Melvin Berkoh
6 May 2022
Professor Bamford
Data Structures

Final Project - Phase 2 Updated

1. Question

```
public interface Question
{
    public String showPrompt() {}
    public void addResult(boolean answer) {}
    public float[] calcResults() {}
}
```

The purpose of this class is to contain the prompt and the responses of each question on the poll.
The constructor accepts a String prompt that is the question to be asked. The class will have two instance fields: a String named prompt and an int array of length two named results. The results will store responses in the format {numberOfNoResponses, numberOfYesResponses}.
Method addResult mutates results depending on the given boolean. If addResult is given false, then it will increment the first element. Else, it will increment the second element.
Method computeResults gives a float array that contains the percentages of no responses and yes responses.
Method showPrompt prints the prompt to standard output for the respondents to see.

2. List<Question>

will store all of our questions in a list of type List<Question>.

3. Scanner

will use the Scanner class to grab the user responses.

4.

```
public interface Poll {
    public void addQuestion(Question question) {}
    public void poll(int numberOfRespondents) {}
    public String generateResultsReport() {}
}
```

The purpose of this class is to encapsulate the List<Question> (#2).
The class has one instance field: a List<Question> named questions. The method addQuestion will add a Question instance to questions. The method poll iterates through questions, calling showPrompt and scanning the standard input numberOfRespondents times while adding each result through addResult. The method generateResultsReport iterates through questions and formats the values given by computeResults to generate a printable summary.

In the main class, I will have a Poll instance and feed it a hardcoded set of questions. Then, poll people by asking the user "How many voters are there?"

After the summusion of phase two my project hasn't really changed that much. I had to find a way to poll more users but not at the same time. The problem was that if multiple users had to vote they would all have to take turns answering one question. This made me realize that my code isn't efficient enough so I took out the "How many voters are there?" and just looped the whole program. A person answered yes to the question, is there another user yes or no. NO will be the end of displaying the results.
This changed my generateResultsReport

```
public String generateResultsReport(boolean isFinal)
   {
        String report;

      if(!isFinal)
         report = " Poll Results:\n";
      else
         report = " The Election is now over!\n Final Results:\n";
      // goes through each question. gives the question and its voting result.
      //the results are rounded to whole number percentages.
      for (int i = 0; i < this.questions.size(); i++)
      {
         Questions question = this.questions.get(i);
         report += "For Question # " + (i + 1) + " : " + question.showPrompt() + "\n";
         float[] results = question.calcResults();
         report += Math.round(results[1] * 100) + "% voted Yes." + "\n";
         report += Math.round(results[0] * 100) + "% voted No." + "\n";
      }
      return report;
   }
```

Also created a boolean variable to help me loop though the program allowing me to end or add more users

```
boolean isMoreVoters = true;

      while (isMoreVoters)
      {
               System.out.println("For all of the following 10 questions please respond with either
yes or"
                                 + " no (Results will be shown at the end)\n");
               //System.out.println("How many users are there? (please enter a numeric value)");
               // polls the number of voters
               election.poll(sc);
```

```
            // display the results
            System.out.println(election.generateResultsReport(false));

            System.out.println("Please enter yes/no if there are additional users");

            boolean loop = true;
            while(loop)
            {
                    String response = sc.nextLine().toLowerCase();
               if (response.equals("yes"))
               {
                  loop = false;
               } else if (response.equals("no"))
               {
                  isMoreVoters = false;
                  loop = false;
               } else
               {
                  System.out.println("Invalid response. Please enter yes or no.");
               }
            }
      }

      System.out.println(election.generateResultsReport(true));
}
```