Java 中的方法体定义要求最好简洁，看看Java源码就很简洁，其实就是最好将大功能模块分割为小功能模块。要高内聚低耦合。

Return有结束方法的功能。

**方法（method）的重载**：需要保证***方法名称和返回值类型***都一致，差异在于参数个数、参数类型、方法体

建议少用递归操作，可能引发栈溢出问题，但是递归处处都在，有些功能甚至只能使用递归来进行处理，（比如数据结构），在Java的系统类库封装中也是用了不少的递归。

封装、继承、多态（为具有统一抽象性质的物体提供一个统一的接口、类别，就像生物学的物种划分命名）

OOA OOD OOP

类与对象的概念越来越贴近人类生活，类的概念提供了代码的重用性。

声明并实例化：类名 实例名=new 类名（）

声明：

类名 实例名=null;

实例化：实例名=new 类名（）;

New用于开辟内存空间，Java中没有delete。所以c与c++是自己管理内存空间，而Java是通过垃圾回收器进行内存的管理。

在Java中如果没有一开始就对属性进行赋值那么Java在对其进行初始化时的默认值：

整型（0）、浮点型（0.0）、布尔型（false）、字符型（’\u0000’）、引用类型（null）

***（！！！！！！！！注意，默认值只有在类的属性定义中有效，在方法中的变量无效）***

***在jdk1.4变量的定义时就必须给出初始值，但是在jdk1.5及其以后，为了灵活编程，可以不用在定义时给出初始值，但是系统并不会给他加默认值，在后面使用改变量时还是必须给出一个值。***

阿里 java 变量命名规范\_阿里巴巴Java开发手册-命名规约

纳兰家的容若 2021-02-25 06:36:41 32 已收藏 1

文章标签： 阿里 java 变量命名规范

版权

1. 【强制】 代码中的命名均不能以下划线或美元符号开始，也不能以下划线或美元符号结束。

反例： \_name / \_\_name / $Object / name\_ / name$ / Object$

2. 【强制】 代码中的命名严禁使用拼音与英文混合的方式，更不允许直接使用中文的方式。

说明：正确的英文拼写和语法可以让阅读者易于理解，避免歧义。注意，即使纯拼音命名方式

也要避免采用。

反例：  DaZhePromotion [ 打折 ] /  getPingfenByName() [ 评分 ] /  int 某变量 = 3

正例：  alibaba /  taobao /  youku /  hangzhou 等国际通用的名称，可视同英文。

3. 【强制】类名使用 UpperCamelCase 风格，必须遵从驼峰形式，但以下情形例外： ( 领域模型

的相关命名 )DO /  BO  /  DTO /  VO 等。

正例： MarcoPolo /  UserDO /  XmlService /  TcpUdpDeal /  TaPromotion

反例： macroPolo /  UserDo /  XMLService /  TCPUDPDeal /  TAPromotion

4. 【强制】方法名、参数名、成员变量、局部变量都统一使用 lowerCamelCase 风格，必须遵从

驼峰形式。

正例：  localValue /  getHttpMessage() /  inputUserId

5. 【强制】常量命名全部大写，单词间用下划线隔开，力求语义表达完整清楚，不要嫌名字长。

正例：  MAX \_ STOCK \_ COUNT

反例：  MAX \_ COUNT

6. 【强制】抽象类命名使用 Abstract 或 Base 开头 ； 异常类命名使用 Exception 结尾 ； 测试类

命名以它要测试的类的名称开始，以 Test 结尾。

7. 【强制】中括号是数组类型的一部分，数组定义如下： String[] args;

反例：使用 String args[] 的方式来定义。

8. 【强制】 POJO 类中布尔类型的变量，都不要加 is ，否则部分框架解析会引起序列化错误。

反例：定义为基本数据类型 Boolean isSuccess； 的属性，它的方法也是 isSuccess() ， RPC

框架在反向解析的时候，“以为”对应的属性名称是 success ，导致属性获取不到，进而抛出异

常。

9. 【强制】包名统一使用小写，点分隔符之间有且仅有一个自然语义的英语单词。包名统一使用

单数形式，但是类名如果有复数含义，类名可以使用复数形式。

正例： 应用工具类包名为 com . alibaba . open . util 、类名为 MessageUtils( 此规则参考

spring 的框架结构 )

10. 【强制】杜绝完全不规范的缩写，避免望文不知义。

反例：  AbstractClass “缩写”命名成 AbsClass；condition “缩写”命名成  condi ，此类

随意缩写严重降低了代码的可阅读性。

11. 【推荐】如果使用到了设计模式，建议在类名中体现出具体模式。

说明：将设计模式体现在名字中，有利于阅读者快速理解架构设计思想。

正例： public class OrderFactory;

public class LoginProxy;

public class ResourceObserver;

12. 【推荐】接口类中的方法和属性不要加任何修饰符号 (public 也不要加 ) ，保持代码的简洁

性，并加上有效的 Javadoc 注释。尽量不要在接口里定义变量，如果一定要定义变量，肯定是

与接口方法相关，并且是整个应用的基础常量。

正例：接口方法签名： void f();

接口基础常量表示： String COMPANY = " alibaba " ;

反例：接口方法定义： public abstract void f();

说明： JDK 8 中接口允许有默认实现，那么这个 default 方法，是对所有实现类都有价值的默

认实现。

13. 接口和实现类的命名有两套规则：

1 ) 【强制】对于 Service 和 DAO 类，基于 SOA 的理念，暴露出来的服务一定是接口，内部

的实现类用 Impl 的后缀与接口区别。

正例： CacheServiceImpl 实现 CacheService 接口。

2 )  【推荐】 如果是形容能力的接口名称，取对应的形容词做接口名 ( 通常是– able 的形式 ) 。

正例： AbstractTranslator 实现  Translatable 。

14. 【参考】枚举类名建议带上 Enum 后缀，枚举成员名称需要全大写，单词间用下划线隔开。

说明：枚举其实就是特殊的常量类，且构造方法被默认强制是私有。

正例：枚举名字： DealStatusEnum， 成员名称： SUCCESS /  UNKOWN \_ REASON 。

15. 【参考】各层命名规约：

A) Service / DAO 层方法命名规约

1 ) 获取单个对象的方法用 get 做前缀。

2 ) 获取多个对象的方法用 list 做前缀。

3 ) 获取统计值的方法用 count 做前缀。

4 ) 插入的方法用 save( 推荐 ) 或 insert 做前缀。

5 ) 删除的方法用 remove( 推荐 ) 或 delete 做前缀。

6 ) 修改的方法用 update 做前缀。

B) 领域模型命名规约

1 ) 数据对象： xxxDO ， xxx 即为数据表名。

2 ) 数据传输对象： xxxDTO ， xxx 为业务领域相关的名称。

3 ) 展示对象： xxxVO ， xxx 一般为网页名称。

4 ) POJO 是 DO / DTO / BO / VO 的统称，禁止命名成 xxxPOJO 。

————————————————

版权声明：本文为CSDN博主「纳兰家的容若」的原创文章，遵循CC 4.0 BY-SA版权协议，转载请附上原文出处链接及本声明。

原文链接：https://blog.csdn.net/weixin\_42495394/article/details/114616042