软件开发原则：SOLID，DRY，KISS 及更多

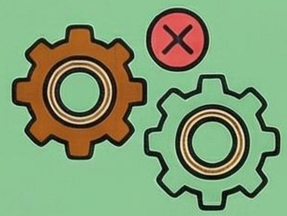
引言与作者

作者：Jean-Jerome Levy，DevOps顾问，专长CI/CD、云。开发原则对项目质量、可维护性至关重要。

关键原则速览

深入探讨SOLID，DRY，KISS等关键开发原则。

SOLID



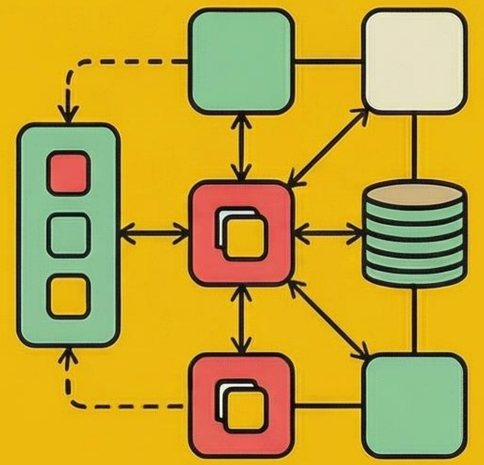
DRY

KISS

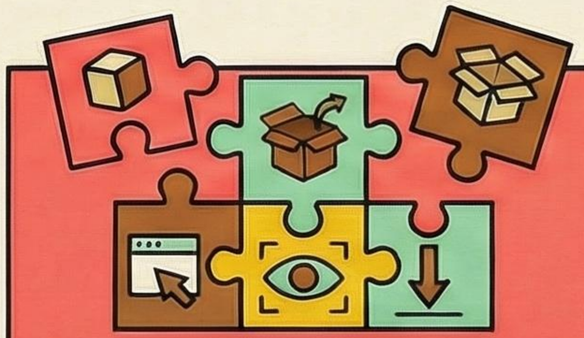
KISS

目标与愿景

旨在实现高质量、易维护、可扩展的代码。

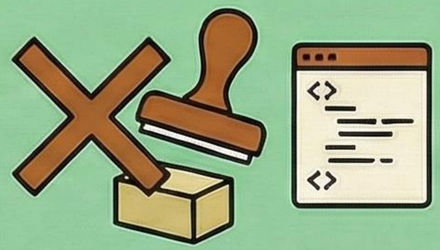


目录



SOLID

- SRP，OCP，LSP，ISP，DIP



DRY

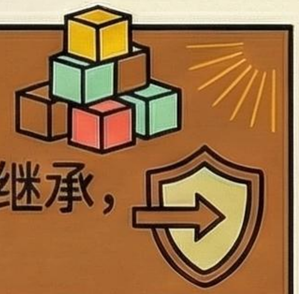
不要重复自己，提升代码质量

KISS

保持简单，避免不必要复杂性

其他重要原则

YAGNI，CoC，组合优于继承，LoD



对创方面

熟练运用常用设计模式

结论与展望

总结并展望未来

SOLID 原则（1/2）

健壮可扩展代码设计基石

单一职责（SRP）

类只负责一个明确职责

- 优点：模块化、易维护、重用性高
- 示例：图书管理中书籍、用户类独立

开放/封闭（OCP）

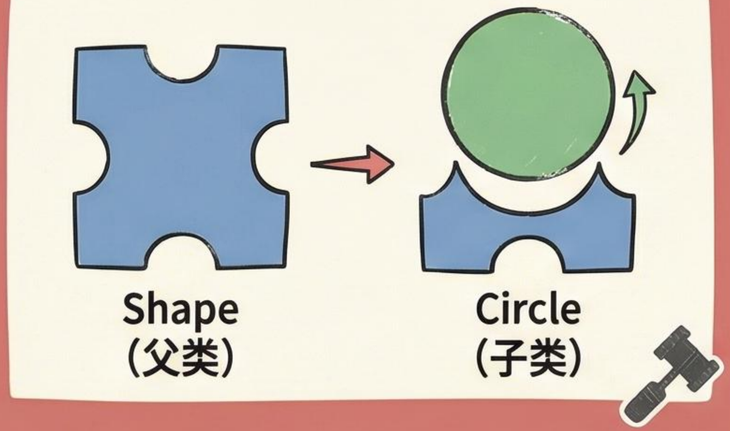
可扩展，不可修改

- 优点：代码灵活、易扩展、利于测试
- 示例：支付应用中抽象方法与子类

主动学习

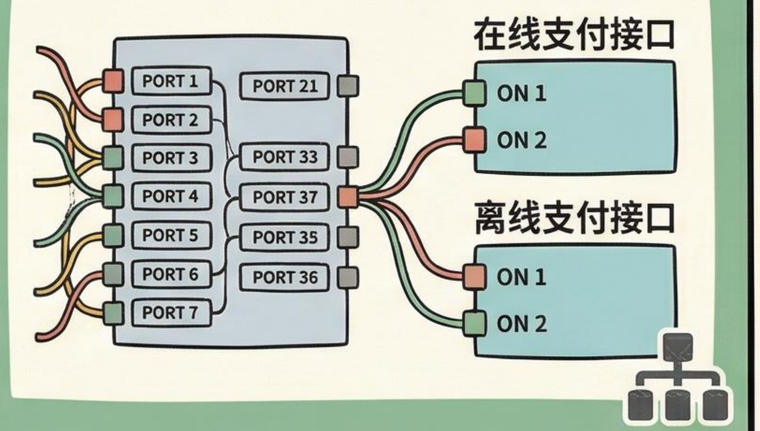
SOLID 原则（2/2）

里氏替换原则（LSP）



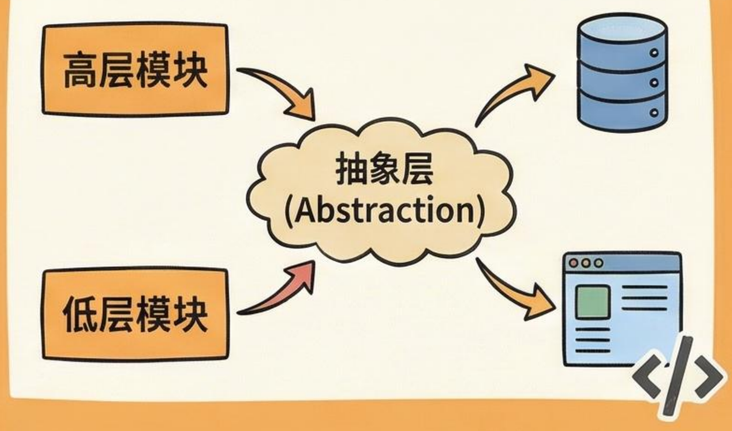
子类可替换父类不影响系统一致性。
优点：促进模块化与重用，确保行为一致。
示例：几何图形子类替代父类（如Circle替代Shape）。

接口隔离原则（ISP）



为客户端定义特定接口，避免单一庞大接口。
优点：接口清晰连贯，维护方便，高灵活性。
示例：电商应用在线与离线支付接口分离。

依赖反转原则（DIP）



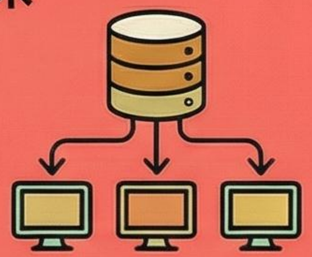
高层不依赖低层，共同依赖抽象。
优点：模块化，易测试，降低模块间耦合。
示例：高层类通过抽象接口依赖低层。

主动学习

DRY（不要重复自己）原则

定义：

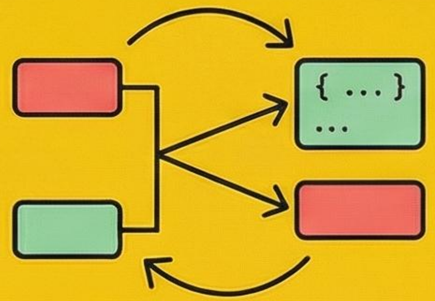
避免代码重复，确保
知识逻辑单一表示

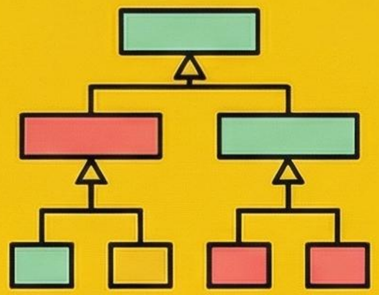


益处：

- 降低复杂性，提升代码可读性
- 简化维护，集中修改和修复
- 促进代码重用和功能封装

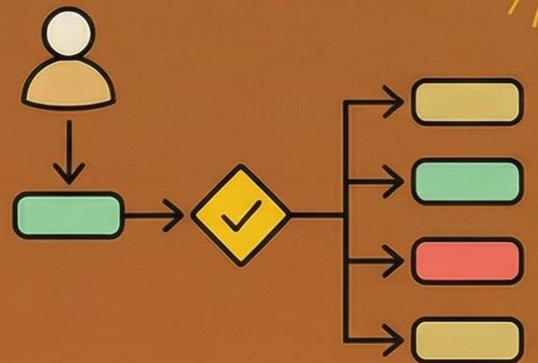
技巧：提取函数，利用类继承、库框架





示例：

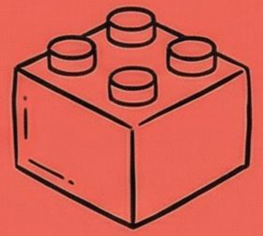
联系人管理中提
取验证逻辑



KISS（保持简单、愚蠢）原则

1. 定义

强调代码简单性，优先选择简单方案。

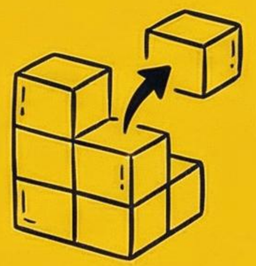


2. 益处：更好的代码理解 & 减少错误

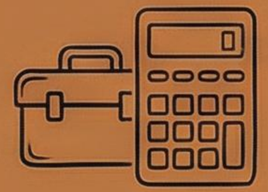
方案直观清晰，易于理解。
简单方案易于测试和验证。

3. 益处：更具扩展性

易于修改，方便添加新功能。



4. 实用技巧与实践示例



实用技巧：
- 避免过度工程与过度抽象
- 命名清晰，避免代码重复（DRY原则）
- 避免过早优化和不必要功能

实践示例：简单计算器，直接函数实现基本运算。

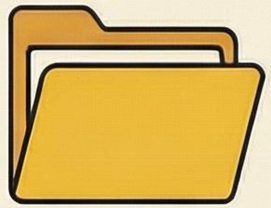
软件原则

其他重要原则（1/2）

YAGNI（你不需要它）

- **定义**：不实现非立即必要的功能或代码
- **优点**：降低复杂性，节省资源，促进迭代开发
- **实践**：添加前自问“现在真的需要吗？”

约定优于配置（CoC）



Java项目目录

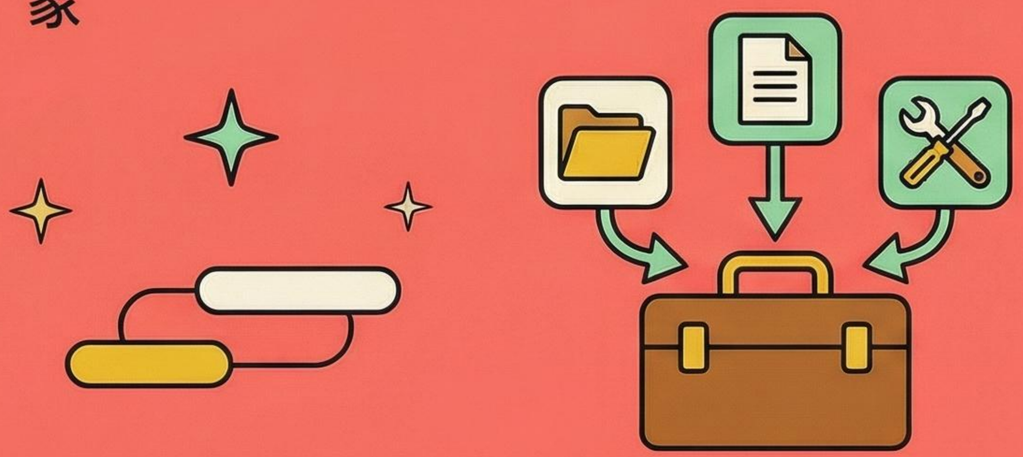
JUnit测试文件

- **定义**：采用预定约定而非显式配置
- **优点**：减少配置，简化开发，提升可读性与协作
- **示例**：Java项目目录、JUnit测试文件命名

其他重要原则（2/2）

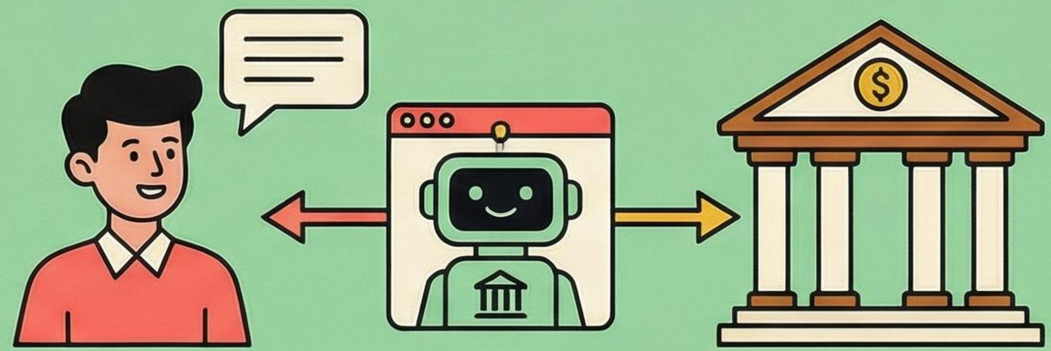
组合优于继承

- 采用组合而非继承，避免硬性依赖
- **优点**：代码灵活重用，提升模块化
- **优点**：降低复杂性，易于维护
- **示例**：文件管理系统组合文件/文件夹对象



迪米特法则（LoD）

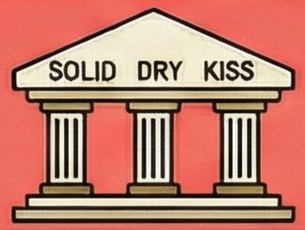
- “只与最亲近朋友交流”原则
- **优点**：促进解耦，减少类间依赖
- **优点**：提高健壮性，降低变更影响
- **示例**：客户端通过接口与银行交互



结论与作者信息

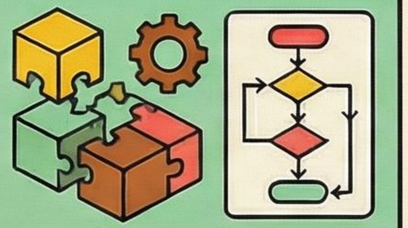
1. 核心原则价值

SOLID，DRY，KISS等是高质量软件基石



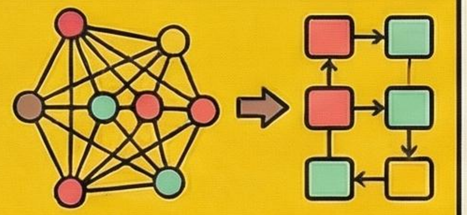
2. 应用优势

构建灵活、可重用、易懂的系统



3. 其他优势

促进模块化，降低复杂性，易于维护



4. 审慎建议

需平衡原则与性能、时间、用户需求

作者介绍

Jean-Jerome Levy，资深DevOps顾问

专业领域

CI/CD, Docker, Kubernetes, Cloud专家

实践经验

擅长创新DevOps实践，20余年经验

合作邀约

欢迎合作创新DevOps项目