

Java入门：代码的坏味道 PDF转换可能丢失图片或格式，建议
阅读原文

https://www.100test.com/kao_ti2020/145/2021_2022_Java_E5_85_A5_E9_97_A8_c104_145555.htm

- 1、 Duplicated Code (重复代码) 如果在一个以上的地点看到相同的程序结构，就应当设法将它们合而为一。
- 2、 Long Method (过长函数) 过长的程序不利于理解。每当感觉需要以注释来说明时，便可将此处代码写入一个独立函数中，并以其用途命名。
- 3、 Large Class (过大类) 如果一个class中出现过多的instance变量，就应当将几个相关的变量提炼到某个组件内。
- 4、 Long Parameter List (过长参数列) 太长的参数列难以理解也容易造成前后不一致、不易使用。
- 5、 Divergent Change (发散式变化) 一个class受到多种变化的影响。针对某一外界变化的所有相应修改，都只应该发生在单一class中，而这个新class内的所有内容都应该反应外界变化。
- 6、 Shotgun Surgury (散弹式修改) 一种变化引发多个classes相应修改。此时应把所有需要修改的代码放进同一个class中。
- 7、 Feature Envy (依恋情结) 对象技术的全部要点在于：将数据和加诸其上的操作行为包装在一起。将总是一起变化的东西放在一起。
- 8、 Data Clumps (数据泥团) 将总是一起出现的数据放进属于它们自己的对象中
- 9、 Primitive Obsession (基本型别偏执)
- 10、 Switch Statements (Switch 惊悚现身) 面向对象程序的一个明显特征就是：少用switch(或case)语句。大多数时候，一看到switch语句就应当考虑以多态来代替它。
- 11、 Parallel Inheritance Hierarchies (平行继承体系)
- 12、 Lazy Class (冗赘类) 如果函数或class的唯一用户是 test cases，就应当将其连同test cases一起删掉。

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