1. Object-oriented part (Java) [8% of your final mark]

Create the classes needed to solve the stable matching problem for residents and programs with the iterative Gale-Shapley algorithm. Your program must be a Java application called `StableMatching` that takes as input the names of the two csv files containing the rank order lists of the residents and the programs.

Since this solution must follow the object-oriented paradigm, your program must be composed of a set of classes. Specifically, it must include, among others, the classes listed below.

- class `Resident` and its attributes:
  - `ID`: the unique number ID for this resident
  - `firstname`
  - `lastname`
  - `rol`: the rank order list enumerating the preferred programs specified using their IDs
  - `matchedProgram`: a reference to the program to which this resident has been matched, `null` if the resident is not matched
  - `matchedRank`: the rank of this resident in the matched program ROL, if it has been matched
- class `Program` and its attributes and methods:
  - `ID`: the unique 3-letter ID for this program
  - `name`
  - `quota`: the number of positions available in this program
  - `rol`: the rank order list enumerating the resident IDs in order of preference
  - `matchedResidents`: a list of references to the residents that have been matched to the program
  - `member(residentID)`: a method that returns true if the resident is included in the ROL of this program
  - `rank(residentID)`: a method that returns the rank of the resident in the program ROL (or `-1` if the resident is not in the list).
  - `leastPreferred()`: a method that returns the reference to the matched resident instance having the highest rank in the program ROL (the least preferred one)
  - `addResident(resident)`: this method will add the resident to the match list of this program if the program has not reached its quota or if this resident is preferred over some of the currently matched residents. This method may have the side effect of removing a resident from the `matchResidents` list (nullifying its `matchedProgram` attribute)

- class `GaleShapley` and its attributes and methods:
  - This is the class that will run the Gale-Shapley algorithm
  - `HashMap<residentID,Resident>`
  - `HashMap<programID,Program>`
  - `loadResidents(filename)`: a method that reads the csv file of the programs
  - `loadPrograms(filename)`: a method that reads the csv file of the programs
- plus any other classes, methods or attributes you judge necessary, including getter and setter methods.

To help you with your project, we provide you with three (incomplete) classes (`GaleShapley`, `Resident`, `Program`) that reads the csv files and creates the two hash maps.

Submission

In addition to the source code of your solution, you must also submit

- A document that includes a UML diagram of all your classes (showing attributes, associations and methods). Do not use static methods, except for the `main` function.
- You must also save and submit the output of your program into a text file (for the `residents.csv` and `programs.csv` files).
- This document must also cite all references used to build your solution.