introduction

The task is to write a pair of applications - the server and the client to perform the authorization "UDP port knocking" method (https: \\ pl.wikipedia.org \ wiki \ Port_knocking)

How the application works

The application consists of two processes, the server and the client.

- First, start the server process. This process opens the number of UDP ports with a given parameter, and then starts listening to packets from clients on them. If the correct sequence of UDP packets sent from that one address (same IP address and port) is detected, it opens a randomly selected port TCP and to the address from which these packets came, sends a UDP message with the number of this TCP port. Then it waits for a TCP connection and after completing simple communication (query-response type), it disconnects, and then starts listening again on UDP ports.
- The client application takes as parameters the server address and port numbers on which it is to "knock". After starting, it opens a UDP port, from which it sends a series of packets to the following ports given as parameters. After sending them, it waits for a return message containing the number of TCP port with which it then establishes a connection. After simple communication with the server (query-response type), it hangs up and quits.

  If the UDP packet sequence is incorrect, no reply will come from the server. In this case, after a certain timeout, the client exits with an error message.

Requirements and assessment method

In order to perform the task, it is necessary to design and implement client and server processes implementing their own communication protocol, enabling the implementation of the above-described functionalities. It is left to the author to decide what content the packets are sent between each process.

1. The server process is started with parameters which are a list of UDP port numbers, it waits for next packets. The list can be of any length and the ports can repeat on it (of course, we only open a port once). We assume used there

are only ports with numbers above 1024. If the server cannot open a given port because it is occupied by another process - it exits with an error.

2. The client's process receives its first parameter with the server's IP address, the next parameters are the UDP port numbers to which subsequent packets should be sent. It should use random (determined by the system) UDP and TCP ports.
3. We write applications in Java according to the Java 8 (JDK 1.8) standard. Only basic UDP communication classes can be used for communication over the network.