



链滴

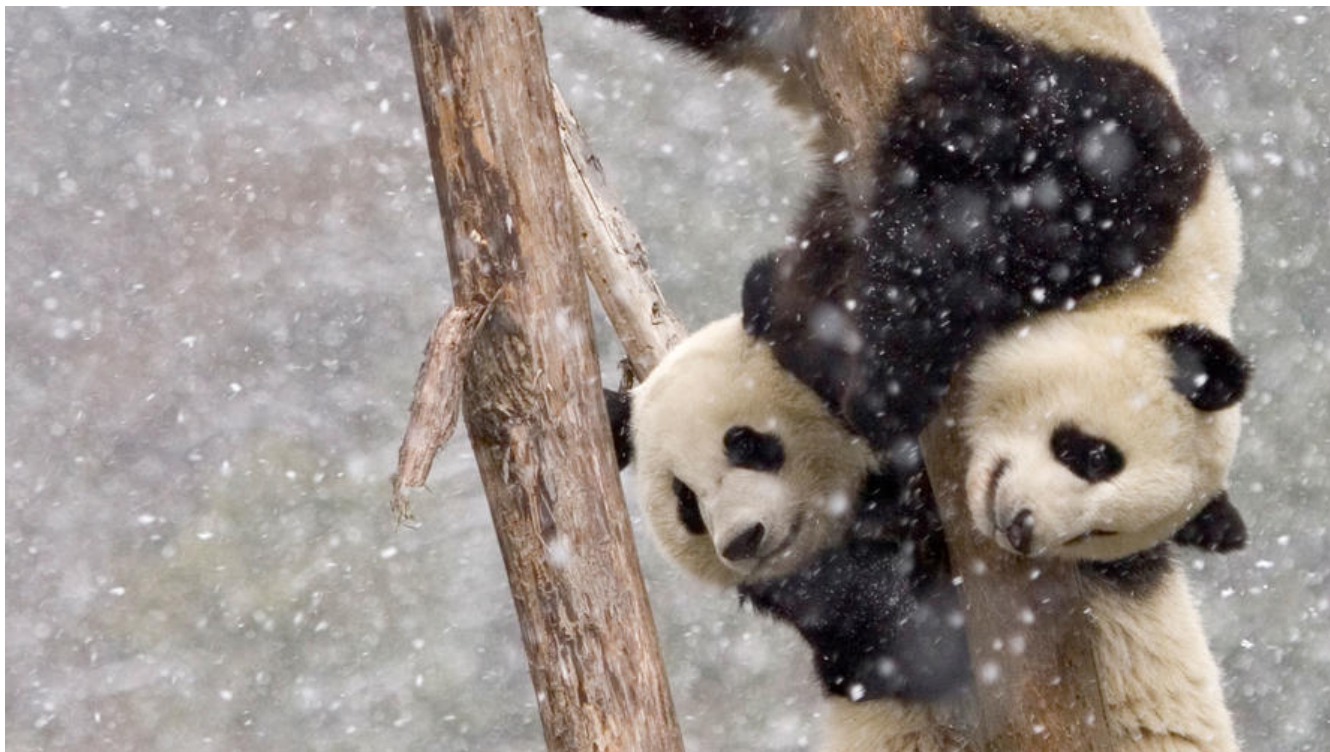
设计模式之七大原则

作者: [2457081614](#)

原文链接: <https://ld246.com/article/1582028553641>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)



使用设计模式的目的

- 代码重用性;
- 代码可读性;
- 可扩展性;
- 可靠性;
- 高内聚、低耦合

七大原则

- **单一职责原则**。对类来说一个类只负责一个职责，如类 A 有两个不同职责：A1 和 A2。当职责 1 求变更而改变 A，可能会导致 A2 报错，所以我们需要将类 A 力粒度拆分为 A1 和 A2，分别负责不同的职责。
- **接口隔离原则**。即一个类对另一个类的依赖应建立在最小的接口之上。
- **依赖倒转原则**。第一，高层模块不应该依赖于底层模块，都应该依赖其抽象。第二，抽象不应该依赖于细节，细节应该依赖抽象，中心思想是面向接口编程。相对于细节的多变性，抽象的东西要稳定的，在 Java 中，抽象指的是接口或抽象类，细节就是具体的实现类。
- **里氏替换原则**。在子类中尽量不要重写父类的方法，在适当的情况下，可以通过聚合，组合，依赖解决问题。
- **开闭原则**。一个类在扩展方面应该是开放的，在修改方面是关闭的。
- **迪米特原则**。迪米特原则又叫最少知道原则，即一个类对自己依赖的类知道的越少越好。也就是说对于被依赖的类不管多么复杂，都尽量将逻辑封装在类的内部。对外除了提供的 public 方法，不对泄露任何信息。
- **合成复用原则**。尽量使用聚合的方式，而不是使用继承。