Atitit保证架构超前性 前瞻性

目录

[第一章 为什么需要修改代码 1](#_Toc29975)

[第一节 业务增加功能 1](#_Toc21984)

[第二节 增加字段 1](#_Toc20230)

[第三节 增加表数据需要查询 修改 1](#_Toc30467)

[第四节 类库升级 1](#_Toc21234)

[第二章 简单抽象原则 道和术的区别 2](#_Toc22753)

[第一节 原则 kiss 2](#_Toc31119)

[第二节 提升到层次哲学层次 2](#_Toc16642)

[第三节 命名也要通用抽象 隐藏细节 2](#_Toc27489)

[第三章 通用性原则 扩展性 2](#_Toc12941)

[第一节 Dsl扩展 2](#_Toc28892)

[第二节 Udf sp扩展机制等 2](#_Toc1642)

[第三节 Host lan engeen机制 dsl脚本引擎机制 2](#_Toc28803)

[第四章 Diy原则保证永久性 2](#_Toc7515)

[第一节 比如rest orm接口。。自己实现的版本永不过时 2](#_Toc21272)

[第二节 更加底层的aipi 2](#_Toc17410)

[第三节 尽可能diy扩展在底层接口上 3](#_Toc9746)

[第五章 贯彻多元化原则 3](#_Toc24556)

[第一节 总有一个是方向 3](#_Toc22257)

[第二节 Diy可宝永久性 3](#_Toc30231)

[第六章 自由化理念 自由化元则 3](#_Toc30263)

[第一节 自由化参数 json map等半结构化参数，dsl等全功能参数 3](#_Toc24208)

[第二节 自由化功能接口 dsl 3](#_Toc9988)

[第三节 自由化返回数据结构json 3](#_Toc22046)

[第七章 适当分层 隔离稳定性层与业务层 4](#_Toc18595)

[第八章 简单化原则 更少的未来修改 4](#_Toc26124)

[第九章 沟通交流 了解业界动态 4](#_Toc4501)

[第十章 模块化原则方便部分替换 4](#_Toc7066)

# 为什么需要修改代码

## 业务增加功能

## 增加字段

## 增加表数据需要查询 修改

## 类库升级

# 简单抽象原则 道和术的区别

## 原则 kiss

大道至简，避免细节。。细节很容易过时，抽象原则永远不会有问题

类似bibile，老子等书籍永不过时。。

设计既要有一定的前瞻性,又要注意可行性,要在两者之间找到平衡。。细节单独出本规范

## 提升到层次哲学层次

## 命名也要通用抽象 隐藏细节

# 通用性原则 扩展性

## Dsl扩展

类似诶sql

## Udf sp扩展机制等

## Host lan engeen机制 dsl脚本引擎机制

# Diy原则保证永久性

## 比如rest orm接口。。自己实现的版本永不过时

## 更加底层的aipi

## 尽可能diy扩展在底层接口上

比如自己jpa接口建立在底层api上

# 贯彻多元化原则

## 总有一个是方向

## Diy可宝永久性

# 自由化理念 自由化元则

## 自由化参数 json map等半结构化参数，dsl等全功能参数

## 自由化功能接口 dsl

## 自由化返回数据结构json

# 适当分层 隔离稳定性层与业务层

# 简单化原则 更少的未来修改

# 沟通交流 了解业界动态

# 模块化原则方便部分替换