

写Java程序最容易犯的21种错误例析 PDF转换可能丢失图片或格式，建议阅读原文

[https://www.100test.com/kao\\_ti2020/138/2021\\_2022\\_\\_E5\\_86\\_99Java\\_E7\\_A8\\_8B\\_c97\\_138426.htm](https://www.100test.com/kao_ti2020/138/2021_2022__E5_86_99Java_E7_A8_8B_c97_138426.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。

13. Temporary Field 一个对象的属性可能只在某些情况下才有意义。这样的代码将难以理解。专门建立一个对象来持有这样的孤儿属性，把只和他相关的行为移到该类。最常见的是一个特定的算法需要某些只有该算法才有用的变量。

14. Message Chain 消息链发生于当一个客户向一个对象要求另一个对象，然后客户又向这另一对象要求另一个对象，再向这另一个对象要求另一个对象，如此如此。这时，你需要隐藏分派。

15. Middle Man 对象的基本特性之一就是封装，而你经常会通过分派去实现封装。但是这一步不能走得太远，如果你发现一个类接口的一大半方法都在做分派，你可能需要移去这个中间人。

16. Inappropriate Intimacy 某些类相互之间太亲密，它们花费了太多的时间去钻研别人的私有部分。对人类而言，我们也许不

应该太假正经，但我们应当让自己的类严格遵守禁欲主义。

17. Alternative Classes with Different Interfaces 做相同事情的方法有不同的函数signature，一致把它们往类层次上移，直至协议一致。

18. Incomplete Library Class 要建立一个好的类库非常困难。我们大量的程序工作都基于类库实现。然而，如此广泛而又相异的目标对库构建者提出了苛刻的要求。库构建者也不是万能的。有时候我们会发现库类无法实现我们需要的功能。而直接对库类的修改有非常困难。这时候就需要用各种手段进行Refactoring.

19. Data Class 对象包括状态和行为。如果一个类只有状态没有行为，那么肯定有什么地方出问题了。

20. Refused Bequest 超类传下来很多行为和状态，而子类只是用了其中的很小一部分。这通常意味着你的类层次有问题。

21. Comments 经常觉得要写很多注释表示你的代码难以理解。如果这种感觉太多，表示你需要Refactoring.

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](http://www.100test.com)