设计模式（Design pattern）是一套被反复使用、多数人知晓、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。

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。

开闭原则具有理想主义的色彩，它是面向对象设计的终极目标。其他几条，则可以看做是开闭原则的实现方法。设计模式就是实现了这些原则，从而达到了代码复用、增加可维护性的目的。

1. **开闭原则**
2. 概念

一个软件实体如类、模块和函数应该对扩展开放，对修改关闭。模块应尽量在不修改原（指原来的代码）代码的情况下进行扩展。

1. 模拟场景

在软件的生命周期内，因为变化、升级和维护等原因需要对软件原有代码进行修改时，可能会给旧代码中引入错误，也可能会使我们不得不对整个功能进行重构，并且需要原有代码经过重新测试。

1. Solution

当软件需要变化时，尽量通过扩展软件实体的行为来实现变化，而不是通过修改已有的代码来实现变化。

4、注意事项

* 通过接口或者抽象类约束扩展，对扩展进行边界限定，不允许出现在接口或抽象类中不存在的public方法
* 参数类型、引用对象尽量使用接口或者抽象类，而不是实现类
* 抽象层尽量保持稳定，一旦确定即不允许修改

5、开闭原则的优点