1. Схема XSD. Простые и сложные типы. JAXB. Маршаллизация и демаршаллизация. Понятие XSL. XSL Transformation.

**Простые типы** – типы, которые не имеют атрибутов и дочерних элементов, они должы должны иметь простой тип данных.

Существуют стандартные простые типы: string, boolean, integer, float, ID, gYearи др.

**Сложные типы** – содержат в себе атрибуты и/или дочерние элементы. Создаются с помощью элемента complexType. Так же как и в простом типе атрибут name задает имя

**JAXB(Java Architecture for XML Binding)** – это мост между XML и Java, Java API для маршалинга объекта в XML и восстановление(демаршалинга) объекта из XML файла. Часто используется в веб-файлах для представления объекта в виде XML схемы и передачи ее по сети. + при обмене данных между системами на разныхязыках.

**Маршаллизация** — механизм преобразования данных из java-объектов в конкретное хранилище

**Демаршаллизация** — обратный процесс преобразования данных из внешних источников в структуру хранения, поддерживаемую виртуальной машиной

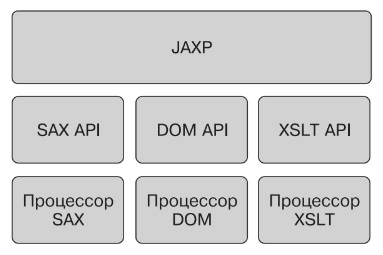
**XSL** (e**X**tensible **S**tylesheet **L**anguage) — семейство рекомендаций [консорциума W3C](https://ru.wikipedia.org/wiki/W3C) (**Консо́рциум Всеми́рной паути́ны**), описывающее языки преобразования и визуализации [XML](https://ru.wikipedia.org/wiki/XML) -документов. Состоит из трех частей:

* XSL Transformations ([XSLT](https://ru.wikipedia.org/wiki/XSLT)) — язык преобразований XML-документов.
* XSL Formatting Objects ([XSL-FO](https://ru.wikipedia.org/wiki/XSL-FO)) — [язык разметки](https://ru.wikipedia.org/wiki/%D0%AF%D0%B7%D1%8B%D0%BA_%D1%80%D0%B0%D0%B7%D0%BC%D0%B5%D1%82%D0%BA%D0%B8) типографских макетов и иных предпечатных материалов.
* [XPath](https://ru.wikipedia.org/wiki/XPath) — язык путей и выражений, используемый в XSLT для доступа к отдельным частям XML-документа.

**XSLT** – расширяемый язык преобразования листов стилей. Служит транслятором, с помощью которого можно свободно модифицировать исходный текст. Широкая область применения.

1. JAXP. Стратегии обработки XML документов: DOM, SAX, StAX – сравнение записи, чтения и поиска. Запись и чтение json

JAXP – набор абстрактных API, упрощающих обработку XML данных с помощью программ, написанных на Java. Усиливает стандартные анализаторы SAX, DOM. Поддерживает преобразование расширяемого языка таблиц стилей XSLT. Дает возможность проводить лексический разбор и модификацию документов XML независимо от конкретной реализации обработки XML.



DOM(Document Object Model) -объектная модель документов, платформенно-независимый программный интерфейс, позволяющий программами скриптам управлять содержимым документов XML и HTML, а также изменять их структуру и оформление. Не накладывает ограничение на структуру документа. Любой документ может быть представлен в виде дерева узлов, каждый узел которого содержит элемент, атрибут, текстовый, графический или любой другой объект. Узлы связаны отношением родитель – потомок.

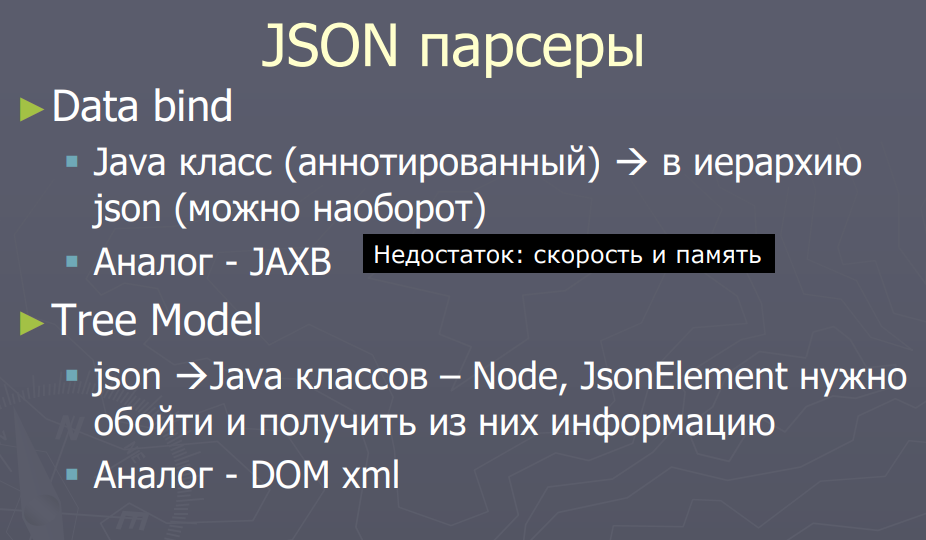
SAX(Simple API for XML) – базируется на модели последовательной одноразовой обработки и не создает внутренних деревьев. При прохождении по XML вызывает соотв. Методы у классов, реализующих интерфейсы, предоставляемые SAX-парсером.

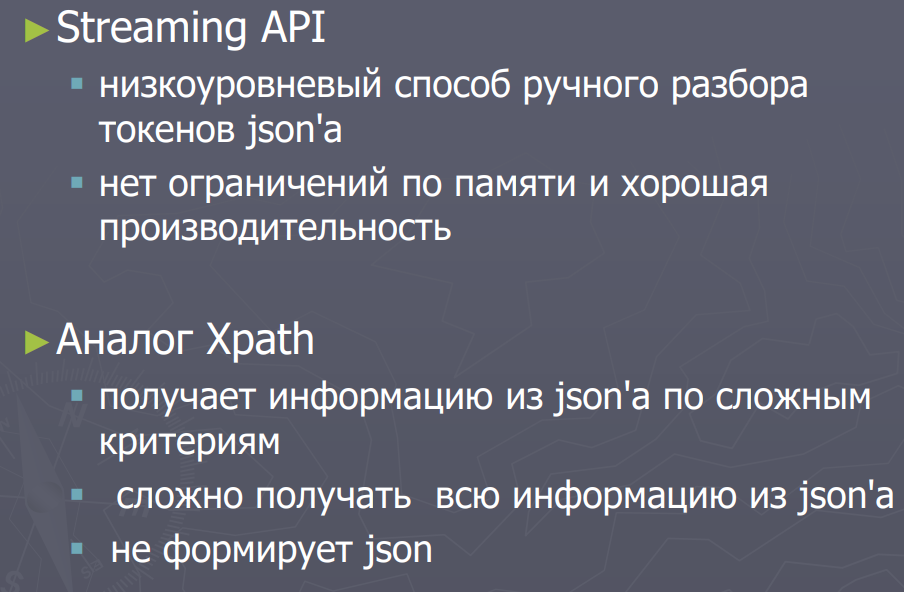
StAX(Streaming API for XML) – не создает дерево объектов в памяти, но в отличие от SAX-парсера, ха переход от одной вершины к другой отвечает приложение, которое запускает разбор документа.

**Что и когда применяется**

DOM-анализаторы следует использовать тогда, когда нужно знать структуру документа и может понадобиться изменять эту структуру либо использовать информацию из XML-документа несколько раз.

SAX/StAX-анализаторы используются тогда, когда нужно извлечь информацию о нескольких элементах из XML-файла либо когда информация из документа нужна только один раз.





**45. JMS. Понятие MOM. Основные понятие и режимы работы.**

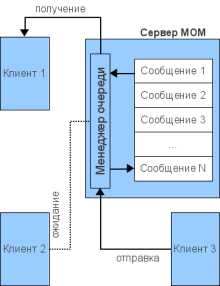
JMS (Java Message Service) является стандартом обмена сообщениями между приложениями. Java приложения, выполненные по технологии Java SE (standalone) или Java EE (WEB) могут создавать, отправлять и получать JMS сообщения. Программное обеспечение, используемое для передачи сообщений между приложениями по стандарту **JMS**, формирует очереди сообщений queue.

**Отправка сообщений** java-приложениями выполняется в асинхронном режиме, т.е. процедура не ждет ответа от получателя. В качестве получателей сообщений, организующих очереди (**queue**), может быть либо программное обеспечение типа WebSphere MQ, связывающее приложения через канал обмена сообщениями, либо WEB-контейнер типа JBoss, GlassFish, обеспечивающих обмен сообщениями между приложениями контейнера, либо по каналам Интернета с использованием [JNDI](http://java-online.ru/jndi.xhtml).

**Получение сообщений** java-приложением осуществляется либо по «подписке» (технология «издатель-подписчик»), либо из очереди («точка-точка»). Для получения сообщений по подписке необходимо должным образом подключиться к соответствующей очереди. При появлении в очереди сообщений они отправляются всем подписчикам. В модели «точка-точка» для получения сообщений необходимо периодически подключаться к соответствующей очереди и читать в ней сообщения.

**Сервисы обработки сообщений** (MOM — message-oriented middleware) —это системы, как правило асинхронные, в которых взаимодействие между клиентом и сервером основано на обмене сообщениями. Сообщения — это текстовые блоки, состоящие из управляющих команд и передаваемых данных. Для передачи сообщений используются байт-ориентированные протоколы, такие как [HTTP](http://www.4stud.info/web-programming/protocol-http.html), POP/SMTP и т.п.

Обмен сообщениями реализуется через API системы MOM. Запросы сервисов ставятся в очередь сообщений и обрабатываются в соответствии с приоритетами и доступностью ресурсов ([рис. 3](http://www.4stud.info/networking/lecture6.html#mom-picture)). Приоритеты сообщений позволяют обеспечить первоочередную доставку важных сообщений, а отложенная доставка осуществляется либо по расписанию, либо при появлении адресата в сети. Ответы сервера содержат информацию об успешном или неуспешном выполнении операции.

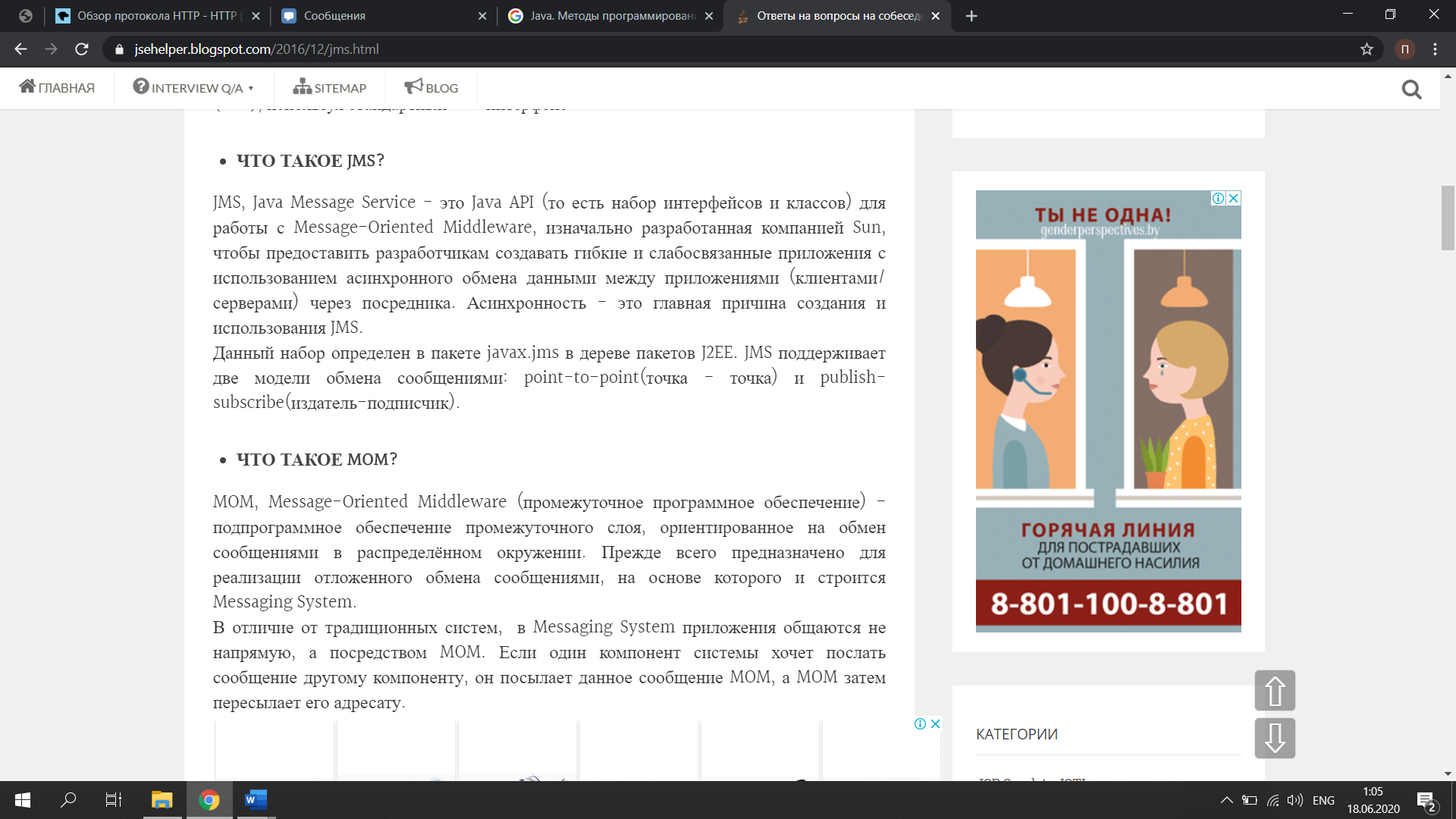


* надежная доставка сообщений (reliable message delivery) — система MOM гарантирует, что в процессе обмена ни одно сообщение не будет утеряно;
* гарантированная доставка сообщений (guaranteed message delivery) — сообщение доставляется адресату немедленно или через заданный промежуток времени, не превышающий определенного значения (в случае, если сеть в данный момент не доступна);
* застрахованная доставка сообщений (assured message delivery) — каждое сообщение доставляется только один раз.

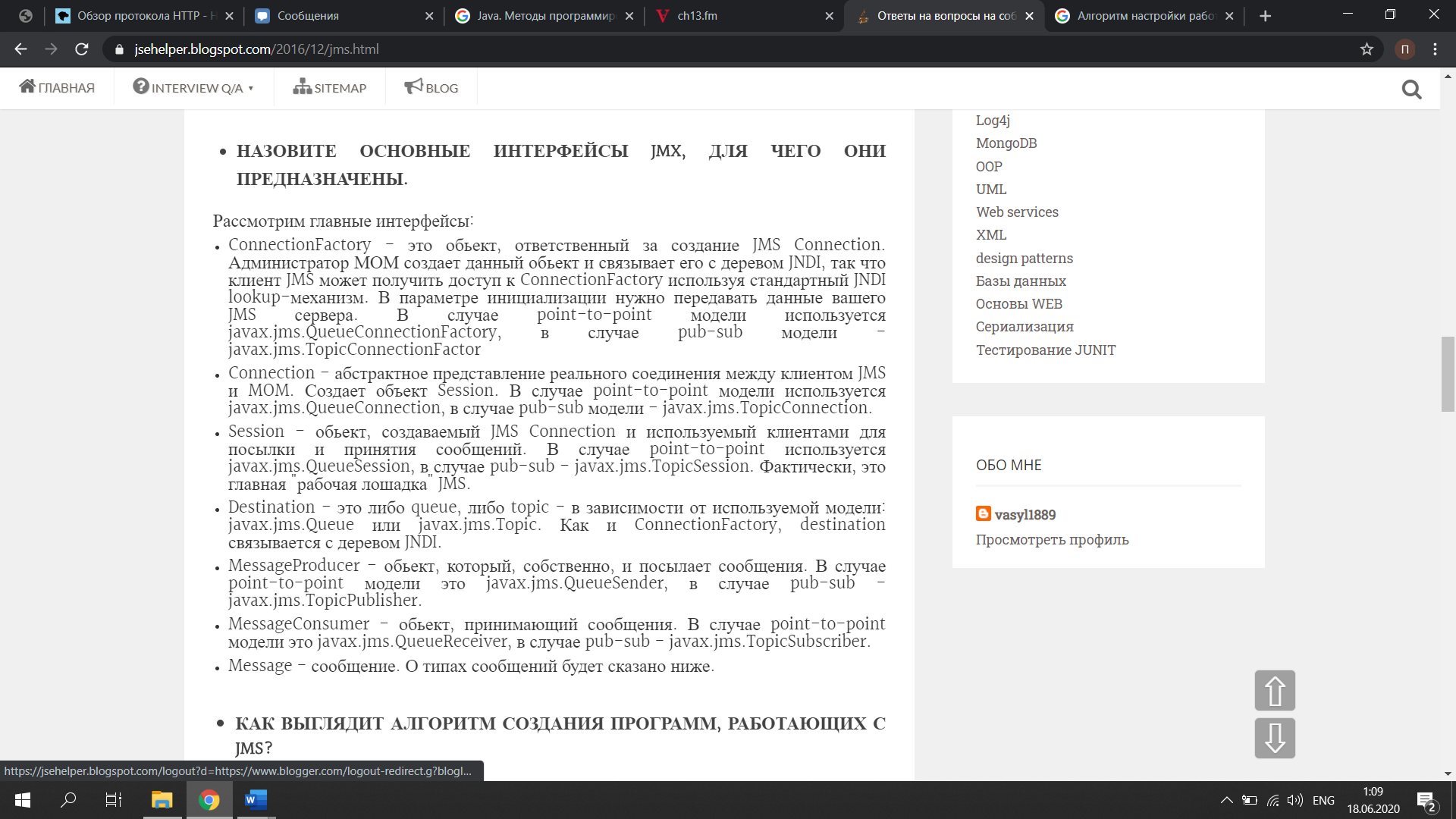
Помимо приведенной, можно сказать классической, схемы с очередями, разработаны и используются сервисы MOM с непосредственной передачей сообщений и на основе подписки.

Системы с непосредственной передачей сообщений (message passing) используют логическое сетевое соединение для обмена сообщениями между взаимодействующими приложениями. Эта схема удобна в тех случаях, когда клиенты и серверы сообщений используются в сильно связанной сетевой инфраструктуре и синхронизированы по времени.

Сервисы MOM, обслуживающие клиентов по подписке/публикации (publish&subscribe) работают по принципу, напоминающему почтовую рассылку: одно приложение публикует информацию в сети, а другие подписываются на эту публикацию для получения необходимых данных. Взаимодействующие таким способом приложения полностью независимы друг от друга, что представляет возможности динамической реконфигурации всей распределенной системы.



46. JMS. Особенности работы P2P, pub/sub. Программные интерфейсы.



47. Алгоритм настройки работы клиентов JMS P2P

48. Алгоритм настройки работы клиентов JMS pub/sub

**49. JMS. Способы обеспечения надежности доставки сообщений**

JMS гарантирует доставку сообщения по своей природе, если сообщение отправлено, то оно будет доставлено потребителю, если есть, независимо от того, что произойдет, MOM предназначена для обеспечения этого факта. В любом случае, доставка не требуется, значит, обрабатывается.

Надежность обеспечивается с помощью различных механизмов:

* первым является сохранение сообщения в очереди (очередь И сообщение должно быть помечено как постоянное, которое является значением по умолчанию), которые гарантируют, что сообщение не будет потеряно в случае прерывания системы.
* тогда у вас есть подтверждение и политика повтора, сообщение будет храниться в очереди до тех пор, пока потребитель не подтвердит это и в случае транзакционного сеанса не будет повторно доставлен до тех пор, пока потребитель не обработает сообщение или не будет достигнута максимальная повторная попытка. После этого сообщение с ошибкой может быть перенаправлено в очередь с мертвой буквой для анализа.

