**开发规范**

1. 类命名
2. 首字母大写
3. 见类名即知其意,比如类名为User,很明确的就可以让其他开发者知道这个是用户类与用户相关的,因此在命名类的时候尽量选择名词
4. 驼峰式命名法,比如UserModel,UserService.将不同单词(或组合单词)的首字母,写成大写,使之清晰明了
5. 类名的最后要加上所属模块.示例:
   1. Model包下有个User类,那么它的命名应该是UserModel
   2. Service包下如果有相同的,命名为UserService
   3. Controller包下有相同的,命名为UserController
6. 如果创建的是接口,那么在最后加上Imp作为标示,示例:
   1. Service包下有个User接口,那么它的命名应该是UserServiceImp
7. 方法命名
8. 首字母小写
9. 驼峰式命名,比如insertUser,除首字母外,其后遇到的每个单词首字母都大写
10. 方法表示某种动作行为,所以在名称上也要体现出来,尽量描述为一种操作,比如getName,SetName
11. 返回值为布尔,并且主要判断某个对象(属性)为是否的时候,用is开头,比如isName(),理解为name是否存在,如果存在则返回true,不存在则返回false
12. 数据操纵语句命名
13. 本次开发主要体现在mapper的xml中
14. 首字母小写,驼峰式命名
15. 查询一条或多条记录的时候以select开头,示例:
    1. selectUser 返回一条或多条user表中的记录
    2. selectGoods 返回一条或多条goods表中的记录
16. 查询一条记录的时候以find开头,示例:
17. findUser 返回一条user表中的记录
18. 插入数据以insert开头,示例:
    1. insertUser 插入一条记录到user表中
19. 删除数据以delete开头,示例:
    1. deleteUser 删除一条user表中的记录
20. 修改数据以update开头,示例:
    1. updateUser 修改user表中的记录
21. 根据某种条件的时候以By为分隔符
    1. selectUserByUsername 根据username字段查询user表中数据,返回一条或者多条
    2. findUserById 根据ID查询user表中数据,返回一条
    3. deleteUserById 根据ID删除user表中的一条记录
    4. updateUserById 根据ID修改user表中的记录
22. 如有其他未包含在内的规定,商讨后另定
23. 变量命名规范
24. 变量名全部小写,示例
    1. name
25. 多个单词中间使用下划线分割,示例:
    1. goods\_name
26. 如没有特殊要求,所有变量全部设为私有类型(private),只对外暴露get于set方法
27. 包
28. 不出意外,本次主要会有4个包,以下:
    1. com.pas.controller controller层,负责业务逻辑,与页面交互
    2. com.pas.dao dao层,负责基本数据操纵
    3. com.pas.service service层,负责通过dao层的方法,写出实际操作
    4. com.pas.model 存储数据库中的实体

如另有其他问题,再进行交流,本文档只确定了最基本的规范.

-侯章旭 于

2017年6月17日01:29:21