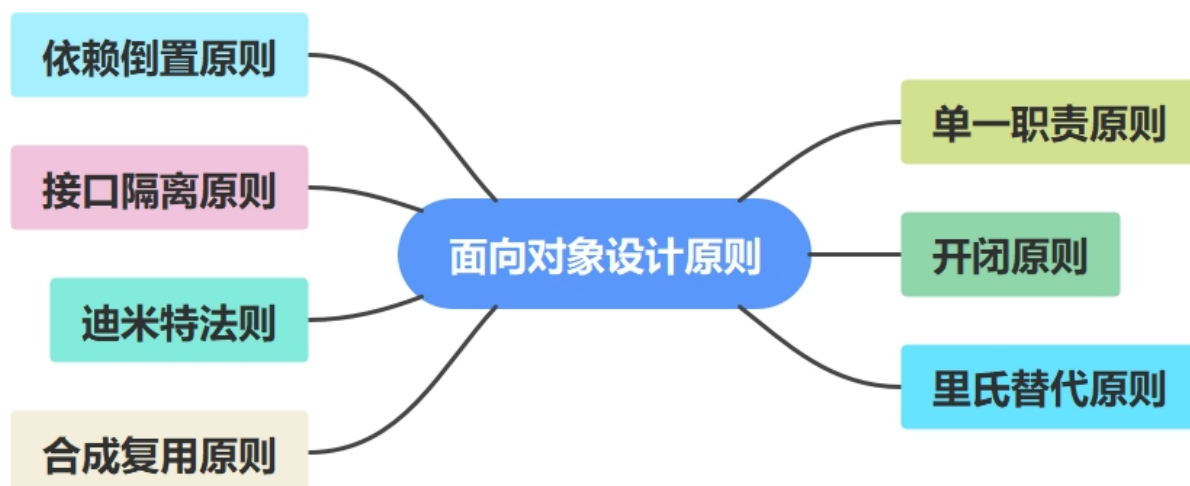


【第18话：中高级Java程序员经常被问到的面向对象设计原则】



1. 单一职责原则

单一职责原则强调的是 一个类只有一种职责。不会因为多种原因而去改变这个类。

如果可能发生多种原因而导致类发生变化，这其实就是高耦合。

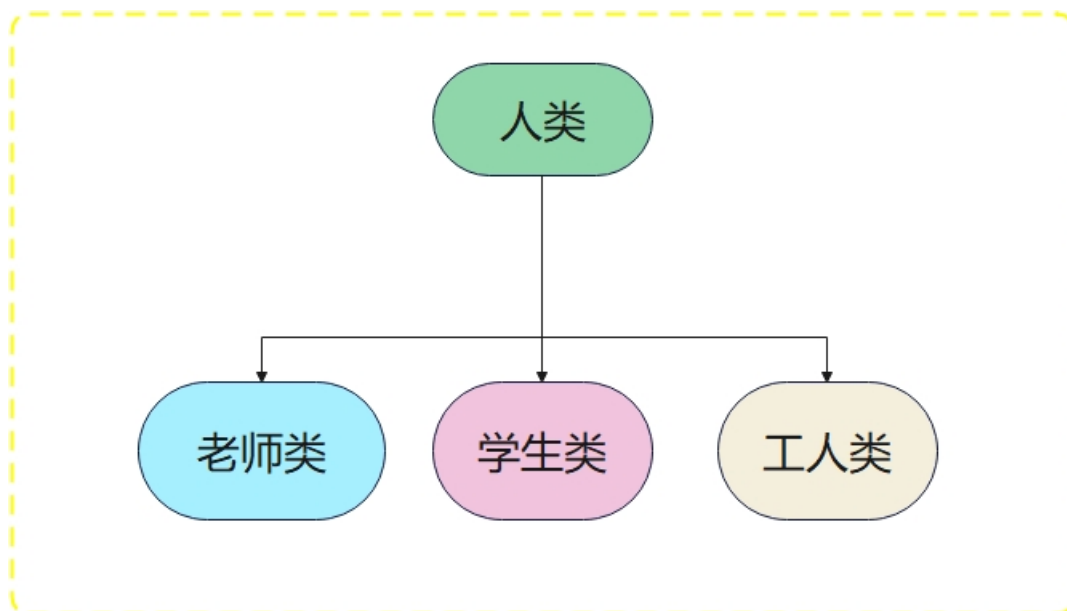
所以单一职责原则是“高内聚、低耦合”的一种指导设计思想。

举例说明：

现实生活中每个人都有自己的职业，包含：老师、学生、工人等。

如果项目设计时只设计一个人类，当老师、学生、工人特性发生变化时都会影响到人类。

所以根据单一职责原则应该设计为：老师类、学生类、工人类。



注意：单一职责原则强调的是—个类只有一种职责。但并不是说类设计时颗粒度越细越好。还是需要看项目的主体。

2.开闭原则

开闭原则强调的是：“软件在添加新功能时，应该在不修改原有代码的基础上，来添加新的功能”。

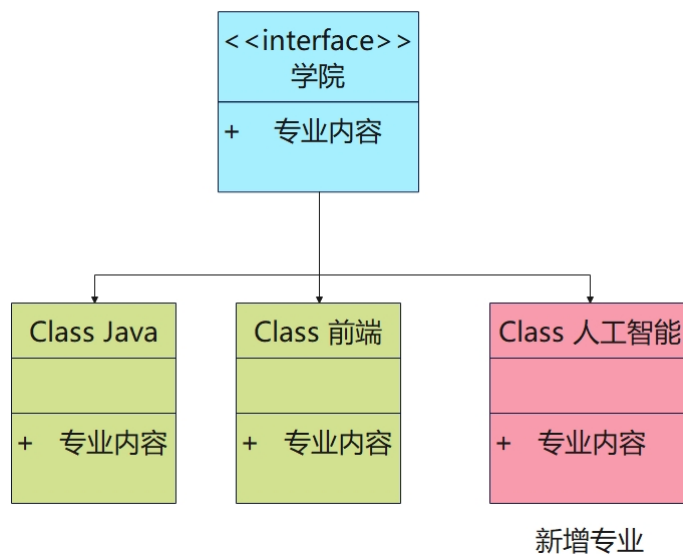
开闭原则是对软件可维护性的一种指导设计思想。

软件设计时想要复合开闭原则，应该尽量围绕抽象类或接口进行设计。

举例说明：

学校开设不同的专业，目前包含：Java学院、前端学院。后期又开设一个新的专业叫做人工智能。

应该设计一个接口，接口中包含专业内容。有实现类Java学院类、前端学院类。后期在添加一个人工智能学院



3.里氏替代原则

里氏替代原则强调的是：“所有基类能出现的地方，子类都可以替代其基类，也可以保证程序正常运行”。

里氏替代指导我们如何设计继承复用，强调不要去破坏继承体系。

放在程序中，应该是子类可以有自己的特有内容，同时可以重写父类的方法。

举个反例：

如果一个类不能重写父类的全部方法，这时应该断开这两个类的继承关系。而变成聚合、组合、依赖的关系。

4.依赖倒置原则

依赖倒置原则强调的是：“高层模块不应该依赖低层模块，两者都应该依赖其抽象；抽象不应该依赖细节；细节应该依赖抽象”。

依赖倒置原则就是我们常常说的面向接口编程。在程序设计时应该先有接口，后有实现类。当一个类想要依赖另一个类时，应该依赖另一个类的接口。

5.接口隔离原则

接口隔离原则强调的是：“不要建立庞大、臃肿的接口。而是让接口颗粒度尽量小，接口中的方法尽量少”。

在程序设计中应该给不同的部分提供不同的专门接口，而不要提供一个总接口。

举例说明：

这点我们在MVC分层开发有着明显的应用。

有Teacher类、Student类。在编写持久层时有TeacherMapper接口和StudentMapper接口。在编写业务层时，有TeacherService接口和StudentService接口。

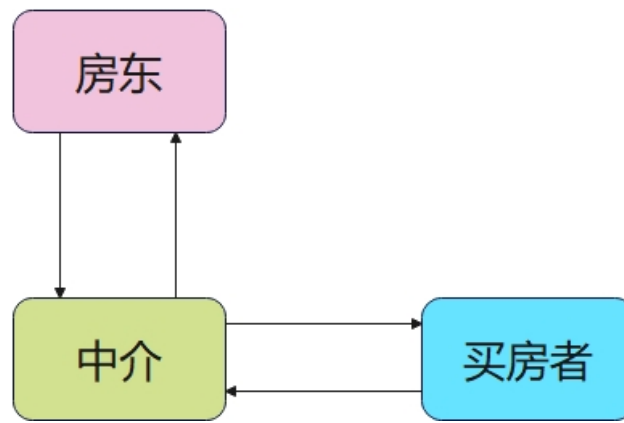
6.迪米特法则

迪米特法则强调的是：“尽量不要和陌生人说话”。

如果两个类没有直接关系，尽量不要给类之间添加聚合、组合、依赖的关系。这样可以降低类之间的耦合度。

举例说明：

生活中，房东的房子交给中介进行出售。中介和房东进行沟通。买房者和中介进行沟通房子，中介给买房者介绍房子。对于这样的关系就不应该在房东中添加买房者的依赖，也不应该在买房者里面添加房东的依赖。



7.合成复用原则

合成复用原则强调的是：“尽量少用继承，而使用组合、聚合、依赖来替代继承”。

因为继承是一种强关系，子类可以知道父类所有实现细节。而组合、聚合、依赖关系中只能知道被依赖类中想要被其他类知道的内容。

举例说明：

Teacher类需要用到Student类。如果让Teacher类继承Student类，可以用到Student类中的方法或属性。但是这样Teacher和Student就是一种强关系。不仅仅不满足合成复用原则，也不满足里氏替代原则。

面向对象设计原则回答示范

