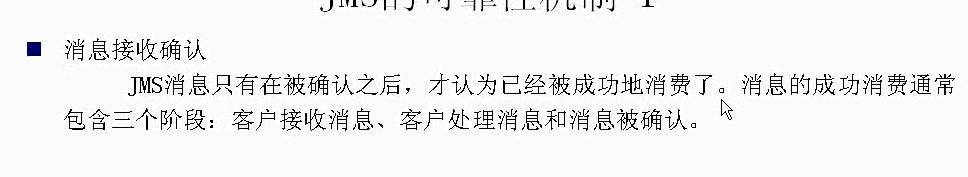
# 1、jms可靠性机制



## 1、事物的使用,查看客户的（消费者）使用消息情况

### 1、不启用事物的时候（每次都会被消费）

|  |
| --- |
| // 获取操作连接  session = connection.createSession(Boolean.*false*, Session.*AUTO\_ACKNOWLEDGE*); |

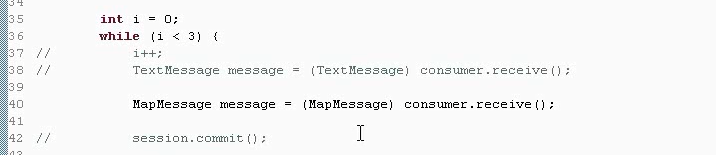
#### 解释：当客户端消费了消息之后，再次消费则不会显示之前的消息

### 2、当使用事物的时候

|  |
| --- |
| // 获取操作连接  session = connection.createSession(Boolean.*true*, Session.*AUTO\_ACKNOWLEDGE*); |

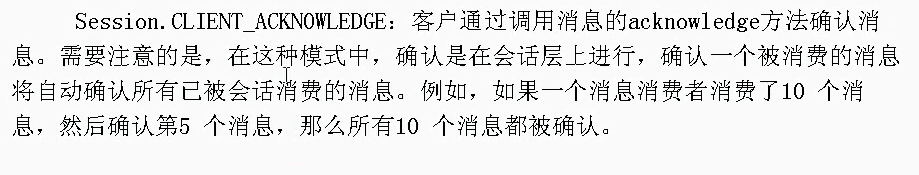
#### 解释：

#### 1、当使用事物的时候，如果commit, 没有使用，则表示没有消费，则客户端（消费者）每次都能够访问到之前所访问到的数据



#### 2、如果每次commit都在使用，则客户端（消费者）每次都能不能够访问之前的数据了，因为已经被消费了。

# 2、使用 Session.*CLIENT\_ACKNOWLEDGE，*不启用事物（1中为开启事物，自动签收），客户端签收



## 1、 message.acknowledge(); 使用时，如果客户端已经消费了消息，则使用该方法，对message进行签收，则会签收所有被使用的消息，再次调用消费者，则不会收到数据

## 2、 message.acknowledge(); 不使用的时候，则所有消费者使用的过的数据，再次调用还会收到数据，因为客户端没有签收。

|  |
| --- |
| **final** Session session = connection.createSession(Boolean.*TRUE*,  Session.*CLIENT\_ACKNOWLEDGE*);  Destination destination = session.createQueue("my-queue3");  ActiveMQMessageConsumer consumer = (ActiveMQMessageConsumer)session.createConsumer(destination);    **int** i = 0;  **while** (i < 3) {  MapMessage message = (MapMessage) consumer.receive();    **if**(i==2){  message.acknowledge();  } |

# 3、