架构规范

1. 目的

为了使软件开发过程顺畅，保证软件质量，于是有了这份开发规范文档。

1. 命名规范
2. 类命名
   1. 类命名采用Pascal命名法
   2. 大写字母开头，各个单词首字母大写
3. 方法命名
   1. 方法命名采用Pascal命名法
   2. 大写字母开头，各个单词首字母大写
   3. 属性的getter和setter方法尽量使用自动，并放在程序的后面
   4. Boolean类型的属性的get方法，形式尽量采用isProperty()
4. 包命名
   1. 包命名采用全小写命名
   2. 项目包命名按照端口分化,例如yhb-api 包命名 com.lrcf.yhb.api+具体业务
   3. Spring boot启动类放在具体业务之前
5. 变量命名
   1. 采用Camel命名法
   2. 小写字母开头，各个单词首字母大写
   3. 特有大写缩写词汇保持大写如：SESSION
   4. 变量名字不宜过长，可适当采用缩减英文元音字母来缩短长度
   5. 假如缩短后名字重复，可以保留其中一个的部分元音字母
6. 资源文件命名
   1. 采用**全小写命名法**
   2. 按照前缀+模块+描述+状态命名，如：btn\_main\_login\_pressed.png
   3. (不一定完全包含如上4个部分，但是要依照顺序命名)
7. 常量命名
   1. 采用**全大写命名法**
   2. 所有字母均大写
8. 项目模块规范
   1. Yhb-common 公共方法
   2. yhb-user-service 用户服务业务
   3. yhb-user-api 用户服务接口
   4. yhb-product-api 产品服务接口
   5. yhb-product-service 产品服务业务
   6. yhb-gift-api 礼券服务接口
   7. yhb-gift-service 礼券服务业务
   8. yhb-company-api 企业服务接口
   9. yhb-company-service 企业服务业务
   10. yhb-media-api 媒体宣传服务接口
   11. yhb-media-service 媒体宣传服务业务
   12. yhb-web-api 后端API请求路径
   13. yhb-tps-api 环迅接口api
   14. yhb-tps-service 环迅接口业务
   15. yhb-dao 统一后端+实体类
   16. ………
9. 注释规范
   1. 参考模板

/\*\*

\* @Desc $desc$ 描述

\* @param $params$ 参数

\* @return $return$ 返回值

\* @Author 孔荣勋【Krx】 作者

\* @Date $date$ $time$ 时间

\*/

4.2.代码注释量必须达到30%以上，不能一点没有，也不是注释很多，只需要注释的代码别人帮你看的时候，能看懂什么逻辑思维。

1. 性能规范
   1. 方法参数类型和参数返回值尽量接口花，以屏蔽具体实现细节，提高系统的可扩展新，例如:public void addUser(List userList){} public List listAllUsers(){}
   2. 方法接口，api不要加修饰符例如 YhbResult login(String name,String pwd,String type) ;
   3. 如果没有必要，不要序列化对象
   4. 垃圾回收和资源释放，可能有异常的操作时必须在try和finally块中释放资源,如数据库连接,I/O操作，Redis
   5. 尽量不面试用Vector和HashTable等旧的集合实现。由于实现实时同步的，避免大量操作带来不必要的性能损失。应使用AyyayList和HashMap来代替。如果一定要使用同步集合类，当时用如下方式:Map map =Collections.synchronizedMap(new HashMap());
2. 代码排版
   1. If,where,for,do,while,case,switch,default等语句后再写代码时必须空格隔开，无论这些语句中代码量有多少必须跟后面代码分割开来，这样阅读代码更加清晰，减少错误产生

实例:

If(money>1000){

isProsty();

}

--空格分开

//\*\*\*\*接下来的代码

Return money;

* 1. 较长的语句或表达式或参数（根据屏幕所占比）要分成多行书写，长表达式要在低优先级操作符处划分新航，操作符放在首位，划分出的新行要进行适当的缩进，使排版整齐，语句清晰

实例:

If（StringUtils.isBlank(name)）{

log.debug(“user name is null, error info message”

+ BaseExcepetion.getMessage());

}

* 1. 不允许把多个短句或定义变量信息放在一行，即一行只写一条语句，让别人改你的代码时阅读更清晰

实例:下面错误实例

String name = new String(); String phone = null;

* 1. 两个变量赋值，或对等操作时，他们之间的操作符之前，之后或者前后要加空格；

实例:

BigDecimal money = new BigDecimal(2000);

BigDecima bMoney = new BigDecimal(2000);

If(money == bMoney){

isProsty();

}

money = bMoney .muply(2);

Return money；