

Nhibernate与代码生成器 PDF转换可能丢失图片或格式，建议
阅读原文

https://www.100test.com/kao_ti2020/472/2021_2022_Nhibernate_c67_472938.htm 原来一直使用代码生成，包括CodeSmith和CodeMatic。最近打算系统学习一下Nhibernate，经过简单的一点探索，发现ORM和代码生成真是有个有千秋。本文侧重比较一下ORM和代码生成二者的优缺点，让二者华山一比较高下，目的为去伪存真，使二者能够相辅相成。从而更好的提高开发效率。本文从下面三个层面对ORM（以NHibernate为首发的O阵营）和代码生成（以CodeMatic为首发的C阵营）进行比较：1）针对数据库二者的架构层次上的异同 2）针对应用程序二者在使用和配置上的异同 3）针对业务逻辑二者在对变化和复杂度上支持度的异同。下面就具体针对这三个层面做一下具体分析，这些分析都来源于自己开发中的一些经验和心得，有些是正确的，有些也存在这样那样的问题。写出来，希望的也只是能抛砖引玉，得到更多朋友，兄弟的帮助和支持。

1) 针对数据库二者在架构层次上异同 ORM针对数据库是由上而下的关系，也就是说ORM并不依赖于数据库。他可以完全从关系数据库中将程序员解放出来，需要程序员小心呵护的是传递给nhibernate的persistent object。这看起来更加OO，而代码生成恰恰相反，代码生成依赖于关系数据库。它总结数据库操作的一些共性，将本来需要程序员手写的代码自动生成出来。从OO的角度来说，代码生成的过程并不体现OO思想，但根据模版或者软件作者的一些逻辑。生成出来的代码却可能具有很好的OO思想。针对数据库来说，ORM是自顶向下的，代码生成则是自下而上。二者方向恰好相反

。 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