单一职责原则(Single Responsibility Principle, SRP)：一个类只负责一个功能领域中的相应职责，或者可以定义为：就一个类而言，应该只有一个引起它变化的原因。

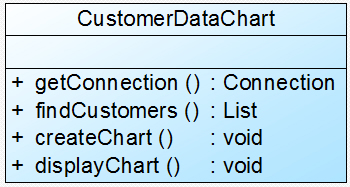
产生背景：

一个类不能太“累”！在软件系统中，一个类（大到模块，小到方法）承担的职责越多，它被复用的可能性就越小，而且一个类承担的职责过多，就相当于将这些职责耦合在一起，当其中一个职责变化时，可能会影响其他职责的运作，因此要将这些职责进行分离，将不同的职责封装在不同的类中，即将不同的变化原因封装在不同的类中，如果多个职责总是同时发生改变则可将它们封装在同一类中。

单一职责原则是实现高内聚、低耦合的指导方针。

实例：Sunny软件公司开发人员针对某CRM系统中客户信息图形统计模块。

一个类的结构如下：



（连接数据库：getConnection()，

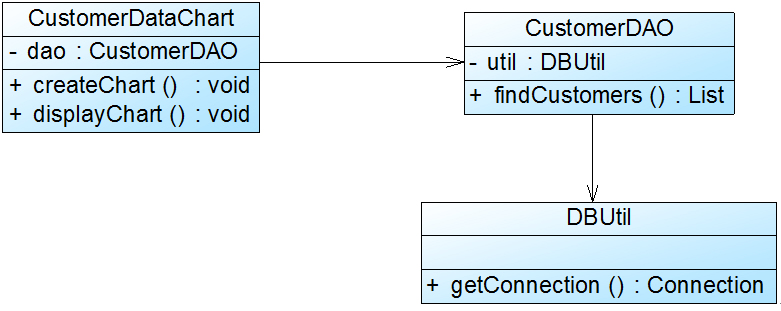
查询所有客户：findCustomers()，

创建图表：createChart()，

显示图表：displayChart()）

可以看出该类承担了太多的职责，不仅代码重用难以实现，而且无论是修改数据库连接方式还是修改图表显示方式都需要修改该类，它不止一个引起它变化的原因，违背了单一职责原则。

根据单一职责原则对该类进行重构：



(1) DBUtil：负责连接数据库，包含数据库连接方法getConnection()；

(2) CustomerDAO：负责操作数据库中的Customer表，包含对Customer表的增删改查等方法，如findCustomers()；

(3) CustomerDataChart：负责图表的生成和显示，包含方法createChart()和displayChart()。

[面向对象设计原则之单一职责原则](http://blog.csdn.net/lovelion/article/details/7536542" \t "https://blog.csdn.net/LoveLion/article/details/_blank)