

实现简介和防御式编程

实现的要点

- 实现的代码给人看：可读性、可维护性、可追溯性
- 实现的代码给机器看：性能、正确性、完整性

软件实现

- 依据软件设计构造可运行系统
- 步骤：程序设计 + 程序实现

软件实现中需要注意的问题

1. 建立组织的软件开发资产库
2. 构件的实现及构件库的管理
3. 中间件的实现及中间件的管理
4. 程序设计风格与编程规范的管理

好的实现的特征

- 可读性、可维护性、性能、可追溯性、正确性、完整性

代码规范

- 编写程序的基本原则、规则和建议
- 事先约定的代码风格
- 软件编程规范示例：华为编程规范、Google C++ Style Guide

注释

- 注释的作用：提高或降低可维护性和可读性
- 6种类型注释：代码的重复、代码的解释、代码中的标记、代码的总结、代码意图的描述、外部引用

防御式编程（Defensive Programming）

- 主要思想：保证对程序的不可预见的使用不会造成之后程序功能的破坏
- 隐喻：防御式驾驶
- 不对用户输入做假定
- 前置和后置条件
- 断言（Assertions）

错误处理技术

- 处理预料中可能要发生的错误情况
- 处理方式：返回中立值、换用下一个正确的数据、返回与前次相同的数据、换用最接近的合法值、把警告信息记录到日志文件中、用语言内建的异常机制抛出一个异常等

异常处理机制

- 异常是把代码中的错误或异常事件传递给调用方代码的一种特殊手段
- 基本结构：子程序使用throw抛出一个异常对象，再被调用链上层其他子程序的try-catch语句捕获

健壮性与正确性

- 正确性：软件按照需求正确执行任务的能力
- 健壮性：软件对于规范要求以外的输入情况的处理能力

进攻式编程

- 主动暴露可能出现错误的态度
- 在开发阶段让它显现出来，在产品代码运行时让它能够自我恢复