



第8章 设计原则

8.7 单一职责原则

刘其成 计算机与控制工程学院 ytliuqc@163.com 2018-09

- ■问题由来
 - -类T负责两个不同的职责:职责P1,职责P2。
 - 当由于职责P1需求发生改变而需要修改类T时, 有可能会导致原本运行正常的职责P2功能发生故 障。

- ■解决方案
 - 遵循单一职责原则 (Single Responsibility Principle, SRP)
 - -分别建立两个类T1、T2,使T1完成职责P1功能, T2完成职责P2功能。
 - -这样,当修改类T1时,不会使职责P2发生故障风险;同理,当修改T2时,也不会使职责P1发生故障风险。

- 类的职责主要包括两个方面
 - -数据职责
 - -行为职责
- 数据职责通过其属性来体现,而行为职责通过其方法来体现。

软件设计与体系结构





8.7.1 概念

■定义

- —一个对象应该只包含单一的职责,并且该职责被完整地封装在一个类中——功能要单一。
- -就一个类而言,应该仅有一个引起它变化的原因。

- 编程时有时会给一个类加各种各样的功能。
 - -比如写一个窗体应用程序,会把各种各样的代码 (算法,数据库访问的SQL语句)都写到窗体类 中。
 - -这样,无论任何需求,你都要更改这个窗体类,可维护性和可复用性都不好。

- 如果一个类承担的职责过多,就等于把这些职责 耦合在一起,一个职责的变化可能会削弱或者抑 制这个类完成其他职责的能力。
- 当变化发生时,设计会遭到破坏。
- 所以,要发现职责并把职责相互分离。

对于前面的计算器的例子中,运用单一职责原则,把计算和显示分开,也就是让业务逻辑与界面逻辑分开,这样它们之间的耦合度就下降了,就容易维护或扩展了。

- 单一职责原则是实现高内聚、低耦合的指导方针
- 在很多代码重构手法中都能找到它的存在,它是最简单但又最难运用的原则,需要设计人员发现类的不同职责并将其分离
- 而发现类的多重职责需要设计人员具有较强的分析设计能力和相关重构经验。

软件设计与体系结构



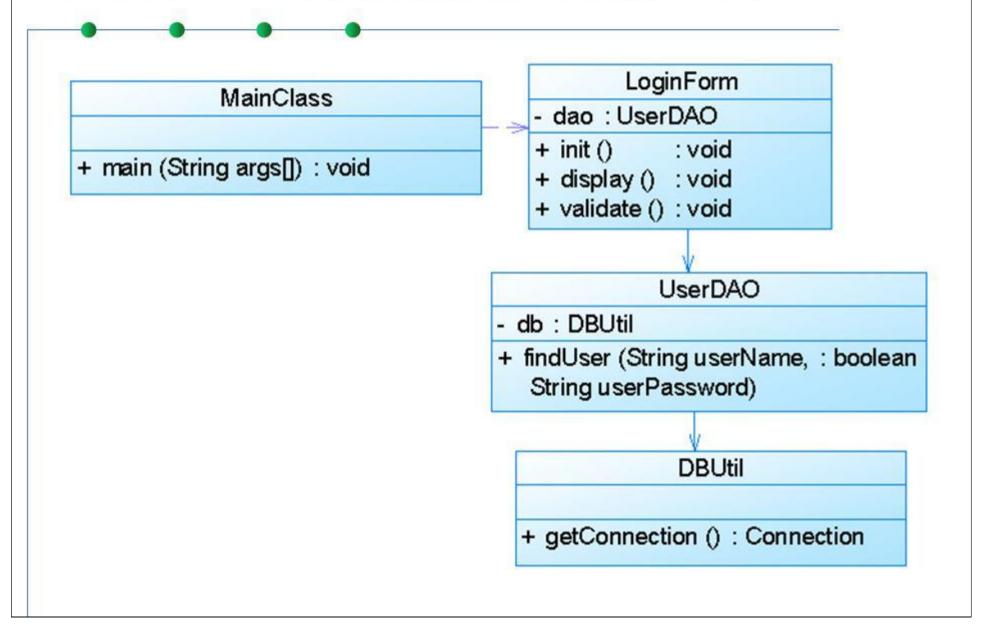


8.7.2 实例

某基于Java的C/S系统的"登录功能"通过如下登录类 (Login)实现:

Login + init () : void + display () : void + validate () : void + getConnection () : Connection findUser (String userName, : boolean String userPassword) + main (String args[]) : void

现使用单一职责原则对其进行重构







谢谢

2018年11月6日