

Programming project: Social Networks

Domantas Meidus

October 22, 2017

Contents

1	Introduction	1
2	First version of the Project	1
2.1	Classes design	1
2.2	Description of the data structures used in the project	2
2.3	Design and implementation of the methods	3

Abstract

Abstract will be written when all operations parts will be implemented.

1 Introduction

The software application to be developed will have to manage a social network. This social network is formed by people that may be linked among each other if there is a friendship relationship among them. For each person the following items will be recorded:

- identifier (unique)
- name
- surname(s)
- birthdate
- birthplace
- home
- studiedat (he/she could have studied at many places)
- workedat (he/she could have worked at many places)
- movies
- groupcode

New people can be added to the network or deleted from it, besides other actions that may be of interest for the project.

2 First version of the Project

2.1 Classes design

Social network program is based on three class: Main, Menu and Operations classes.

Main: This class is the main Social Network class. This class have one function - invokes Menu class.

Menu: This is the class which is responsible for collecting user keystrokes and displaying project Menu text. Menu class invokes Operations.

Operations: This class is responsible for all Social Network project functions. All Social Network used classes are displayed in the project class diagram(Figure 1)

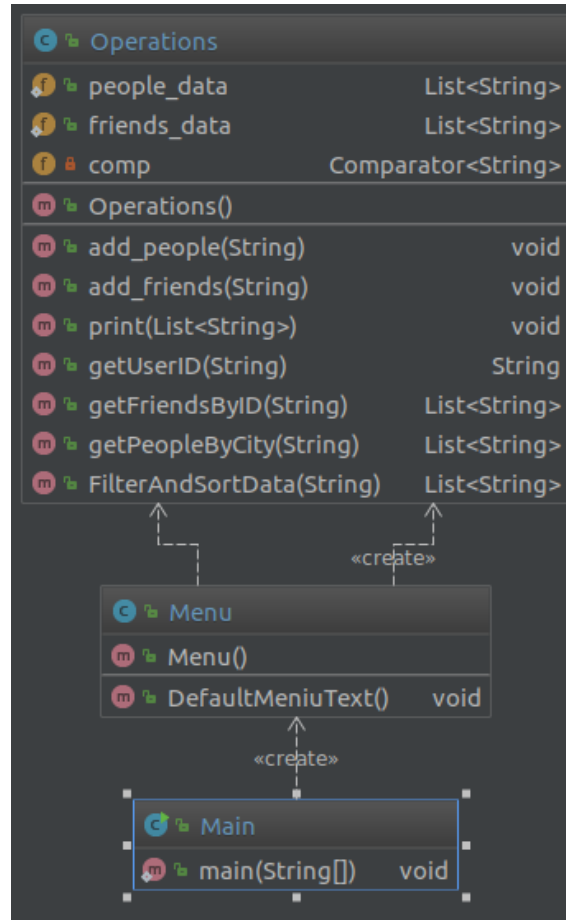


Figure 1: Project Class diagram.

2.2 Description of the data structures used in the project

The main data structures used in the Social Network project are type of Lists: **people_data** and **friends_data**.

- **people_data.**

Usage: List is used for storing all the people in the social network.

Data type: String.

Access modifiers: public static

Class: Operations

Implementation: **public static List<String> people_data = new ArrayList<String>();**

- **friends_data.**

Usage: List is used for storing all the friends clique in the social network.

Data type: String.

Access modifiers: public static

Class: Operations

Implementation: **public static List<String> friends_data = new ArrayList<String>();**

Some Operations class functions have implemented lists data structures in order to store results or display information.

2.3 Design and implementation of the methods

- **Menu class:**

public Menu(): Menu class constructor which invokes Operations class in order to access Social Network functions. Constructor analyze user inputs in order to achieve Social Network functions. The diagram of all available functions which are available for user are displayed in the figure 2. Available operations according user input value:

input = 0: Exit the Social Network program.

input = 1: Take the data from a person and add him/her to the network. This function does not read any data from Console. This function receives the data as parameters.

input = 2: Upload people's data (people.txt) from a file in our directory and add them to the social network. Use the function developed in the previous point.

input = 3: Print out a listing to a text file of the people on the network.

input = 4: The task is to implement an operation that relates them in the network, that is, the relationship has to be stored in the network.

input = 5: On the social network: given a person surname, retrieve his/her friends.

input = 6: On the social network: given a city, retrieve all people who were born there.

input = 7: On the social network: retrieve the people who were born between dates D1 and D2, sorted by birthplace, surname, name. The order relationship will be implemented according to the lexicographic order (dictionary's order) of the strings used for the attributes.

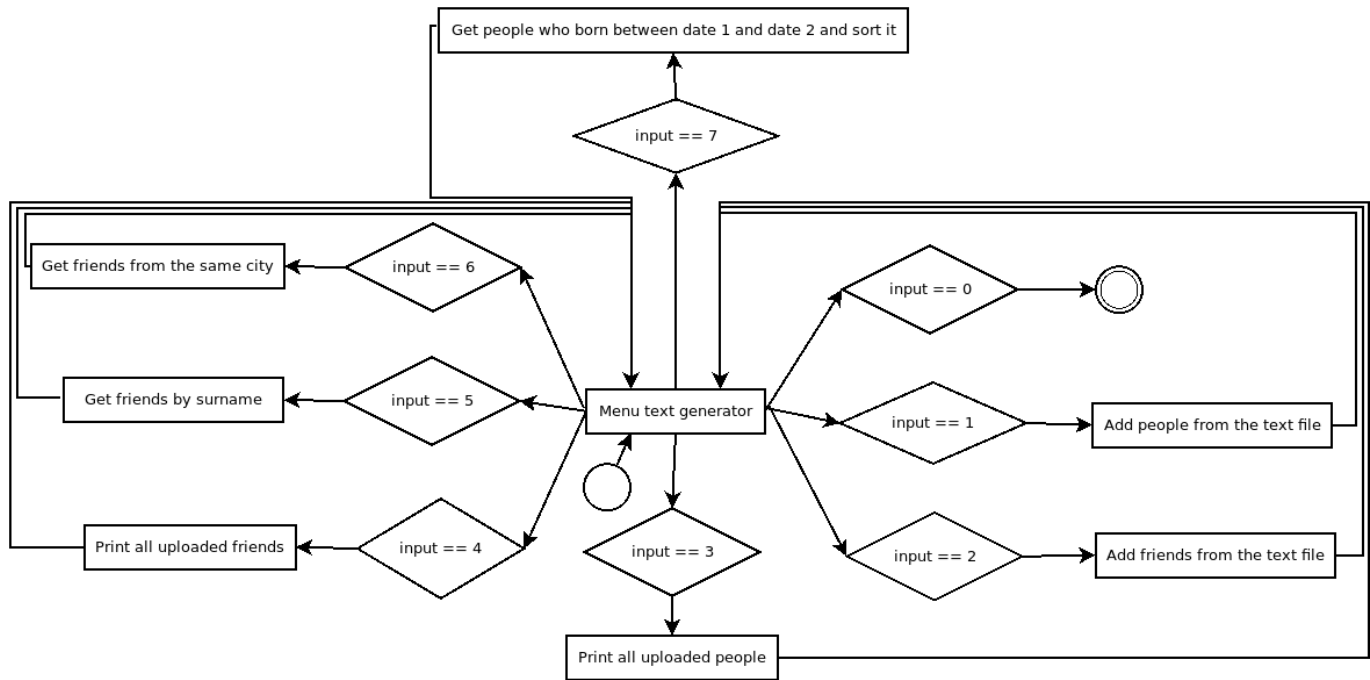


Figure 2: Menu operations scheme.

public void DefaultMenuText(): This is a void methods which only display text of menu for the user. Menu text in the program can be seen in the figure 3.

```

*****Social Network Menu*****
[1] To upload the people file
[2] To upload the friend file
[3] Print all People
[4] Print all Friends
[5] Retrieve friends by surname
[6] Retrieve all people who born in the same city
[7] Retrieve all people who born between D1 and D2 dates and sort data by birthday, surname, name
[0] To exit the program

```

Figure 3: Displayed default Menu text in the Social Network program.

- **Operations class:**

public void addPeople(String path): This Method is responsible for reading the data of people file using scanner and add the data to the list(friends_data) data structure. It has one argument: **String path** which is the path of the people text file in the local computer file system.

public void addFriends(String path): This Method responsible for reading the data of friends file using scanner and add the data to the list(friends_data) data structure. It has one argument: **String path** which is the path of the friends text file in the local computer file system.

public void print(List<String> data): This method prints the data in the terminal of the List. It has one argument: **List<String> data** - is a list of stored data. In program this method is used for displaying elements of **people_data**, **friends_data** or other lists which could be created during in operations implementation.

public String getUserID(String surname): Method for getting UserID attribute value from people_data list. This method iterate between every element in people_data list and search for a surname which is passed from program User. It has one argument: **String surname** - which is an program user written input text value.

public List<String> getFriendsByID(String userID): Method for displaying all friends of person which UserID attribute matches with given userID. It has one argument: **String userID** - which is the person code. It returns a list of all person friends surnames.

public List<String> getPeopleByCity(String city): Method for displaying all people in the social network which birthday place matches with city text value. It has one argument: **String city** - which is the city of people birthday place. It returns a list of all person which birthday place text value matches with city text value.

public List<String> getPeopleByCity(String city): Method for displaying all people in the social network which birthday place matches with city text value. It has one argument: **String city** - which is the city of people birthday place. It returns a list of all person which birthday place text value matches with city text value.

public List<String> FilterAndSortData(String date): Method for creating new list for storing filtered and sorted people_data list data. It has one argument: **String date** - is written date range for filtering in the String data type written as an input from program User. Format of this method date should be like this: **date = "dd-mm-yyyy dd-mm-yyyy"**; for example: **date = "12-09-1982 15-04-1992"**. At first date should be less than the second date. Sorting operation is performed after data has been filtered. Filtered data is sorted in ascending way according 3 people data attributes:

1. **Person Birthday date**
2. **Person surname**
3. **Person name**

References