

1 单一职责原则 (SRP)

Single Responsibility Principle

OO 设计的 SOLID 原则: 为实现易维护、可扩展的软件系统，应遵循一定的设计原则。

缩写	全称	翻译
SRP	The Single Responsibility Principle	单一职责/责任原则
OCP	The Open Closed Principle	开放封闭原则
LSP	The Liskov Substitution Principle	里氏替换原则
ISP	The Interface Segregation Principle	接口分离原则
DIP	The Dependency Inversion Principle	依赖倒转原则

应只有一类原因可以引发对类的修改 (there should never be more than one reason for a class to change)。换句话说，就是让一个类只承担一种类型责任，当这个类需要承担其他类型的责任的时候，就需要分解这个类。

目的:

- 为了实现模块化，即高内聚、低耦合。
- 是任何程序设计方法都遵循的原则，如面向过程方法中的函数设计。一个函数只实现一种功能; 一个函数的代码行数不宜过长 (30 行?/一屏?)。

遵循 SRP 带来的好处: 可以降低类的复杂度，提高类的可读性; 控制变更影响的范围; 综合以上两点，SRP 可提高系统的可维护性; SRP 是所有原则中的最简单的，同时也是最难正确运用的。对象识别与职责划分，是 OO 设计永恒的主题。

```
interface Room
{
    void service();
};
class Kitchen implements Room
{
    void service()
    { //实现做饭的功能; }
};
class Bedroom implements Room
{
    void service()
    { //实现休息的功能; }
};
```