

一篇文章入门求解器开发设计模式（11/30/2020更新）

原创 www.cae-sim.com [多物理场仿真技术](#)



求解器是仿真软件的最核心模块，开发求解器是一项复杂繁琐的工作，包含了较多的数据结构，业务逻辑，专业算法和数值计算，合理的使用软件设计模式能优化求解器的架构设计，提升求解器的质量。

但现实是，由于求解器开发者大部分都是理工科出身，比如力学，流体等有较强的专业背景，而非计算机专业，因此在业务逻辑处理上较强，而在数据组织，软件架构上有所欠缺；而一般的程序员又缺乏仿真专业知识背景。

因此写了设计模式在求解器中应用系列，希望在专业业务开发和软件架构设计直接搭起一个桥梁。本文介绍了几种常见的软件设计模式在求解器开发中的应用，并在已有设计模式基础上，针对有限元数据结构特点，提出了一种属性模式。

设计模式的六大原则

总原则：开闭原则（Open Close Principle）

开闭原则就是说对扩展开放，对修改关闭。在程序需要进行拓展的时候，不能去修改原有的代码，而是要扩展原有代码，实现一个热插拔的效果。所以一句话概括就是：为了使程序的扩展性好，易于维护和升级。想要达到这样的效果，我们需要使用接口和抽象类等。

1、单一职责原则（Single）

不要存在多于一个导致类变更的原因，也就是说每个类应该实现单一的职责，如若不然，就应该把类拆分。

2、里氏替换原则（Liskov Substitution Principle）

里氏代换原则(Liskov Substitution Principle LSP)面向对象设计的基本原则之一。里氏代换原则中说，任何基类可以出现的地方，子类一定可以出现。LSP是继承复用的基石，只有当衍生类可以替换掉基类，软件单位的功能不受到影响时，基类才能真正被复用，而衍生类也能够在基类的基础上增加新的行为。里氏代换原则是对“开-闭”原则的补充。实现“开-闭”原则的关键步骤就是抽象化。而基类与子类的继承关系就是抽象化的具体实现，所以里氏代换原则是对实现抽象化的具体步骤的规范。

历史替换原则中，子类对父类的方法尽量不要重写和重载。因为父类代表了定义好的结构，通过这个规范的接口与外界交互，子类不应该随便破坏它。

3、依赖倒转原则（Dependence Inversion Principle）

这个是开闭原则的基础，具体内容：面向接口编程，依赖于抽象而不依赖于具体。写代码时用到具体类时，不与具体类交互，而与具体类的上层接口交互。

4、接口隔离原则（Interface Segregation Principle）

这个原则的意思是：每个接口中不存在子类用不到却必须实现的方法，如果不然，就要将接口拆分。使用多个隔离的接口，比使用单个接口（多个接口方法集合到一个的接口）要好。

5、迪米特法则（最少知道原则）（Demeter Principle）

就是说：一个类对自己依赖的类知道的越少越好。也就是说无论被依赖的类多么复杂，都应该将逻辑封装在方法的内部，通过public方法提供给外部。这样当被依赖的类变化时，才能最小的影响该类。

SRP	The Single Responsibility Principle	单一责任原则
OCP	The Open Closed Principle	开放封闭原则
LSP	The Liskov Substitution Principle	里氏替换原则
DIP	The Dependency Inversion Principle	依赖倒置原则
ISP	The Interface Segregation Principle	接口分离原则

以上也就经常说有的 **SOLID**原则。

最少知道原则的另一个表达方式是：只与直接的朋友通信。类之间只要有耦合关系，就叫朋友关系。耦合分为依赖、关联、聚合、组合等。我们称出现为成员变量、方法参数、方法返回值中的类为直接朋友。局部变量、临时变量则不是直接的朋友。我们要求陌生的类不要作为局部变量出现在类中。

6、合成复用原则（Composite Reuse Principle）

原则是尽量首先使用合成/聚合的方式，而不是使用继承。

[FEM之在求解器中使用设计模式\(1\)---Bridge模式](#)

[FEM之在求解器中使用设计模式\(2\)---Abstract Factory模式和Builder模式](#)

[FEM之在求解器中使用设计模式\(3\)---Composite模式](#)

[FEM之在求解器中使用设计模式\(4\)---Builder模式](#)

[FEM之在求解器中使用设计模式\(5\)---Adaptor模式](#)

[FEM之在求解器中使用设计模式\(6\)---Strategy模式](#)

[FEM之在求解器中使用设计模式\(7\)---属性模式](#)

[FEM之在求解器中使用设计模式\(8\)---Command和Observer模式](#)

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

[FEM之在求解器中使用设计模式\(10\)---Visitor访问者模式](#)