# RabbitMQ学习总结

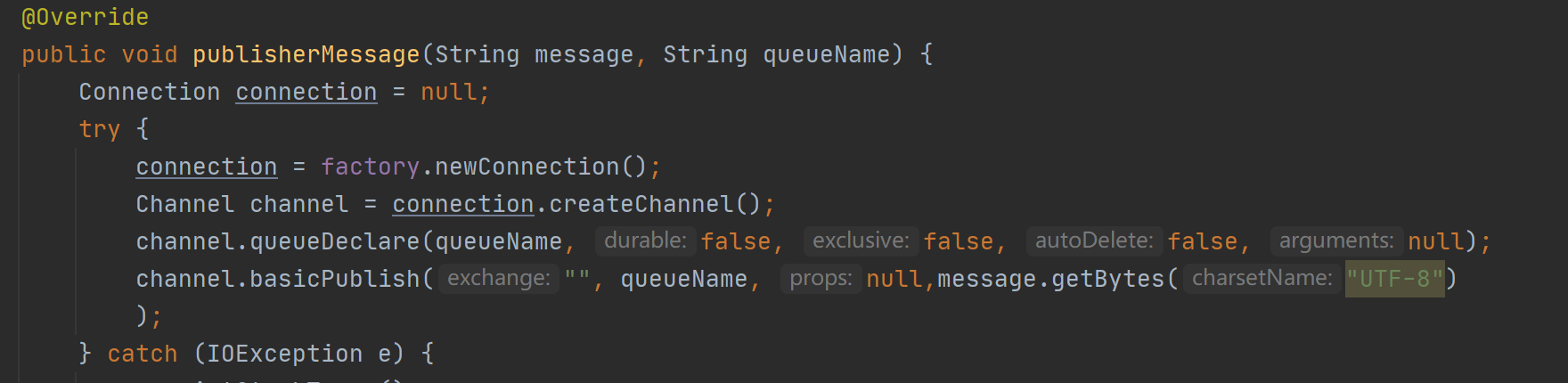
## RabbitMq基础篇

### 简单模式

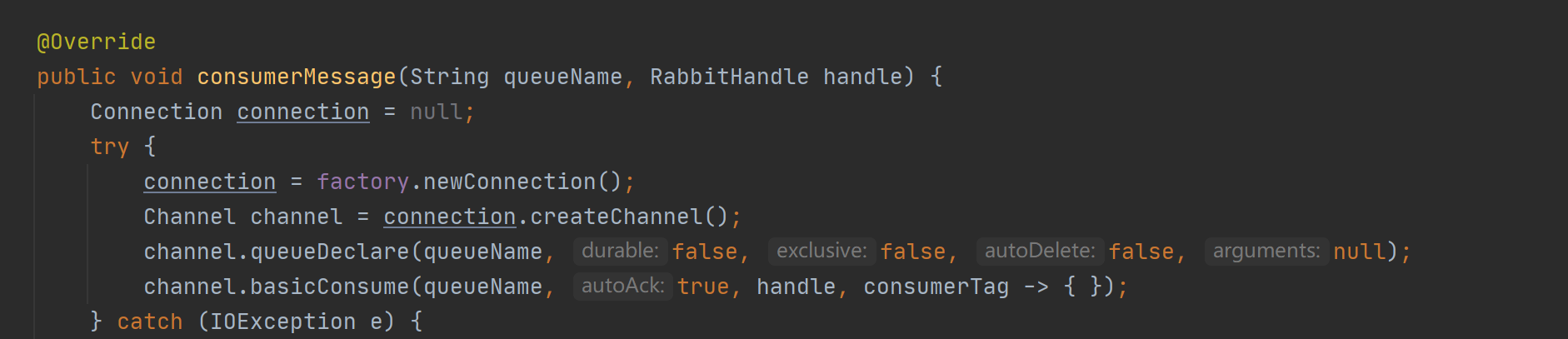
一个生成者一个消费组

IMG_256

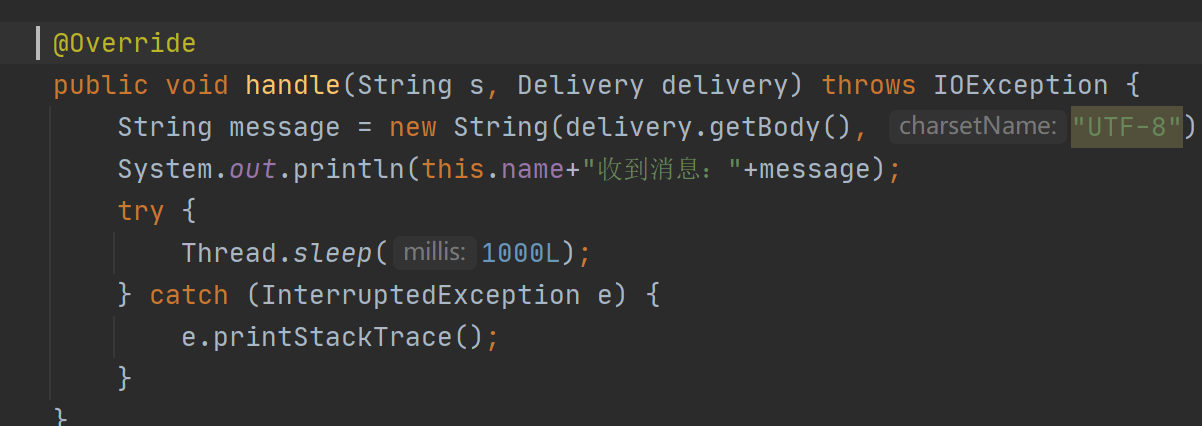
生成者：创建一个链接，一个channel，声明一个队列



消费组代码：

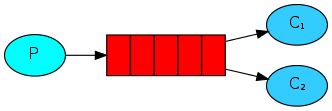


处理器代码：

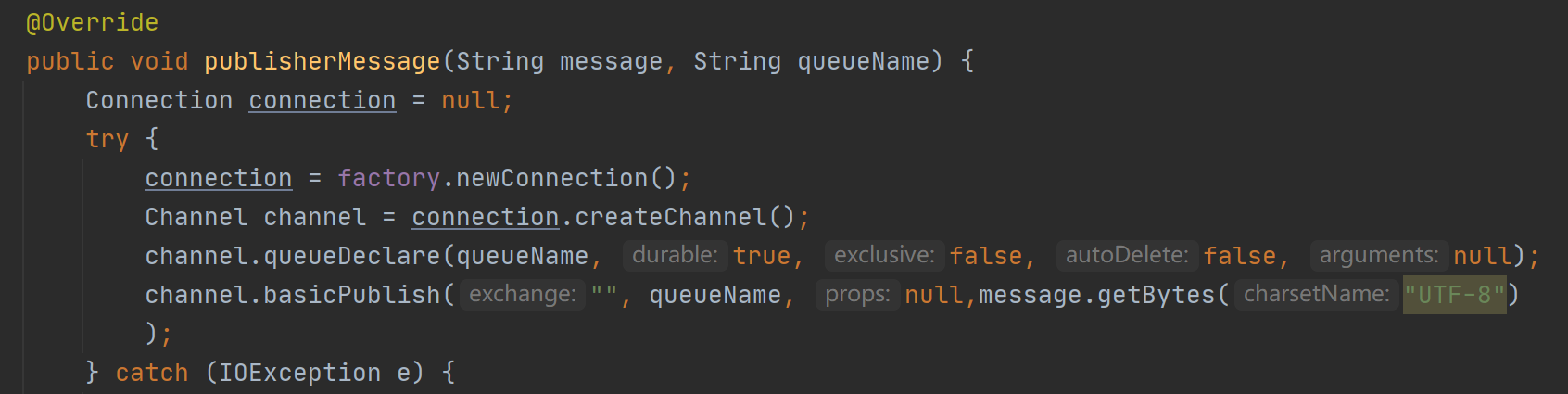


### 工作模式

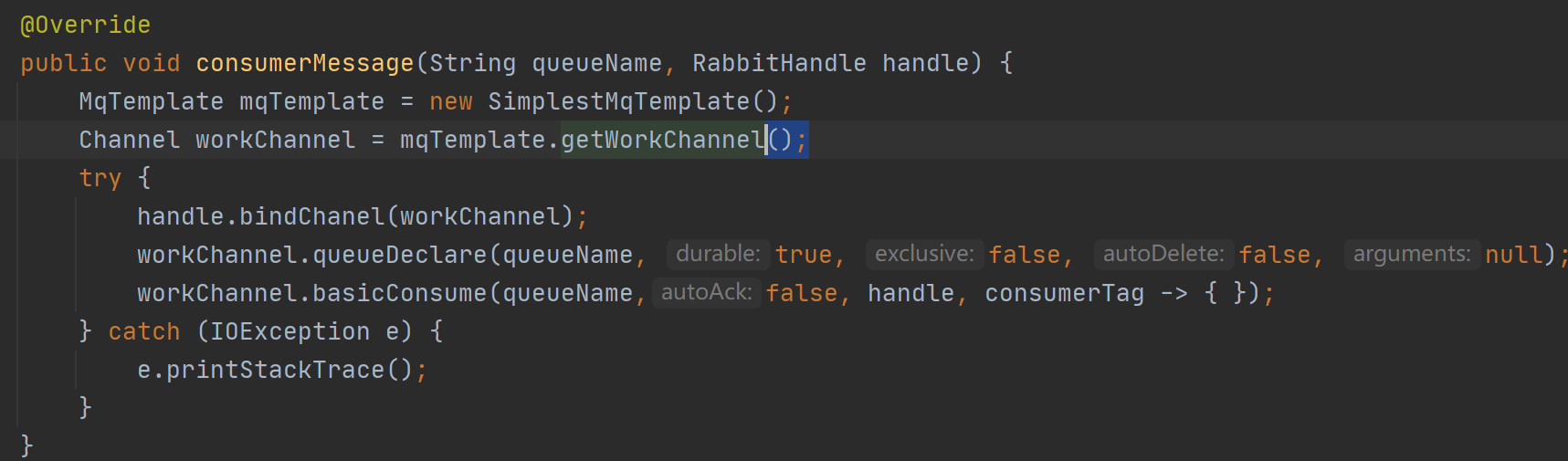
多个消费组共同消费



生产者发布消息代码：



消费组获取消息代码：

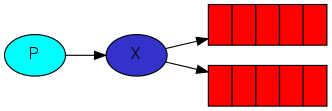


出来逻辑：

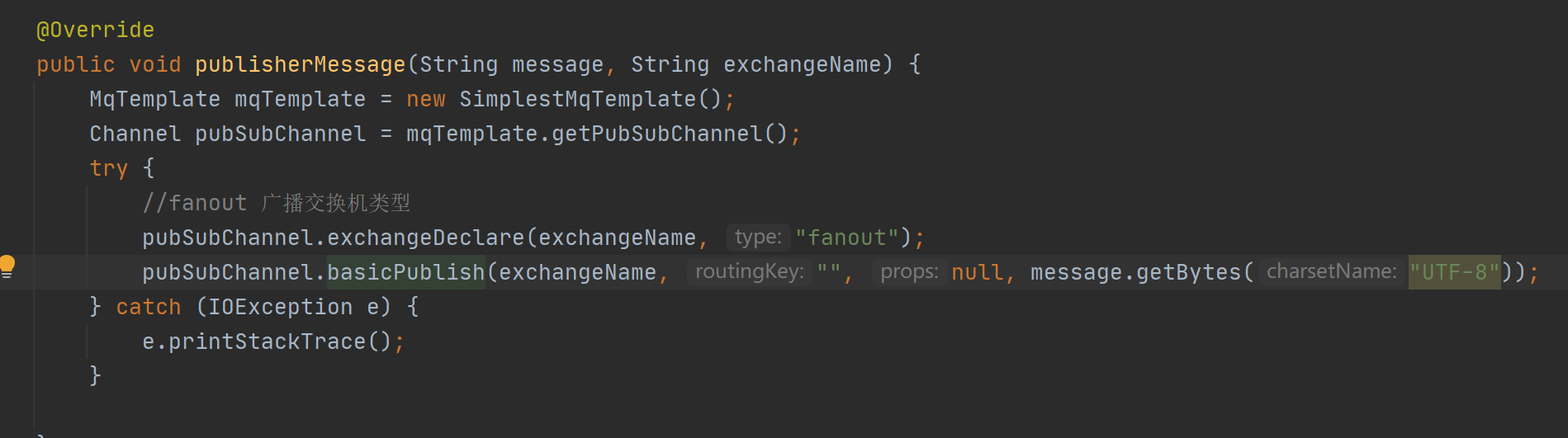


### 订阅模式

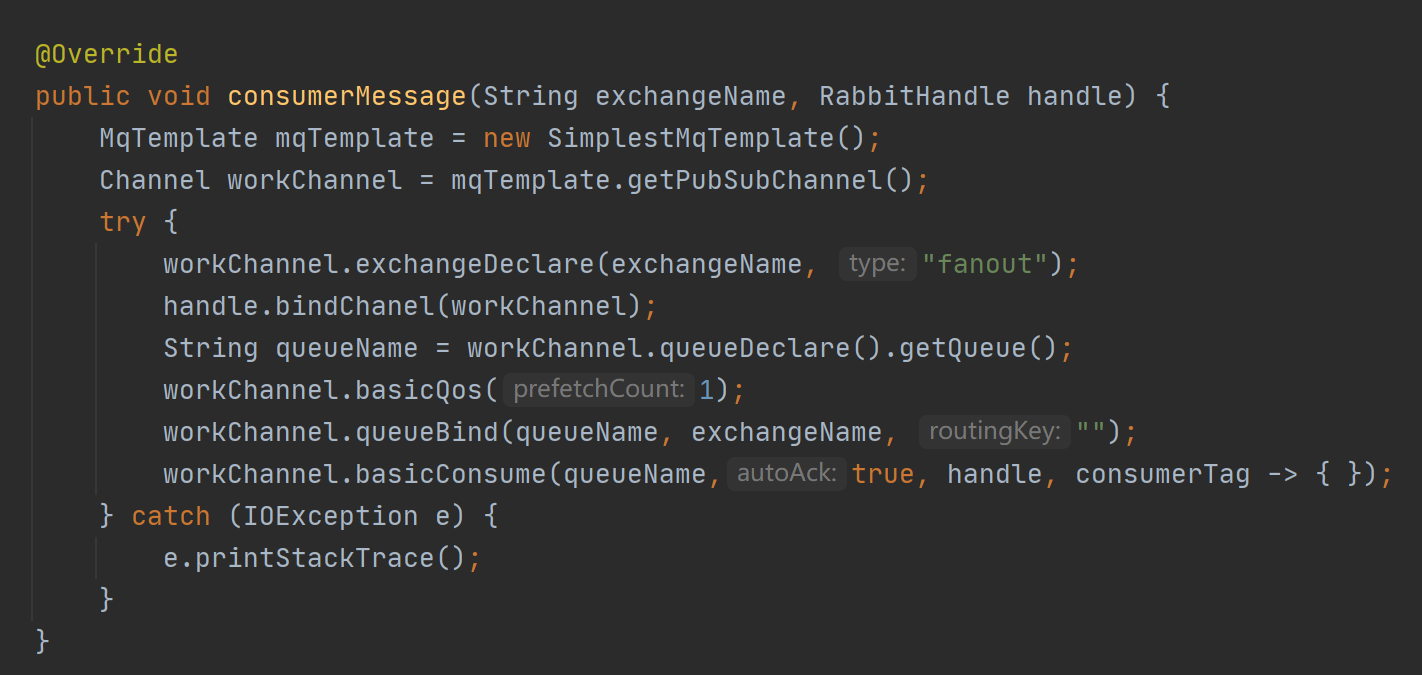
绑定交换机（fanou），绑定该交换机的所有队列都会收到该生产者的消息



生产者发布消息代码



消费组接受消息代码：

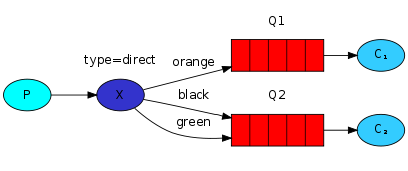


处理消息部分：

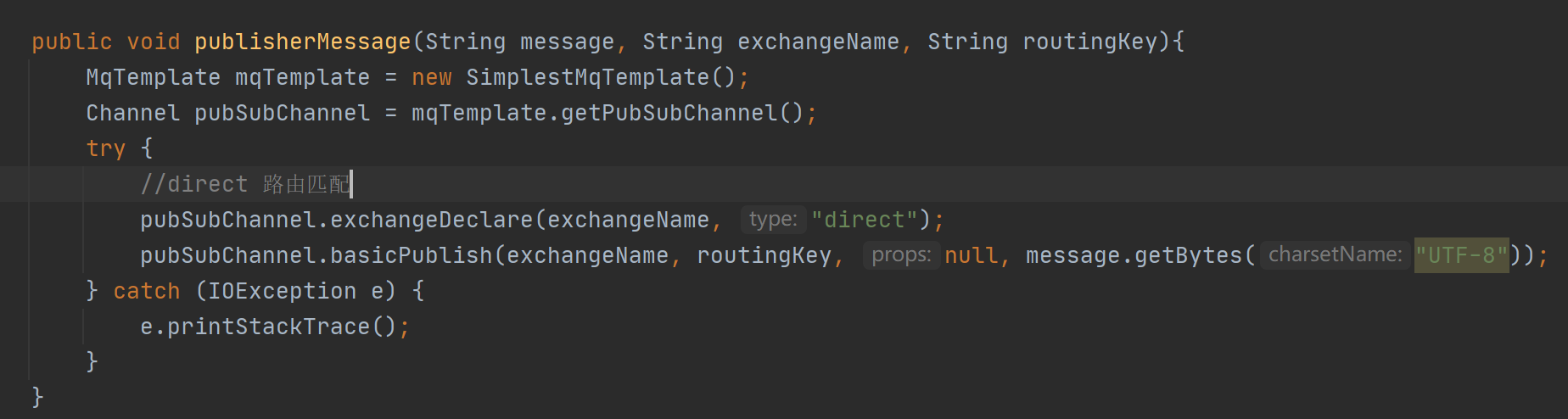


### 路由模式

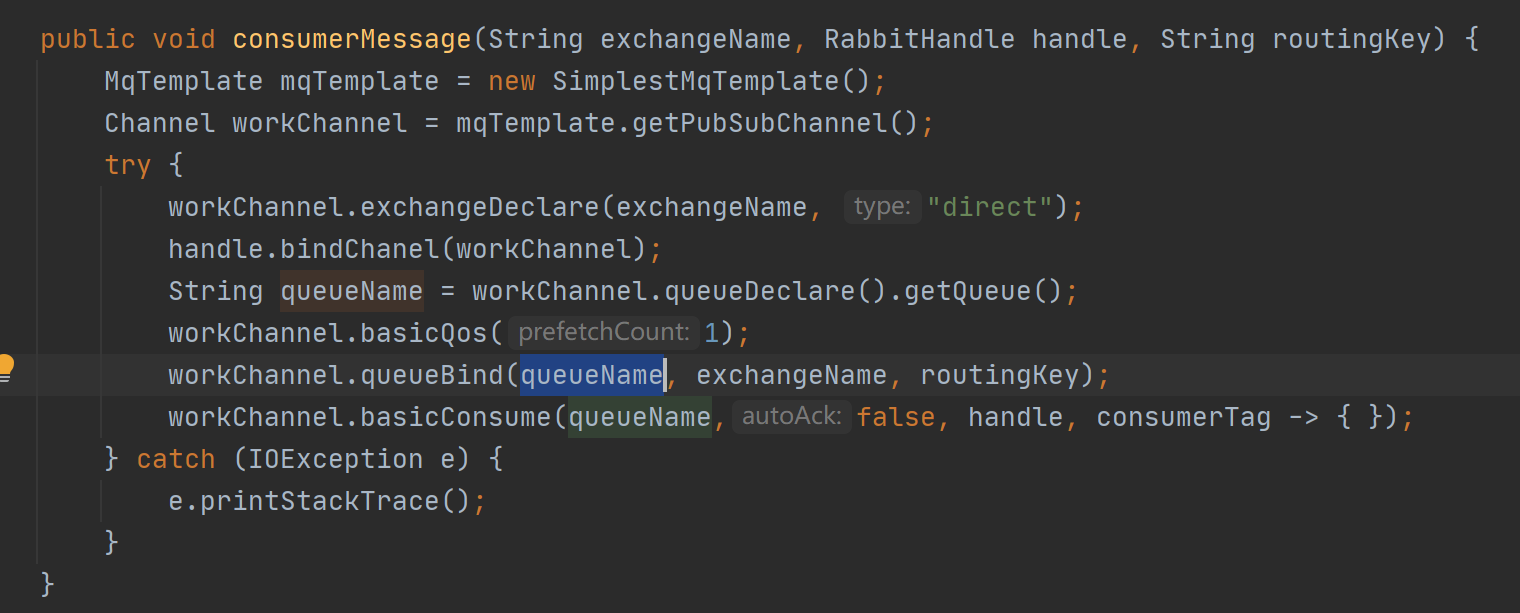
绑定交换机（direct），根据路由key(routingKey),匹配队列，符合的队列接受匹配的信息，该routingKey完全匹配不支持模糊



生产者发布消息代码：



消费组处理消息代码：

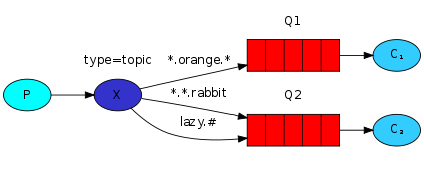


消费组处理消息代码：



### 主题（topic）模式

绑定交换机（topic），根据路由key(routingKey),匹配队列，符合的队列接受匹配的信息，该routingKey支持模糊,\*表示单个任意字符，#表示零个或者多个字符



此处逻辑与路由模式代码差不多，将交换机类型缓存topic类型，根据匹配符匹配路由key（routingKey）就可以

**注意：接交换机有4中上述只讲述了三种，其中有一种叫做headers，不需要路由key(routingKey)，通过属性去匹配，性能差，可以单独了解不做说明；这里还有第六章模式[Publisher Confirms](https://www.rabbitmq.com/tutorials/tutorial-seven-java.html)这里不介绍，后面高级篇会有说明**

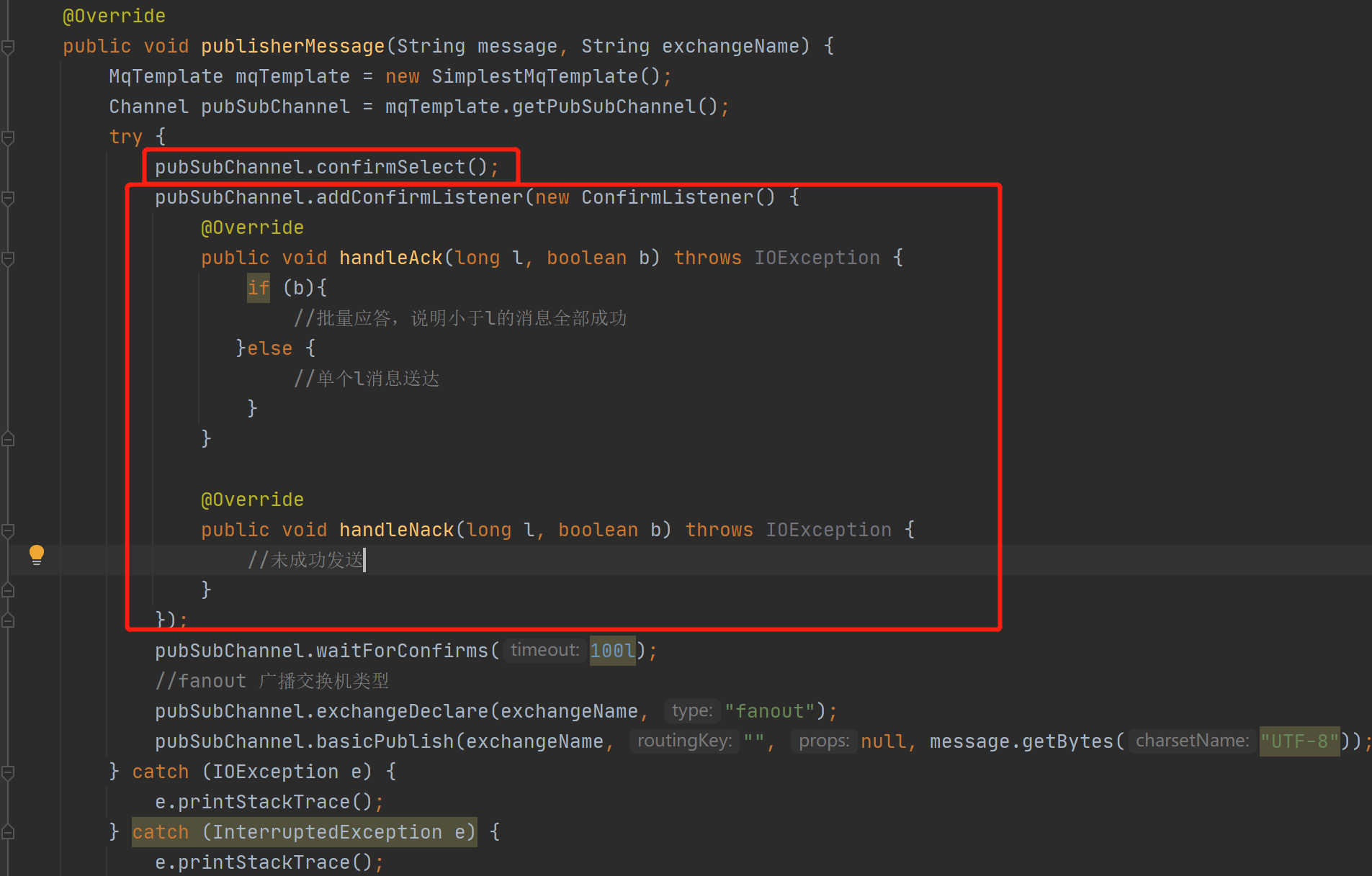
## 高级篇

### RabbitMQ消息的可靠传输

消息的丢失分为三部分：生产者未发送成功，消息队列丢失，消费组异常丢失

**生产者问题：**

Publisher comfirms模式（上文提到的）：channel进入confirm模式，所有在该信道上发布的消息都将会被指派一个唯一的ID（从1开始），一旦消息被投递到所有匹配的队列之后；rabbitMQ就会发送一个ACK给生产者（包含消息的唯一ID），这就使得生产者知道消息已经正确到达目的队列了；如果rabbitMQ没能处理该消息，则会发送一个Nack消息给你，你可以进行重试操作



transaction机制就是说：发送消息前，开启事务（channel.txSelect()）,然后发送消息，如果发送过程中出现什么异常，事务就会回滚（channel.txRollback()）,如果发送成功则提交事务（channel.txCommit()）。然而，这种方式有个缺点：吞吐量下降

**队列消息丢失问题：**

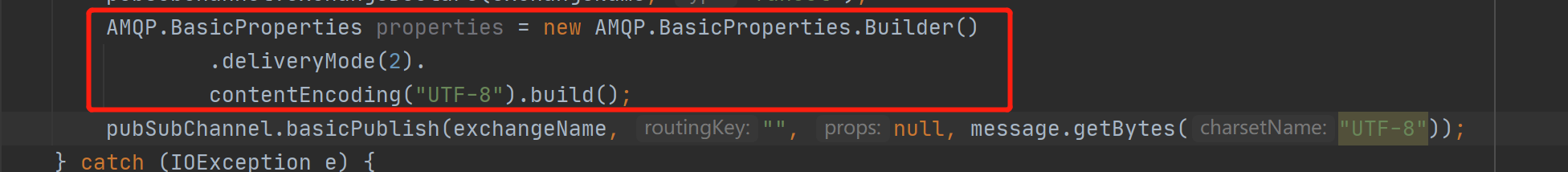
处理消息队列丢数据的情况，一般是开启持久化磁盘的配置。

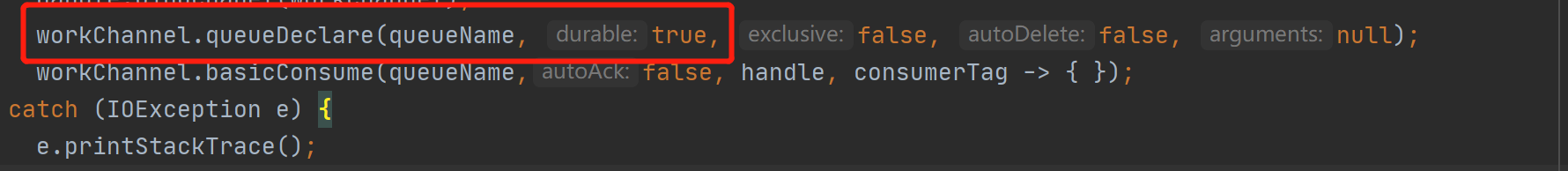
这个持久化配置可以和confirm机制配合使用，你可以在消息持久化磁盘后，再给生产者发送一个Ack信号。这样，如果消息持久化磁盘之前，rabbitMQ阵亡了，那么生产者收不到Ack信号，生产者会自动重发。

那么如何持久化呢？

这里顺便说一下吧，其实也很容易，就下面两步

1. 将queue的持久化标识durable设置为true,则代表是一个持久的队列
2. 发送消息的时候将deliveryMode=2





**消费者丢失消息：**

消费者丢数据一般是因为采用了自动确认消息模式，改为手动确认消息即可！

消费者在收到消息之后，处理消息之前，会自动回复RabbitMQ已收到消息；

如果这时处理消息失败，就会丢失该消息；

解决方案：处理消息成功后，手动回复确认消息。

### RabbitMQ重复消费的问题

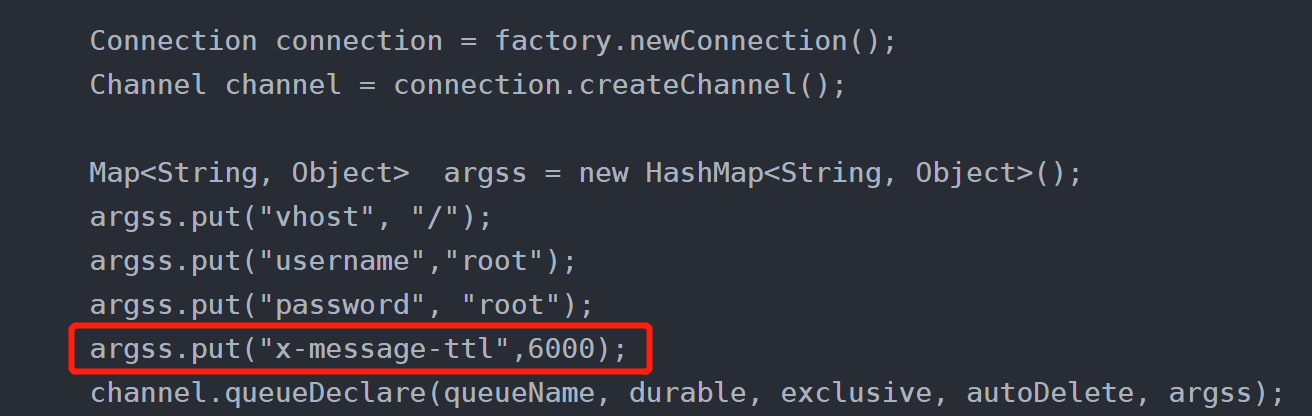
因为网络传输等等故障，确认信息没有传送到消息队列，导致消息队列不知道自己已经消费过该消息了，再次将消息分发给其他的消费者。

针对以上问题，一个解决思路是：保证消息的唯一性，就算是多次传输，不要让消息的多次消费带来影响；保证消息等幂性；

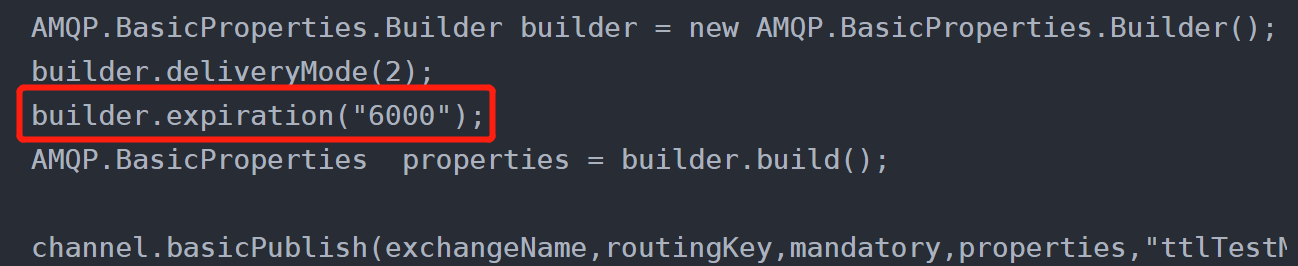
比如：在写入消息队列的数据做唯一标示，消费消息时，根据唯一标识判断是否消费过

### RabbitMQ之TTL（Time-To-Live 过期时间）

1. 针对队列设置过期时间：



1. 针对消息设置过期时间：



### RabbitMQ之死信队列

### RabbitMQ延时队列