“单一职责原理”感想

在用面向对象的思想编程的时候，把一个类比喻为一个事物，事物具备的功能则是这个类的操作，书既可以用来读，也可以用来垫桌脚，在寒冷的时候还可以点燃取暖，但是在单一职责原理中，书的不同功能就是引起类变化的不同原因，可以分为三类，如果混合在一起，修改其中一个职责的时候，有可能会对另一个职责产生影响。当一个类只用一个职责的时候，另一个不会使用的职责就浪费了资源。

单一职责的优点：降低类的复杂度，一个类只负责一个原则；提高类的可读性，提高系统的可维护性；降低类的职责变更带来的风险。在面向对象的程序开发中，高内聚、低耦合是软件设计追求的目标，单一职责则可以看做是高内聚、低耦合的引申，将职责定义为引起变化的原因，以提高内聚性，用来减少引起变化的原因。职责愈多，可能引起变化的原因就会变多，相互之间的影响，会极大的损害内聚性和耦合度，类不要过多的功能可以用来保证实体变化的原因点只有一个。在划分类的职责的时候又涉及到类的粒度，当职责的粒度变小，更加细化的时候，就会与单一职责原理发生冲突，那么只有逻辑足够简单，才可以在代码级别上违反单一职责原则；只有类中方法数量足够少，才可以在方法级别上违反单一职责原则。