

常用的面向对象设计原则包括:

原则	说明
单职原则	设计功能单一的类。本原则与结构化方法的高内聚原则是一致的。
开闭原则	对扩展开放, 对修改封闭。
李氏替换原则	子类可以替换父类。(也称里氏替换原则, 子类可以扩展父类的功能, 但不改变父类原有的功能, 子类继承父类时, 除添加新方法外, 尽量不重写父类的方法)。
依赖倒置原则	要依赖于抽象, 而不是具体实现; 要针对接口编程, 不要针对实现编程。
接口隔离原则	使用多个专门的接口比使用单一的总接口要好。
组合重用原则	要尽量使用组合, 而不是继承关系达到重用目的。
迪米特原则(最少知识法则)	一个对象应当对其他对象有尽可能少的了解。本原则与结构化方法的低耦合原则是一致的。

3. 设计模式

设计模式是前人经验的总结, 它使人们可以方便地复用成功的软件设计。

根据处理范围不同, 设计模式可分为类模式和对象模式。

根据目的和用途不同, 设计模式可分为创建型模式、结构型模式和行为型模式三种。

- ①创建型模式主要用于创建对象
- ②结构型模式主要用于处理类或对象的组合
- ③行为型模式主要用于描述类或对象的交互以及职责的分配。

4. 软件实现(掌握)

1. 软件配置管理

无重点

2. 软件编码

无重点

3. 软件测试

软件测试方法可分为静态测试和动态测试。

静态测试是指被测试程序不在机器上运行, 而是采用人工检测和计算机辅助静态分析的手段对程序进行检测。静态测试包括对文档的静态测试和对代码的静态测试, 对文档的静态测试主要是以检查单的形式进行, 而对代码的静态测试一般采用桌前检查、代码走查和代码审查。

• **桌前检查** (Desk Checking): 对程序执行情况进行人工模拟, 用逐步检查源代码中是否有逻辑或语法错误的办法来检测故障。

• **代码走查**: 在代码走查的过程中, 开发人员向其他人来阐述他们的代码。

• **代码审查**: 对计算机源代码系统化地审查, 常用软件同行评审的方式进行, 找出及修正在软件开发初期未发现的错误。

动态测试是指在计算机上实际运行程序进行软件测试。一般采用白盒测试和黑盒测试方法。

白盒测试也称为结构测试, 主要用于软件单元测试中。它的主要思想是, 将程序看作成一个透明的白盒, 测试人员完全清楚程序的结构和算法, 按照程序内部逻辑结构设计测试用例。白盒测试方法主要有控制流测试、数据流测试和程序变异测试等。另外, 使用静态测试的方法也可以实现白盒测试。例如, 使用人工检查代码的方法来检查代码的逻辑问题, 也属于白盒测试。白盒测试方法中, 最常用的技术是逻辑覆盖, 即使用测试数据运行被测试程序, 考察对程