1. MOM (Message Oriented Middleware) - это промежуточное ПО, используемое для обмена сообщениями между приложениями. JMS (Java Message Service) - это API для работы с MOM на языке Java.
2. Принцип работы MOM основан на использовании посредников (брокеров), которые обрабатывают и направляют сообщения между приложениями. Существуют две основные архитектуры MOM: point-to-point (P2P) и publish/subscribe (pub/sub).
3. В режиме point-to-point (P2P) каждое сообщение адресуется конкретному получателю, который получает сообщение и удаляет его из очереди. Таким образом, каждое сообщение может быть обработано только одним получателем.
4. В режиме publish/subscribe (pub/sub) сообщения публикуются на тему (topic), которая может быть подписана несколькими получателями. Каждый подписчик получает копию сообщения, которое может быть обработано независимо от других получателей.
5. Для запуска и настройки брокеров в Open MQ необходимо установить и настроить сервер Open MQ, создать конфигурационный файл, определить настройки подключения к брокеру и запустить брокер.
6. Администрируемые объекты в MOM - это объекты, используемые для настройки и управления брокером и его компонентами. Они могут включать в себя очереди, темы, подписчиков и другие компоненты.
7. JMS API включает в себя классы и интерфейсы для создания, отправки и получения сообщений, управления очередями и темами, управления сессиями и транзакциями, и т.д.
8. Администрируемые объекты JMS - это объекты, которые могут быть созданы и настроены администратором системы для управления компонентами JMS, такими как очереди, темы, подписчики и другие.
9. Структура сообщения JMS включает заголовок, свойства и тело сообщения.
10. Типы сообщений JMS могут включать в себя текстовые, бинарные, объектные, карты свойств и т.д.
11. При получении сообщений синхронно приложение блокируется до тех пор, пока не будет получено сообщение. При получении сообщений асинхронно приложение продолжает работу, а сообщения обрабатываются в отдельном потоке.
12. Алгоритм обмена сообщениями в JMS выглядит следующим образом:

* Приложение отправляет сообщение в очередь или топик брокера сообщений.
* Брокер сообщений принимает сообщение и сохраняет его в соответствующей очереди или топике.
* Получатель извлекает сообщение из очереди или подписывается на топик и ожидает получения сообщения.
* Когда новое сообщение появляется в очереди или топике, брокер сообщает об этом получателю, и тот получает сообщение.

1. При создании подписки на топик в JMS можно выбрать долговременную или недолговременную подписку. При долговременной подписке получатель будет получать все сообщения, которые были отправлены в топик после того, как получатель подписался на него. Даже если получатель был оффлайн, когда сообщение было отправлено, он все равно получит это сообщение, когда подключится к топику. При недолговременной подписке получатель будет получать только те сообщения, которые были отправлены в топик в момент, когда получатель был подключен к топику.
2. Модели подтверждения доставки сообщений в JMS бывают двух типов:

* AUTO\_ACKNOWLEDGE - автоматическое подтверждение доставки. После получения сообщения JMS-провайдер автоматически подтверждает его доставку.
* CLIENT\_ACKNOWLEDGE - подтверждение доставки вручную. Получатель сообщения должен явно подтвердить доставку сообщения, вызвав метод acknowledge() объекта Message.

1. Селекторы сообщений (фильтры) позволяют выбирать сообщения, которые соответствуют определенным критериям, в результате чего получатель получает только те сообщения, которые ему действительно нужны. Селекторы сообщений определяются при создании подписки или приемника сообщений и могут быть выражены на языке SQL-92.
2. MDB (Message-Driven Bean) - это компонент EJB, который слушает и обрабатывает сообщения, поступающие в очередь или топик брокера сообщений. Когда новое сообщение появляется в очереди или топике, MDB получает его и запускает метод обработки сообщения. MDB позволяет обрабатывать сообщения асинхронно и отвязывает приложение от конкретной реализации протокола обмена сообщениями.