清华电脑学堂

UML

面向对象设计与分析 基础教程

牛丽平 郭新志 宋强 杨继萍 等编著

- 总结了作者多年 UML 教学经验
- 全面介绍 UML 应用知识
- 理论讲解虚实结合，简明实用
- 提供丰富的实验指导和习题

清华大学出版社

清華电脑学堂

UML

面向对象设计与分析 基础教程

牛丽平 郭新志 宋强 杨继萍 等编著

清华大学出版社
北 京

内 容 简 介

本书全面介绍使用UML进行软件设计、分析与开发的知识。UML适合于以体系结构为中心、用例驱动、迭代式和渐增式的软件开发过程，其应用领域非常广泛。本书内容包括面向对象的分析方法和设计方法，面向对象分析的三层设计，用例图、类图、对象图和包图、活动图、顺序图和协作图、状态图、构造组件图和部署图等，UML核心语义以及扩展机制的三个重要组成部分：构造型、标记值和约束，使用与UML紧密结合的RUP进行软件开发，对象约束语言，UML在Web应用程序中的应用，使用C++语言实现UML模型（重点介绍类图模型的实现）的基本原理和方法。

本书适合作为普通高校计算机专业教材，也可以作为软件设计人员和开发人员的参考资料。

图书在版编目(CIP)数据

UML面向对象设计与分析基础教程/牛丽平等编著. —北京：清华大学出版社，2007.7
ISBN 978-7-302-15429-7

Ⅰ. U…　Ⅱ. 牛…　Ⅲ. 面向对象语言，UML－程序设计－教材　Ⅳ. TP312

中国版本图书馆CIP数据核字（2007）第086861号

责任编辑：夏兆彦　王冰飞
责任校对：张　剑
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
c－service@tup.tsinghua.edu.cn
社 总 机：010-62770175　**邮购热线**：010-62786544
投稿咨询：010-62772015　**客户服务**：010-62776969
印 刷 者：北京市清华园胶印厂
装 订 者：三河市李旗庄少明装订厂
经　　销：全国新华书店
开　　本：185×260　**印　张**：20.75　**字　数**：488千字
版　　次：2007年7月第1版　　**印　次**：2007年7月第1次印刷
印　　数：1～5000
定　　价：29.80元

前言

FOREWORD

20 世纪 90 年代，人们推出了许多不同的面向对象设计和分析方法。这些不同的面向对象的方法具有不同的建模符号体系，这些不同的符号体系极大地妨碍了软件的设计人员、开发人员和用户之间的交流。因此，有必要在分析、比较不同的建模语言以及总结面向对象技术应用实践的基础上，建立一个标准的、统一的建模语言。UML 就是这样的建模语言，UML 在 1997 年 11 月 17 日被对象管理组织 OMG 采纳成为基于面向对象技术的标准建模语言。统一建模语言 UML 不仅统一了面向对象方法中的符号表示，而且在其基础上进一步发展，并最终被统一为被人们所接受的标准。

UML 相当适合于以体系结构为中心的、用例驱动的、迭代式和渐增式的软件开发过程，其应用领域颇为广泛，除了可用于具有实时性要求软件系统建模以及处理复杂数据的信息系统建模外，还可用于描述非软件领域的系统。

UML 适用于系统开发过程中从需求分析到完成测试的各个阶段：在需求分析阶段，可以用用户模型视图来捕获用户需求；在分析和设计阶段，可以用静态结构和行为模型视图来描述系统的静态结构和动态行为；在实现阶段，可以将 UML 模型自动转换为用面向对象程序设计语言实现代码。

本书以渐进的顺序来介绍 UML，从需求分析开始，然后再构建和部署系统。

第 1 章　主要介绍什么是面向对象的分析方法和设计方法，面向对象分析的三层设计；然后介绍面向对象分析的工具和方法——UML，以及 UML 的主要构成。

第 2 章　主要介绍什么是用例图，用例图的组成，以及如何使用用例图对系统进行需求分析。

第 3 章　介绍类图、对象图和包图的基本概念，重点介绍了类与类之间的关系以及如何建模类图。

第 4 章　主要介绍活动图的相关知识和活动图在 UML 建模中发挥的作用，并辅以图书馆管理系统活动图实例。

第 5 章　介绍系统交互之一的顺序图，其中主要介绍系统顺序图的作用，以及顺序图的组成。UML2.0 在 UML1.x 的基础上，为管理复杂交互添加了顺序片段部分。

第 6 章　主要介绍通信图，通信图也是描述系统交互的动态视图，其在 UML1.x 中称为协作图，在本章介绍了构成通信图的主要部件，以及如何实现通信图与顺序图之间的转换。

第 7 章　本章主要介绍 UML2.0 新增的交互视图——时序图。当正在建模的系统对时间有需求时，就需要使用时序图对其进行交互建模。

第 8 章　本章主要介绍交互概况图和组合结构图，交互概况图将顺序图、通信图和时序图组合在一起，使用各种类型的交互图的特长为用例进行建模。组合结构图则从另一个方面描述了类之间的组成结构。

第 9 章　主要介绍了状态图的基础知识，并着重介绍状态图中的重要元素，最后给出了图书馆管理系统中用到的状态图。

第 10 章　介绍如何构造组件图和部署图。

第 11 章　介绍如何通过与 UML 紧密结合的 RUP 进行软件开发，重点介绍了 RUP 的二维空间和 RUP 的核心工作流程。

第 12 章　介绍如何根据 UML 模型进行数据库设计。

第 13 章　本章由浅入深地介绍了对象约束语言，包括对象约束语言的结构、语法、集合的使用和 OCL 标准库等。

第 14 章　本章从 UML 四层体系结构入手，详细介绍了 UML 核心语义及扩展机制的三个重要组成部分：构造型、标记值和约束。

第 15 章　介绍用 C++语言实现 UML 模型（重点介绍类图模型的实现）的基本原理和方法。

第 16 章　介绍使用 UML 分析一个比较完整的案例——图书管理系统，这一章是对前面基础部分的总结，展示了如何使用 UML 为系统建模。

第 17 章　主要介绍嵌入式系统的分析，以及嵌入式系统的技术特点和开发过程，并通过一个案例——MP3 播放器，介绍 UML 在嵌入式系统中的应用。

第 18 章　主要介绍 UML 在 Web 应用程序中的应用，通过本章的介绍，将使读者对 Web 应用程序的开发有一个全新的认识。

本书特色

本书是一本完整介绍 UML 在软件设计和开发过程中应用的教程，在编写过程中我们精心设计了丰富的体例，以帮助读者顺利学习本书内容。

- **理论紧密结合实践**　全书提供了 3 个完整的分析案例，通过示例分析、设计过程讲解 UML 的应用知识。
- **图文并茂**　UML 理论知识比较抽象，本书绘制了大量 UML 图，帮助读者直观理解抽象内容。
- **网站互动**　我们在网站上提供了本书案例和扩展内容的资料链接，便于读者继续学习相关知识；授课教师也可以下载本书教学课件和其他教学资源。
- **思考与练习**　简答题测试读者对各章内容的掌握程度；分析题理论结合实际，引导读者深入掌握 UML 理论知识。

读者对象

本书在多家院校成熟教案以及自编教材的基础上整合编写，全面介绍使用 UML 进行软件设计、分析与开发的知识，适合作为普通高校计算机专业教材，也可以作为软件设计人员和开发人员的参考资料。

本书作者均从事软件分析、开发和教学工作，拥有丰富的 UML 开发案例。参与本书编写人员除了封面署名人员之外，还有吴俊海、张瑞萍、董志鹏、祝红涛、王海峰、郝相林、刘万军、杨宁宁、郭晓俊、康显丽、辛爱军、牛小平、贾栓稳、王立新、苏静、赵元庆、王蕾、亢凤林、韦潜、郝安林等人。由于时间仓促，书中错误在所难免，敬请读者批评指正。读者可以通过清华大学出版社网站 www.tup.tsinghua.edu.cn 与我们联系。

编　者

目 录

CONTENTS

第1章　UML与面向对象

UML（Unified Modeling Language，统一建模语言）是软件和系统开发的标准建模语言，它主要以图形的方式对系统进行分析、设计。任何大规模的系统设计难度都比较大。从简单的单机桌面程序设计到多层的企业级系统，任何系统都可以分解成为多个软件和硬件。那么，面对如此庞大复杂的结构，我们如何与客户沟通，了解客户对系统的需求？如何在开发人员之间共享设计，以确保各个部分能够无缝地协作？在开发复杂的系统时，如果缺乏相应的帮助工具，则很容易曲解或遗忘许多细节，这就是使用 UML 的原因。

面向对象是一种软件开发方法。软件系统，特别是由多个人开发的大型软件系统，应使用某种方法来开发。甚至由一个人开发的小型软件系统也应通过某种方法进行改进。所有的专业开发人员都相信，合适的开发方法是软件系统开发的基础。因此，在过去的几十年中，人们发明了许多开发方法，如瀑布开发方法、螺旋式开发方法、迭代式开发方法等。面向对象的设计方法是一种新兴程序设计方法，其基本思想是使用对象、类、封装、继承、关联、消息等基本概念来对系统进行分析与设计。

从 20 世纪 80 年代末到 90 年代中出现了一大批面向对象的分析与设计方法。各种流派的方法相互之间各不相同，它们之间的差异为面向对象的程序开发方法的发展和应用带来了不便。在这种情况下，UML 应运而生。UML 是在多种面向对象分析与设计方法相互融合的基础上形成的，是一种专用于系统建模的语言。它为开发人员与客户之间，以及开发人员之间的沟通与理解架起了“桥梁”。

本章学习要点：

- 理解面向对象概念
- 了解 OO 开发
- 熟悉 OO 开发的优点
- 掌握 OO 开发三层设计
- 了解模型的作用
- 了解面向对象的主要概念
- 了解 UML 的发展
- 掌握 UML 四层结构
- 了解统一的含义
- 理解 UML 视图和图的关系
- 掌握 UML 模型元素内容
- 理解 UML 通用机制
- 了解 UML 建模在软件开发中的应用

1.1 面向对象开发

面向对象开发作为一种新兴的软件开发方法，正在逐渐取代传统的方法，日益成为当前软件工程领域的主流方法。

1.1.1 理解面向对象开发

面向对象（Objec-Oriented，OO）不仅是一些具体的软件开发技术与策略，而且是一整套关于如何看待软件系统与现实世界的关系，用什么观点来研究问题并进行求解，以及如何进行系统构造的软件方法学。

概括地说，面向对象方法的基本思想包括两个主要方面。一方面是从现实世界中客观存在的事务出发来构造软件系统，并在系统的构造中尽可能地运用人类的自然思维方式。开发软件是为了解决某些问题，这些问题所涉及的业务范围称为该软件的问题域。面向对象方法强调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为系统中的对象，以对象作为系统的基本构成单位。这可以使系统直接映射问题域，保持问题域中的事物及其相互关系的本质。

另一方面是面向对象方法比以往的方法更接近人类的自然思维方式。虽然结构化开发方法也采用了符合人类思维习惯的原则与策略（如自顶向下、逐步求解等方法），但是与传统的结构化开发方法不同，面向对象方法更加强调运用人类在日常生活中的逻辑思维中采用的思想方法，例如抽象、分类、继承等。这使得开发人员能够更有效地解决问题，并以其他人也能理解的方法将自己的想法表达出来。

如前所述，软件开发是对问题求解的过程。按照软件工程学的观点，可以将软件开发过程分为几个周期，主要包括分析、设计、编程、测试和维护等主要阶段。在软件工程学出现以前，软件开发主要是指编程，在这个阶段软件开发的成功于否完全依赖于开发人员的经验。随着计算机应用邻域的拓广，问题域的复杂性急剧膨胀，而且由于人类思维的局限性，使得软件系统的复杂性和其中包含的错误已经达到开发人员无法控制的程度。于是就出现了 20 世纪 60 年代人们所说的“软件危机”。

软件危机的出现促进了软件工程学的形成和发展。如果将编程技术比作工匠的盖房技术，则软件工程则是一套完整的建筑学体系。这样当要建造一座大楼时，首先需要设计人员根据实地情况对大楼进行设计，然后再由工匠建造。与此类似，当开发大型软件系统时，同样需要开发人员进行分析、设计，最后才是编程实现，以及测试和维护等一系列过程。在整个软件开发的过程中，这需要一整套软件工程理论与技术。

传统的软件开发方法是指所有非面向对象的软件开发方法。在传统的软件工程方法中，其主要特点是将软件开发过程分为如下几个阶段。

- **需求分析** 需求分析是指理解用户需求，就软件功能与客户达成一致，估计软件风险和评估项目代价，最终形成开发计划的一个复杂过程。这个过程为以后的软件设计打下基础。
- **总体设计** 在总体设计阶段，以需要分析的结果为出发点，试图构造一个具体的系统设计方案。主要决定系统的模块结构，包括模块的划分、模块间的数据传递及调用关系。

- **详细设计** 详细设计则是在总体设计的基础上，分析每个模块的内部结构及算法，最终产生每个模块的程序流程图。
- **编程和测试** 编程阶段又称为实现阶段，其主要工作是用一种编程语言开发能够被机器理解和执行的系统。测试则是发现和排序程序的错误，最终形成正常的系统。
- **维护** 软件维护主要分为两种：一种是在软件的使用过程中发现了错误而进行的修改；另一种是因为用户需求发生了变化而进行的维护。

面向对象的软件工程方法的基础是面向对象的编程语言。一般认为诞生于 1967 年的 Simula-67 是第一种面向对象的编程语言。尽管该语言对后来许多面向对象语言的设计产生了很大的影响，但它没有后继版本。继而 20 世纪 80 年代初 Smalltalk 语言掀起了一场“面向对象”运动。随后便诞生了面向对象的 C++、Eiffel 和 CLOS 等语言。尽管在当时面向对象的编程语言在实际使用中具有一定的局限性，但它仍吸引了广泛的注意，一批批面向对象编程书籍层出不穷。直到今天面向对象编程语言数不胜数，在众多领域发挥着各自的作用，如 C++、Java、C#、VB.NET 和 C++.NET 等等。随着面向对象技术的不断完善，面向对象技术逐渐在软件工程邻域得到了应用。

面向对象的软件工程方法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）等内容。

（1）面向对象的分析

OOA 就是应用面向对象方法进行系统分析。OOA 是面向对象方法从编程领域向分析领域发展的产物。从根本上讲，面向对象是一种方法论，不仅仅是一种编程技巧和编程风格，而是一套可用于软件开发全过程的软件工程方法，OOA 是其中的第一个环节。OOA 的基本任务是运用面向对象方法，从问题域中获取需要的类和对象，以及它们之间的各种关系。

（2）面向对象的设计

OOD 指面向对象设计，在软件设计生命周期中发生于 OOA 后期或者之后。在面向对象的软件工程中，OOD 是软件开发过程中的一个大阶段，其目标是建立可靠的、可实现的系统模型；其过程是完善 OOA 的成果，细化分析。其与 OOA 的关系为：OOA 表达了“做什么”，而 OOD 则表达了“怎么做”，即分析只解决系统“做什么”，不涉及“怎么做”；而设计解决“怎么做”的问题。

（3）面向对象的编程

OOP 就是使用某种面向对象的语言，实现系统中的类和对象，并使得系统能够正常运行。在理想的 OO 开发过程中，OOP 只是简单地使用编程语言实现了 OOA 和 OOD 分析和设计模型。

为了加深理解面向对象的开发，下面将比较面向对象开发与传统的软件开发。面向对象的开发方法把完整的信息系统看成对象的集合，用这些对象来完成所需要的任务。对象能根据情况执行一定的行为，并且每个对象都有自己的数据。另一方面，软件系统的传统开发方法则把系统看成一些与数据交互的过程，这些数据与过程隔离保存在不同文件中，当程序运行时，就创建或修改数据文件。图 1-1 显示了面向对象开发与传统软件开发之间的区别。

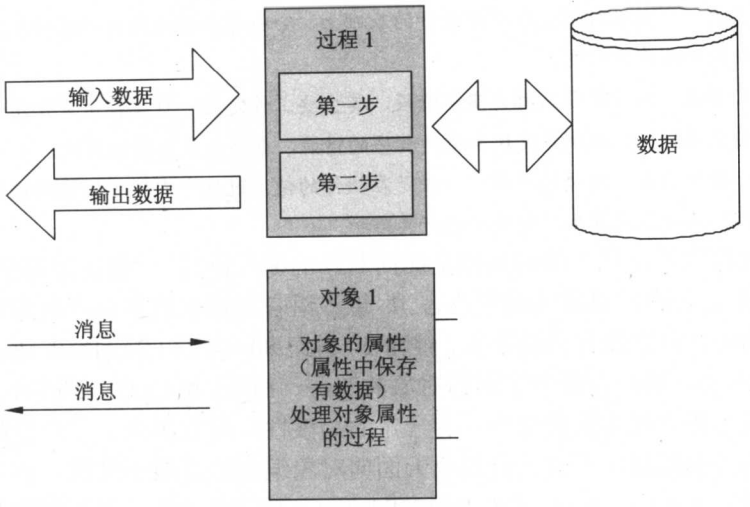

图 1-1 传统方法与面向对象方法的比较

过程通过接收输入的数据，然后对它进行处理，随后保存数据或输出数据。面向对象则是通过接收消息来更新它的内部数据。这些差别虽然看起来简单，但对于整个系统的分析、设计和实现来说却非常重要。

任何一种开发方法，在开发任何系统之前，开发人员都会对此项目先进行分析。系统需求分析就是对系统需求的研究、了解和说明。系统需求定义了系统要为从事业务活动的用户完成的任务。这些需求一般用图表来描述，对图表进行规范化后就构成了该系统的基本需求模型。系统分析过程中建立的模型被称为逻辑模型，因为它仅仅描述了系统的需求，而不涉及到如何实现此需求。系统设计就是建立一个新模型，该模型展示了组成软件系统所使用的技术。系统设计过程中所建立的模型也称为物理模型。

在传统的结构化分析和设计中，开发人员也使用图形模型，如数据流图（DFD）用来表示输入、输出和处理，还要建立实体关系图（ERD）以表示有关存储数据的详细资料。它的设计模型主要由结构图等构成。

在 OO 开发中，因为需要描述不同的对象，所以 OO 开发中所建立的模型不同于传统的模型。例如，OO 开发不仅需要用数据和方法来描述建模，还需要用模型来描述对象之间的交互。OO 开发中使用 UML 来构造模型。

OO 开发方法不仅在模型上与传统的开发方法不同，在系统开发生命周期也有不同。系统开发生命周期是开发一个项目的管理框架，它列出了开发系统时的每个阶段和在每个阶段所要完成的任务。几个主要阶段有计划、分析、设计、实现和支持。系统开发生命周期最初用在传统的系统开发中，但它也能用于 OO 开发中。OO 开发人员经常使用迭代开发方法来分析、设计和实现。

迭代开发方法就是先分析、设计，编写部分程序完成系统需求的一部分。然后再分析——设计——编程完成其他需求。图 1-2 演示了迭代开发方法。

迭代开发方法和早期瀑布开发方法形成了鲜明对比。在瀑布开发方法中，在开始设计前要完成所有的需求分析，然后在需求分析的基础上进行系统设计，编程工作要在系

统分析和设计完成后才进行。虽然传统的开发方法也使用了迭代开发方法，但是因为每个迭代过程都涉及到改进和增加模块的功能，而且在 OO 开发过程中每次迭代增加一个类，所以 OO 开发比传统的开发更适用于迭代开发。

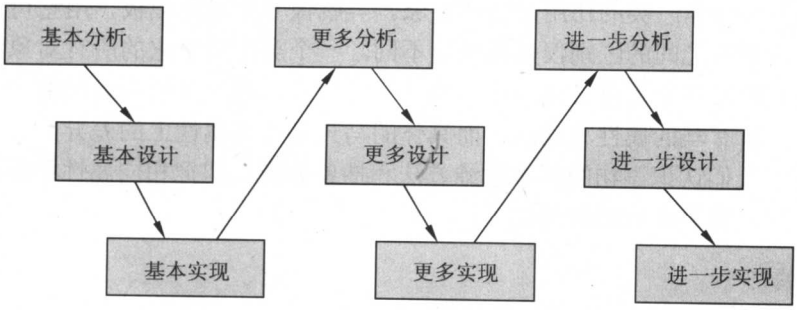

图 1-2 迭代开发

OO 方法在建造系统模型和系统如何工作方面和传统的编程不同，在系统开发生命周期和项目管理中，OO 开发仍然和传统的系统开发有相似之处。

1.1.2 面向对象的主要概念

为了进一步理解面向对象的内涵，下面将逐一介绍面向对象的主要概念。

1. 对象

对象（Object）从一般意义上来讲，它是现实世界中一个实际存在的事物，它可以是看得见摸得着的东西，如汽车；也可以是日常生活中一个抽象的概念，如课程。对象具有自己的状态特性和动作。状态特性即该对象区别于其他对象的特征，它可以用某种数据来描述，例如汽车的型号、载重、颜色等。动作为该对象所表现的行为或对象具有的功能，例如汽车可以移动、拐弯等。

从上面的介绍可知，现实世界中对象是无处不在的。但是在开发一个系统时，通常只需要一定范围内与系统目标有关的事物（即问题域中所涉及的对象），并且用系统的对象抽象地表示。

上面介绍的对象为现实世界中的对象，对于软件系统中的对象而言，对象描述了客观事物中的一个实体，它是构成系统的一个基本单元，由一组属性和操作组成。对象的属性是描述对象状态特性的数据项。对象的操作是对象具有动作的描述，它通常是一组可执行的语句或过程。一个对象可以具有多个属性和多个操作。

2. 类

类（Class）将众多的事物归纳、划分，使相同的事物归为一类。分类所依据的原则是抽象，即忽略事件的非本质特征，只考虑那些与当前需要有关的本质特性，从而找出事物的共性，以便把有共同性质的事物划分为一类。例如马、牛、石头等一些抽象概念，它们描述了该类事件所具有的共同性质。

而在 OO 方法中的类定义为：类是一组具有相同属性和操作的对象集合，它为所有属于该类的对象提供了统一的描述。

在面向对象的编程语言中，类是一个独立的程序单元，它具有类名和该类对象应具有的所有属性和操作。类的作用是创建对象。类就像一个对象模板，用它可以创建许多对象，对象与对象之间的区别仅是属性值不同。一个类为属于它的所有对象给出了统一的定义，而它的对象则是符合这一定义的一个实体。因此对象也称为类的一个实例。例如马类描述了所有马的属性和功能，而具体的马只是在其属性上的差异。

类体现了人们认识事物的基本思维方法和抽象分类，即把相同属性和操作的对象划分为一类，用类作为这些对象的描述。

3. 封装

封装是面向对象的一个重要原则。封装指将对象属性和操作结合在一起，构成一个独立的对象。它的内部信息是隐蔽的，不允许外界直接存取对象的属性，而只能通过指定的接口与对象联系。

封装使得对象属性和操作紧密给合在一起，这反映了事物的状态特性与动作是事物不可分割的特征。系统中把对象看成其属性和操作的结合体，就使对象能够集中而完整地描述一个事物。这避免了将数据和功能分离开进行处理，使得系统的组成与现实世界中的事物具有良好的对应性。

封装的信息隐蔽作用反映事物的独立性。这样使得对于对象外部而言，只需要注意对象对外呈现的行为，而不必关心其内部的工作细节。封装可以使软件系统的错误局部化，因而大大减少了查错和排错的难度。另一方面，当修改对象内部时，由于它只通过操作接口对外部提供服务，因此大大减少了内部修改对外部的影响。

4. 继承

继承是指子类可以拥有父类的全部属性和操作。继承的 OO 方法的一个重要的概念，并且是 OO 技术可以提高软件开发效率的一个重要原因。

在建造系统模型时，可以根据所涉及到的事物的共性抽象出一些基本类，在此基础上再根据事物的个性抽象出新的类。新类既具有父类的全部属性和操作，又具有自己独特的属性和操作。父类与子类的关系为一般与特殊的关系。

继承机制具有特殊的意义。由于子类可以自动拥有父类的全部属性和操作，这样使得定义子类时，不必重复定义那些在父类中已经定义过的属性和操作，只需要声明该类是某个父类的子类，将精力集中在定义子类所特有的属性和操作上。这样提高了软件的可重用性。

断承具有传递性。如果子类 B 继承了类 A，而子类 C 又继承了类 B，则子类 C 可以继承类 A 和类 B 的所有属性和操作。这样子类 C 的对象除了具有该类的所有特性外，还具有全部父类的所有特性。

如果限定每个子类只能继承单独一个父类的属性和操作，则这种继承称为单继承。在有些情况下，一个子类可以同时继承多个父类的属性和操作，这种继承称为多重继承。

5. 消息

消息是指对象之间在交互中所传递的通信信息。当系统中的其他对象需要请求该对象执行某个操作时，就向其发送消息，该对象接收消息并完成指定的操作，然后把操作结果返回到请求服务的对象。

一个消息一般应该含有如下信息：接收消息的对象、请求该对象提供的服务、输入信息和响应信息。

消息在面向对象的程序中具体表现为函数调用，或其他类似于函数调用的机制。对于一个顺序系统，由于其不存在并发执行多个任务，其操作是顺序执行的，因此其消息实现目前主要为函数调用。而在并发程序和分布式程序中，消息则为进程间的通信机制和远程调用过程等其他通信机制。

6. 多态性

在面向对象的开发中，多态性是指在父类中定义的属性和操作被子类继承后，可以具有不同的数据类型或表现出不同的行为。例如，在定义一个父类“几何图形”时，为其定义了一个绘图操作。当子类“椭圆”和“矩形”都继承了几何图形类的绘图操作时，该操作根据不同的类对象，将执行不同的操作，在“椭圆”类对象调用绘图操作时，该操作将绘制一个椭圆，而当“矩形”类对象执行该操作时将绘制一个矩形。这样当系统的其他部分请求绘制一个几何图形时，同样的“绘图”操作消息因为接收消息的对象不同，将执行不同的操作。

在父类与子类的类层次结构中，利用多态性可以使不同层次的类之间共享一个方法名，而各自有不同的操作。当一个对象接收到一个请求消息时，所采取的操作将根据该对象所属的类决定。

在继承父类的属性和操作的名称时，子类也可以根据自己的情况重新定义该方法。这种情况称为重载。重载是实现多态性的方法之一。

7. 关联

在现实世界中，事物不是孤立的、互相无关的，而是彼此之间存在着各种各样的联系。例如在一个学校中，有教师、学生、教室等事物，他们之间存在着某种特定的联系。在面向对象的方法中，用关联来表示类或对象集合之间的这种关系。在面向对象中，常把对象之间的连接称为链接，而把存在对象连接的类之间的联系称为关联。

如果在OOA和OOD阶段定义了一个关联，那么在实现阶段必须通过某种数据结构来实现它。关联还具有多重性，多重性表示参加关联的对象之间数量上的约束，有一对一、一对多、多对多等不同的情况。

8. 聚合

现实世界中既有简单的事物，也有复杂的事物。当人们认识比较复杂的事物时，常用的思维方法为：把复杂的事物分解成若干个比较简单的事物。在面向对象的技术中像这样将一个复杂的对象分解为几个简单对象的方法称为聚合。