**Лабораторная 5**

**Развертывание и работа с Message-oriented middleware**

Используя RabbitMQ продемонстрировать асинхронное взаимодействие нескольких систем/подсистем (приложений) на основе очереди.

1. Реализовать два варианта очереди: [Producer/Consumer](http://www.enterpriseintegrationpatterns.com/patterns/messaging/CompetingConsumers.html) (Point-to-Point) и [Publish/Subscribe](http://www.enterpriseintegrationpatterns.com/patterns/messaging/PublishSubscribeChannel.html) (Topic)
2. Реализовать следующую логику - клиент отправляет сообщение в очередь, один из консьюмеров его вычитывает, модифицирует и кладет в ответную очередь клиенту, который выполнял отправку, клиент вычитывает ответ и отображает его
3. Показать и настроить и варианты предоставляемые MOM, связанные с:
   * подтверждением доставки/обработки сообщений клиентом (*Message Acknowledgment*),
   * сохранности очереди сообщений (Message Persistence - *Durable queue*),
   * время пребывания сообщения в очереди (*Message TTL),*
   * максимальная длина очереди (*Max length)* (что происходит с сообщениями когда очередь заполнена)
4. Для варианта Producer/Consumer показать случай, когда Consumer берет из очереди сообщение на обработку, но не может его обработать: падает/не возвращает Ack/возвращает негативный Ack. Показать, будет ли при этом данное необработанное сообщение взято на обработку другим Consumer или окажется потерянным.

Материалы по RabbitMQ:

* <https://www.rabbitmq.com/getstarted.html>
* <https://www.cloudamqp.com/blog/2015-05-18-part1-rabbitmq-for-beginners-what-is-rabbitmq.html>