对于面向对象软件系统的设计而言，在支持可维护性的同时，提高系统的可复用性是一个至关重要的问题，如何同时提高一个软件系统的可维护性和可复用性是面向对象设计需要解决的核心问题之一。在面向对象设计中，可维护性的复用是以设计原则为基础的。每一个原则都蕴含一些面向对象设计的思想，可以从不同的角度提升一个软件结构的设计水平。

面向对象设计原则为支持可维护性复用而诞生，这些原则蕴含在很多设计模式中，它们是从许多设计方案中总结出的指导性原则。面向对象设计原则也是我们用于评价一个设计模式的使用效果的重要指标之一，在设计模式的学习中，大家经常会看到诸如“XXX模式符合XXX原则”、“XXX模式违反了XXX原则”这样的语句。

最常见的7种面向对象设计原则如下表所示：

表1 7种常用的面向对象设计原则

设计原则名称	**定 义**	**使用频率**
单一职责原则 (Single Responsibility Principle, SRP)	一个类只负责一个功能领域中的相应职责	★★★★☆
开闭原则 (Open-Closed Principle, OCP)	软件实体应对扩展开放，而对修改关闭	★★★★★
里氏代换原则 (Liskov Substitution Principle, LSP)	所有引用基类对象的地方能够透明地使用其子类的对象	★★★★★

依赖倒转原则
(Dependence Inversion Principle, DIP)
抽象不应该依赖于细节，细节应该依赖于抽象
★★★★★
接口隔离原则
(Interface Segregation Principle, ISP)
使用多个专门的接口，而不使用单一的总接口
★★☆☆☆
合成复用原则
(Composite Reuse Principle, CRP)
尽量使用对象组合，而不是继承来达到复用的目的

★★★★☆
迪米特法则
(Law of Demeter, LoD)
一个软件实体应当尽可能少地与其他实体发生相互作用
★★★☆☆