Kuanysh Zhunussov
12BD02076
Lab #5

In this laboratory work I implemented messaging pattern to build a “numberprocessing” service. Which does the following tasks: Get the number(s) and do some mathematical operations. This project was written on Java as Android app. For working with message queuing I used CloudAMPQ (RabbitMQ as cervice) and rabbitmq library for Java.

Firstly I've settup Connection and create channel from it. Then set parameters for channel:

channel.basicQos(1) – only 1 message for a worker at a time
channel.queueBind() - bind queue to exchange
channel.basicConsume() - set up consumer

In program I have my own queue which contain messages sent from EditText (if message was sended with publicToAmqp() remove from queue).

Then I create 5 threads: 1 for publish and 4 for workers (all workers do same things).

I publish message with function channel.basicPublish() and wait with channel.waitForConfirmsOrDie() (if I catch Exception, that is message not published → add message to my own queue again).

Workers get messages with consumer.nextDelivery(). Then I handle message → make numerical operaton for O(n) and add it to my View after calculating. Since I didn't use channel.basicAck(true), in my program workers doesn't send acknowledges.