Java RMI. (9 points)

This homework consists of two parts: *i*) to become familiar with the basic concepts of JAVA RMI by implementing a simple client/server application (the same service that sorts a string) and *ii*) implement a load balancing with a proxy server.

Please read the details about JAVA RMI and get an example classes in Java:

https://docs.oracle.com/javase/7/docs/technotes/guides/rmi/hello/hello-world.html

The solution for this homework must be uploaded to OLAT until Monday, **16.04.2018**, at **08:00**, and will be evaluated on 17th of April!

Part 1: A Simple Client / Server Application using JAVA RMI (Invoke a remote object).

Implement a server that offers a simple service *String* ***sort****(inputString)*, which returns the sorted *inputString*.

a. Implement a test *Server's* service ***sort****(inputString)* and a test *Client* **(2 points)**
b. Implement another *Server's* service ***compute****(inputTime)* that will simulate a computational-intensive service (Simulate it simply by waiting *inputTime* seconds). **(1 point)**
c. Determine experimentally whether your server is capable of handling concurrent client requests from the same client or from multiple clients. Write the results of the observations in a text file. Compare the observation with Homework 2. **(1 point)**

Part 2: A Proxy Server.

Extend the solution of Part 1 such that you create a proxy server or dispatcher between clients and end-point servers. Namely, clients with access rights *sort* should access to ***sort****(inputString)*, while clients with access rights *compute* should access to ***compute****(inputTime)*.

a) Develop
    a. two clients; one with access rights *sort* and the other with access rights *compute*.
    b. the end-point server (similar as the server of Part 1).
    c. an appropriate proxy server such that will prevent the clients to have access to the service for which they do not have access rights.
b) Demonstrate that your application is functional with at least two clients with different access rights, one proxy and two end-point servers that serve different services.

(5 points)