Очереди, AMQP, задания для **функционального** тестирования. Все описания в шапке классов (если интересно).

Все jar файлы запускаются командой: *java –jar some.jar*

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 queue-rabbit-disconnect-task.jar, дождитесь надписи **'Execution finished'** в стандартном выводе.

Text

Description automatically generated

3) Проверьте состояние очереди **'query-name-1'**. Сколько сообщений ушло, сколько осталось (если осталось).

5 ушло, 5 осталось

Для выполнения данного задания понадобится доступ в консоль:  
 docker container exec -it rabbitmq /bin/sh

Text

Description automatically generated  
  
4) Перезапустите контейнер с брокером. Проверьте состояние очереди. Сохранилось ли ее состояние? Почему сохранилось/не сохранилось?

Очередь не сохранилась, т.к. очереди хранятся в памяти

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 queue-rabbit-persist-disconnect-task.jar, дождитесь надписи **'Execution finished'** в стандартном выводе.

Text

Description automatically generated

3) Проверьте состояние очереди **'pers-query-1'**. Сколько сообщений ушло, сколько осталось (если осталось).

40 ушло, 10 осталось

Для выполнения данного задания понадобится доступ в консоль:  
 docker container exec -it rabbitmq /bin/sh

Text

Description automatically generated

4) Перезапустите контейнер с брокером. Проверьте состояние очереди. Сохранилось ли ее состояние? Почему сохранилось/не сохранилось?

Очередь сохранилась, т.к. вероятно, было выставлено свойство persisted.

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 queue-rabbit-round-robin-task.jar, дождитесь надписи **'Execution finished'** в стандартном выводе.

Graphical user interface, application

Description automatically generated  
  
3) Посмотрите на сообщения в стандартном выводе.  
 а) Какой получатель (id) начал обработку (прием) сообщений?

rcvId = 78  
 б) Какой получатель (id) завершил обработку (прием) сообщений?

rcvId = 90  
 в) Почему какой-то промежуток времени сообщения никем не обрабатывались?

Какое-то время worker не работал на прием сообщений и их никто не забирал, потом был запущен другой worker и прием сообщений продолжился.