# 代码的坏味道

|  |  |
| --- | --- |
| **坏味道** | **常用重构** |
| Alternative Classes with Different Interfaces  (异曲同工类) | Rename Method ,  Move Method |
| Comments  (过多注释类) | Extract Method ,  Introduce Assertion |
| Data Class  (纯稚的数据类) | Move Method ,  Encapsulate Field ,  Encapsulate Collection |
| Data Clumps  (数据泥团) | Extract Class ,  Introduce Parameter ,  Preserve Whole Object |
| Drivergent Change  (发散式变化) | Extract Class |
| Duplicated Code  (重复代码) | Extract Method ,  Extract Class ,  Pull Up Method,  From Template Method |
| Feature Envy  (依恋情节) | Move Method ,  Move Field ,  Change Bidirectional ,  Association to Unidirectional ,  Replace Inheritance with Delegation ,  Hide Delegate |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

备注：

Alternative n. 二中择一；供替代的选择

adj. 供选择的；选择性的；交替的

Assertion n. 断言，声明；主张，要求；坚持

Encapsulate n. 压缩

Preserve n. 保护区；禁猎地；加工成的食品

vt. 保存；保护；维持；腌；禁猎

Whole n. 整体；全部

adj. 完整的；纯粹的

Bidirectional adj. 双向的；双向作用的

3.1 重复代码（Duplicated Code）

坏味道行列中首当其冲的就是Duplicated Code，如果你在一个以上的地点看到相同的程序结构，那么肯定：设法将他们合而为一，程序会变得更好。

最单纯的Duplicated Code就是“同一个类的两个函数含有相同的表达式”。这时候你需要做的就是采用Extract Method 提炼出重复的代码，然后让这两个地点都调用被提炼出来的那一段代码。

另一种常见的情况就是“两个互为兄弟的子类内含有相同表达式”。避免这种情况，只需要对两个类都使用 Extract Method ， 然后再对被提炼出来的代码使用 Pull Up Method ，将它推入超类内。如果代码之间只是类似，并非完全相同，那么就运用Extract Method 将相似的部分和差异部分割开，构成单独的一个函数。然后你可能发现运用 From Template Method 获得一个 Template Method 设计模式。如果有些函数以不同的算法做相同的事情，你可以选择其中较清晰的一个，并使用 Substitute Algorithm 将其他函数的算法替换掉。

如果两个毫不相关的类出现 Duplicated Code,你应该考虑对其中一个使用 Extract Class，将重复代码提炼到一个独立的类中，然后在另一个类内使用这个新类。但是，重复代码所在的函数也可能的确只应该属于某个类，另一个类只能调用它，抑或这个函数可能属于第三个类，而另两个类应该引用这第三个类。你必须决定这个函数放在哪儿合适，并确保它被安置后就不会再在其他任何地方出现。