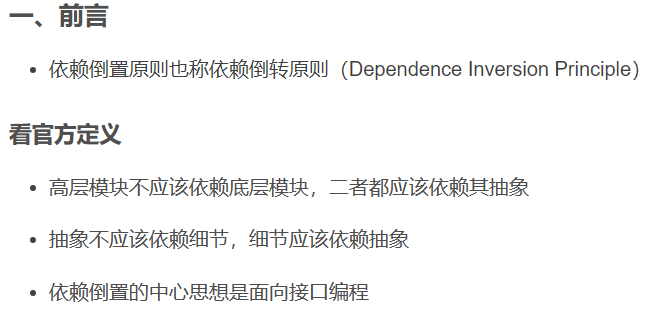
# 面向对象设计原则之五大原则

* 单一职责原则（The Single Responsibility Principle，简称SRP）
* 开放－封闭原则（The Open-Close Principle，简称OCP）
* Liskov替换原则（The Liskov Substitution，简称LSP）

里氏代换原则（LSP）：子类型必须能够替换掉它们的父类型。[ASD]

* 依赖倒置原则（The Dependency Inversion Principle，简称DIP）
* 接口隔离原则（The Interface Segregation Principle，简称ISP）

## 依赖倒置原则



### 基础知识

**1.什么是依赖？**

就是对不同模块、对象或个体之间的相互作用或相互关联的描述。

实现类与抽象之间所表达的关系。

**2.依赖关系有哪些(UML的六种依赖)？**

继承、接口实现、组合(初始化依赖),聚合(字段数组依赖), 关联(字段依赖), 依赖

(构造参数依赖)。

实现类与抽象之间所表达的关系。

**3.抽象和细节具体指代什么？**

抽象：接口、抽象类。

细节：具体实现类

**4.解决什么问题？**

通过抽象解决强依赖问题。

### 原则解释

**1.高层模块不应该依赖低层模块，二者都应该依赖其抽象**。

含义：模块与模块之间不能直接依赖，需要通过抽象进行依赖。

有点：降低模块之间的耦合度，提高系统的灵活性和可维护性

**2.抽象不应该依赖细节，细节应该依赖抽象。**

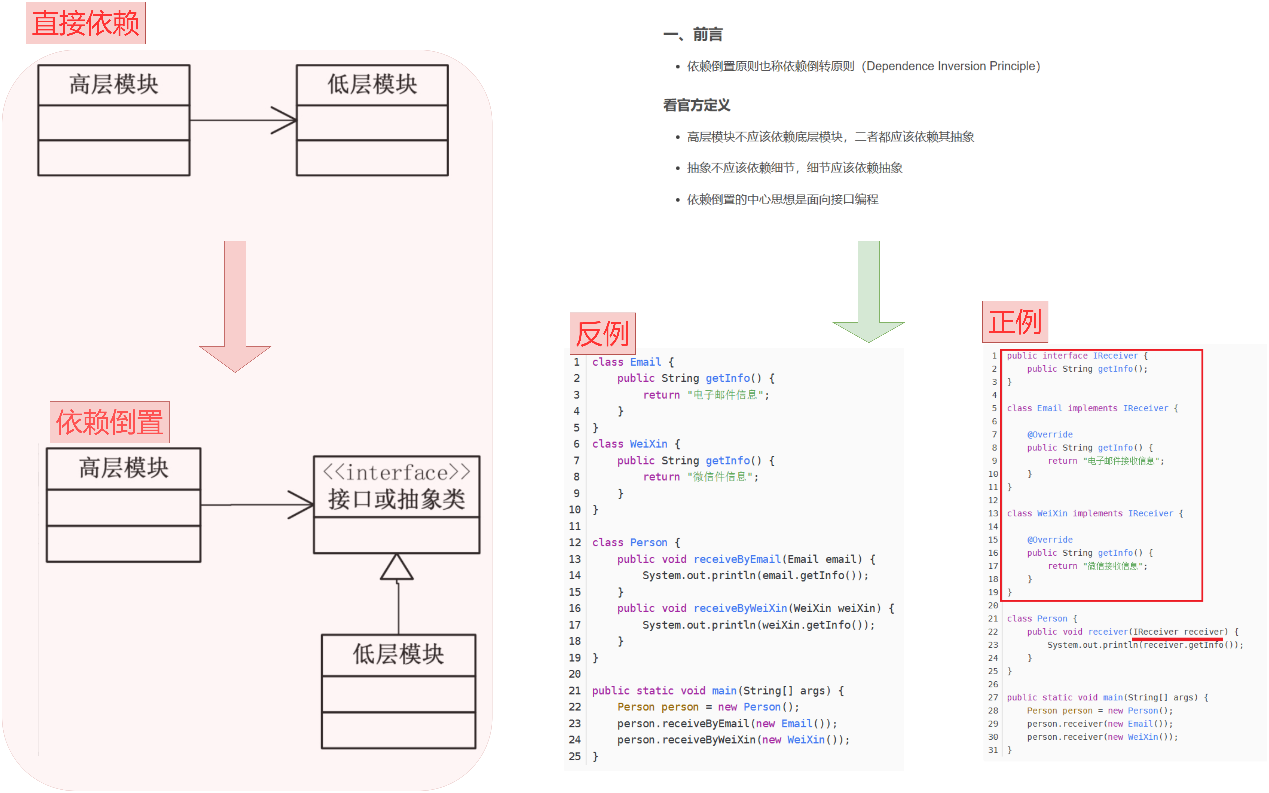
含义：***抽象类的定义与具体的实现细节无关(不能产生依赖关系)，是对事物规律***

***的一个抽象（就是定义一个抽象类或接口）。***

优点：提供一组稳定的接口，及模块间解耦.

**3.依赖倒转的中心思想是面向接口编程。**

### 实现方法



### 名字由来

描述了依赖关系的调整。

