

SRP - 单一职责原则

- » 原则: 一个类只有一个改变的理由
- » 职责 = “改变的理由”
- » 如果一个类有多个职责, 那么职责就会彼此耦合。
- » 如果模块具有多个职责, 内聚性就低。
- » 良好设计的内聚性应该很高

内聚性

- » 内聚性
 - 衡量一个模块中的源代码行如何协同工作以提供特定功能的指标。面向对象编程中, 内聚性指方法实现单一功能的程度。实现单一功能的方法具有高内聚性。

内聚性

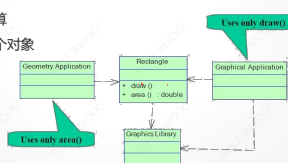
- » 低内聚性的表现:
 - 一个类的方法很少相关。
 - 方法执行不相关的活动, 通常使用不相关的数据集。

内聚性

- » 低内聚性的缺点
 - 模块理解困难。
 - 系统维护困难。
 - 模块重用困难。

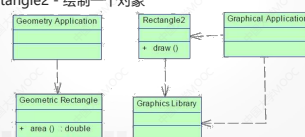
SRP 例子 I

- » 矩形 - 有两个职责的类
 - 几何计算
 - 绘制一个对象



SRP 例子 I: 更好的设计

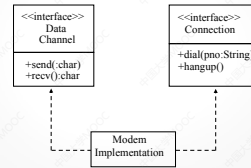
- » 将职责分离, 创建2个类
 - 几何矩形 - 几何计算
 - Rectangle2 - 绘制一个对象



SRP 例子 II

```
interface Modem
{
    public void dial(String
pno);
    public void hangup();
    public void send(char c);
    public char recv();
}
```

SRP 例子 II



SRP - 总结

- » 注意：如果职责不会独立发生变化，职责无需分离。否则，会造成“不必要的复杂性”。

内聚性

- » 高内聚性的原则可以应用于不同层面。目前的例子主要侧重类的内聚性。
- » 内聚性也可以是方法、包和子系统层面的。具体可参考：<http://www.javaworld.com/jw-05-1998/jw-05-techniques.html>

练习

- » 举一个违反SRP原则的例子并解释原因。
- » 如何修改以符合SRP?