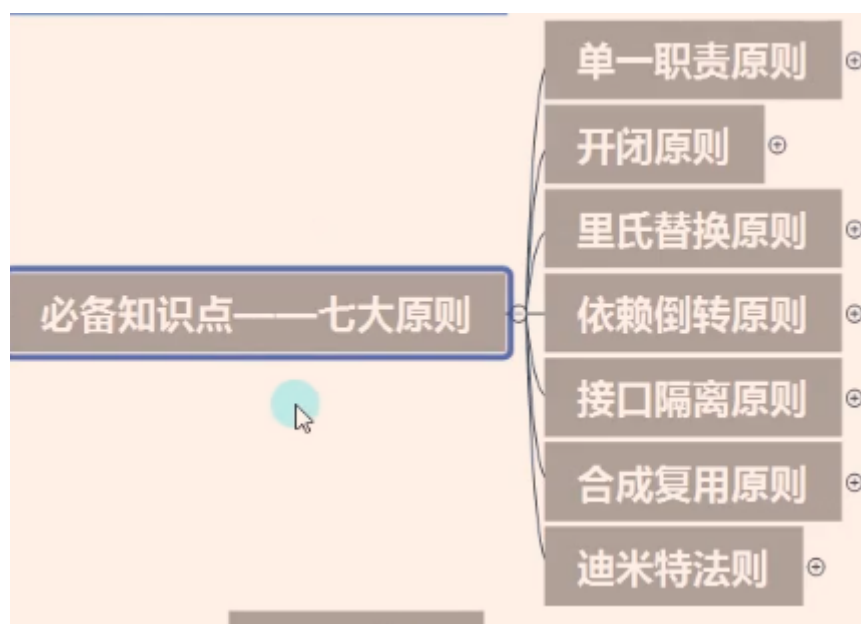


任务54：面向对象七大原则 知识点



学习目标

了解面向对象七大原则基本概念
在之后实践应用中
要基于七大原则去设计程序

为什么有七大原则

七大原则总体要实现的目标是：
高内聚、低耦合
使程序模块的可重用性、移植性增强

高内聚低耦合

从类角度来看，高内聚低耦合：
减少类内部，对其他类的调用

从功能块来看 高内聚低耦合：
减少模块之间的交互复杂度

七大原则

单一职责原则

接口隔离原则

开闭原则

合成复用原则

里氏替换原则

依赖倒转原则

迪米特法则

单一职责原则

SRP(Single Responsibility Principle)

类被修改的几率很大，因此应该专注于单一的功能。

如果把多个功能放在同一个类中，功能之间就形成了关联，改变其中一个功能，有可能中止另一个功能。

举例：假设程序、策划、美术三个工种是三个类，他们应该各司其职，在程序世界中只应该做自己应该做的事情。

开闭原则

OCP(Open-Closed Principle)

对拓展开发，对修改关闭

拓展开放：模块的行为可以被拓展从而满足新的需求

修改关闭：不允许修改模块的源代码（或者尽量使修改最小化）

举例：继承就是最典型的开闭原则的体现，可以通过添加新的子类 and 重写父类的方法来实现

里氏替换原则

LSP(Liskov Substitution Principle)

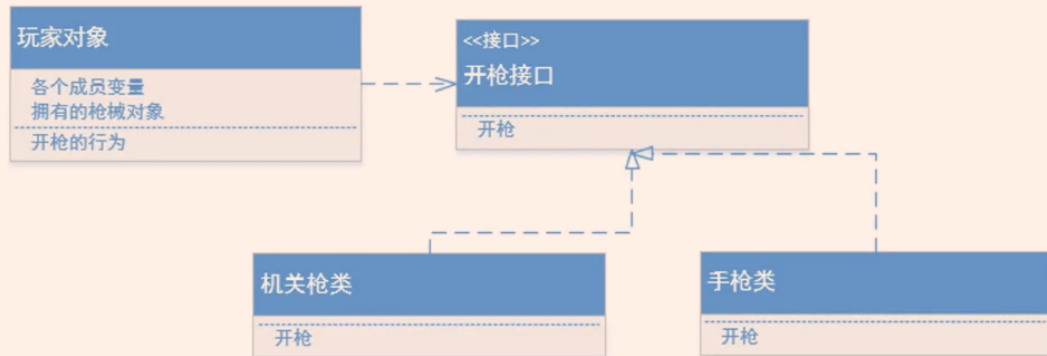
任何父类出现的地方，子类都可以替代

举例：用父类容器装载子类对象，因为子类对象包含了父类的所有内容

依赖倒转原则

DIP(Dependence Inversion Principle)

要依赖于抽象，不要依赖于具体的实现



迪米特原则

LoP(Law of Demeter)

又称最少知识原则

一个对象应当对其它对象尽可能少的了解

不要和陌生人说话

举例：一个对象中的成员，要尽可能少的直接和其它类建立关系
目的是降低耦合性

moon, www.taobao.com

接口分离原则

ISP(Interface Segregation Principle)

不应该强迫别人依赖他们不需要使用的方法

一个接口不需要提供太多的行为，一个接口应该尽量只提供一个对外的功能，让别人去选择需要实现什么样的行为，而不是把所有的行为都封装到一个接口当中

举例：飞行接口、走路接口、跑步接口等等虽然都是移动的行为
但是我们应该把他们分为一个一个单独的接口，让别人去选择使用

合成复用原则

CRP(Composite Reuse Principle)

尽量使用对象组合，而不是继承来达到复用的目的

继承关系是强耦合，组合关系是低耦合

举例：脸应该是眼镜、鼻子、嘴巴、耳朵的组合，而不是依次的继承
角色和装备也应该是组合，而不是继承

注意：不能盲目的使用合成复用原则，要在遵循迪米特原则的前提下

七大原则

单一职责原则：一个类只处理自己应该处理的内容，不应该啥都写在一起

开闭原则：对拓展开放，对修改封闭。新加功能尽量是加处理而不是改代码

里氏替换原则：任何地方子类都能替代父类，父类容器装子类

依赖倒转原则：不要依赖具体的实现，要依赖抽象（接口）

迪米特法则：一个类要尽量减少对别的类的了解，尽量少用别的类和自己关联

接口隔离原则：一个接口一个行为，不要一个接口n个行为

合成复用原则：除非设计上需要继承，否则尽量用组合复用的形式

WELCO

如何使用这些原则

在开始做项目之前，整理UML类图时

先按自己的想法把需要的类整理出来

再把七大原则截图放在旁边，基于七大原则去

优化整理自己的设计

整体目标就是：高内聚，低耦合

WELCO www.falr.com