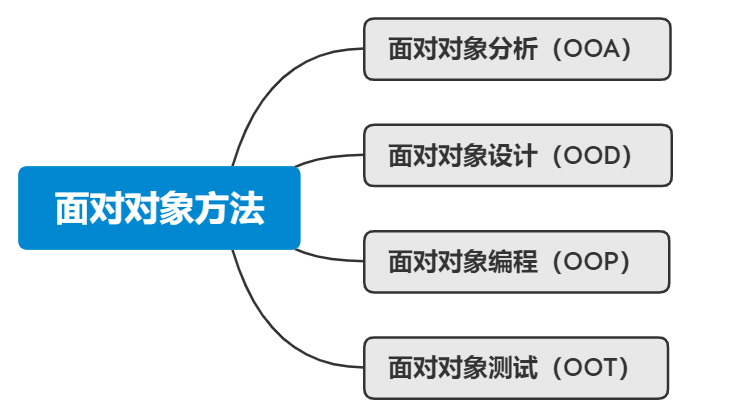
# 2.2 面对对象开发

面对对象方法（OO：Object Oriented）具有很强的的类的概念，因此它能很直观地模拟人类对客观世界的认识方式。也能模拟人类在认知过程中的由一般到特殊或由特殊到一般的归纳功能。前面介绍的类的概念既能够反映出对象的本质属性，又提供了实现对象共享机制的理论依据。

如果遵照面对对象方法的思想进行软件系统的开发，其过程分为以下4个阶段：



## 1. 系统调查和需求分析：OOA

**1.什么是面对对象分析（OOA：Object-Oriented Analysis）？**

对用户的开发需求以及要开发的系统所面临的问题进行调查和研究。针对复杂的问题领域，抽象出对象及其属性和方法。

**2.OOA需要做什么？**

OOA阶段主要目的是提取系统的需求，分析出为了满足用户的需求，系统必须“做什么”（系统能提供的功能），而不是“怎么做”（系统如何实现）。

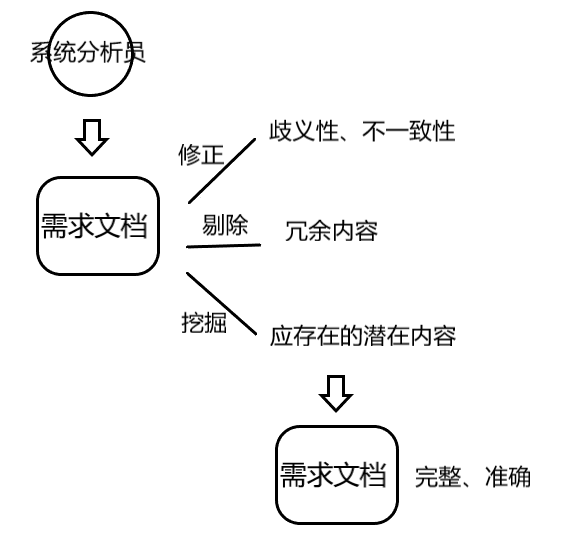
**3..OOA分析什么？**

实例需求文档。需求文档也叫需求陈述或问题陈述。对于要开发的任何一个系统，需求陈述是首要任务。因为系统最终是要由用户使用，而在该过程中，主要是陈述用户的需求，即该系统应该“做什么”，而不是“怎么做”，即系统要完成的任务是什么，而不是解决问题的方法。

在进行需求陈述时，必须要清楚所要解决问题的目标。如果目标模糊，将会影响整个系统分析、设计和实现等后续开发阶段的所有工作。也就是说，需求质量的好坏直接影响到整个系统的质量，是个很关键的过程，如果想准确表达用户的要求，在对需求分析进行陈述时需要分析人员和用户一起研究和讨论。

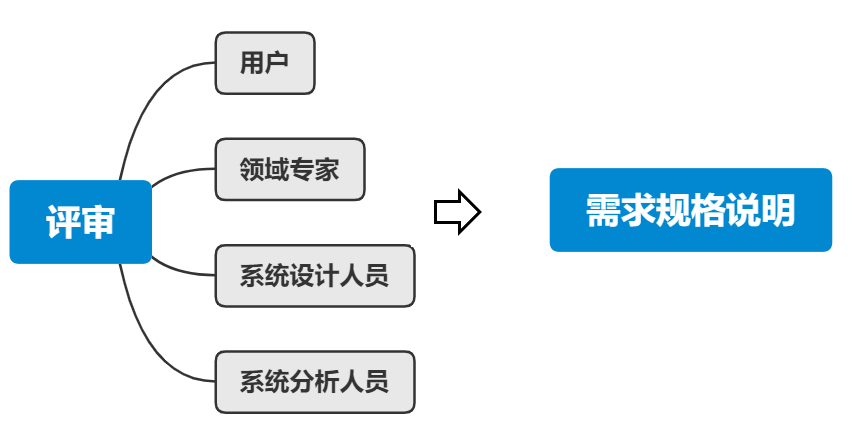
**4.OOA过程概述**

分析：在OOA阶段，系统分析员要对需求文档进行分析。通过分析可以发现并对需求文档中的歧义性、不一致性进行修正，剔除那些冗余内容，挖掘系统中应存在的潜在内容，弥补系统中的不足，从而使需求文档更完整和准确。



建模：对需求文档进行分析和整理后，为了给面对对象分析过程提供依据，要进行需求建模。这是系统分析员根据提取的用户需求，对用户的需求进行深入地理解，识别出问题领域内的对象，并分析对象之间的关系，抽象出目标系统需要完成的任务，这样就可以利用OOA模型准确地表示出来，即用面对对象观点建立对象模型、动态模型和功能模型。

评审：经过需求的分析和建模，最后对所得的需求进行评审。通过用户、领域专家、系统分析员和系统设计人员的评审，并进行反复修改后，最终确定目标系统的需求规格说明。



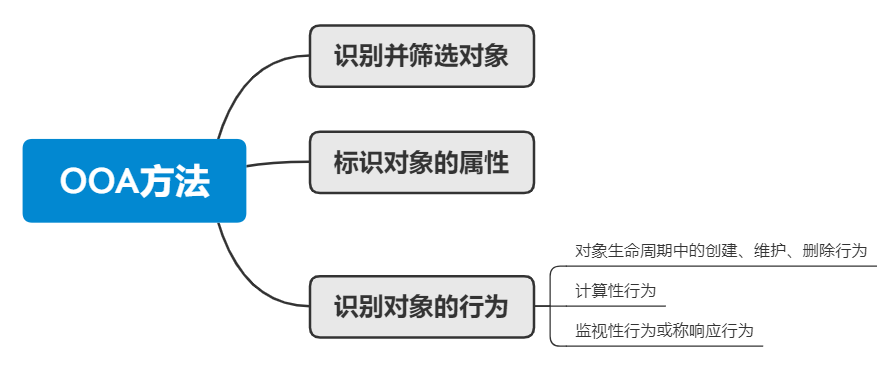
**5.OOA方法（面对对象的分析方法）**

面对对象的分析方法，指的是按照面对对象的概念和方法，在对任务的分析中，根据客观存在的事物以及事物之间的关系，归纳出相关的对象，包括对象的属性、行为及对象之间的联系，并将具有共同属性和行为的对象用一个类来表示。

通过面对对象的分析，建立一个能反映真实工作情况的需求模型。在这个阶段所形成的的模型只是一个比较粗略的模型。OOA所强调的是在系统调查资料的基础上，进行对象的归类分析和整理。

使用OOA方法对系统调查和需求分析进行分析处理时，一般要遵循前面所讲述的抽象、封装、继承、消息通信等原则。

**OOA方法分为以下几个阶段：**



**1.识别并筛选对象**

对象的定义：

（1）对象应该是实际问题域中有意义的个体或概念实体；

（2）对象具有目标软件系统所关系的属性；

（3）对象应该以某种方式与系统发生关联，即对象必须与系统中其他有意义的对象进行消息传递，并提供外部服务；

如何识别与筛选：

（1）通过对用户需求分析文档的分析找出所有名词或者名词短语，合并同义词；

（2）除去具有动作含义的名词，这些动词将被描述为对象操作而不是对象本身；

**2.标识对象的属性**

属性的定义：

（1）属性是对问题域中对象性质的一个描述；

（2）对象在系统中所有可能的状态就是属性的取值；

如何正确识别属性：

（1）在识别属性的过程中，对于问题领域中的某个实体，不但要求其取值有意义，而且它本身在系统中必须要是独立存在。这时应该将该实体看做一个对象，而不能作为另一对象的属性；

（2）简洁性：为了保持需求模型的简洁性，一般将省略对象的一些导出属性。

例如：“年龄”可由“出生日期”和当前时间导出，因此，“年龄”属于实体“人”的导出属性，所以人基本属性中不包含“年龄”。

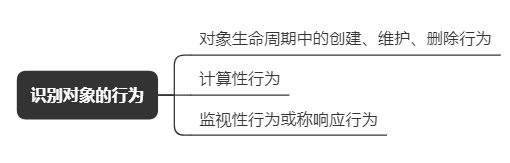
**3.识别对象的行为**

行为的定义：对象的行为可以简单地理解为对象对外提供的所有功能。

比如：一个对象处理另一个对象的请求、查询或命令，即响应外部的事件，要完成某项操作，这种操作将该对象自身的属性值或系统的状态，这些都是对象的行为。

可以把整个系统看成是对象之间的相互通信，以及在通信过程中引发的动作。

一般将对象的行为分为以下三类：



（1）对象生命周期中的创建、维护、删除行为

例如图书管理系统中图书信息的创建、删除和修改等行为。

（2）计算性行为

典型的计算性行为包括：利用基本的对象属性值计算派生出的属性值，以及为了响应其他对象的请求，完成某些数据处理功能，并将结果返回。

这类计算性行为往往完成的是数据处理功能，即对象提供的外部的计算性行为。

分析人员可以根据在定义对象的外部行为时，针对其他对象发出的消息请求提取计算性行为。

（3）监视性行为或称响应行为

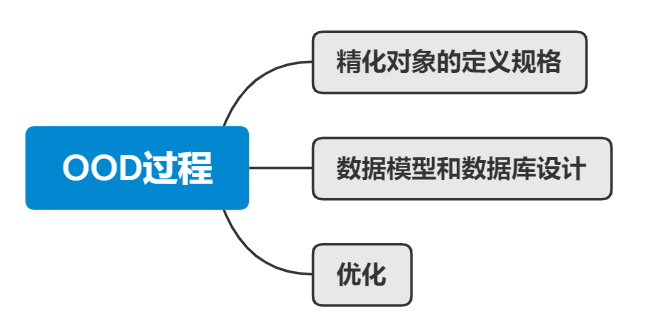
为了提取对象的响应行为，分析人员需要对对象的主要状态进行定义。对于每一个状态，列出可能的外部事件，预期的反应，并进行适当的精化。例如，“图书”的对象状态可以为“借出”、“在库”等，在每一个状态可处理的事件及预期反应可以表示为响应行为。

## 2. 面对对象设计方法：OOD

**1.什么是面对对象设计（OOD：Object-Oriented**  **Design）？**

对OOA的结果进行进一步的抽象、归类整理。对每一部分进行分别的具体设计，先是进行类的设计，类的设计可能包含多个层次（利用继承、派生）。然后在这些类的基础上，提出程序设计的思路和方法，对算法进行设计。在设计阶段，不牵扯某一种具体的计算机语言，而是用一种更通用的描述工具进行描述。

**2.OOD的设计过程：**



（1）精化对象的定义规格

对于OOA所抽象出来的对象和类以及分析过程中产生的分析文档，OOD需要有一个根据设计要求整理和求精的过程，使之更能符合OOP的需要。这个整理和求精过程主要有两个方面：

一是要根据面向对象的概念模型整理分析所确定的对象结构、属性、方法等内容，改正错误的内容，删去不必要和重复的内容等。

二是进行分类整理，以便于下一步数据库设计和程序处理模块设计的需要。整理的方法主要是进行归类，对类和对象、属性、方法和结构、主题进行归类。

1. 数据模型和数据库设计

数据模型的设计需要确定类和对象属性的内容、消息连接的方式、系统访问、数据模型的方法等。最后每个对象实例的数据都必须落实到面向对象的库结构模型中。

1. 优化

OOD的优化设计过程是从另一个角度对分析结果和处理业务过程的整理归纳，优化包括对象和结构的优化、抽象、集成。对象和结构的模块化表示OOD提供了一种范式，这种范式支持对类和结构的模块化。这种模块符合一般模块化所要求的所有特点，如信息隐蔽性好，内部聚合度强和模块之间耦合度弱等。集成化使得单个构件有机地结合在一起，相互支持。

## 3. 程序实现：OOP

**什么是面对对象编程（OOP：Object-Oriented Programming）？**

利用面对对象的程序设计语言，进行系统的实现，即面对对象编程（OOP），是一种计算机编程架构。OOP 的一条基本原则是计算机程序是由单个能够起到子程序作用的单元或对象组合而成。OOP 达到了软件工程的三个主要目标：重用性、灵活性和扩展性。为了实现整体运算，每个对象都能够接收信息、处理数据和向其它对象发送信息。

## 4. 系统测试：OOT

**1.什么是面对对象测试（OOT：Object-Oriented Test）？**

系统开发好后，在交付用户使用前，必须对程序进行严格的测试。测试的主要目的就是发现程序中的错误，进行改正，使得系统更加健壮。面对对象测试时，采用面对对象的方法进行测试，以类作为测试的一个基本单元，即面对对象测试（OOT）

# 2.3 软件建模概述

**1.软件建模的概念**

模型是对现实存在的实体的抽象和简化，模型提供了系统的蓝图。模型过滤非本质的细节信息，抽象出的问题本质，使问题更容易理解。抽象是一种允许我们处理复杂问题的方法。为了建立复杂的软件系统，我们必须抽象出系统的不同视图，使用精确的符号建立模型，验证这些模型是否满足系统的需求，并逐渐添加细节信息把这些模型转变为实现。这就是软件建模。

软件建模是开发优秀软件的一个核心工作，其目的是把要设计的结构和系统的行为联系起来，并对系统的体系结构进行可视化和控制。

可视化的建模的是使用一些图形符号进行建模，可视化建模的作用如下：它可以捕捉用户的业务过程，可以作为一种很好的交流工具，可以管理系统的复杂性，可以定义软件的架构，还可以增加重用性。

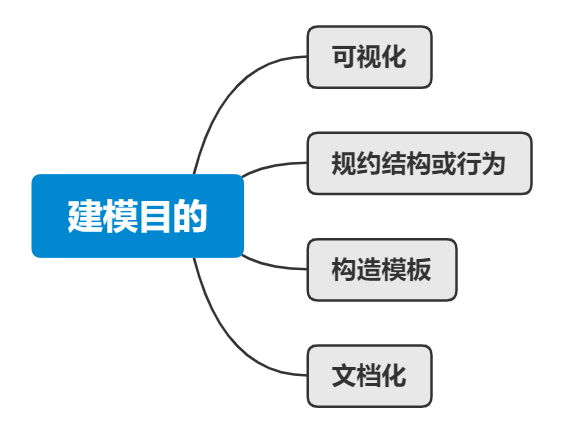
**2.建模目的**

（1）模型有助于按照实际情况或按照所需要的样式对系统进行可视化；

（2）模型能够规约系统的结构或行为；

（3）模型给出了指导构造系统的模板；

（4）模型对作出的决策进行文档化；



**3.软件建模的用途**

现在的软件越来越大，大多数软件的功能都很复杂，使得软件开发只会变得更加复杂和难以把握。

解决这类复杂问题最有效的方法之一就是分层理论：即将复杂问题分为多个问题逐一解决。

软件模型就是对复杂问题进行分层，从而更好地解决问题。

这就是为什么要对软件进行建模的原因。有效的软件模型有利于分工与专业化生产，从而节省生产成本。为了降低软件的复杂程度，便于提早看到软件的将来，便于设计人员和开发人员交流使用了建模技术。对于软件人员来说，模型就好像是工程人员的图纸一样重要。只是目前来看软件模型在软件工程中的重要性还远远没有达到图纸的在其它工程中地位。

**4.软件建模的优点（其实就是软件建模的目的）**

（1）使用模型便于从整体上、宏观上把握问题，可以更好的解决问题。

（2）可以加强工作人员之间的沟通，可以更早的发现问题或疏漏的地方。

（3）模型为代码生成提供依据，帮助我们按照实际情况对系统进行可视化。

（4）模型允许我们详细说明系统的结构或行为，模型给出了一个指导我们构造系统的模板，并对我们做出的决策进行文档化。