

Clean Code Model

I 基础级

- 格式
 - 横向格式
 - 纵向格式
- 注释
 - 好的注释
 - 不好的注释
- 物理设计
 - 编译自满足
 - 接口文件设计职责单一
 - 仅包含需要的文件
 - 仅公开用户需要的接口

II 进阶级

- 命名
 - 关注点
 - 文件夹 | 包
 - 文件
 - 函数 | 类方法 | 类
 - 参数 | 变量
 - 风格统一的命名规范
 - 避免在命名中使用编码
 - 名称贴合业务语义
 - 名称贴近问题域而非解决域
- 测试
 - 风格统一的测试场景描述
 - 每个测试用例测试一个场景
 - 一组测试场景封装为一个测试套
 - 尝试使用DSL表达测试场景

- 符合F(ast).I(ndependent).R(epeatable).S(elf-validating).T(imely)原则

• 对象和数据结构

- 区分数据结构与对象的使用场景
- 避免给数据结构添加getter & setter
- 避免在对象中暴露成员变量

III 高级

• 函数

- 每个函数只做一件事
- 函数内语句同一抽象层次
- 尽量避免三个以上的函数参数
- 区分查询函数与指令函数
- 消除重复的函数

• 类

- 设计职责单一的类
- 设计依赖于抽象的类
- 设计对扩展开放，对修改关闭的类
- 避免方法过多的类（上帝类）

• 系统

- 合理的对系统进行分层
- 定义清晰的模块边界及职责
- 分离构造与使用
- 考虑系统性能