5. 【强制】二方库里可以定义枚举类型，参数可以使用枚举类型，但是接口返回值不允许使用枚举类型或者包含枚举类型的 POJO 对象。

6. 【强制】依赖于一个二方库群时，必须定义一个统一的版本变量，避免版本号不一致。
   说明：依赖 springframework-core,-context,-beans，它们都是同一个版本，可以定义一个变量来保存版本：${spring.version}，定义依赖的时候，引用该版本。

7. 【强制】禁止在子项目的 pom 依赖中出现相同的 GroupId，相同的 ArtifactId，但是不同的 Version。
   说明：在本地调试时会使用各子项目指定的版本号，但是合并成一个 war，只能有一个版本号出现在最后的 lib 目录中。可能出现线下调试是正确的，发布到线上却出故障的问题。

8. 【推荐】所有 pom 文件中的依赖声明放在<dependencies>语句块中，所有版本仲裁放在<dependencyManagement>语句块中。
   说明：<dependencyManagement>里只是声明版本，并不实现引入，因此子项目需要显式的声明依赖，version 和 scope 都读取自父 pom。而<dependencies>所有声明在主 pom 的<dependencies>里的依赖都会自动引入，并默认被所有的子项目继承。

9. 【推荐】二方库不要有配置项，最低限度不要再增加配置项。

10. 【参考】为避免应用二方库的依赖冲突问题，二方库发布者应当遵循以下原则：
    1）**精简可控原则**。移除一切不必要的 API 和依赖，只包含 Service API、必要的领域模型对象、Utils 类、常量、枚举等。如果依赖其它二方库，尽量是 provided 引入，让二方库使用者去依赖具体版本号；无 log 具体实现，只依赖日志框架。
    2）**稳定可追溯原则**。每个版本的变化应该被记录，二方库由谁维护，源码在哪里，都需要能方便查到。除非用户主动升级版本，否则公共二方库的行为不应该发生变化。

（三）服务器

1. 【推荐】高并发服务器建议调小 TCP 协议的 time_wait 超时时间。
   说明：操作系统默认 240 秒后，才会关闭处于 time_wait 状态的连接，在高并发访问下，服务器端会因为处于 time_wait 的连接数太多，可能无法建立新的连接，所以需要在服务器上调小此等待值。
   正例：在 linux 服务器上请通过变更/etc/sysctl.conf 文件去修改该缺省值（秒）：

   ```
   net.ipv4.tcp_fin_timeout = 30
   ```

2. 【推荐】调大服务器所支持的最大文件句柄数（File Descriptor，简写为 fd）。
   说明：主流操作系统的设计是将 TCP/UDP 连接采用与文件一样的方式去管理，即一个连接对应于一个 fd。主流的 linux 服务器默认所支持最大 fd 数量为 1024，当并发连接数很大时很

容易因为 fd 不足而出现“open too many files”错误，导致新的连接无法建立。 建议将 linux 服务器所支持的最大句柄数调高数倍（与服务器的内存数量相关）。

3. 【推荐】给 JVM 环境参数设置-XX:+HeapDumpOnOutOfMemoryError 参数，让 JVM 碰到 OOM 场景时输出 dump 信息。

   说明：OOM 的发生是有概率的，甚至相隔数月才出现一例，出错时的堆内信息对解决问题非常有帮助。

4. 【推荐】在线上生产环境，JVM 的 Xms 和 Xmx 设置一样大小的内存容量，避免在 GC 后调整堆大小带来的压力。

5. 【参考】服务器内部重定向使用 forward；外部重定向地址使用 URL 拼装工具类来生成，否则会带来 URL 维护不一致的问题和潜在的安全风险。

七、设计规约

1. 【强制】存储方案和底层数据结构的设计获得评审一致通过，并沉淀成为文档。
   说明：有缺陷的底层数据结构容易导致系统风险上升，可扩展性下降，重构成本也会因历史数据迁移和系统平滑过渡而陡然增加，所以，存储方案和数据结构需要认真地进行设计和评审，生产环境提交执行后，需要进行 double check。
   正例：评审内容包括存储介质选型、表结构设计能否满足技术方案、存取性能和存储空间能否满足业务发展、表或字段之间的辩证关系、字段名称、字段类型、索引等；数据结构变更（如在原有表中新增字段）也需要进行评审通过后上线。

2. 【强制】在需求分析阶段，如果与系统交互的 User 超过一类并且相关的 User Case 超过 5 个，使用用例图来表达更加清晰的结构化需求。

3. 【强制】如果某个业务对象的状态超过 3 个，使用状态图来表达并且明确状态变化的各个触发条件。
   说明：状态图的核心是对象状态，首先明确对象有多少种状态，然后明确两两状态之间是否存在直接转换关系，再明确触发状态转换的条件是什么。
   正例：淘宝订单状态有已下单、待付款、已付款、待发货、已发货、已收货等。比如已下单与已收货这两种状态之间是不可能有直接转换关系的。

4. 【强制】如果系统中某个功能的调用链路上的涉及对象超过 3 个，使用时序图来表达并且明确各调用环节的输入与输出。
   说明：时序图反映了一系列对象间的交互与协作关系，清晰立体地反映系统的调用纵深链路。

5. 【强制】如果系统中模型类超过 5 个，并且存在复杂的依赖关系，使用类图来表达并且明确类之间的关系。
   说明：类图像建筑领域的施工图，如果搭平房，可能不需要，但如果建造蚂蚁 Z 空间大楼，肯定需要详细的施工图。

6. 【强制】如果系统中超过 2 个对象之间存在协作关系，并且需要表示复杂的处理流程，使用活动图来表示。
   说明：活动图是流程图的扩展，增加了能够体现协作关系的对象泳道，支持表示并发等。

7. 【推荐】需求分析与系统设计在考虑主干功能的同时，需要充分评估异常流程与业务边界。
   反例：用户在淘宝付款过程中，银行扣款成功，发送给用户扣款成功短信，但是支付宝入款时由于断网演练产生异常，淘宝订单页面依然显示未付款，导致用户投诉。

8. 【推荐】类在设计与实现时要符合单一原则。
   说明：单一原则最易理解却是最难实现的一条规则，随着系统演进，很多时候，忘记了类设计的初衷。

9. 【推荐】谨慎使用继承的方式来进行扩展，优先使用聚合/组合的方式来实现。
   说明：不得已使用继承的话，必须符合里氏代换原则，此原则说父类能够出现的地方子类一定能够出现，比如，“把钱交出来”，钱的子类美元、欧元、人民币等都可以出现。

10. 【推荐】系统设计时，根据依赖倒置原则，尽量依赖抽象类与接口，有利于扩展与维护。
    说明：低层次模块依赖于高层次模块的抽象，方便系统间的解耦。

11. 【推荐】系统设计时，注意对扩展开放，对修改闭合。
    说明：极端情况下，交付的代码都是不可修改的，同一业务域内的需求变化，通过模块或类的扩展来实现。

12. 【推荐】系统设计阶段，共性业务或公共行为抽取出来公共模块、公共配置、公共类、公共方法等，避免出现重复代码或重复配置的情况。
    说明：随着代码的重复次数不断增加，维护成本指数级上升。

13. 【推荐】避免如下误解：敏捷开发 = 讲故事 + 编码 + 发布。
    说明：敏捷开发是快速交付迭代可用的系统，省略多余的设计方案，摒弃传统的审批流程，但核心关键点上的必要设计和文档沉淀是需要的。
    反例：某团队为了业务快速发展，敏捷成了产品经理催进度的借口，系统中均是勉强能运行但像面条一样的代码，可维护性和可扩展性极差，一年之后，不得不进行大规模重构，得不偿失。

14. 【参考】系统设计主要目的是明确需求、理顺逻辑、后期维护，次要目的用于指导编码。
    说明：避免为了设计而设计，系统设计文档有助于后期的系统维护，所以设计结果需要进行分类归档保存。

15. 【参考】设计的本质就是识别和表达系统难点，找到系统的变化点，并隔离变化点。
    说明：世间众多设计模式目的是相同的，即隔离系统变化点。

16. 【参考】系统架构设计的目的：
    - 确定系统边界。确定系统在技术层面上的做与不做。
    - 确定系统内模块之间的关系。确定模块之间的依赖关系及模块的宏观输入与输出。
    - 确定指导后续设计与演化的原则。使后续的子系统或模块设计在规定的框架内继续演化。
    - 确定非功能性需求。非功能性需求是指安全性、可用性、可扩展性等。

附 1：版本历史

版本号	更新日期	备注
1.0.0	2017.2.9	阿里巴巴集团正式对外发布
1.0.1	2017.2.13	1）修正 String[]的前后矛盾。2）vm 修正成 velocity。3）修正 countdown 描述错误。
1.0.2	2017.2.20	1）去除文底水印。2）数据类型中引用太阳系年龄问题。3）修正关于异常和方法签名的部分描述。4）修正 final 描述。5）去除 Comparator 部分描述。
1.1.0	2017.2.27	1）增加前言。2）增加<? extends T>描述和说明。3）增加版本历史。4）增加专有名词解释。
1.1.1	2017.3.31	修正页码总数和部分示例。
1.2.0	2017.5.20	1）根据云栖社区的“聚能聊”活动反馈，对手册的页码、排版、描述进行修正。2）增加 final 的适用场景描述。3）增加关于锁的粒度的说明。4）增加“指定集合大小”的详细说明以及正反例。5）增加卫语句的示例代码。6）明确数据库表示删除概念的字段名为 is_deleted
1.3.0	2017.9.25	增加单元测试规约（PDF 终极版），阿里开源的 IDE 代码规约检测插件：点此下载 更多及时信息，请关注《阿里巴巴 Java 开发手册》官方公众号： 
1.3.1	2017.11.30	修正部分描述；采用和 P3C 开源 IDE 检测插件相同的 Apache2.0 协议。
1.4.0	2018.5.20	增加设计规约（详尽版）