

AOP：通过面向方面编程提高代码的封装和复用性 PDF转换  
可能丢失图片或格式，建议阅读原文

[https://www.100test.com/kao\\_ti2020/140/2021\\_2022\\_AOP\\_EF\\_BC\\_9A\\_E9\\_80\\_9A\\_E8\\_c100\\_140047.htm](https://www.100test.com/kao_ti2020/140/2021_2022_AOP_EF_BC_9A_E9_80_9A_E8_c100_140047.htm) 摘要面向方面编程 (AOP)

是施乐公司帕洛阿尔托研究中心 (Xerox PARC) 在上世纪 90 年代发明的一种编程范式，它使开发人员可以更好地将本不该彼此纠缠在一起的任务（例如数学运算和异常处理）分离开来。AOP 方法有很多优点。首先，由于操作更为简洁，所以改进了性能。其次，它使程序员可以花费更少的时间重写相同的代码。总之，AOP 能够为不同过程提供更好的封装性，提高未来的互操作性。是什么使软件工程师都希望自己能成为硬件工程师呢？自从函数发明以来，程序员花费了大量时间（及其老板的大多数资金）试图设计这样的系统：它们不过是一些组合模型，由其他人创建的部件构成，布置成独特的形状，再覆盖上一些悦目的颜色。函数、模板、类、组件等等一切，都是软件工程师自己创建“软件集成电路”（模拟硬件设计师的电子器件）的种种尝试。我把这些都归咎于 Lego（乐高玩具）。把两个玩具块（即组件）拼起时发出的悦耳的咔哒声很让人上瘾，会促使许多程序员发明一种又一种新的封装和重用的新机制。这方面最新的进展就称为面向方面编程 (AOP)。AOP 的核心是安排（一个摞在另一个之上）组件的一种方式，可以获得其他种类基于组件的开发方法无法得到的重用级别。这种安排是在客户端和对象之间的调用堆栈中进行的，其结果是为对象创建了一种特定的环境。这种环境正是 AOP 程序员主要追求的东西，继续阅读本文，您将了解这一点。随本文一起提供的代码示例分为两部分

: COM 部分和 Microsoft .NET 部分。COM 部分创建了一种基础结构，可以在 COM 对象中添加方面，提供用户界面来配置类的方面，还给出了在我们提供的基础结构上创建的一个示例方面实现。.NET 部分说明了如何使用内置于 .NET 基础结构来完成 COM 版本同样的任务，但是所用代码更少，选择也更多。也提供了适合此基础结构的示例方面。100Test 下载频道开通，各类考试题目直接下载。详细请访问 [www.100test.com](http://www.100test.com)