The game of Mafia

In this assignment you will write a program that plays the game Mafia. There are many variants of the game, and ours runs as follows. The game is held in a little town in Italy. Some of the townspeople belong to the mafia. Mafia members all know each other, but the rest of the townspeople (the innocents) don’t know who belongs to the mafia. The game begins at night (the night phase). At night the mafia decide which of the innocents in the town to kill. In the day phase that follows the townspeople can accuse a person of belonging to the mafia. If there is a majority that agrees, the accused is executed. Any person in the town can accuse someone of belonging to the mafia, and all the townspeople that are alive participate in the vote.

In our version of the game during the night phase a mafia member is chosen, and he randomly chooses an innocent person to be executed. In the day phase a random townsperson is chosen as an accuser, and he picks a random townsperson (not himself) to accuse as belonging to the mafia. Voting proceeds as follows: Mafia members vote yes for killing an innocent, and no for killing a mafia member. The townspeople vote randomly—there should be a fifty-fifty chance for voting to execute a particular person.

Classes you need to implement

The Person class models a generic towns-person. It should contain the following methods:

- A constructor that receives a string as parameter (the name of the person).
- boolean vote(String name): returns a boolean value, returning true 50% of the time.
- String accuse(String [] names): returns a random name out of the array of names of potential people to accuse (each name should be equally likely to be picked).

The Innocent class models an innocent towns-person. It should extend the Person class, and use all its predefined methods.

The MafiaMember class models a member of the mafia. It needs to override the vote method of the Person class, voting no (false) for killing a mafia member, and yes (true) for an innocent. To model the fact that mafia members know each other, each MafiaMember will keep an ArrayList of the names of the members of the mafia, and a method boolean isMafiaMember(String name) that determines if a given person belongs to the Mafia. The list of members of the mafia can be shared across the mafia members.

The MafiaMember
class also needs to override the `accuse` method, such that it randomly picks an innocent person. A `MafiaMember` also has a `kill` method with the same signature as the `accuse` method which picks an innocent towns-person to be killed (i.e. its signature is `String accuse(String [] names)`).

The `Town` class is a collection of townspeople. It contains an ArrayList of type `Person` that contains the people in town. In support of the game it needs to provide the following methods:

- A `constructor` `Town(String [] names, int numMafia)` that receives an array of names and an integer that specifies how many members the local mafia should have. It then instantiates the required number of mafia members picking their names randomly from the given list of names. The rest of the names are assigned to the town’s innocents. Store the town’s people in an ArrayList called people that accepts objects of type `Person`.

- `boolean alive()` indicates whether there is at least one innocent and at least mafia member left alive in town, so that the game can continue.

- `getMafiaMember()` returns a random `MafiaMember` from the members of the local mafia.

- `getTownPerson()` returns a random `Person`.

- `void remove(String name)` removes the person with the given name from the list of people in town. Note that if the person is a member of the mafia, the lists of mafia members in each `MafiaMember` objects need to be updated.

- `int getNumInnocents()` returns the number of innocents in town.

- `int getNumMafia()` returns the number of mafia members.

- `String [] getPeople()` returns the names of the people in town who are still alive.

- `int vote(String name)` runs a vote among the town’s people on whether to execute the person with the given name (does this by tallying up the votes from each person in town).

- `boolean isMajority(int votes)` returns true if the given number of votes is a majority (greater than half of the number of people in town).

A driver class is provided. Your code will be run with this driver code.
Specifications, notes, and hints

Your program needs to meet the following specifications:

- Submit the files Person.java, MafiaMember.java, Innocent.java, and Town.java. Make sure each program has the specified methods, so our driver code would work.

- When commenting your code use Javadoc style comments at the beginning of each method.

- Put comments at the top of the file with your name, EID, date and course, and a short (one or two line) description of what the program does. We will be testing the code on the machines in the CS computer lab, so make sure your code runs on those machines.

- Submit your source code files via the checkin program by the due date (read the course syllabus for the late policy).