



讨论：好的代码

袁剑

■ 什么是好的代码？

- 可读性
 - 唯一不变的就是变化（需求、接口、人员……）
 - 理解和后续更新、优化
- 高性能
 - 能省则省，0.1ms也很重要
 - 权衡，99%的努力 Vs 1%的性能
- 优雅？！
 - 语言风格一致，特定的语言有特定的规范（C#，JS，C/C++）
 - 把代码当文章（文如其人），精雕细琢

■ 变量命名

- 有意义的名称

■ 继承

- 组合优于继承
- 不要用继承来实现方法重用

■ 设计模式

- 不要刻意滥用
- 熟悉各个模式的应用场景

- 重复的代码 (Duplicated Code)
 - 抽取公共代码
- 过长的方法 (Long Method)
 - 按功能抽取成小方法
- 过大的类 (Large Class)
 - 按功能抽取成小类
 - 子类/模板模式/适配器模式
- 过多的注释 (Comments)
 - “为什么这样”比“如何这样”更重要
 - 代码自注释

- 每一个方法应该有一个明确、清晰的目的，好的程序应该是由一系列目的明确的、不重复的方法组成。
- 尽量使用语言的简单功能，不要使用不那些自己还不肯定的复杂功能，简单才是美
- 尽量避免复制和粘贴，不断优化并超越自己是程序员的追求。
- 避免绝对化，保持怀疑的态度，数据说话。
- 保持阅读量，定期总结并应用到工作中。
- 增强责任心，你是代码的Owner，你得对它负责到底

■ 推荐：重构-改善既有代码的设计

<http://book.douban.com/subject/1229923/>