

Design and implement a program processes voting data of presidential election for the country of College where the said country consists of 4 states. At this time, there are 6 candidates are running for the position as seen in the candData.txt text file. The results of the votes are sent to you for processing, see sample in voteData.txt. It lists each candidate's name and the votes they received by states. You should have a nicely formatted output of the election results with the winner's name printed.

voteData.txt

Your program driver should generate a randomly generated voteData based on the given candidate names, random state selection (1 - 4), and random votes (0 - 50) per state.

You should generate 10,000 - 1,000,000 entries in the voteData.txt before processing the data. In your design (UML/Flowchart, UML/pseudo code) have a BigO analysis for your algorithms (where it applies) and justify your selection of algorithms based on speed and storage characteristics.

Candidate

This class implements a candidate as an object.

Methods (besides standard methods, implement the specific methods especially their post condition requirements)

public void method_name_chsen_by_you(int state, int votes)

Method to set the votes of a candidate for a particular state.

Postcondition: The votes specified by the parameter votes are assigned to the state specified by the parameter state.

public void method_name_chsen_by_you(int state, int votes)

Method to update the votes of a candidate for a particular state.

Postcondition: The votes specified by the parameter votes are added to the state specified by the parameter state.

public boolean method_name_chsen_by_you(Candidate otherCan)

Method to determine if two candidates have the same name.

Postcondition: Returns true if the name of this candidate is the same as the name of the candidate specified by the parameter otherCan.

public int method_name_chsen_by_you(Candidate otherCan)

Method to compare the names of two candidates.

Postcondition: Returns 0 if the name of this candidate is the same as the name of otherCan; returns < 0 if the name of this candidate is less than the name of otherCan; returns > 0 if the name of this candidate is greater than the name of otherCan.

ElectionResult

This class contains methods for processing voting data for presidential election.

Methods (besides standard methods, implement the specific methods especially their post condition requirements)

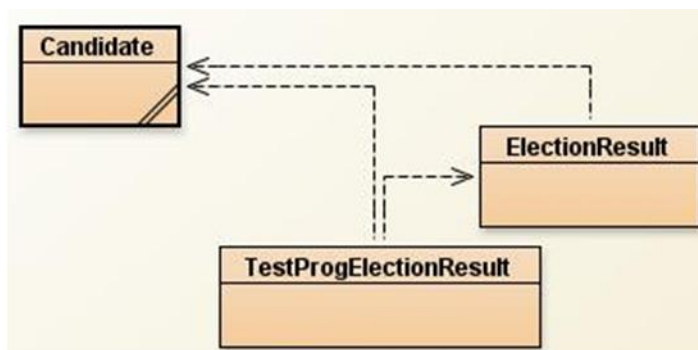
public static void method_name_chsen_by_you(Candidate[] list, int length)

Algorithm post condition: list objects are in ascending order.

public static int method_name_chsen_by_you(Candidate[] list, int length, Candidate searchItem)

Algorithm post condition: If searchItem is found in the list, it returns the location of searchItem; otherwise it returns -1.

To help you get started, here is a basic diagram of classes that you should design and implement.



candData.txt :

Mia
Mickey
Donald
Peter
Danny
Ashley

voteData.txt:

Mia 2 34
Mickey 1 56
Donald 2 56

Mia 1 78
Danny 4 29
Ashley 4 78
Mickey 2 63
Donald 1 23
Peter 2 56
Danny 1 25
Peter 4 23
Danny 4 12
Ashley 4 82
Mickey 3 67
Donald 2 67
Mia 3 67
Ashley 1 23
Mia 1 56
Donald 2 35
Peter 1 27
Danny 2 34
Peter 4 23
Mickey 4 89
Peter 1 23
Danny 3 89
Mia 3 89
Peter 1 67
Danny 2 37
Ashley 2 89
Mia 2 78
Donald 1 87
Peter 1 90
Danny 4 56