。作为教材,本书每章都附有习题,其中有的是复习思考题,有的可作为课外作业或课程设计的内容。高等教育出版社的网站 <http://www.hep.edu.cn> 或 <http://www.hep.com.cn> 载有与本书配套的 Power Point 格式的电子教案,以供需用者下载。

全书由甘仞初主编,各章节的主要编者如下:

甘仞初 第一、二、三、四章,第七章(部分)

颜志军 第五章

杜晖 第六章

龙虹 第七章（部分）

第三、四章中系统开发实例部分取自北京理工大学甘仞初与陈永红共同承担的一个开发项目，由陈永红执笔整理而成。

本书作为教材，总结了作者多年的教学和信息化实践经验的同时，学习、参考了国内外大量的书籍与文献资料，主要的已列在本书的参考文献中。在此谨向有关作者致以深深的谢意。北京理工大学管理与经济学院系统与信息研究室王作春、冯海旗、史俊峰、朱容华、许航宇、谢莹、王海涛、曹聪梅、甘明鑫、陈琳、吴菊华、赵玉明、俞静等在本书的编写过程中做了大量工作，在此一并表示诚挚的感谢。编者特别感谢高等教育出版社的有关编辑为本书出版付出的辛勤劳动。本书编写仓促，书中错误与不妥之处，敬希读者指正。

编 者

2003年2月于北京理工大学

# 目 录

<b>第 1 章 信息系统建设导论</b> .....	1
1.1 变革时代的信息系统.....	2
1.2 信息系统发展的四个阶段.....	4
1.3 信息系统的作用与组成.....	10
1.4 信息系统的功能结构.....	12
1.5 信息系统的空间分布结构.....	19
1.6 信息系统的类型.....	27
1.7 信息系统的生命周期.....	32
1.8 结构化方法.....	40
1.9 系统建设方法的发展概述.....	45
小结.....	54
习题.....	55
<b>第 2 章 信息系统规划与业务流程改革</b> .....	57
2.1 系统规划的目标和工作内容.....	58
2.2 信息系统规划各阶段的工作内容.....	61
2.3 业务流程的识别与改革.....	66
2.4 信息系统总体结构规划的方法与步骤.....	77
2.5 信息系统规划实例——某集团公司管理信息系统规划概要.....	85
小结.....	114
习题.....	115
<b>第 3 章 结构化系统分析</b> .....	116
3.1 系统分析的目标和主要活动.....	117
3.2 结构化系统分析方法和工具.....	118
3.3 系统分析阶段各项活动的内容.....	142
3.4 系统分析中的信息收集.....	150

3.5 结构化信息系统分析实例——新亚纸制品公司管理信息系统分析概要	156
小结	167
习题	168
<b>第4章 信息系统设计与实施</b>	<b>169</b>
4.1 信息系统设计概述	170
4.2 系统总体布局	172
4.3 软件系统的总体结构设计	173
4.4 数据存储的总体结构设计	195
4.5 计算机与网络系统方案的选择	197
4.6 系统详细设计	198
4.7 系统实施	222
4.8 结构化信息系统设计实例——新亚纸制品公司管理信息系统开发概要（续）	232
小结	239
习题	241
<b>第5章 面向对象方法</b>	<b>243</b>
5.1 面向对象的基本概念	244
5.2 经典的面向对象方法	250
5.3 面向对象的分布式技术	254
5.4 统一建模语言 UML 概述	263
5.5 UML 建模技术	266
5.6 UML 在系统开发中的应用	282
5.7 面向对象的信息系统开发实例	294
小结	302
习题	303
<b>第6章 基于 Web 的信息系统开发</b>	<b>304</b>
6.1 基于 Web 的信息系统开发概述	305
6.2 基于 Web 的信息系统软件运行环境	305
6.3 基于 Web 的信息系统开发技术	309
6.4 基于 Web 的信息系统安全	323

---

6.5 基于 Web 的信息系统开发工具	329
6.6 基于 Web 的信息系统的发展	333
6.7 基于 Web 的信息系统开发实例——远程销售管理系统开发概要	339
小结	348
习题	349
<b>第 7 章 信息系统管理概论</b>	<b>351</b>
7.1 信息系统建设的项目管理	352
7.2 信息系统的运行管理	377
7.3 系统维护	387
7.4 系统评价	389
7.5 系统的可靠性和安全性概述	391
小结	393
习题	394
参考文献	395

# 第 5 章

## 面向对象方法

---

本章论述了面向对象方法的基本概念，对几种主要的面向对象方法的基本思想和原则进行了讨论，介绍了面向对象方法的发展历史和面向对象方法发展过程中的一些经典方法，包括 OOA/OOD/OOP、OMT、OOSE、Booch 方法。在此基础上，本章对目前流行的面向对象的分布式技术 CORBA、DCOM、EJB 进行了简要介绍，论述了面向对象的建模方法。章中还介绍了统一建模语言 UML 的发展历程，对 UML 中的基本词汇、语法和扩展机制进行了阐述，利用案例说明了 UML 中的图形表示法。在建模语言的基础上，本章讨论了统一开发过程(RUP)，对 RUP 的四个阶段、六个工作流进行了阐述，并介绍了建模工具 Rational Rose。最后，本章给出了两个利用 UML 来进行信息系统分析设计的例子。通过本章学习，可以掌握信息系统开发的面向对象方法的基本概念、基本方法和技术。

## 5.1 面向对象的基本概念

### 一、概述

面向对象的思想最早起源于一种名为 Simula 的计算机仿真语言,这种于 1967 年左右出现的语言被公认为第一个面向对象的语言,尽管如此,Smalltalk、C++、Objective C 等通用面向对象语言直到 20 世纪 80 年代才出现,这些面向对象的语言在编程方法上和以前的语言有很大的不同,它们不再采用传统的编程语言,如 Cobol、Ada 的结构化、确定性和串行的编程模式,而采用了 Simula 首创的对象、属性、责任和消息的编程模式。由于 Smalltalk-80 和 C++ 语言的推出,使面向对象的程序设计语言趋于成熟,并为越来越多的人所理解和接受,从而形成了面向对象编程(Object-Oriented Programming, OOP)这一新的程序设计方法。也是在这个时候,业界才真正开始认真考虑使用面向对象的语言来开发系统。信息系统的开发并不仅仅是程序设计,它还包括系统分析、系统设计、系统运行与维护、项目管理等多个环节。因此,只解决程序设计一个环节的问题是不够的。OOP 解决问题的思路是从对象(人、地方和事情)入手,而不像传统方法和结构化方法一样从功能入手,或像信息工程方法一样从信息角度入手。由于面向对象的编程解决问题的策略比较特殊,因此需要相应的面向对象的系统分析和设计方法。由于老的成熟的方法论集中考虑功能或数据,而不是对象,这些旧的方法在处理信息系统模型时常常存在着某些局限性,而面向对象编程语言所体现的思想和方法可以克服。这样,从 20 世纪 80 年代中、后期开始,人们进行了在系统开发各个环节中应用面向对象概念和方法的研究,并逐渐出现了面向对象分析(Object-Oriented Analysis, OOA)、面向对象设计(Object-Oriented Design, OOD)等涉及系统开发其他环节的方法和技术,它们与面向对象编程(OOP)结合在一起,形成了一种新的系统开发方式模型,即面向对象(Object-Oriented, OO)方法。

面向对象的方法中主要涉及以下一些基本概念。

### 二、对象(object)

在面向对象方法中,对象是最基本的概念,简而言之,对象是信息系统必须觉察到的问题域中的人、地点和事物的抽象。在用户眼中,对象是相当于现实世界的某类事物,而在系统分析员眼中,对象则是描述该事物的一组属性数据和作用于这些属性数据之上的操作方法。所以一个对象是把事物的属性和对

属性数据的操作方法结合成的整体。可以把某个信息系统所要处理的问题涉及的领域(范围)叫做问题空间(问题域),而对象是问题空间中与系统问题处理相关的事物的抽象体。对象表示真实的事物,它可以是视觉可见的东西(如房屋、电脑),也可以是抽象概念(如思想和策略),它既可以是主动的、可以启动或控制过程的事物(如传感器和人),也可以是被响应服务请求的事物(如电梯和字典)。

对象有三个特性:预定义行为、状态和标识。预定义行为是对象初始时所定义的动作序列。一个对象可以有任意个属性或域,对象的状态和存储在它的域中的值相对应。而标识是对象的惟一编号,用来与其他对象相区别。对象如何响应接收的消息是由预定义行为和它的状态共同决定的,并通过惟一的标识来与其他对象相区别,从而接收其他对象发送的消息。

一个对象包括一组预定义的行为,每一个消息都会由预定义行为来响应。当对象接收到一个消息时,它的响应方式和结果是由预定义的行为和对象的当前状态共同决定的,状态的不同对响应消息的行为和行为的执行结果会产生影响。对象响应消息时,它的状态也随之变化,这也就意味着它的域值的改变。

对象的标识不依赖于对象的状态和行为,也不随对象状态的变化或行为的发生而变化。这就如同虽然某个学生的状态变化了(如年龄的增长)或发生了一个行为(如放假回家),但他并不会变成另一个人。即使同一个类中的两个对象状态相同,它们仍然是两个不同的对象。

### 三、消息(message)与操作方法(method)

对象之间进行通信的数据叫做消息。当一个消息发送给某个对象时,包含要求接收到消息的对象去执行某些活动的信息,接收到消息的对象对其进行解释,然后予以响应。这种通信机制叫做消息传递。在此过程中,发送消息的对象不需要知道接收消息的对象如何对请求予以响应。

一个消息由下列三个元素组成:

- (1) 消息名。
- (2) 零个或多个参数列表,为接收对象提供数据信息。
- (3) 对接收对象的引用。

当对象收到一个消息时,它就按照预定义的行为完成响应,这些行为是由接收对象的方法定义的。方法就是与一个对象有关的过程,即实现某一操作的一段代码。一个对象可以有任意数目的方法,当一个对象收到一个消息时,它就执行它的某个方法来响应,这个消息的参数作为输入传递给该方法。对象收到一个消息后,决定调用哪个方法来响应该消息的过程称为方法绑定。方法的

选择是由消息名、消息的参数数目和类型共同决定的。方法绑定就是找到型构 (signature) 和消息的元素相匹配的方法：方法名与消息名相匹配，方法的参数表与消息的参数表在类型和数目上相匹配。执行与消息对应的预定义行为可能会产生三种结果：

- (1) 返回一个值给消息发送者。
- (2) 改变接收者的状态。
- (3) 改变作为参数传给接收者的对象的状态。

每类对象都定义一组服务，也就是定义一组作用于该类对象上的操作方法，从而传统的控制结构的功能都可以通过对象及其相互间的消息传递来实现。图 5.1 为对象分解图。

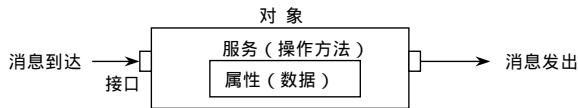


图 5.1 对象的分解图

#### 四、对象接口 (interface)

对象可响应的消息是由对象接口决定的。对象的接口是以一组操作方法的形式出现的，每一个操作都对应于在响应某个消息时对象所完成的预定义行为，客户通过对象的接口来理解对象支持的各种行为，并根据接口来发送消息。

调用对象的操作方法要遵守一定的规则，这些规则描述了如何使用对象的每一个操作。客户要想和对象进行正确交互，就必须遵守对象的接口和规则。例如，当一个 Person 对象接收到带有一个参数的 setAge() 消息时，这个参数必须是整型数值对象，下面的消息就违反了对象调用规则：

```
p.setAge ( null );
```

对象接口可以分为公有接口和限制型接口，其中对象的公有接口强调的是任何类型的对象都可以使用它的操作，也就是说公有接口定义了一组可供任何对象发送的消息和调用的操作方法。而对象提供的限制型操作仅供某些类型的客户使用，这种只供特定类型客户使用的操作称为限制型接口。

#### 五、类 (class)

类是面向对象方法中最重要的概念之一，这种把一组对象的共同特征加以抽象并存储在一个类中的思想正是面向对象方法中重要的一点。类是在对象之上的抽象，一个类定义了一组大体上相同的对象。一个类所包含的操作方法

和属性数据描述了一组对象的共同行为和属性。对象是类的具体化，是类的实例。如学生是一个类，学生丁三、李四是两个对象，是学生类的实例。

每一个对象都属于某个类。类不仅决定了对象的类型，还决定它的域和方法。域和方法在类定义中表达出来。创建一个新对象时，与对象类型相对应的类的定义决定了对象的结构和行为。

一个类的上层可以有超类，下层可以有子类，这样就形成了一种层次结构，在这种类层次结构中，一个类可以有多个超类，也可以有多个子类。在面向对象方法中，类（包括超类、子类）具有对象的全部特征。从性质上看，各种形式的类也是不同层次的对象，类的层次结构就是对象的层次结构。强调类的概念和类层次结构是为了强调对于大体相同的一批对象的共性进行抽取，以形成具有继承关系的类层次结构。而继承性正是面向对象方法建立的系统结构最优越的特性之一。

简单地说，一个类对应着一个概念，它的子类对应着概念中满足某一特殊形式的子概念，每个子类的实例也是父类的实例。类和它的子类之间通常有两种主要的层次结构：

#### （1）分类结构

即一般—具体结构关系，或者是“is a”关系，如汽车、交通工具都是对象类。汽车是子类，汽车是一种（is a）交通工具。这类分类结构即是下面所讨论的继承性的基础，也是面向对象技术中抽象（abstraction）和泛化（generalization）机制的体现。

#### （2）组装结构

即整体—部分结构关系，或者是“a part of”关系。如汽车和车轮都是对象类。车轮是子类，车轮是汽车的一个组成部分。这是一种由部分聚集成整体的聚集关系（aggregation）。对象之间除了层次关系，还有关联关系。关联关系表示对象之间行为的依赖关系和消息传递关系等。关联有多重性，其中包括一对一、一对多、多对多的关联关系。

概括而言，类的主要功能有三个：第一，类定义了类型，即一组对象共有的属性和方法。第二，类提供了它的类型的实现，它定义了实例如何表示以及根据选定的表示方法如何实现。第三，类提供了初始化实例的构造器，构造器可以用来创建和初始化新的实例。

## 六、继承性（inheritance）

继承性是类层次结构中的一个重要特点，是超类和子类之间共享数据和操作方法的机制。通常在定义和实现一个新类时，可以在一个现有的类的基础上进行，把这个已经存在的类所定义的内容作为自己的内容，并加入若干新的内

容,图 5.2 所示为超类 A 和它的子类 B 之间的继承关系。图中类 B 只从其父类 A 得到继承,叫做“单重继承”。如果一个子类有两个以上的父类,则称为“多重继承”。

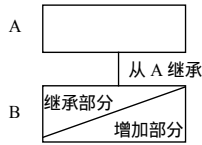


图 5.2 类的继承关系

继承性是面向对象程序设计语言区别于其他语言的重要标志之一,是其他语言所没有的特性。

继承是软件重用的一个基本机制。当通过继承定义一个新类时,新类获得了已存在类的域和行为。新类称为子类,而已存在类称为超类或父类。子类又可以是其他类的父类。这就形成了继承层次结构,就像图 5.3 中的类图所描绘的样子。在这个类图中,每一个父类都与它的子类用线相连,并有一个空心三角形箭头指向父类。Figure 类位于这个图的根;LineSegment、Point 和 Region 是类 Figure 的子类;而 Circle、Ellipse 和 Rectangle 类又是 Region 的子类。

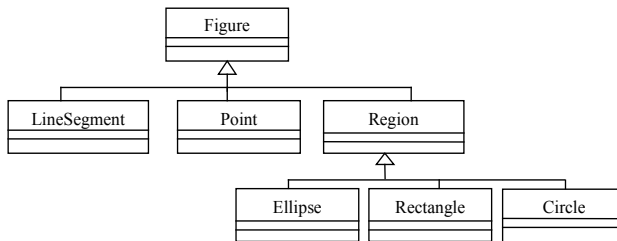


图 5.3 类的层次结构

继承有下面三种使用方式：

- (1) 子类定义新的属性和方法作为对它所继承的属性和方法的补充。
- (2) 子类重新实现所继承的一个或多个方法。被子类重定义的方法称为被覆盖了；也可以说子类覆盖了特定的方法。方法被覆盖后,类的行为发生了变化：对同一个消息,子类和父类的实例调用不同的方法,产生不同的行为。
- (3) 子类实现一个或多个其父类声明但没有实现的方法。这种情况下,这个父类就是一个抽象类,它声明但没有实现的方法称为抽象方法。

使用继承可以创建一个类家族。家族里的每个成员共享位于继承层次根部

的父类定义的操作，每一种类都是父类的子类。在图5.3中，Ellipse类就是以下两种类的子类：Figure、Region。图中所有的类都支持父类Figure定义的操作，而且其中有些子类还有自己的补充操作。

当利用继承定义新类时，这个新子类就可以被客户使用，软件系统也因增加新的类型而扩展了。虽然有了新的类型，但软件系统还是如同以前一样工作，类的父类和使用它们的客户都不会因为子类的加入而受影响或发生变化。每个类都可以通过继承来修改，从这方面说它是开放的；但从客户的角度来看它们却是封闭和固定的，因为这些变化不会影响客户。继承可以在不影响类本身和其他相关元素的情况下达到重用。

## 七、封装性（encapsulation）

封装是把一组相关软件元素组织到一起的方法，其目的在于将对象的使用者和设计者分开，使用者不必知道对象行为实现的细节，只须用设计者提供的对象接口来访问对象。封装的定义为：

对所有对象的内部软件范围的边界进行限定；

对各对象之间相互作用的接口进行描述；

对每个对象的内部实现（操作方法和数据）进行保护。

封装的基本单位是对象，对象的规格说明或接口则作为对象的外部界面，指明该对象所能接收的消息。在对象的内部，每个消息对应一个服务（操作方法），它实施对属性数据的操作，对消息进行响应。

利用封装来隐藏那些不属于对象公有接口的软件元素称为信息隐藏。抽象和信息隐藏是相辅相成的概念。抽象揭示了一个客户要使用一个对象必须了解的内容；信息隐藏则为了防止客户滥用对象，把它没有必要知道的内容隐藏起来。抽象和信息隐藏给出了对象提供的服务——对象接口，与此同时隐藏了是如何完成这些服务的——对象方法的实现。

在过程型程序设计中，中心模块是过程和被过程所封装的数据。这种数据处理方式的标准机制是向函数传递参数并得到返回值。在面向对象程序设计中，中心模块是类。数据和处理数据的过程可以被封装或包含在一个类中。换言之，封装的数据和过程都是类的成员。被封装在类中的域通常是隐藏的，以便为那些可以在类的外部访问的构造函数和方法提供后台支持。

## 八、多态性（polymorphism）

术语多态来自希腊语，意思是“有多种形式”。在收到消息时，对象要予以响应，不同的对象收到同一消息可产生多种不同的结果，即会有多种不同形式，这就是多态。在使用多态时，用户可以发送一个通用的消息，而实现的细

节则由接收对象自行决定，这样，同一消息就可以由不同的对象调用不同的方法来响应，从而产生不同的响应结果。

多态的实现受到继承性的支持，利用类层次的继承关系，把具有通用功能的消息存放在高层次，而实现这一功能的不同的行为放在较低层次，则在这些低层次上生成的对象就能给通用消息以不同的响应。

应用面向对象方法进行系统开发也可分为系统分析、系统设计和系统实施（含编程和测试）等阶段。在面向对象方法中，系统模型的基本单元是对象，是问题空间中客观事物（实体）的抽象。而系统的功能是通过对象之间的消息传递来实现的。客观事物在复杂多变的环境和用户需求的变更中是相对稳定的，因而用面向对象方法建立的系统具有较强的应变能力，各组成部分可重用性好。并且，面向对象方法在系统分析、设计和实施阶段均采用以对象为基本单元的统一模型，只是随着系统开发的进展对模型进行逐步细化与扩充，开发阶段之间有着良好的衔接。这就是面向对象方法在 20 世纪 80 年代后期以来得到迅速发展和广泛应用的主要原因。本章后面将阐述面向对象的主要技术、一般思路 and 面向对象设计与面向对象编程的其他知识。

## 5.2 经典的面向对象方法

### 一、OOA/OOD

OOA/OOD 方法是由 Coad 和 Yourdon 二人共同提出的。这种方法的特点是概念清晰，简单易学。

#### 1. 面向对象分析（OOA）

在系统开发过程中进行了系统业务调查后，就可以按照面向对象的思想来分析问题。OOA 强调在系统调查资料的基础上，针对 OO 方法所需的素材进行归类分析和整理，而不是对管理业务现状和方法的分析。

面向对象的分析方法主要包括：分析发现对象；定义它们的类，然后建立类之间的关系；找出重用类，最后用重用类的实例—对象构造系统框架。这种方法通过对问题空间的若干实例进行抽象，将外部实体的静态特性和动态行为定义为描述对象的属性和服务，实现了系统问题空间与解空间之间分析、表达的一致性以及两个空间之间映射的同态性，从而满足系统对适应需求变化的稳定性。

OOA 可以按照以下五个步骤来进行：

标识对象