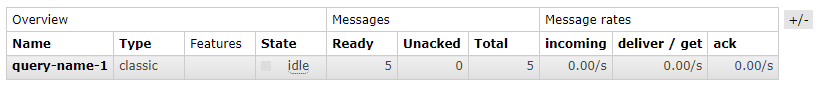
Очереди, AMQP, задания для **функционального** тестирования. Все описания в шапке классов, которые надо запустить. Список классов ниже.

1. com.epam.queue.tasks.rabbit.simple.tasks.FuncBasicDisconnectTask

Функциональне тестирование. Задание 1.

Цель: знакомство с очередями и основными инструментами мониторинга.  
  
1) запустите брокер в докере:  
 docker run -d --name rabbitmq -p 5672:5672 -p 15672:15672 rabbitmq:3-management  
  
2) Запустите класс, дождитесь надписи **'Execution finished'** в стандартном выводе.  
  
3) Проверьте состояние очереди **'query-name-1'**. Сколько сообщений ушло, сколько осталось (если осталось).

Для выполнения данного задания понадобится доступ в консоль:  
 docker container exec -it rabbitmq /bin/sh  
  
4) Перезапустите контейнер с брокером. Проверьте состояние очереди. Сохранилось ли ее состояние? Почему сохранилось/не сохранилось? 

Text

Description automatically generated

5 доставлено 5 осталось

Graphical user interface, text

Description automatically generated

Состояние не сохранилась, non Durable очереди не переживают рестарт брокера

1. com.epam.queue.tasks.rabbit.queue.tasks.FuncPersistDisconnectTask

Функциональне тестирование. Задание 2.  
  
Цель: знакомство с ack.  
  
1) запустите брокер в докере:  
 docker run -d --name rabbitmq -p 5672:5672 -p 15672:15672 rabbitmq:3-management  
  
2) Запустите класс, дождитесь надписи **'Execution finished'** в стандартном выводе.  
  
3) Проверьте состояние очереди **'pers-query-1'**. Сколько сообщений ушло, сколько осталось (если осталось).

Для выполнения данного задания понадобится доступ в консоль:  
 docker container exec -it rabbitmq /bin/sh  
  
4) Перезапустите контейнер с брокером. Проверьте состояние очереди. Сохранилось ли ее состояние? Почему  
 сохранилось/не сохранилось?

40 доставлено, 10 осталось

Text

Description automatically generated with medium confidence

Состояние сохранилось, очередь durable

Text

Description automatically generated

1. com.epam.queue.tasks.rabbit.queue.tasks.FuncRoundRobinTask

Функциональне тестирование. Задание 3.  
  
Цель: знакомство с round-robin.  
  
1) запустите брокер в докере:  
 docker run -d --name rabbitmq -p 5672:5672 -p 15672:15672 rabbitmq:3-management  
  
2) Запустите класс, дождитесь надписи **'Execution finished'** в стандартном выводе.  
  
3) Посмотрите на сообщения в стандартном выводе.  
 а) Какой получатель (id) начал обработку (прием) сообщений?  
 б) Какой получатель (id) завершил обработку (прием) сообщений?  
 в) Почему какой-то промежуток времени сообщения никем не обрабатывались?

**Начал [rcvId = 37]**

**Завершил [rcvId = 89]**

**Пауза была в момент когда первый консьюмер отключился, а второй ещё не подключился.**

Очереди, AMQP, задания для **автоматизированного** тестирования. Все описания в шапке классов, которые надо запустить. Список классов ниже.

1. com.epam.queue.task.ImplementReceiverTaskIT

Очереди для автоматизаторов. AMQP. Задание 1.  
  
Задача: реализовать интерфейс {@link com.epam.queue.tasks.interfaces.Receiver} и передать объект в  
{@link ImplementReceiverTask#applyReceiver(Receiver)} (смотрите метод {@link #testReceiver()}  
  
Если тест прошел, то значит вы все сделали правильно.  
  
Материалы, которые могут помочь в этом нелегком деле:  
<a href=**"https://www.rabbitmq.com/tutorials/tutorial-one-java.html"**>

+

Text

Description automatically generated

1. com.epam.queue.task.FixReceiverTaskIT

Очереди для автоматизаторов. AMQP. Задание 2.  
  
Класс {@link PersistReceiverToFix} реализует интерфейс {@link com.epam.queue.tasks.interfaces.Receiver}.  
  
Задача: следать так, чтобы тест {@link #testAckReceiver()} проходил. Для этого нужно пофиксить  
{@link PersistReceiverToFix#receive()}.  
  
Материалы, которые могут помочь в этом еще более нелегком деле:  
<a href=**"https://www.rabbitmq.com/tutorials/tutorial-two-java.html"**>  
Обратить внимание на ack (что это и зачем).

+

boolean autoAck = false;

Text

Description automatically generated

1. com.epam.queue.task.RoundRobinWorkflowTaskIT

Очереди для автоматизаторов. AMQP. Задание 3.

Запускается тест, начинается отправка и прием сообщений, но потом происходит что-то ужасное и тест падает.  
  
Задача: следать так, чтобы тест {@link #testRoundRobin()} проходил. Для этого нужно положить что-то в  
{@link RoundRobinWorkflowTask#addMagic(Object)}.  
  
Материалы, которые могут помочь в этом максимально нелегком деле:  
<a href=**"https://www.rabbitmq.com/tutorials/tutorial-two-java.html"**>

+  
Text

Description automatically generated