

## Java 编程思想 -- 面向对象逻辑思维

什么是面向对象的设计思想？也许有不少初学者对这个概念还有许多不明白的地方，特别是这个处于新旧思想交替的时代，许多人刚刚学完现在看来是快要淘汰的只是面向过程的语言。他们的脑子还没有脱离面向过程思想的束缚，抬头却发现，“面向对象”早已经流行开来，这个陌生的词漫天飞舞。随便拿起一本流行计算机技术书籍，那里会没有“面向对象”的字眼！于是心中便惶惑了：什么叫“面向对象”？不感兴趣者，一带而过；有志于在这方面发展的，匆忙找到一本有关书籍来啃究竟什么是“面向对象”。然而，要突破思想的束缚，说难也不难，说到要深刻认识却也不容易。笔者在做了一些轻量级的工作后，颇以为有点心得，不怕贻笑大方，写出已供广大同行批评指正。

“对象（Object）”一词，早在十九世纪就有现象学大师胡塞尔提出并定义。对象是世界中的物体在人脑中的映象，是人的意识之所以为意识的反映，是做为一种概念而存在的先念的东西，它还包括了人的意愿。举例说明吧。当我们认识到一种新的物体，它叫树，于是在我们的意识当中就形成了树的概念。这个概念会一直存在于我们的思维当中，并不会因为这棵树被砍掉而消失。这个概念就是现实世界当中的物体在我们意识当中的映象。我们还可以有我们自己的意愿，虽然我们并不需要付诸实现——只要在你的脑中想着把这棵树砍掉做成桌子、凳子等——我们就把它叫做意向。于是，对象就是客观世界中物体在人脑中的映象及人的意向。只要这个对象存在我们的思维意识当中，我们就可以籍此判断同类的东西。譬如，当我们看到另外一棵树是，并不会因为所见的第一棵树不在了失去了供参照的模板而不认识这棵树了。当我们接触某些新事物时，我们的意识就会为这些事物确立一个对象。当然这个过程是怎么形成的，那就不是我们所能讨论的问题了。上面所说的对象研究的是一般意义上的问题，因而它可以外推到一切事物。我们经常所说的“对象”，一班指的

是解决信息领域内所遇到问题的方法。特别是应用软件技术来决问题的方法。如我们经常碰到的面向对象的编程 ( Object-Oriented Programming )、面向对象的分析 ( Object-Oriented Analysis )、面向对象的设计 ( Object-Oriented Design ) 等。应用前面所介绍的关于对象的概念,可以对这些问题做进一步的分析。在面对较复杂的系统,我们可以将它作为一个对象来进行分析。一个系统 ( 解决某个问题的全套解决方案 ) 作为一个对象,可以由多个部分组成。同样,这个对象也可以由多个对象组成。对于同类的事物,可以由一个对象来表示。这样做的益处是显而易见的,它灵活而高效,可以大大减轻设计人员的工作量,简化实际的模型。举一个例子。在关系型数据库的设计当中,我们可以把一个元组当作对象,给它定义一组操作方法。这些方法将适用于所有元组,从而我们不必在更大的范围内去细致的考虑不同的元组 ( 如判断一个元素是否合法 ) : 因为它们有一组公共的面向本身的方法,它们“自己”可以“解决”自己的问题。更上一层的对象可以是一个表、视图等。表对象在元组对象的基础上又有它们自己的方法,如增加、删除等。从这个层面上讲,它也只需要做“自己”的事情,因为有元组对象的支持,它无须去考虑像元素是否合法这类的事情。甚至,有时为了满足我们还可以将元素或表群当作对象并定义它们自己的方法。这样,更能显示面向对象的优势。

上面所讨论的可以说是面向对象的分析方法。在具体的设计过程当中,还应该采用适当的方式。因为面向对象的思想固然很先进,如果做得不好的话,同样不能达到预期的效果。这主要表现在处理对象与对象的关系上没有做好,对象与对象的层次不分明。如上面所举得关系型数据库得例子,如果在元组层面上得对象过多得考虑一个表对象得因素,或一个表层面上对象过多地考虑一个元组对象的因素,甚至去考虑元素层面上的因素,这些都不是好的面向对象的设计方法。这一点,在语言实现方面,Java 比 C++ 更有优势,因为它不允许多

重继承，从而使对象之间的关系更明确。谁也不会否认 C++ 的功能更强大，但是它也要由次付出巨大代价——当现在代码库发展到一定程度、一定规模时，个对象之间的层次关系将变得异常复杂，给后继使用者得学习、理解带来很大的困难，应用上很难把握。另外，虽然 C++ 具备面向对象的处理能力，但它还是保留了很多面向过程的东西。用 C++ 完全可以不用面向对象的思想来进行程序设计，当然人们不会这样去做——除了那些只是把 C++ 看成是 C 扩充的初学者，这就为以后的发展埋下了隐患。在者一方面，Java 的限制更多一点。就者一点还远远不够。搞开发的是人，开发方法是由人决定的。要应用面向对象的方法开发出优秀的软件，必须要求开发人员具有良好的面向对象的思想。好的工程师可以利用适当的工具开发出优秀的软件——而不在乎他所使用的语言工具——Java、C++、Object Pascal、Ada 等。