

# FEM之在求解器中使用设计模式(9)---Facade模式和Proxy模式

原创 [www.cae-sim.com](http://www.cae-sim.com) [多物理场仿真技术](#)



再介绍软件开发中常用的两种设计模式 Facade（门面）模式和Proxy（代理）模式。

Facade模式个人认为是和单例一样，是比较简单也容易理解的一种模式。

Facade的含义是当外部和一个子系统沟通时，可以通过一个统一的门面对象进行。

比如客户发现了软件bug，客户可以和软件团队的架构师，高级工程师，测试工程师，产品工程师任意一个人反馈，但最好的是直接反馈给产品经理，产品经理再和团队内部的人员沟通。这就保证客户和研发部门始终只通过产品经理沟通，而产品经理就是门面。

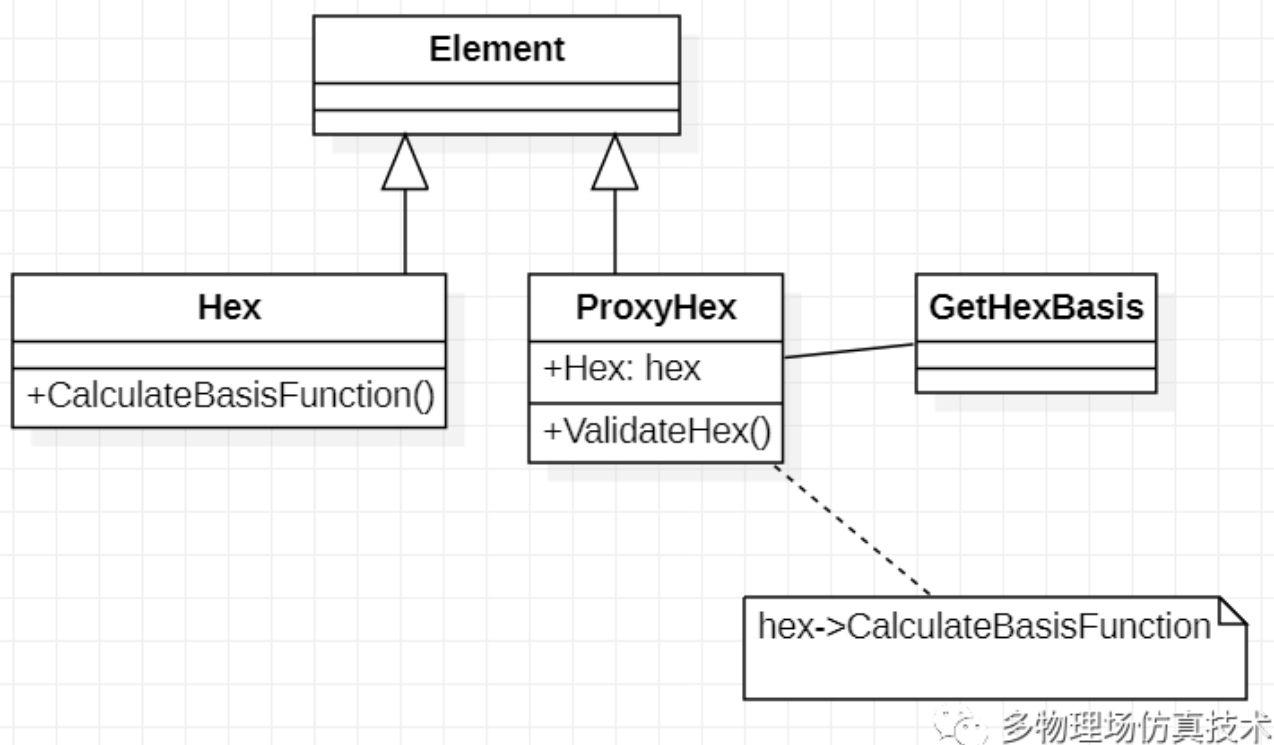
所以从这个意义看，Facade是比较容易理解和实现的模式。在求解器开发中可以尽量使用这种模式。比如在组装刚度矩阵过程中，需要频繁取网格中的各种几何和属性数据，每种数据都有自己的数据结构，这时候可以构造一个网格管理器，通过管理器的接口来存取数据，避免直接存取数据。这样做的好处是组装刚度矩阵只需要关注组装逻辑，而不用花太多精力去管理存取数据，同时也便于任务的拆分。Facade比较简单，就不画UML图了。

## Proxy（代理）模式

代理模式的目的是为对象提供一种代理以控制对象的访问。

代理模式在数据管理中也是一种常见的模式，比如我们定义的网格数据：点的坐标。设计点的类通常包含三维坐标xyz，如果需要计算一个点到另一个点的距离，有两种方法，一是在点的类内部实现，也可以添加另外的工具类来实现；求解器的数据中有很多类似的结构需求。

在类的内部实现功能，类似函数多的时候，类的设计臃肿，且和类本身定义没有多大关联；如果使用工具类，缺乏对类本身细度的控制，而Proxy代理模式能很好解决这些问题。



上图显示了使用ProxyHex代理类来计算六面体Hex的基函数，该代理类使用ValidateHex函数检验Hex数据的有效性，如果有效则调用Hex的基函数的计算函数，从而简化了Hex类的设计。