JMS是java的消息服务，JMS的客户端之间可以通过JMS服务进行异步的消息传输

* 1. P2P模式图 
  2. 涉及到的概念
     1. 消息队列（Queue）
     2. 发送者(Sender)
     3. 接收者(Receiver)
     4. 每个消息都被发送到一个特定的队列，接收者从队列中获取消息。队列保留着消息，直到他们被消费或超时。
  3. P2P的特点
     1. 每个消息只有一个消费者（Consumer）(即一旦被消费，消息就不再在消息队列中)
     2. 发送者和接收者之间在时间上没有依赖性，也就是说当发送者发送了消息之后，不管接收者有没有正在运行，它不会影响到消息被发送到队列
     3. 接收者在成功接收消息之后需向队列应答成功

如果你希望发送的每个消息都应该被成功处理的话，那么你需要P2P模式。

1. Pub/Sub
   1. Pub/Sub模式图 
   2. 涉及到的概念
      1. 主题（Topic）
      2. 发布者（Publisher）
      3. 订阅者（Subscriber）   
         客户端将消息发送到主题。多个发布者将消息发送到Topic,系统将这些消息传递给多个订阅者。
   3. Pub/Sub的特点
      1. 每个消息可以有多个消费者
      2. 发布者和订阅者之间有时间上的依赖性。针对某个主题（Topic）的订阅者，它必须创建一个订阅者之后，才能消费发布者的消息，而且为了消费消息，订阅者必须保持运行的状态。
      3. 为了缓和这样严格的时间相关性，JMS允许订阅者创建一个可持久化的订阅。这样，即使订阅者没有被激活（运行），它也能接收到发布者的消息。

如果你希望发送的消息可以不被做任何处理、或者被一个消息者处理、或者可以被多个消费者处理的话，那么可以采用Pub/Sub模型

消息的消费   
在JMS中，消息的产生和消息是异步的。对于消费来说，JMS的消息者可以通过两种方式来消费消息。   
○ 同步   
订阅者或接收者调用receive方法来接收消息，receive方法在能够接收到消息之前（或超时之前）将一直阻塞   
○ 异步   
订阅者或接收者可以注册为一个消息监听器。当消息到达之后，系统自动调用监听器的onMessage方法