Socket programming

I have implemented a client/server application over TCP with threads for each connection. It would be also possible to use UDP and avoid implementing threads (because the operation “divide” is very fast) but it is a too simple solution, it is more interesting to combine threads with sockets for learning or practising.

The implemented protocol:

1) Client sends first operand as integer
2) Client sends second operand as integer
3) Server divides
4) Server sends error code as integer
    1. If error code is 0, the operation has been successful
    2. If error code is 1, the operation has been unsuccessful (remote exception)
5) Server sends result as float if no error has happened

Implemented classes:

- DivideService: Listen on port 9000 and creates new threads to attend connections
- DivideThread: Receives two integers through a socket, divides them and return the error code and the result following the above protocol.
- Client: Requires two arguments and sends them to port 9000 of localhost. Receives error code and result if the operation has been successful.

Execution:

- Server:
  ```
  java DivideService
  ```
- Client:
  ```
  java Client 50 2
  ```

Author: Sergio Blanco Cuaresma