

## 5.1 软件编程

---

- 良好的编程实践

软件编程是一个复杂而迭代的过程，其除了编写代码之外，还包括**代码审查**、**单元测试**、**代码优化**、**集成调试**等一些列工作

软件编程使用程序设计语言将软件设计描述（SDD）转化为程序代码。

软件编码包括程序设计和程序实现

- 软件编码规范

与特定语言相关的描写如何编写代码的规则集合

- 提高编码质量，避免不必要的程序错误
    - 增强程序代码的可读性、可重用性和可移植性

- 自文档化代码

代码拥有可阅读性

- 程序模板
    - 注释
    - 编写相对较短的函数

- 浮点运算的精度

- 代码静态分析工具

- Pylint
      - 检查代码风格是否符合PEP8规范
      - 检查代码是否存在常见的错误和违反最佳实践
      - 检查重复的代码
    - Java
      - Checkstyle
      - FindBugs
      - PMD
      - Jtest
    - C++
      - VS
      - Lint
    - WEB
      - JSHint
      - CSSLint
      - HTMLHint

- 代码审查

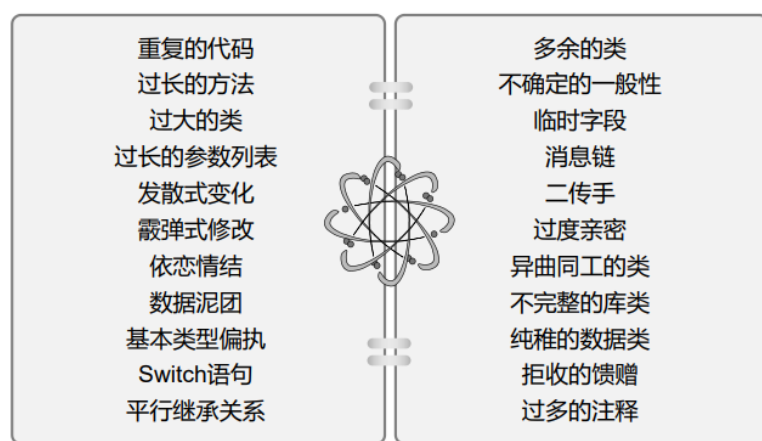
代码审查是一种用来确认**方案设计和代码实现的质量保证机制**，**通过阅读代码**来检查源代码与编码规范的符合性以及代码的质量

- 作用
  - 检查设计合理性
  - 了解别人的代码，防止开发人员离职无人接手
  - 分享知识、设计、技术
  - 增加代码可读性
  - 处理代码中的“地雷区”
- 审查方式
  - 桌面检查（非正式）
    - 由一个程序员人工阅读代码
  - 代码走查
    - 设计或编码人员组成走查小组，阅读文档和代码并提问和讨论
  - 代码审查（正式）
    - 若干编程人员和测试人员组成审查小组，以会议形式阅读、讨论和争议对代码进行静态分析
- 缺陷检查表

## • 代码重构

**重构**是对**软件内部结构**的一种调整，其目的是在**不改变软件功能和外部行为**的前提下，提高其**可理解性、可扩展性和可重用性**

- 不适合重构的情况
  - 代码设计混乱，设计完全错误
  - 时间紧迫
  - 重构工作量影响任务估算时间
- 重构与添加新功能的区别
  - 前者不添加新功能，只对程序结构进行改进；后者不修改既有代码，只添加新功能
- 代码较差的情况



- 重复的代码
  - 拥有相同形式或相同作用的代码
    - 措施
      - 抽取方法
        - 将不同方法内部相同的代码抽取成一个函数

- 抽取类

如果两个不相干的类存在重复代码，则将重复代码抽取成一个独立的类中

- 替换算法

对于相同的功能使用相同的比较清晰的函数统一实现

- 过长的函数

看到超过 N 行代码的方法，就检查是否能够重构

- 措施

- 抽取方法

- 如果需要注释来说明，则可以将注释内容抽取成函数

- 发散式变化

某个类因为不同的原因在不同的方向上变化

- 让调用者找到类方法的目标对象，类本身不负责查找对象
  - 为不同的决策抽取不同的类
  - 如果多个类共享相同类型的决策，则可以进行合并

- 霰弹式修改

发生一次改变时，需要修改多个类的多个地方

- 找到或抽取负责这些修改的类
  - 使用移动字段和移动方法将功能置于所选的类中

- 数据泥团

同样的两至三项数据频繁一起出现类和参数表中；代码声明了某些字段，并声明了处理字段的方法，随后定义了更多字段和方法；各组字段名以类似的子串开头或结束

- 如果项是类中的字段，则抽取到新类中
  - 如果值共同出现在方法的签名中，则引入参数对象
  - 将这些项移至新的对象中

- 纯稚的数据类

类仅由字段构成，或者只有简单的赋值方法和取值方法构成

- 采用封装字段阻止对字段直接访问（仅允许通过赋值方法和取值方法）
  - 对可能的方法尽量采用移除设置方法进行重构。
  - 采用封装集合方法去除对所有集合类型字段的直接访问。
  - 查看对象的各个客户，如果客户试图对数据做同样的工作，则对客户采用抽取方法，然后将方法移到该类中。
  - 在完成上述工作后，可能发现类中存在多处相似的方法，使用相应的重构技术，以协调签名并消除重复。
  - 对字段的大多数访问都不再需要，因为所移动的方法涵盖了其实际应用，因此可以使用隐藏方法来消除对赋值方法和取值方法的访问。