## 一、练习题

### 第一部分

1. 下面属于RocketMQ架构中的角色的是：ABCD
2. Producer
3. Consumer
4. Broker
5. NameServer
6. Zookeeper
7. 下面属于RocketMQ特性的是：ABCEFGHIJK
8. 订阅与发布
9. 消息有序
10. 消息过滤
11. RocketMQ“仅一次”语义
12. 回溯消费
13. 事务消息
14. 定时消息
15. 消息重试
16. 消息重投
17. 流控
18. 死信队列
19. 关于RocketMQ的消息推拉说法正确的是：ABCD
20. RocketMQ支持消息推拉两种模式
21. RocketMQ的推消息本质上是拉消息
22. Push模式实时性高
23. Pull模式由客户端控制消费的频率
24. RocketMQ角色和属于表述正确的是：ABE
25. RocketMQ的消息模型包括生产者、消费者、Broker，其中每个Broker可以存储多个主题的消息，每个主题的消息可以分片存储于不同的Broker上。MessageQueue是主题分片的最小单元，多个生产者可以属于同一个生产组，多个消费者可以属于同一个消费组。
26. 生产组中的生产者发送同一类型的消息，且发送逻辑一致。如果发送的是事务消息，原生产者崩溃之后，Broker可以联系同一生产组的其他生产者实例提交或回滚。
27. 同一消费组中消费同一类消息且消费逻辑可以不一致。消费者也不需要订阅完全相同的主题。
28. 广播消费表示一条消息可以被多个消费者消费，但是同一消费组的消费者只能被一个消费者消费一次。
29. 集群消费表示一个消费组中的消费者平摊同一个主题中的消息。

### 第二部分

1. 消息发送的返回状态包括：ABD
2. FLUSH\_DISK\_TIMEOUT
3. FLUSH\_SLAVE\_TIMEOUT
4. FLUSH\_NOT\_AVAILABLE
5. SEND\_OK
6. 关于提升生产者写入性能的说法正确的是：ABCD
7. 可以采用Oneway方式发送
8. 可以增加生产者提高消息发送的并发量
9. 推荐使用EXT4文件系统，IO调度算法使用deadline算法
10. 顺序写CommitLog无论是HDD还是SSD磁盘，都能保持较高的写入性能
11. 关于消息存储，下列说法正确的是：ABCE
12. 消息真正的物理存储文件是CommitLog，ConsumeQueue是消息的逻辑队列
13. 目前的高性能磁盘顺序写超过了一般网卡的传输速度。
14. ConsumeQueue保存了指定主题下队列消息在CommitLog中的起始物理偏移量，消息大小以及消息标签的HashCode值
15. CommitLog在滚动的时候，每个新文件的名称就是该新文件中第一条消息的时间戳
16. IndexFile文件名fileName是以创建时的时间戳命名的。
17. 下面关于零拷贝原理说法正确的是：ACD
18. RocketMQ既使用了HeapByteBuffer，也使用了DirectByteBuffer
19. Java中有HeapByteBuffer和DirectByteBuffer两个类对应。
20. HeapByteBuffer的数据维护在JVM堆空间，写入数独快，回收方便
21. DirectByteBuffer数据维护在堆外内存，可以省略数据从JVM拷贝到系统内存的步骤。
22. 操作系统不能直接操作JVM内存。
23. 关于消息重试，下列说法正确的是：ADE
24. 消息的重试只针对集群消费方式，对广播方式无效
25. RocketMQ的消息队列默认允许每条消息最多重试18次，时间依次延长。
26. 在消息重试的时候，需要客户端维护消息的MessageID，否则会引起混乱
27. 在客户端需要重试的场合，可以返回Null，可以抛异常，对于并发消费，推荐使用ConsumeConcurrentlyStatus.RECONSUME\_LATER。
28. 顺序消息在消费失败后，RocketMQ会不断进行消息重试，可能引起应用阻塞。

### 第三部分

1. NameServer的设计特点包括：ABD
2. NameServer相互独立，彼此没有通信，单台NameServer宕机不影响其他NameServer。
3. NameServer不主动推消息
4. Broker默认每隔10s与NameServer发送心跳包，保持活跃状态。
5. Consumer随机与一个NameServer建立长连接，如果NameServer断开，则从NameServer列表中查找下一个NameServer进行连接。
6. 关于故障对消息的影响，正确的是：AD
7. Broker正常关闭，启动，如果是同步发送，不会引起消息的丢失。
8. Broker异常宕机，然后启动，不会引起消息的丢失。
9. 磁盘损坏，一定会造成消息丢失。
10. 机器损坏，但能马上恢复供电，如果MASTER、SLAVE都配置为SYNC\_FLUSH，可以不丢失消息。
11. 关于消费者说法，正确的是：ABCD
12. RocketMQ无法避免消息重复。
13. 可以提高消费者数量达到加快消费的目的。
14. 可以使用批量方式消费，加快消费速度。
15. 可以跳过不重要的消息，提高消费速度。

### 第四部分

1. 动态增减NameServer机器，下列说法正确的是：ABCD
2. 可以在客户端通过setNamesrvAddr的方式扩展Namesrv
3. 可以在Java启动参数中设置，扩展Namesrv
4. 可以通过Linux环境变量，扩展Namesrv
5. 可以通过HTTP服务，扩展Namesrv
6. 新增一个主题的消费组，无法消费历史消息，该如何解决？BC
7. setConsumeFromWhere(ConsumeFromWhere.CONSUME\_FROM\_LAST\_OFFSET)
8. setConsumeFromWhere(ConsumeFromWhere.CONSUME\_FROM\_FIRST\_OFFSET)
9. setConsumeFromWhere(ConsumeFromWhere.CONSUME\_FROM\_TIMESTAMP)

consumer.setConsumeTimeout(System.currentTimeMillis() - 3600000);

1. 关于刷盘机制，下列说法正确的是：ABCD
2. RocketMQ所有消息都是持久化的
3. 同步刷盘中，消息首先写入PageCache，然后线程等待，通知刷盘线程刷盘，前端线程等待。
4. 异步刷盘中，消息写入到PageCache，前端线程返回给生产者。
5. 如果内存不足，则尝试丢弃干净的Page，如果还内存不足，则OS尝试刷盘部分脏页。

### 第五部分

1. 关于死信消息的特性，正确的是：BCE
2. 死信不会被消费者消费
3. 死信有效期与正常消息相同，均为3天，3天后被自动删除。
4. 一个死信队列对应一个GroupId
5. 如果GroupId没有产生死信消息，则RocketMQ也要为其创建相应的死信队列。
6. 死信队列保存了对应的GroupId的所有死信，不管属于哪个主题。
7. 关于定时消息，下列说法正确的是：ADE
8. messageDelayLevel有18个级别
9. messageDelayLevel是主题的属性
10. messageDelayLevel的值可以小于0
11. messageDelayLevel的值可以大于18
12. 定时消息暂存于SCHEDULE\_TOPIC\_XXXX主题中。
13. 下列关于事务消息说法正确的是：BCD
14. 生产者向RocketMQ发送“待确认”消息，订阅了该主题的消费者即可看到该消息
15. 生产者发送完待确认消息，收到RocketMQ的确认之后，执行本地事务逻辑。
16. 生产者根据本地事务逻辑的执行结果，提交或回滚事务消息
17. Broker可以对生产者的本地事务结果发起“回查”作为事务的补偿手段。

## 二、实战题

基于RocketMQ设计秒杀。

要求：

1. 秒杀商品LagouPhone，数量100个。
2. 秒杀商品不能超卖。
3. 抢购链接隐藏
4. Nginx+Redis+RocketMQ+Tomcat+MySQL

提示：



