

FEM之在求解器中使用设计模式(1)--Bridge模式

原创 www.cae-sim.com [多物理场仿真技术](#)
收录于合集
#设计模式 8
#求解器开发 17



首先很佩服工程出身的人能编写代码，事实上，很多优秀的求解器不是计算机出身的人开发的。之所以想写《在求解器中使用设计模式》系列有关技术细节的文章，是因为自己在参考很多开源求解器代码时，发现很少有开发者能基于软件工程，从代码的维护性，扩展性，稳定性，移植性，兼容性方面来考虑设计。

且不说用100数组定义一个单元的未知自由度有多么耗内存，也不说如何阅读有100多个参数的函数，更不用说如何在如下代码添加新单元：判断单元类型用switch语法加上20多个case，每个case的代码有一百多行。单是看着一个简单的用线性三角单元求解拉普拉斯方程分别用Matlab, C, Fortran, VB, Java写上一遍就觉得很无语：在一件简单事情上做了太多重复性的工作。

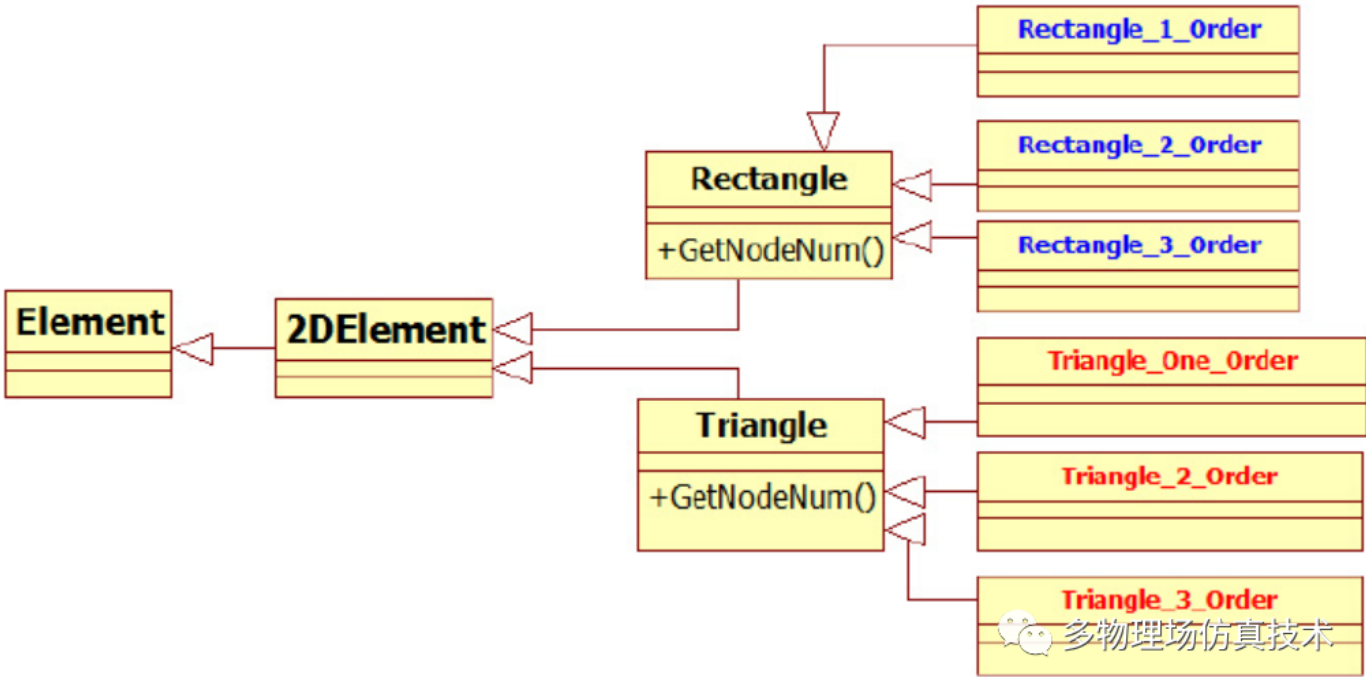
设计模式 (Design pattern) 是一套被反复使用、多数人知晓的、经过分类编目的、代码设计经验的总结。使用设计模式是为了可重用代码、让代码更容易被他人理解、保证代码可靠性。毫无疑问，设计模式于己于他人于系统都是多赢的；设计模式使代码编制真正工程化；设计模式是软件工程的基石脉络，如同大厦的结构一样。

首先介绍Bridge模式（桥接模式）：

考虑如下问题：

有限元单元按照阶次可以分为线性，二次，高次单元。以2D三角单元为例，线性单元（3个点），二次单元（6个点，每边中点有一个点）

类图通常设计如下图1：

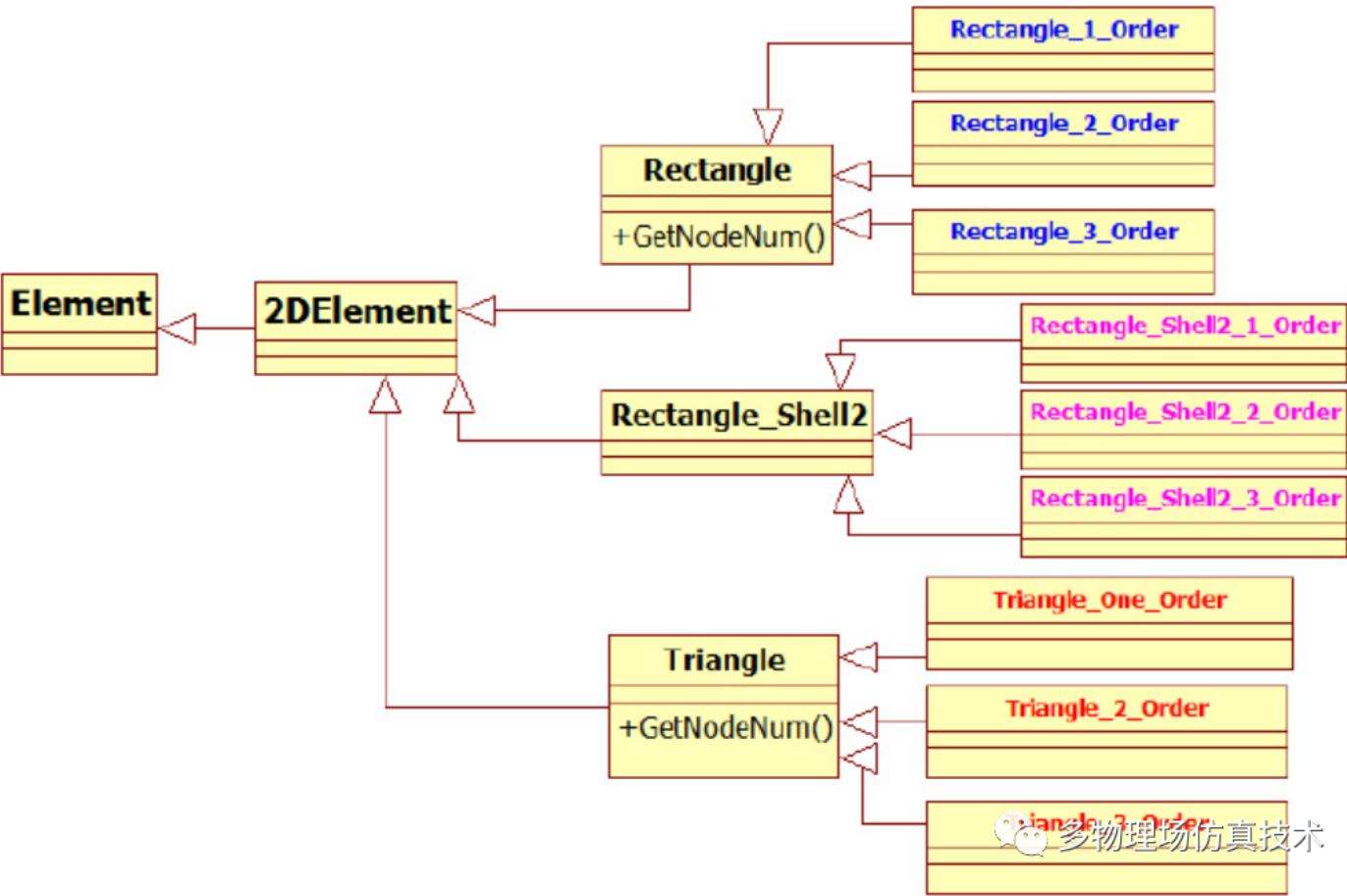


GetNodeNum()用来获得单元的节点数目：

以四边形Shell单元为例：

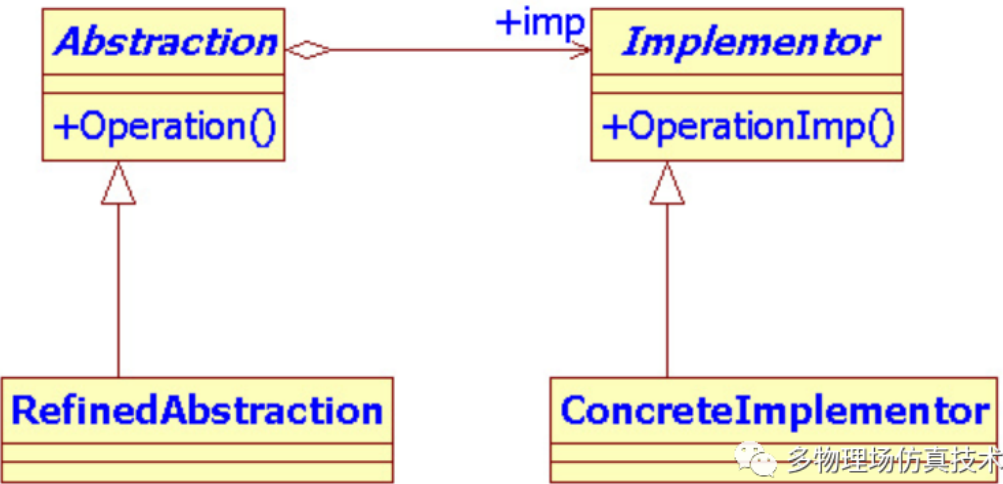
Rectangle_1_Order()=4; 1阶 4个点

Rectangle_2_Order()=8; 2阶8个点
Rectangle_3_Order()=12; 3阶12个点
现在需要添加一种新的四边形单元类型Rectangle_Shell2。
图1变为如下图2:



可以看出增加一种单元类型，要增加4个类，还不考虑更高次的单元（已有软件最高可以达到8次，Ansys中仅Plane/Shell单元就有40多种，如果全部添加则需要8*40=320个类，而且还只考虑了长方形这一种单元，显然不合理）

- 针对这个问题有两种解决办法：
1. 在每个单元内部根据阶次计算节点个数。这种办法肯定可行，但是增加了单元的复杂性，也造成了代码的重复。
 2. 利用Bridge模式，将单元节点计算和单元抽象分离。
- 典型的Bridge图如下：



【GOF95】在提出桥梁模式的时候指出，桥梁模式的用意是"将抽象化(Abstraction)与实现化(Implementation)脱耦，使得二者可以独立地变化"。这句话有三个关键词，也就是抽象化、实现化和脱耦。

1.抽象化：

存在于多个实体中的共同的概念性联系，就是抽象化。作为一个过程，抽象化就是忽略一些信息，从而把不同的实体当做同样的实体对待。

2.实现化：

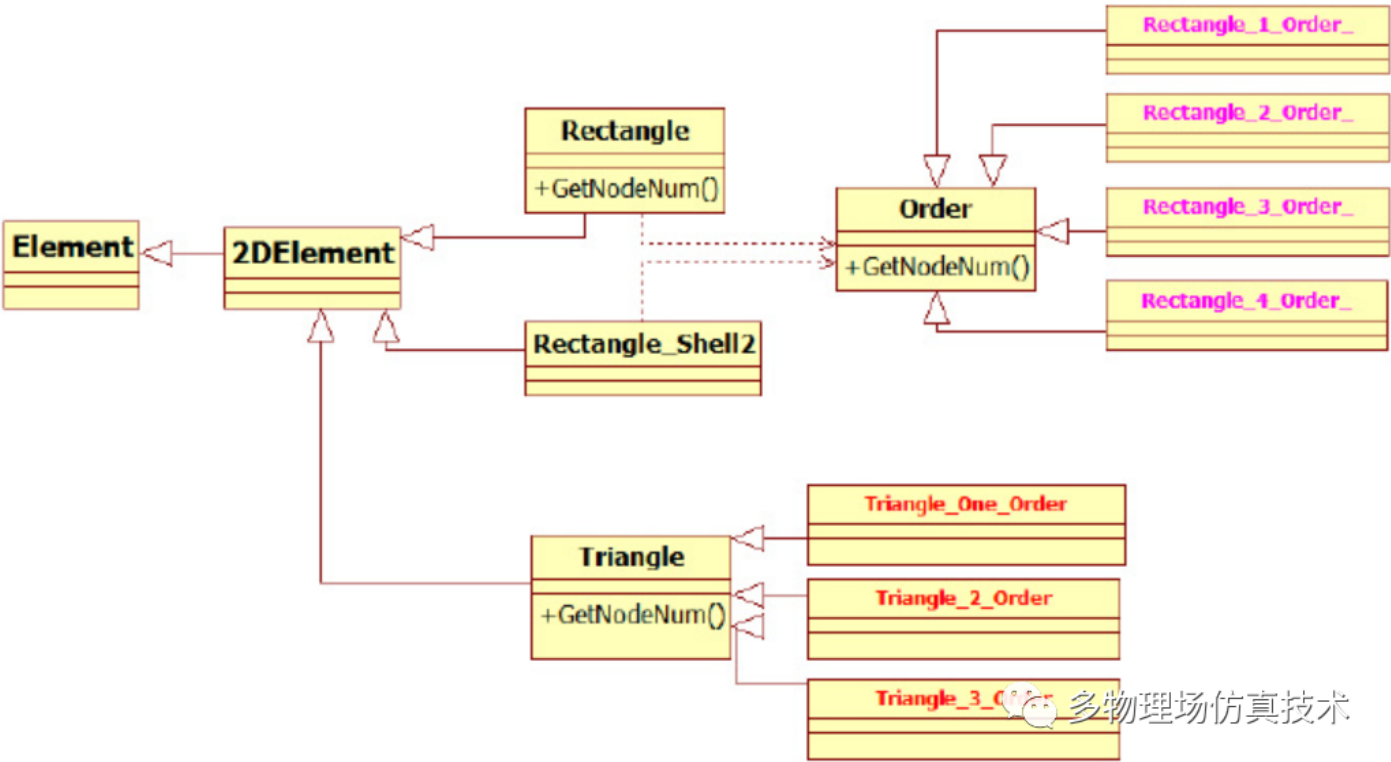
抽象化给出的具体实现，就是实现化。

3.脱耦：

所谓耦合，就是两个实体的行为的某种强关联。而将它们的强关联去掉，就是耦合的解脱，或称脱耦。在这里，脱耦是指将抽象化和实现化之间的耦合解脱开，或者说是将它们之间的强关联改换成弱关联。

将两个角色之间的继承关系改为聚合关系，就是将它们之间的强关联改换成为弱关联。因此，桥梁模式中的所谓脱耦，就是指在一个软件系统的抽象化和实现化之间使用组合/聚合关系而不是继承关系，从而使两者可以相对独立地变化。这就是桥梁模式的用意。

采用Bridge模式后，图2变成如下图3：



通过Bridge模式 实现了有限元单元和阶次的解耦，二者可以独立的变化，基于Bridge模式，可以在有限元单元中应用更多的变量，比如维度：0维到3维；高斯积分精度，从0到高阶；针对多物理场的单元：电磁，结构，热，声，流体，耦合单元等等，使得有限元单元设计更加灵活，通用性，扩展性更强。

现在网上关于设计模式资源很多，更多Bridge的具体实现可以百度