编程规约

本章是传统意义上的代码规范，包括变量命名、代码风格、控制语句、代码注释等基本的编程习惯，以及从高并发场景中提炼出来的集合处理技巧与并发多线程的注意事项。

1.1 命名风格

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

   反例：_name / __name /$name / name_ / name$ / name__

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，即使是纯拼音命名方式也要避免采用。

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

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

3. 【强制】类名使用 UpperCamelCase 风格，但 DO/BO/ DTO/ VO/AO/PO 等情形例外。

   正例：MarcoPolo / UserDO / XmlService / TcpUdpDeal / TaPromotion/QrCode

   反例：macroPolo / UserDo / XMLService / TCPUDPDeal / TAPromotion/QRCode

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，单词间用下画线隔开，力求语义表达完整清楚，不要嫌名字长。

   正例：MAX_STOCK_COUNT/PRIZE_NUMBER_EVERYDAY

   反例：MAX_COUNT/PRIZE_NUMBER

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名开始，以 Test 结尾。

7. 【强制】类型与中括号之间无空格相连定义数组。

   正例：定义整形数组 int[] arrayDemo;

   反例：在 main 参数中，使用 String args[]来定义。

8. 【强制】POJO 类中布尔类型的变量都不要加 is 前缀，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 isDeleted; 的属性，它的方法也是 isDeleted()，RPC 框架在反向解析的时候，“误以为”对应的属性名称是 deleted，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，则类名可以使用复数形式。

正例：应用工具类包名为 com.alibaba.ai.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，避免词不达义。

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读性。

11 【推荐】为了达到代码自解释的目标，任何自定义编程元素在命名时，使用尽量完整的单词组合来表达其意。

正例：从远程仓库拉取代码的类命名为 PullCodeFromRemoteRepository

反例：变量 int a; 的随意命名方式。

12 【推荐】如果模块、接口、类、方法使用了设计模式，应在命名时体现出具体模式。

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理念。

正例：
```
public class OrderFactory;
public class LoginProxy;
```