

写Java程序最容易犯的21种错误实例分析[1] PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/206/2021_2022__E5_86_99Java_E7_A8_8B_c104_206931.htm

1.Duplicated Code 代码重复几乎是最常见的异味了。他也是Refactoring的主要目标之一。代码重复往往来自于copy-and-paste的编程风格。与他相对应OAOO是一个好系统的重要标志。

2.Long method 它是传统结构化的“遗毒”。一个方法应当具有自我独立的意图，不要把几个意图放在一起。

3.Large Class 大类就是你把太多的责任交给了一个类。这里的规则是One Class One Responsibility.

4.Divergent Change 一个类里面的内容变化率不同。某些状态一个小时变一次，某些则几个月一年才变一次；某些状态因为这方面的原因发生变化，而另一些则因为其他方面的原因变一次。面向对象的抽象就是把相对不变的和相对变化相隔离。把问题变化的一方面和另一方面相隔离。这使得这些相对不变的可以重用。问题变化的每个方面都可以单独重用。这种相异变化的共存使得重用非常困难。

5.Shotgun Surgery 这正好和上面相反。对系统一个地方的改变涉及到其他许多地方的相关改变。这些变化率和变化内容相似的状态和行为通常应当放在同一个类中。

6.Feature Envy 对象的目的是封装状态以及与这些状态紧密相关的行为。如果一个类的方法频繁用get方法存取其他类的状态进行计算，那么你要考虑把行为移到涉及状态数目最多的那个类。

7.Data Clumps 某些数据通常像孩子一样成群玩耍：一起出现在很多类的成员变量中，一起出现在许多方法的参数中，这些数据或许应该自己独立形成对象。

8.Primitive Obsession 面向对象的新手通常习惯

使用几个原始类型的数据来表示一个概念。譬如对于范围，他们会使用两个数字。对于Money，他们会用一个浮点数来表示。因为你没有使用对象来表达问题中存在的概念，这使得代码变的难以理解，解决问题的难度大大增加。好的习惯是扩充语言所能提供原始类型，用小对象来表示范围、金额、转化率、邮政编码等等。

9. Switch Statement 基于常量的开关语句是OO的大敌，你应当把他变为子类、state或strategy.

10. Parallel Inheritance Hierarchies 并行的继承层次是shotgun surgery的特殊情况。因为当你改变一个层次中的某一个类时，你必须同时改变另外一个层次的并行子类。

11. Lazy Class 一个干活不多的类。类的维护需要额外的开销，如果一个类承担了太少的责任，应当消除它。

12. Speculative Generality 一个类实现了从未用到的功能和通用性。通常这样的类或方法唯一的用户是testcase.不要犹豫，删除它。

100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com