Distributed Systems
Exercise Sheet 5, Thursday
Klingemann, SS 2022

Deadline: 23rd June 2022

2nd Assessed Exercise

1. Date-Service using Java RMI
Understand the example code and test the programs. Client and server can run on the same machine. The nameservice has to be started by means of the rmiregistry command.

2. Distributed system for the management of conferences using Java RMI
Extend your simple system for the management of conferences from Sheet 3 to create a client-server-based system using Java RMI. Therefore, you have to transform the objects of the classes conference and participant into remote objects. All conference- and participant-objects exist only on the server! For each participant there exists a separate object. Similar to the description in Sheet 3, we require that each conference has two private attributes: a name and a set of participant objects. This set should represent the people that participate in the conference. (Note that it is a set of objects and not just a number!) Each participant has three private attributes: a name, a country and a company.
A conference should have methods with the following functionality. All these methods can be invoked from the client.

- Search for a participant with a particular name. You can assume that there exists at most one. The method returns a reference to the corresponding participant object.
- Add a new participant to the conference. The method has three parameters: a name, a country and a company. A corresponding participant object is created and added to the set of participants of the conference.
- Return the set of all participant objects of the conference. (The return value has to be a collection of references to participant objects and not just a number!)
- Return the name of the conference.

A participant should have methods with the following functionality. All these methods can be invoked from the client.

- Return the name
- Return the country
- Return the company
- Change the company

Implement a client that is calling all methods of the objects on the server in a sensible manner. In particular, your client should be able to calculate based on the methods above the total number of all participants from Germany. Your application should use two conference objects. The nameservice has to be started by means of the rmiregistry command.

Organisatorical matters

- You have to solve the exercise completely on your own! (No working in groups!)
- It is necessary but not sufficient to present a working program. Moreover, you have to be able to explain all parts of your program, be able to answer questions with respect to your program and make small extensions of you program.
- Your program has to be created completely within the exercise slot.
- If you violate one of the rules above, this implies that you definitely fail in this exercise.
- You can only present solutions that correspond to the exercise slot you are assigned to.
- It is in your responsibility to present your solution in time before the deadline. The assessment of your solution can only be guaranteed if you finish your program 60 minutes before the end of the exercises.
- To take part in the exam it is required to solve at least three of five assessed exercise sheets.