

Object-Oriented Programming Exercises

2 *Social Network (Rede Social)* Application Development

2.1 Problem statement

The goal of this exercise is to develop an application similar to the popular Facebook. This application will manage a set of people and a set of friendship relations among them. Each *person* has an associated *name*, an *email* a sentence describing his *status* and the person's set of *friends*. Note that friendship relationships are simetric, that is, if “Ana” is friends with “Bruno”, then “Bruno” is friends with “Ana”.

Assume that typically there are no more than 500 registered people in the system and that each person has no more than 50 friends. The application must support:

1. Check if a given person is registered in the system (command `CONSULTAPESSOA`). This command receives the person's name as argument. The operation allways succeeds offering as feedback (“Pessoa registada.”), if the person is registered, or (“Sem registo.”), if the person is not registered.
2. Register a person in the system (command `REGISTA`). This command receives the person's name, email and initial status. In case of success, the person is registered (“Pessoa registada com sucesso.”). The operation fails if: (1) there is already a person registered with the same name (“Pessoa registada.”)
3. Check if a friendship is registered in the system (command `CONSULTAAMIZADE`). This command receives the names of the two people. The operation always succeeds, returning the message (“Amizade existente.”), if the friendship already exists, or (“Amizade inexistente.”), if the friendship does not exist.
4. Establish a new friendship between two persons (command `AMIGOS`). This command receives the names of the two persons. If successful, the friendship is created (“Amizade criada.”). This operation fails if: (1) one of the names does not match a registered user (“Sem registo.”); (2) the friendship already exists (“Amizade existente.”); or (3) there is an attempt to register a new friendship relationship with a single user (i.e. a user tries to become friends with that same user) (“Amizade invalida.”).
5. List the friends of a user (command `CONSULTAAMIGOS`). The command receives the name of the user. If that user has no friends, the program outputs the message “Nao tem amigos registados.” Otherwise, this operation prints a header (“Lista de amigos:”) followed by the name and email address of all that user's friends, sorted by insertion order. This operation fails if: (1) the name does not match the name of any registered user (“Sem registo.”).
6. Update the user's status (command `NOVOESTADO`). The command receives the user name and new status. If successful, the status is updated (“Estado alterado.”). This operation fails if: (1) the user name does not match a registered user (“Sem registo.”).
7. Print the status of a user (command `CONSULTAESTADO`). The command receives the user's name. If successful, the program prints the user's status. This operation fails if: (1) the name does not match a registered user (“Sem registo.”).

8. List registered users in the application (command `PESSOAS`). This operation always succeeds. If there are no registered users, the program presents the message “Rede Social vazia.” Otherwise, the operation presents the header (“Lista de pessoas registradas:”) followed by the name and email address of the registered users. The list is sorted by insertion order.
9. Quit the program execution (command `SAIR`). This operation always succeeds (“Adeus.”).

2.2 Program interaction example

Please develop your program so that it ensures the interaction model illustrated in the following example (the character `↵` represents a change of line):

PESSOAS↵
Rede Social vazia.↵
↵
REGISTA↵
Ana↵
ana@gmail.com↵
Bom dia mundo!↵
Pessoa registada com sucesso.↵
↵
REGISTA↵
Ana↵
ana@gmail.com↵
A trabalhar...↵
Pessoa registada.↵
↵
REGISTA↵
Bruno↵
bruno@gmail.com↵
A trabalhar...↵
Pessoa registada com sucesso.↵
↵
CONSULTAAMIZADE↵
Bruno↵
Maria↵
Amizade inexistente.↵
↵
CONSULTAPESSOA↵
Bruno↵
Pessoa registada.↵
↵
CONSULTAPESSOA↵
Daniel↵
Sem registo.↵
↵
AMIGOS↵
Ana↵
Bruno↵
Amizade criada.↵
↵
CONSULTAAMIGOS↵
Bruno↵
Lista de amigos:↵
Ana; ana@gmail.com↵
↵

AMIGOS↵
Ana↵
Ana↵
Amizade invalida.↵
↵
NOVOESTADO↵
Ana↵
Boa tarde!↵
Estado alterado.↵
↵
CONSULTAESTADO↵
Ana↵
Boa tarde!↵
↵
CONSULTAAMIZADE↵
Bruno↵
Ana↵
Amizade existente.↵
↵
CONSULTAESTADO↵
Daniel↵
Sem registo.↵
↵
NOVOESTADO↵
Daniel↵
Surfing...↵
Sem registo.↵
↵
REGISTA↵
Carlos↵
carlos@gmail.com↵
Bem disposto!↵
Pessoa registada com sucesso.↵
↵
AMIGOS↵
Ana↵
Carlos↵
Amizade criada.↵
↵
AMIGOS↵
Ana↵
Carlos↵
Amizade existente.↵
↵
PESSOAS↵
Lista de pessoas registadas:↵
Ana; ana@gmail.com↵
Bruno; bruno@gmail.com↵
Carlos; carlos@gmail.com↵
↵
CONSULTAAMIGOS↵
Ana↵
Lista de amigos:↵
Bruno; bruno@gmail.com↵
Carlos; carlos@gmail.com↵
↵
SAIR↵
Adeus.↵
↵

2.3 Development

Develop your program according to the following phases:

1. In a first phase, specify the interface(s) supporting the program *Social Network*. Draw a diagram to illustrate the different involved interfaces.
2. In a second phase implement and test the program *Social Network*.
3. Submit your source code to *Mooshak*.

Test files

The tests in Mooshak will verify the implementation of the several commands:

- Test file: `1_in_test.txt` (15 points)
- Test file: `2_in_test.txt` (15 points)
- Test file: `3_in_test.txt` (15 points)
- Test file: `4_in_test.txt` (15 points)
- Test file: `5_in_test.txt` (15 points)
- Test file: `6_in_test.txt` (15 points)
- Test file: `7_in_test.txt` (10 points)

2.4 Additional commands

11. Add *public post* (command `POSTARPUBLICO`). This command receives the name of the message author and the message to be published. If successful, the message is published in the author's wall ("Post registado."). This operation fails if: (1) there is no registered user with the given name ("Sem registo.).
12. Add *public post* on a friend's wall (command `POSTAR`). This command receives the post author name, the post to be published and the wall's owner name. In case of success, the post is published ("Post registado."). This operation fails if: (1) there is no registered user with the given name ("Sem registo."); (2) there is no friendship between the author and the wall owner ("Amizade inexistente.).
13. Visit a friend's wall (command `MURAL`). This command receives the visitor's name, and the wall owner's name. If successful, the operation presents the header ("Mural de <dono>:") followed by the several visible posts, including the author of each post. This operation fails if: (1) there is no registered user, or wall owner with the given names ("Sem registo."); (2) the user and the wall owner are not friends ("Amizade inexistente.).