Lambton College | In Toronto

Final Project

[Total Marks: 100]

Class: *2019F_ MAD 3463_2*
Instructor: Mohammad Kiani

Evaluation: 35%
Due Date: Wednesday, 2nd October 2019, 17:00 PM

Submission:

- The final project can be completed in a team of maximum 3 members.
- Create a PDF document for your report named as your group name that contains all the requested answers.

Project name: Election

The goal of this exercise is to simulate the election of a leader by the members of his (or her) party.

The code to be produced

Your task is to complete the provided code in accordance with the following description.

the **Applicant** class: We start by implementing a class Applicant that will be used to represent each person applying for the position of leader. An applicant is characterized by:

- the number of voters (number of people that will vote for him/her).
- a name;

You are required to satisfy the following constraints when implementing the Applicant class:

1. the constructor will initialize the applicant's *name* and the number of voters given values taken as parameters (in that order). The number of voters will be 0 by default. The constructor must be compatible with the provided main method;

2. the Applicant class will contain:
    - a copy constructor;
    - a method elect, compatible with the provided main method, that increments the number of voters associated to the applicant;
    - a method init that resets the number of voters to zero;
    - a getter getVotes that will return the number of voters that are backing the applicant;
    - a getter getName that returns the applicant's name;

3. the class must be well encapsulated.

The **Ballot** class: The election of a leader takes place through the organisation of a vote.

A Ballot is characterized by:

- the set of applicants for the position of leader;
- the total number of available voters (all members in the party);
- the date (day) of the vote (an integer).

Your task is to implement the Ballot class in order to model this concept while respecting the following constraints:

1. the class, in particular its constructor, must be compatible with the provided main method; each applicant shall be a copy of the one taken as argument. The constructor will also take a Boolean as final argument; if this Boolean is true, each applicant shall have its associated number of voters reset to zero; the value of this argument must be true by default;

2. the class must be well encapsulated;
3. the set of applicants must be represented using an ArrayList;
4. the class must implement the following public interface:

    - a method computeVoters that returns the actual number of voters (an integer). This value is computed as the sum of the voters associated to each applicant;
    - a method winner that returns the name (a String) of the applicant that received the highest number of votes. In the case of a tie, the last such applicant will be returned;
    - a method results that will display the results of the vote in the following format (must be strictly respected):

```
Participation rate -> <participation rate of the vote> per cent
Effective number of voters -> <actual number of voters>
The chosen leader is -> <name of the elected leader>
Voter distribution
<name of applicant 1> -> <percent 1> percent of voters
.....
<name of applicant n> -> <percent n> percent of voters
```

(an empty line must appear at the end of the display).
The<participation rate of the vote>is the ratio, in percent, between the actual number of voters (as computed by the previous method) and the total number of available voters. You shall then use the method winner described above to determine the<name of the elected leader>.
<percent i> is the percentage of voters that backed applicant number i (ratio, in percent, of the number of voters backing the applicant and the actual number of voters).
If the actual number of voters is zero, the method results will simply display a message indicating that the vote is canceled, namely: "Canceled ballot, no voting" followed by an end-of-line character.

All numerical values must be displayed with a single decimal digit. This can be achieved by using the following instructions:
`System.out.format("The result is %.1f", value);`
where value is a double that should be displayed with a single decimal digit.

Vote hierarchy**:**

The members of a party express their choice through a ballot (the class Vote). A Vote is characterized by:

- the name of the *applicant* (the person for whom the vote is cast, a String);
- the actual date when the vote was cast (for simplicity, only the day, an integer);
- the limit date after which the vote can no longer be cast (again, the day).

The Vote class must provide:

- an invalidity testing method: Boolean isInvalid() that cannot be defined concretely for an arbitrary ballot;
- a constructor that initializes the applicant's name, the actual cast date and the limit date given values taken as parameters (in that order);
- the getters getDate and getLimitDate;
- the redefinition of the toString method that produces a String representation of the ballot strictly respecting the following format:

`for <name of the applicant> -> invalid` if the ballot is invalid and
`for <name of the applicant> -> valid` if it is valid where <name of the applicant> is the name of the applicant for which this vote has been cast. Please note the leading space character in the above representations.

A ballot can consist in a *paper ballot* or an electronic vote. A paper ballot can furthermore be submitted *by mail*.

Invalidity of a ballot is determined based on its type:

1. an electronic vote is invalid if its cast date is strictly superior to the limit date minus two (electronic votes must be sent in earlier than the others);
2. a paper ballot is invalid if the ballot hasn't been signed: a Boolean attribute will indicate this being or not the case. This attribute shall be initialized by a constructor taking a Boolean value as the last argument (true means that it is signed);
3. a vote sent in by mail is invalid if it hasn't been signed or if its cast date is strictly superior to the limit date.

The classes that require a checking dates to determine validity shall implement an interface CheckBallot that requires the implementation of a method Boolean checkDate(). This method should be used whenever it is necessary to check dates to determine validity of a ballot. This method will return true when the date is valid.

The subclasses of Vote must also redefine the toString method in order to produce String representations strictly respecting the following formats:

- for the paper ballots:
  `vote by paper ballot for <name> -> invalid` if the ballot is invalid and
  `vote by paper ballot for <name> -> valid` if it is valid;
- for the votes sent in by mail:
  `mailing of a paper ballot vote by mail for <name> -> invalid` if the ballot is invalid and
  `mailing of a paper ballot vote by mail for <name> -> valid`
  if it is valid;
- and for electronic votes:
  `e-voting for <name> -> invalid` if the ballot is invalid and
  `e-voting for <name> -> valid` if it is valid.

where <name> is the name of the applicant for which the vote has been cast.

Lambton College | In Toronto

Vote simulation

Your final task is to enrich the Ballot class in order to handle a set of ballots.

In order to implement this feature, add to your Ballot class:

- a votes attribute that represents a set of Vote instances (use an ArrayList). Be wary of the x potential changes this may imply for the existing constructor;
- a method countVotes that updates the number of voters for each applicant given the contents of votes: for each valid ballot in votes, increment the number of voters of the applicant for which the vote was cast.
- a method simulate that takes in parameter a participation rate and a vote date. This method will simulate an election based on the following algorithm:

1. compute the actual number of voters as the total number of voters (members of the party) multiplied by the participation rate (and then, convert the result to an int);
2. for each voter i, draw an integer at random candNum between 0 and the number of applicants (minus 1) using the provided method `Utils.randomInt`;
3. if i%3 returns 0, add an electronic vote to votes in favor of the candidate with index candNum in the list of applicants;
4. ifi%3returns1, add a paper ballot to votes in favor of the candidate with index candNum in the list of applicants;
5. if i%3 returns 2, add a vote sent in by mail votes in favor of the candidate with index candNum in the list of applicants;
6. display the vote you obtained using the correct representation as described previously.

To simplify, all paper ballots from odd voters will be signed, and all others will not. All ballots will be dated using the parameter from the simulate method. Remember, the definition of the Ballot class has an attribute for the date of the vote!

Execution example:

```
Test part I:
--------------
Participation rate -> 36.7 percent
Effective number of voters -> 11
The chosen chef is -> Angel Anerckjel

Voter distribution
Tarek Oxlama -> 18.2 percent of voters
Nicolai Tarcozi -> 27.3 percent of voters
Vlad Imerbutin -> 18.2 percent of voters
Angel Anerckjel -> 36.4 percent of voters
```

Lambton College | In Toronto

```
Test part II:
---------------
e-voting for Nicolai Tarcozi -> valid
vote by paper ballot for Vlad Imirboutine -> valid
mailing of a paper ballot vote for Vlad Imirboutine -> invalid
e-voting for Tarek Oxlama -> valid
vote by paper ballot for Vlad Imirboutine -> invalid
mailing of a paper ballot vote for Nicolai Tarcozi -> valid
email voting for Angel Anerckjel -> valid
paper ballot vote for Angel Anerckjel -> valid
mailing of a paper ballot vote for Nicolai Tarcozi -> invalid
email voting for Angel Anerckjel -> valid
vote by paper ballot for Tarek Oxlama -> invalid
mailing of a paper ballot vote for Tarek Oxlama -> valid
e-voting for Tarek Oxlama -> valid
paper ballot vote for Angel Anerckjel -> valid
mailing of a paper ballot vote for Vlad Imirboutine -> invalid
Participation rate -> 50.0 percent
Effective number of voters -> 10
The chosen chef is -> Angel Anerckjel

Voter distribution
Tarek Oxlama -> 30.0 percent of voters
Nicolai Tarcozi -> 20.0 percent of voters
Vlad Imerbutin -> 10.0 percent of voters
Angel Anerckjel -> 40.0 percent of voters

e-voting for Tarek Oxlama -> invalid
vote by paper ballot for Vlad Imirboutine -> valid
mailing of a paper ballot vote for Vlad Imirboutine -> invalid
e-voting for Vlad Imirboutine -> invalid
vote by paper ballot for Tarek Oxlama -> invalid
mailing of a paper ballot vote for Tarek Oxlama -> valid
e-voting for Tarek Oxlama -> invalid
paper ballot vote for Tarek Oxlama -> valid
mailing of a paper ballot vote for Tarek Oxlama -> invalid
e-voting for Nicolai Tarcozi -> invalid
vote by paper ballot for Tarek Oxlama -> invalid
mailing of a paper ballot vote for Vlad Imirboutine -> valid
e-voting for Nicolai Tarcozi -> invalid
vote by paper ballot for Vlad Imirboutine -> valid
mailing of a paper ballot vote for Vlad Imirboutine -> invalid
Participation rate -> 25.0 percent
Effective number of voters -> 5
The chosen chef is -> Vlad Imirboutine

Voter distribution
Tarek Oxlama -> 40.0 percent of voters
Nicolai Tarcozi -> 0.0 percent of voters
Vlad Imerbutin -> 60.0 percent of voters
Angel Anerckjel -> 0.0 percent of voters
```

```
e-voting for Nicolai Tarcozi -> invalid
paper ballot vote for Tarek Oxlama -> valid
mailing of a paper ballot vote for Tarek Oxlama -> invalid
e-voting for Tarek Oxlama -> invalid
vote by paper ballot for Tarek Oxlama -> invalid
mailing of a paper ballot vote for Angel Anerckjel -> valid
e-voting for Tarek Oxlama -> invalid
vote by paper ballot for Nicolai Tarcozi -> valid
mailing of a paper ballot vote for Tarek Oxlama -> invalid
e-voting for Nicolai Tarcozi -> invalid
vote by paper ballot for Angel Anerckjel -> invalid
mailing of a paper ballot vote for Nicolai Tarcozi -> valid
email vote for Angel Anerckjel -> invalid
vote by paper ballot for Nicolai Tarcozi -> valid
mailing of a paper ballot vote for Nicolai Tarcozi -> invalid
Participation rate -> 25.0 percent
Effective number of voters -> 5
The chosen chef is -> Nicolai Tarcozi

Voter distribution
Tarek Oxlama -> 20.0 percent of voters
Nicolai Tarcozi -> 60.0 percent of voters
Vlad Imerbutin -> 0.0 percent of voters
Angel Anerckjel -> 20.0 percent of voters
```