Social Network

Develop an application to support a social network. All classes must be in package "`Social`".

R1 - Subscription

The interaction with the system is made using class `Social`.
You can register new account using function `addPerson()` which receives as parameters a unique code, name and surname.
The function throws the exception `PersonExistsException` if the code passed is already associated to a subscription.
The function `getPerson()` returns a string containing code, name and surname of the person, in order, separated by blanks. If the code, passed as a parameter, does not match any person, the method throws the exception `NoSuchCodeException`.

R2 - Friends

Any person, registered in the social, should have possibility to add friends. Friendship is bidirectional: if person A is friend of a person B, that means that person B is friend of a person A. Friendship is created using function `addFriendship()` that receives as parameters the codes of both. The function throws the exception `NoSuchCodeException` if one or both codes do not exist.
Function `listOfFriends()` receives as a parameter code of a person and returns the collection of his/her friends. The exception `NoSuchCodeException` is thrown if the code does not exist. If a person has no friends an empty collection is returned.
The function `friendsOfFriends` receives as a parameter the code of a person and returns the collection of the friends of his/her friends, i.e. friends of the second level. The exception `NoSuchCodeException` is thrown if the code do not exist. If the list is empty the method returns an empty colletion. The list should not contain the person whose code was passed as parameter in this method. ("remove yourself from the list")
The function `friendsOfFriendsNoRepitition()` returns the list of codes of friends of the second level (friends of the friends), like the previous method, with the difference that it should delete duplications. For example, Friend A has Friend B and Friend D. Both B and D have Friend C. The function `friendsOfFriendsNoRepitition()` called for A should return C only once. If the list is empty the function should return and empty collection. The exception `NoSuchCodeException` is thrown if the code do not exist.

R3 - Groups

It is possible to register a new group using function `addGroup()`. Name of the group should be a single word.
The function `listOfGroups()` returns the list of names of all registered groups. If there are no groups in the list the function should return an empty collection.
A person can subscribe to a group using function `addPersonToGroup()` that receives as parameters the code of the person and the name of the group.
Function `listOfPeopleInGroup()` returns collection of codes of people subscribed to a given group.

R4 - Statistics

Function `personWithLargestNumberOfFriends()` returns code of a person that has largest amount of friends (first level). Do not consider the case of ties.
Function `personWithMostFriendsOfFriends()` returns code of a person that has largest amount of the friends of friends (second level). Do not consider the case of ties.
Function `largestGroup()` returns name of the group with largest number of members. Do not consider the case of ties.
Function `personInLargestNumberOfGroups()` returns code of a person that is subscribed to largest number of groups. Do not consider the case of ties.